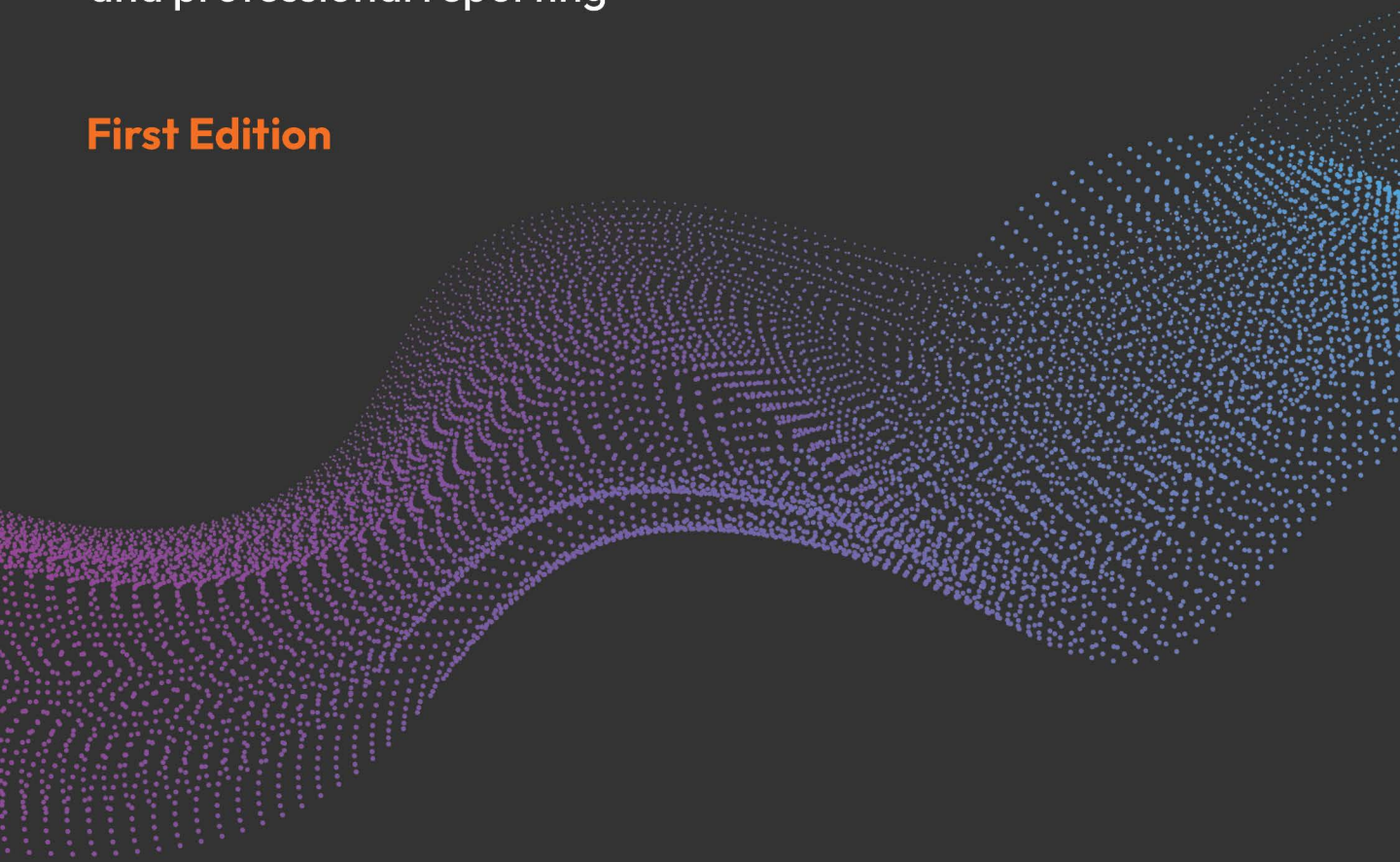
EXPERT INSIGHT

The Ultimate Ethical Hacker's Guide

Build job-ready skills with hands-on labs in recon, exploitation, and professional reporting

First Edition



Glen D. Singh

‹packt›

The Ultimate Ethical Hacker's Guide

Build job-ready skills with hands-on labs in recon, exploitation, and professional reporting

Glen D. Singh

‹packt›

The Ultimate Ethical Hacker's Guide

Copyright © 2026 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.
Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.
Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

Portfolio Director: Vijin Boricha
Senior Portfolio Manager: Rahul Nair
Program Manager: Ankita Thakur
Content Engineer: Tazeen Shaikh
Technical Editor: Nithik Cheruvokodan
Copy Editors: Tazeen Shaikh and Swathi Ajayakumar
Indexer: Pratik Shirodkar
Proofreader: Tazeen Shaikh
Production Designer: Alishon Falcon
Growth Lead: Ankita Thakur

First published: June 2026

Production reference: 1290626

Published by Packt Publishing Ltd.
Grosvenor House
11 St Paul's Square
Birmingham
B3 1RB, UK.

ISBN 978-1-80666-115-2
www.packtpub.com

I would like to dedicate this book to the people in our society who have always worked hard in their field of expertise and who have not been recognized for their hard work, commitment, sacrifices, and ideas, but who, most importantly, believed in themselves when no one else did. This book is for you. Always have faith in yourself. With commitment, hard work, and focus, anything can be possible. Never give up, because great things take time.

– Glen D. Singh

Contributors

About the author

Glen D. Singh is an information security professional, enterprise strategist, and educator specializing in security governance, cyber incident response, and enterprise cyber operations. He is the author of more than 12 books covering ethical hacking, penetration testing, cyber operations, and network engineering. With over a decade of academic experience, Glen has served as a lecturer in information security and network engineering. He holds an MSc in Cyber Security and certifications from EC-Council, Cisco, and Check Point. Passionate about mentoring others, Glen is dedicated to advancing cybersecurity awareness and digital resilience in Trinidad and Tobago.

I would like to thank God, the preserver of the universe, for all His divine grace and guidance. I would also like to thank Rahul Nair, Tazeen Shaikh, Ankita Thakur, and the wonderful team at Packt Publishing, who have provided amazing support throughout this journey. To the technical reviewer, Mayur Parmar, thank you for your outstanding contribution to making this an amazing book.

About the reviewer

Mayur Parmar is a Team Lead and Penetration Tester with 6+ years of experience in web, mobile, API, thick-client security testing, and red team operations. He holds an M.Sc. in Digital Forensics and Information Security from the National Forensic Sciences University, Gandhinagar, and is a PortSwigger Burp Champion, APIsec Ambassador, and Hack The Box SME.

Active in the bug bounty community since 2019, he has contributed across Synack, HackerOne, Bugcrowd, and Open Bug Bounty platforms and is ranked among Yogosha's Top 50 researchers globally. Recognized among India's Top 100 Cybersecurity Influencers by CyberFrat, he holds certifications including eWPTX, CRTP, CRTA, eCPPT, eJPT, PT1, CEH, ICCA, ASCP, CASA, and APIsec Certified Practitioner.

Table of Contents

Preface **xiii**

Free benefits with your book ..... **xx**

Chapter 1: Getting Started with Ethical Hacking **1**

Understanding information security **2**

Information security plans • 3

Pillars of information security • 4

Understanding the methodologies and frameworks of threat actors **7**

The Cyber Kill Chain • 7

Reconnaissance • 8

Weaponization • 9

Delivery • 9

Exploitation • 10

Installation • 10

Command and control (C2) • 10

Actions on objectives • 11

Tactics, techniques, and procedures • 12

Tactics • 12

Techniques • 13

Procedures • 14

Penetration testing methodologies • 14

Ethical hacking concepts **17**

Classes of hackers • 18

Phases of hacking • 20

Reconnaissance • 20

Scanning • 21

Gaining access • 21

Maintaining access • 22

Clearing tracks • 22

Scoping and limitations • 23

Information security controls 24

Information security laws and standards 26

Payment card industry data security standard (PCI DSS) • 27

ISO/IEC 27001 • 28

HIPAA • 28

SOX • 29

DMCA • 29

FISMA • 29

Summary 29

Further reading 30

Chapter 2: Setting up an Ethical Hacking Lab 33

Technical requirements 34

Ethical hacking lab design 35

Setting up a virtualized environment 40

Part 1 – Installing the hypervisor • 40

Part 2 – Creating a virtual network • 44

Part 3 – (Ubuntu users only) Installing Oracle VirtualBox and creating a virtual network • 46

Setting up attack machines 47

Part 1 – Setting up Kali Linux • 48

Part 2 – Configuring the network adapter for Kali Linux • 52

Part 3 – Configuring Kali Linux • 55

Part 4 – Implementing local AI power with Ollama on Kali Linux • 60

Setting up vulnerable machines 64

Part 1 – Building a vulnerable Windows-based virtual machine • 65

Part 2 – Building a vulnerable Linux-based virtual machine • 71

Part 3 – Setting up the Metasploitable 2 virtual machine • 74

Setting up a vulnerable wireless lab 78

Summary **82**
Further reading **82**

Chapter 3: Reconnaissance and Footprinting **83**

Technical requirements **84**
Footprinting concepts and methodology **84**
Organization information • 88
Network information • 89
System information • 90
Reconnaissance methodology • 91
Reconnaissance using search engines and web services **92**
Lab: Using Google dorking techniques to locate hidden systems • 95
Lab: Identifying organization's subdomains and exposed assets • 99
Part 1: Using Netcraft to profile the public infrastructure of an organization • 99
Part 2: Using Subfinder and DNSx to enumerate subdomains • 102
Part 3: Working with online tools to automate subdomain enumeration • 105
Lab: Using geospatial data to locate an object • 107
Email and web footprinting **111**
Lab: Performing email footprinting • 113
Lab: Identifying sensitive systems using deep web search • 114
Lab: Setting up alerting for a target • 120
WHOIS and DNS footprinting **121**
Lab: Performing WHOIS footprinting • 122
Lab: Performing DNS footprinting • 126
Reconnaissance countermeasures **129**
Summary **131**
Further reading **132**

Chapter 4: Scanning **133**

Technical requirements **134**
Fundamentals of network scanning **134**
Port scanning • 135

Network scanning • 135
Vulnerability scanning • 136
Host discovery concepts and techniques 137
ICMP scanning concepts • 137
ARP scanning concepts • 139
UDP scanning concepts • 141
TCP scanning concepts • 141
IP scanning concepts • 143
Lab: Identifying live hosts using Nmap • 144
Lab: Working with Zenmap for host discovery • 151
Service port and operating system discovery 153
TCP Connect scan • 154
TCP stealth scan • 154
TCP Inverse scan (with FIN/Xmas/Null inside) • 156
TTL values • 157
Lab: Using TCP scans to identify open ports • 159
Lab: Identifying specific open ports • 164
Lab: Identifying running services and operating systems • 166
Lab: Perform scanning techniques using AI • 168
Scanning beyond IDS and firewalls 169
Lab: Evade threat detection systems with Nmap • 172
Scanning countermeasures 175
Summary 176
Further reading 177

Chapter 5: Enumerating Systems 179

Technical requirements 179
Enumeration concepts 180
NetBIOS and SNMP enumeration 182
Lab: Performing NetBIOS enumeration • 185
Lab: Performing SNMP enumeration • 189
SMTP and DNS enumeration 193

Lab: Performing SMTP enumeration • 197
Lab: Performing DNS enumeration • 201
Common network service enumeration 204
Lab: Enumerating Microsoft RPC services • 206
Lab: Enumerating SSH • 208
Enumeration countermeasures 210
Summary 213

Chapter 6: Unlock the Code Bundle and the PDF Version 215

Unlock this book's free benefits in three easy steps 216

Other Books You May Enjoy 220

Index 223

Preface

As organizations continue to add more technology into their environment to support the growing demand of their workforce and meet their business objectives, they're also increasing their attack surface from a cybersecurity perspective. While many companies have either an internal Information Technology (IT) unit or are outsourcing their technical services to a third-party provider, these teams are responsible for configuring systems and network architectures. However, quite often IT professionals may not always apply best practices such as using secure baselines or ensuring systems have the latest firmware or patch installed. Such malpractices create security vulnerabilities that enable threat actors to exploit those flaws and compromise the confidentiality, integrity, and/or availability of the system or network.

Therefore, organizations can implement a proactive approach by performing regular security assessments on their assets and network architecture to determine the risk if a real cyber-attack or threat were to materialize. Security assessments help organizations identify gaps in their security architecture, determine whether their existing security controls are configured and operating correctly to prevent or mitigate various cyber-attacks. Additionally, security assessments help senior management such as the executive team to better understand the type and severity of cyber-risks that exists and the potential impact to their business if a real threat were to occur.

Hence, the need for ethical hacking remains essential as organizations continue to use technology to enhance their operations. Ethical hackers use their skillset to legally help organizations identify hidden security vulnerabilities in their systems and provide recommendations on how to improve their security posture. Yet, there are many organizations that are developing new solutions, expanding their network architectures and implementing new devices but aren't performing regular ethical hacking assessments to determine whether they are introducing new vulnerabilities within their network or not.

Spending over a decade as an information security professional in various industries, I've seen many organizations implement various systems, from servers and networking devices to even network security solutions, with misconfigurations and running outdated firmware. As a cybersecurity educator, I've met and taught many aspiring and seasoned IT professionals who often admit their oversight of the need for ethical hacking and understanding how this skillset can help them improve their cyber defenses. Furthermore, meeting with many business leaders and having insightful discussions on how their infrastructure can be compromised

helps them to better appreciate the proactive approach and need for security assessments and ethical hackers on their technical teams.

I've written this book to help IT professionals who are interested in ethical hacking and simulating real-world cyber-attacks and threats to identify hidden security vulnerabilities on systems and networks. It builds practical ethical hacking skills through realistic attack scenarios that reflect how security assessments are performed in real organizations. It helps learners move beyond theory by showing how vulnerabilities are discovered, exploited responsibly, and reported clearly.

This book begins by introducing the fundamentals of information security such as the importance of protecting confidentiality, integrity, and availability of assets. It explores how ethical hackers adopt common attack frameworks that are mapped to the **Tactics, Techniques and Procedures** (**TTPs**) of real adversaries. Using these TTPs helps ethical hackers simulate similar attacks and threats on organizations to determine whether security vulnerabilities exist and better understand their security posture. Then, we take a deep dive into both passive and active reconnaissance techniques and learn how ethical hackers use common tools to assist with collecting and analyzing sensitive information about their targets. You will explore common scanning techniques to identify misconfigurations, vulnerabilities in running services, open ports, and operating system versions of live hosts on a network.

Furthermore, you will be delving into vulnerability management to understand how ethical hackers perform vulnerability assessments on targeted systems and networks while using common vulnerability scanners in the industry. The information collected during vulnerability assessment phases helps ethical hackers move to the system hacking phase. During the system hacking phase, you will discover how common tools and techniques enable you to exploit discovered security vulnerabilities, gain unauthorized access, set up, and establish persistent access on compromised devices and networks. You will explore the fundamentals of malware threats and how threat actors use evasion techniques to bypass security controls, and intercept communication channels to capture sensitive information that leads to unauthorized access to systems and networks.

The last part of the book explores the security vulnerabilities that exist within wireless networks and how ethical hackers can determine whether their organization is susceptible and apply recommended countermeasures. Lastly, this book concludes by providing you with a structural framework for writing a security assessment report that contains your findings during each phase of your ethical hacking engagement.

By the end of this book, you will be able to perform ethical hacking engagements from reconnaissance to reporting with confidence.

Who this book is for

This book is for aspiring ethical hackers, penetration testers, and cybersecurity professionals who want hands-on experience with real-world attack techniques and defensive awareness. Basic knowledge of networking and operating systems is recommended.

What this book covers

Chapter 1, *Getting Started with Ethical Hacking*, introduces the information security pillars and the importance of protecting assets, explores various security frameworks and how they're used by security professionals, ethical hacking concepts, and the need for various security controls.

Chapter 2, *Setting up an Ethical Hacking Lab*, provides the methodology for building a virtualized lab environment for safely developing and sharpening ethical hacking skills.

Chapter 3, *Reconnaissance and Footprinting*, explores common tools and techniques used by ethical hackers to passively gather sensitive information about targeted systems and networks for planning an attack.

Chapter 4, *Scanning Networks*, dives into using active reconnaissance tools that help ethical hackers to profile systems, identify operating systems, discover open ports, running services, and use advanced scanning techniques to evade network-based security solutions.

Chapter 5, *Enumerating Systems*, focuses on extracting sensitive information from misconfigured systems that helps ethical hackers to identify common attack vectors that lead to system exploitation.

To get the most out of this book

- It's recommended to have a solid foundation in networking fundamentals, implementation, and operations to help bridge the gap in understanding how systems communicate with each other
- Having the fundamental knowledge of security solutions such as firewalls, email security, and endpoint- and network-based threat protection systems provides additional insights on their role and function in a network architecture
- Experience working with or using hypervisors for setting up and managing virtual machines and networks will be beneficial
- Familiarity with Linux-based and Windows-based operating systems is beneficial

> **Note**
>
> **Disclaimer**
>
> The information within this book is intended to be used only in an ethical manner. Do not use any information from the book if you do not have written permission from the owner of the equipment. If you perform illegal actions, you are likely to be arrested and prosecuted to the full extent of the law. Neither Packt Publishing nor the author of this book takes any responsibility if you misuse any of the information contained within the book. The information herein must only be used while testing environments with proper written authorization from the appropriate persons responsible.

Download the example code files

This book includes a complete downloadable code bundle containing all the example projects and files used throughout the chapters. We recommend downloading the bundle so you can follow along smoothly and experiment with the examples.

Use the bundle as a practical starting point. Modify it, extend it, and apply what you learn by creating your own variations as you progress through the chapters.

Get the code bundle

If you bought the book directly from Packt:

1. Go to **packtpub.com**
2. Click your **profile picture** and select **Your Orders**
3. Find this book and click **Download Code**

If you bought this book from Amazon or any other channel partner:

1. Go to `packtpub.com/unlock` or scan the following QR code



2. Search for this book
3. Sign up or log in to your free Packt account
4. Upload your proof of purchase and download the code bundle locally

Usage note: You're free to use and modify this code for personal learning and non-commercial projects.

Download the color images

Your purchase includes a color, DRM-free PDF copy of this book, ideal for viewing color images, screenshots, and diagrams. Refer to the *Free benefits with your book section* at the end of the *Preface* to unlock your PDF copy.

Conventions used

There are a number of text conventions used throughout this book.

`CodeInText`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and X (formerly known as Twitter). For example: " To power off this virtual machine, use the `sudo halt` command "

A block of code is set as follows:

```
#!/bin/bash
domain=$1

echo "[*] WHOIS Lookup"
whois $domain> whois_$domain.txt
```

Any command-line input or output is written as follows:

```
cd .vagrant.d\boxes
vagrant box add rapid7/metasploitable3-ub1404
```

Bold: Indicates a new term, an important word, or words that you see on the screen. For instance, words in menus or dialog boxes appear in the text like this. For example: "Ensure your processor supports **VT-x/AMD-V**. These are hardware-assisted technologies that support virtualization "

Note

Warnings or important notes appear like this.

Tip

Tips and tricks appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book or have any general feedback, please email us at `customercare@packt.com` and mention the book's title in the subject of your message.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you reported this to us. Please visit `http://www.packt.com/submit-errata`, click **Submit Errata**, and fill in the form. All valid errata will be timely updated on GitHub.

Piracy: If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at `copyright@packt.com` with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit .

Share your thoughts

Once you've read *The Ultimate Ethical Hacker's Guide*, we'd love to hear your thoughts! Scan the QR code below to go straight to the Amazon review page for this book and share your feedback.



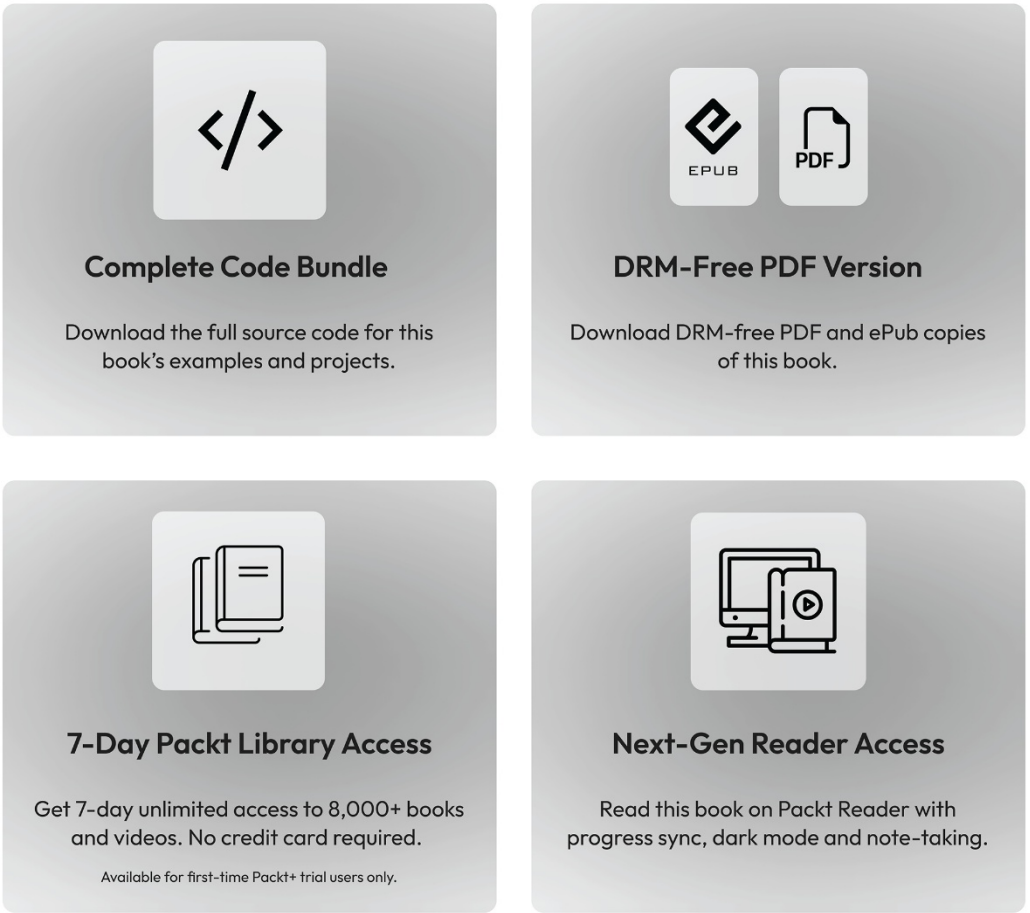
https://packt.link/r/1806661152

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.

Free benefits with your book

This book includes free benefits designed to support your learning and help you apply what you learn effectively. Activate them now for instant access (see the *How to unlock* section for instructions).

Here's a quick overview of what you can instantly unlock with your purchase:



How to unlock

Scan the QR code (or go to `packtpub.com/unlock`). Search for this book by name, confirm the edition, and then follow the steps on the page.





Note: Have your invoice handy. Purchases made directly from the Packt website don't require an invoice.

1

Getting Started with Ethical Hacking

Threat actors are always seeking ways to identify and exploit security vulnerabilities on targeted systems, networks, and applications. Threat actors are the person or group responsible for performing a cyber-attack or threat on an application, system, or network, while cybersecurity professionals are continuously working on staying at least one step ahead of those threat actors. However, many people may think hackers only seek new methods to compromise their targets, it's important to remember that many organizations around the world still have lots of unpatched systems with misconfigurations and even vulnerable applications that contain old security vulnerabilities that are easy to compromise.

Can you imagine the potential impact and damage that can occur if a threat actor finds a critical vulnerability on a mission-critical server of a large organization? The *confidentiality* of data, the *integrity* of the system, and the *availability* of the services and resources of the server are compromised. These are known as the **CIA Triad**, the three foundational pillars of information security that help security professionals to govern how assets are safeguarded from threat actors. Therefore, any user who depends on the services and resources of a compromised server will be affected.

One of the most popular and trending fields within the cybersecurity industry is ethical hacking. **Ethical hacking** is the practice of using hacking techniques of real adversaries to identify and exploit security vulnerabilities on targeted systems, with legal permission. **Ethical hackers**, commonly referred to as **white-hat hackers**, use their skills to simulate real-world cyber-attacks to identify hidden security vulnerabilities and report them to the organization with recommendations on how those security flaws can be mitigated. Furthermore, ethical hacking exercises help organizations to identify whether their existing security controls and countermeasures are working as expected or if any gaps exist.

In this chapter, you will learn about the fundamentals of information security, why it's important and how it helps organizations to develop a governance framework. You will explore various phases of the **Cyber Kill Chain** and how ethical hackers align their procedures to simulate real-world cyber-attacks and threats. Additionally, you will also gain a deep dive into understanding why ethical hackers use **Tactics, Techniques, and Procedures** (**TTPs**) similar to real adversaries during security assessments.

Furthermore, you will learn about various information security controls and common information security laws, industry standards, and best practices for organizations that operate within the payment card industry and those that want to ensure their systems and networks are compliant with security industry standards.

This chapter covers the following topics:

- Understanding information security
- Understanding the methodologies and frameworks of threat actors
- Ethical hacking concepts
- Information security controls
- Information security laws and standards

Note

Download the code bundle and the PDF version of this book

Your purchase includes a DRM-free PDF copy of this book, the code bundle, and a range of exclusive benefits. To unlock everything, follow the *Free benefits with your book* section in the *Preface*.

Understanding information security

As many organizations around the world continue to adapt to the digital age to support business growth, reach new customers, and go beyond traditional borders, they are becoming susceptible to various types of cyber-attacks and threats on the internet and even internally within their organization. For instance, organizations store a lot of data of internal and external users in various places, such as on employees' laptops, on-premise servers, cloud platforms, **Network Attached Storage** (**NAS**), and even networking devices. This data is also shared via different networks, thus opening itself to various cyber-attacks and threats.

Threat actors are aware that data has become most valuable to many organizations. So, its value continues to rise. While threat actors perform traditional cyber-attacks on their targets such as **Distributed Denial-of-Service** (**DDoS**), others are motivated by financial gain and

have developed more sophisticated threats and business models. One such business model is offering **Ransomware-as-a-Service (RaaS)** on the dark web. This enables anyone with access to the dark web to either subscribe, pay a one-time fee or become affiliated with the ransomware developer to use their services or malicious code for their personal agenda and gain. Hence, it's important for organizations to use information security frameworks and practices to prevent and mitigate against new and existing cyber-attacks and threats.

> **Note**
>
> Keep in mind, as an ethical hacker you are not required to understand the operational side of the **RaaS** but only the general concept as it's related to modern threats.

By moving toward a state of information security, organizations are better equipped to protect their assets against various types of cyber-attacks and threats, as well as unauthorized access and use of their electronically stored information. **Assets** are anything that has value to an organization or a person. The following are common types of assets:

- **Tangible** – Tangible assets are anything in a physical form, such as servers, computers, networking devices like routers and switches, and security appliances such as firewalls
- **Intangible** – Intangible assets are assets without a physical form such as intellectual property, software license keys, and data
- **People** – These are the employees, stakeholders, suppliers/vendors, and customers of an organization

Without developing an information security governance, organizations will not have clear goals or objectives that are measurable with strategic plans, tactical plans, and operational plans. Now, let's learn about common information security plans, frameworks, and industry's best practices.

Information security plans

The *strategic plan* of an organization is the long-term vision and goals for information security to ensure it aligns with the business objectives and helps stakeholders to better understand the importance of implementing information security best practices. For instance, an organization may want to adopt the **National Institute of Standards and Technology (NIST) Cybersecurity Framework (CSF) 2.0** to improve their cyber defense. Implementing a framework does not occur over days; sometimes the process can span months and sometimes years depending on the available resources. In this instance, the implementation of the NIST CSF 2.0 can be one of many goals within the organization's strategic plan.

The *tactical plan* is a short-term plan that focuses on specific tasks and allocation of resources for meeting strategic goals. As compared to the strategic plan that may span a period of 5 years,

the tactical plan is usually 12 – 18 months. Being a short-term plan, it enables the organization and information security professionals to create actionable steps, determine the best procedures and technologies with realistic timelines for executing each step to meet strategic goals. Lastly, the *operational plan* focuses on the daily operations and tasks with detailed activities implementing the actions of the tactical plan.

The following diagram shows each of these information security plans and how they are interconnected:

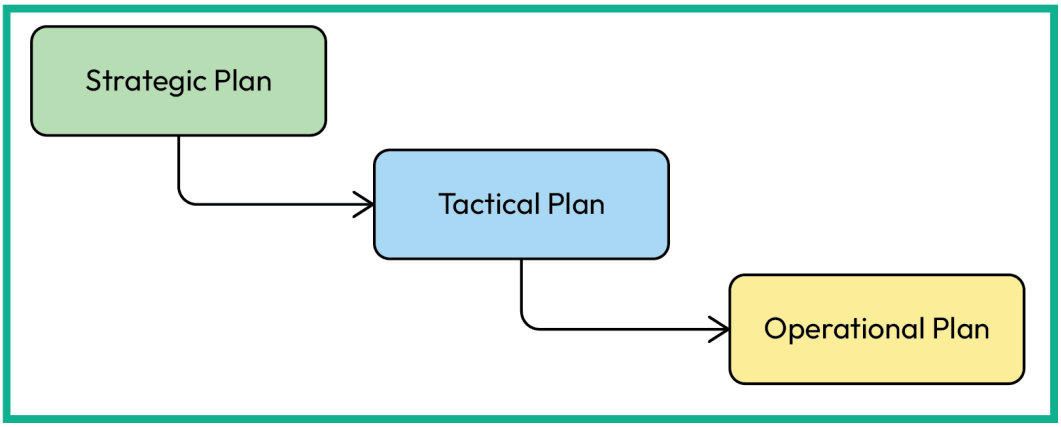


Figure 1.1 – Information security plan

Imagine an organization wanting to improve their data privacy posture to ensure unauthorized users are unable to access confidential information or systems. Therefore, the organization creates a strategic goal on long-term data protection called "*Implementing a Zero Trust Architecture that improves data privacy.*" The tactical plan can focus on identifying the current gaps and what's needed to fill those gaps. This would include creating actions and timelines to implement a **Multi-Factor Authentication** (**MFA**) solution on all systems within the organization. Lastly, the operational plan would focus on day-to-day actions such as monitoring and tuning the MFA solution to operate efficiently and enrolling and training staff on the MFA policies and procedures.

Now that we've discussed what an information security plan consists of, let's learn more about its pillars and their roles in keeping systems, networks, and applications safe from malicious activities.

Pillars of information security

Information security relies on some key elements to reduce cyber-risks such as tampering of data or configurations, theft of information, and disruption of services and resources to

legitimate users. The following are the five elements that information security frameworks and standards rely upon:

- **Confidentiality**: Confidentiality focuses on ensuring only authorized users have access to data. For instance, a breach in confidentiality can be due to missing or inadequate data encryption technologies that enable threat actors to intercept network communication and access sensitive data that's being transmitted over the network. In another instance, an entry-level employee may have access to sensitive corporate records due to misclassification of information within the organization and lack of information systems auditing and enforcement of security controls.
- **Integrity**: Integrity focuses on ensuring data is not altered or modified by an unauthorized user. Consider an example of a breach in integrity due to a malware infection on server that stores sensitive information such as financial records. The malware can be designed to corrupt the files on the server, therefore compromising the integrity of the data itself.
- **Availability**: Availability focuses on ensuring that systems and resources are available to authorized users. A breach in availability can be intentional and unintentional. For instance, a threat actor can perform a **Denial of Service** (**DoS**) attack on a targeted web application server that causes the server to become unresponsive to legitimate users on the internet, which would be the intent of the threat actor. However, a server can become unavailable if any of its hardware components are no longer working or the power source is affected. System administrators, for example, can implement **Redundant Array of Independent Disks** (**RAID**) to provide redundancy on the server's storage disks and implement **Uninterruptible Power Supply** (**UPS**) to provide backup power if the main power source is unavailable.
- **Authenticity**: Authenticity focuses on ensuring the data is genuine and has originated from a trusted source. When software developers publish their applications, they usually include a digital certificate that helps both customers and systems to determine the identity of the vendor which builds trust. A digital certificate is a digital identification card that helps users and devices to determine the identity of another user or system over an untrusted network such as the internet. A breach in authenticity would be if threat actors are sending phishing email campaigns to targeted victims. The phishing emails would seem to originate from a trusted sender such as a bank, service provider or even an e-commerce organization. The intention would be to trick the victims into believing the email is legitimate, either open the malicious attachment or click the obfuscated link in the message. The intention of the threat actor would be

to collect sensitive information from the victims, compromise their systems and/or expand their cyber-attack.

- **Non-repudiation**: Non-repudiation provides the reassurance that a user is unable to deny their action on a system. For instance, it provides the reassurance that a sender of an email cannot deny they've sent the email, and the recipient of the message cannot deny they've received the email. For instance, using digital signatures within an email message proves evidence the email originated from the real sender, hence the sender is unable to repudiate/deny sending the message.

While there are five elements of information security, three of them make up the **CIA Triad**, which is *confidentiality*, *integrity* and *availability* as shown in the following diagram:

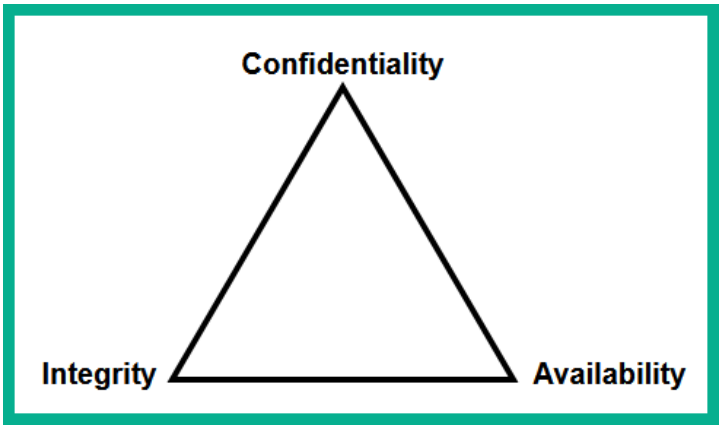


Figure 1.2 – CIA Triad

The goal is to create a balance of each element in the CIA Triad, but this can be challenging. For instance, if an organization focuses on availability such as easy access to systems and applications for the end users, the security controls to enforce confidentiality and integrity would be reduced or less strict. Similarly, if an organization focuses more on confidentiality or integrity, the availability of resources and services would impact the authorized users. Hence, its importance to create a well-balanced security ecosystem that enables functionality among users within an organization.

There are information security attacks, methodologies, and frameworks that are commonly used in industry, which help organizations improve their systems and networks. We will now discuss those in detail.

Understanding the methodologies and frameworks of threat actors

Behind every cyber-attack or threat, there's always a threat actor who is responsible for their malicious actions. In the threat actor's mindset, the targeted system has something that's valuable, which leads to creating a goal and working toward it. As an aspiring ethical hacker and cybersecurity professional, it's important to understand common goals of adversaries.

The following are common motives of adversaries, but are not limited to:

- Information theft
- Compromise brand reputation
- Service interruption
- Create financial loss for the target
- Spread social, political, or religious beliefs
- Revenge
- Sabotage
- Espionage

Each threat actor always has a purpose or rationale for their cyber-attack, whether they've realized it or not. The common reasons for cyber-attacks, but not limited to financial gain, steal and leak trade secrets, personal vendettas, and proving their skills. Before proceeding to attack their targets, threat actors determine whether there are sufficient resources available to complete the attack. If resources are insufficient, the threat actor finds a way to obtain the required amount. The start of the attack may be postponed depending on how long it takes to acquire them.

In addition, the threat actor usually identifies at least one security vulnerability on the targeted system prior to the attack. A *vulnerability* is a security weakness or flaw within an application, system or network. The threat actor either develops or acquires an exploit that can take advantage of the vulnerability and compromise the targeted system. Without the right skills and capabilities of the threat actor, their likelihood of success is reduced. Hence, some threat actors will collaborate with others who have the required skillset and work together with a mutual arrangement of perhaps sharing the loot.

The Cyber Kill Chain

How can aspiring ethical hackers and cybersecurity professionals better visualize and map each phase of an attack by a real adversary? This is where you'll be introduced to the **Cyber Kill Chain**, an industry-recognized adversary attack map that helps cybersecurity professionals

better understand how attackers are able to perform various types of attacks and enables organizations to improve their cyber defenses. The Cyber Kill Chain is a seven-layer framework that provides visibility into each phase of an attack.

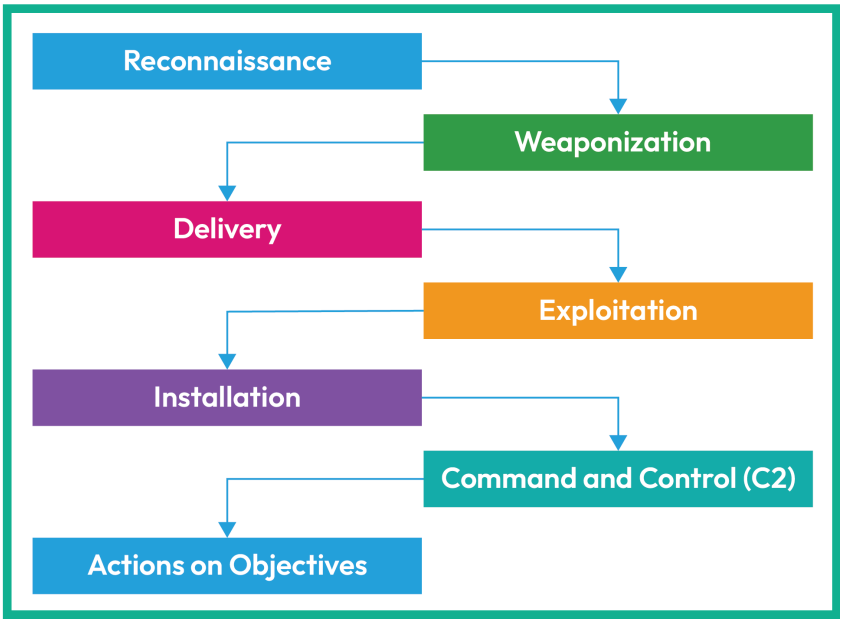


Figure 1.3 – The Cyber Kill Chain

In the following sub-sections, you will learn about the role and function of each phase of the Cyber Kill Chain.

Reconnaissance

During the **reconnaissance** phase, the adversary or threat actor focuses on information gathering, collecting as much information as is valuable to planning the attack on the target. Some techniques would include passive information gathering, also known as **Open Source Intelligence** (**OSINT**), from public data sources that help the attacker to identify any security vulnerabilities.

Many organizations don't always realize how they are unintentionally leaking sensitive data on the internet which can be leveraged by threat actors. Active information gathering focuses on making direct contact with the target, perhaps sending specially crafted probes (packets) to the targeted servers to identify open service ports or scanning a web application for known security vulnerabilities. Keep in mind, the reconnaissance phase is very important as the information gathered and analyzed helps the threat actor in the next phase of the Cyber Kill Chain.

Weaponization

Once the threat actor has sufficient information about the target and has identified at least one security vulnerability, they will develop or acquire an exploit to compromise each of the security vulnerabilities found on the targeted systems. If the threat actor has the skillset, they can develop their own exploit or malicious code. Then, they build a test environment to simulate the targeted system and execute their malicious code within it to determine whether the exploit is working as expected or if further improvement is needed prior to delivering the exploit to the target. Additionally, a threat actor can attempt to acquire the exploits or malicious code from various sources on the clear web/internet or the dark web.

> **Note**
>
> A word of caution: it's important to remember that the author of the exploit or malicious code can have more nefarious intentions and may have designed their code to compromise the systems of anyone who uses it.

Furthermore, the threat actor usually couples a payload onto the exploit that is designed to perform post-exploitation functions such as establishing a backdoor onto the compromised system or even setting up a reverse shell to the attacker's machine. Payloads are usually referred to as the code that executes after an exploit has successfully compromised a targeted system. Hence, this phase of the Cyber Kill Chain focuses on developing a weapon to compromise the confidentiality, integrity, and/or availability of the target.

Delivery

During the **delivery phase**, the threat actor focuses on delivering the exploit and payload to the targeted system. There are various delivery methods such as the following:

- **USB** – The exploit or malicious code can be copied onto USB flash drives and dropped at strategic locations within the vicinity of the target
- **Email** – Sending a phishing email to the target with a malicious attachment or tricking the target into clicking an obfuscated link to download and install the malicious code
- **Wi-Fi** – Either setting up a rogue access point within the vicinity of the target with a malicious captive portal that tricks the victim into downloading and installing malicious code onto their system
- **Physical** – The threat actor has physical/local access to the targeted system and delivers the malicious code via a portable storage device or downloads over a network connection

Without the delivery phase, the threat actor won't be able to transport the exploit with payload onto the targeted system. Once the exploit with payload is successfully delivered, the threat actor moves to the next phase, exploitation.

Exploitation

The **exploitation phase** focuses on executing the exploit and payload on the targeted system. However, if the exploit works within the threat actor's environment but not on the real targeted system, the threat actor usually revisits the weaponization phase once more or even the reconnaissance phase. If the exploit is successful, then the threat actor has compromised the targeted system and will be able to execute malicious code on them.

During this phase, the exploit can cause instability of the targeted system and result in system crashes, which can easily alert the cybersecurity team of the organization. Furthermore, sometimes an exploit would fail when it's executed the first time. A threat actor does not give up at this phase; they try again because they would have identified the value of the target and their goals for planning and executing the cyber-attack on the target.

Installation

The **installation phase** focuses on installing malicious software (malware) onto the compromised system. During the weaponization phase, the threat actor may have developed additional payloads that are installed during this stage to perform various post-exploitation actions such as creating backdoor access, escalating privileges, and even setting up persistent remote access on the compromised system. A common practice by seasoned threat actors is to set up multiple backdoors on compromised systems. Therefore, if one backdoor is not available, another one can be used to gain unauthorized access to the targeted system.

Command and control (C2)

The **Command and Control** (**C2**) phase focuses on setting up, establishing, and maintaining a two-way communication channel between the compromised systems and the attacker's command server. This enables the threat actor to easily manage compromised systems, exchange data, and further expand the attack on the network. The following image shows an example of C2 operations:

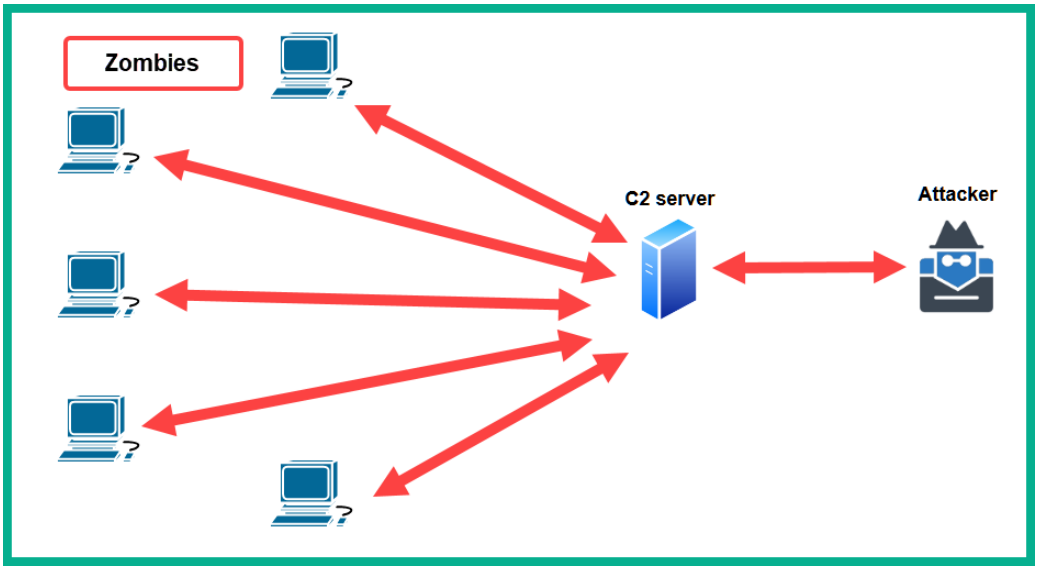


Figure 1.4 – C2 communication channel

Not all adversaries establish C2 during their attack; some threat actors may entirely skip this phase because it's not always part of their plan for meeting their goals. But in large-scale cyber-attacks, threat actors who want to control multiple compromised machines will set up a centralized C2 server and install C2 agents on each victim machine. However, we will not dive into this topic as it is out of scope in this book.

Actions on objectives

In **objectives phase**, the threat actor has accomplished their goals of the attack on the victim/targeted system. At this stage of the Cyber Kill Chain, the cybersecurity team of the compromised organization was unable to stop the attacker as they had worked their way through each phase to meet their goals. For cybersecurity teams, it's important to identify system and network-based intrusions as soon as possible and activate the organization's incident response procedures/playbooks to quickly contain the threat to reduce its potential impact on business operations.

As an aspiring ethical hacker, it's important to understand each phase of this framework as it helps you to better understand the methodology used by real adversaries and improve your security testing on targeted systems. You also should be aware of the patterns that are associated with the malicious activities of threat actors in the industry, which will be discussed next.

Tactics, techniques, and procedures

Cybersecurity researchers and professionals observe common patterns of malicious activities used by threat actors around the world; this is commonly referred to as **tactics, techniques, and procedures** (**TTPs**). Cybersecurity professionals leverage the TTPs of real adversaries to profile specific threat actors, understand how a threat actor behaves, and the tools and methodologies used to compromise their victims. Such information provides cybersecurity professionals with **cyber threat intelligence** (**CTI**) that enables organizations to improve their information security governance, incident response plans and procedures, and overall cyber defenses to safeguard their assets from malicious actors. Let's look at the purpose and differences between TTPs.

Tactics

The **tactics** are the goals or objectives of the threat actor or can easily be described as what the threat actor or threat actor group wants to accomplish with the cyber-attacks on the target. For instance, the **MITRE ATT&CK** framework lists the following tactics:

- **Reconnaissance** – Specializes in collecting as much useful information as possible to create a profile of the target
- **Resource development** – Focuses on developing or acquiring the resources needed to support an attack
- **Initial access** – Concentrates on common entry vectors to gain an initial foothold on the target
- **Execution** – Focuses on ensuring the malicious code, exploit and payloads run on the targeted system
- **Persistence** – This tactic focuses on ensuring the threat actor always has access to the compromised systems and network, even after reboots or credentials are changed on the systems
- **Privilege escalation** – Concentrates on ensuring higher levels of privileges are obtained on the compromised systems on a network
- **Defense evasion** – Focuses on avoiding threat detection systems during the attack on the target
- **Credential access** – The threat actor focuses on stealing credentials from compromised systems
- **Discovery** – This tactic focuses on obtaining information about the systems and network architecture of the target

- **Lateral movement** – Consists of methods to access and manage remote systems on the internal network of the target
- **Collection** – Focuses on gathering information that's of interest to meet their goals of the attack
- **Command and Control** – This tactic focuses on establishing and maintaining a 2-way communication channel between the threat actor and compromised systems on the target's network
- **Exfiltration** – Focuses on techniques that enable the threat actor to steal data from the compromised network
- **Impact** – This tactic focuses on disrupting the availability or integrity of a targeted system

While there are many goals, threat actors develop strategic methods to achieve their goals and objectives; these are commonly referred to as techniques.

Techniques

Techniques are the specific methodologies that are used by threat actors to achieve their goals (tactics). They help cybersecurity professionals to understand how an adversary performs their attack on a target. Whether the threat actor has developed malware and delivered it onto a targeted server or sends malicious code over a network to a remote server that exploits a security vulnerability within the operating system.

For instance, if an attacker's goal is reconnaissance on a targeted organization, the following are some techniques for achieving the goal:

- Active scanning
- Gather victim host information
- Gather victim identity information
- Gather victim network information

Let's consider another scenario: if the attacker's goal is credential access, the common techniques for meeting this goal would include brute force, credentials from password stores, input capture, and network sniffing.

Ethical hackers can leverage the techniques of real adversaries to simulate real-world cyber-attacks during their security assessments and exercises. By using real techniques, organizations can test whether their security controls and cybersecurity team are able to detect and prevent a real cyber-attack if one were to occur.

Procedures

Procedures are the detailed step-by-step actions performed by a threat actor to implement their techniques. This enables cybersecurity professionals to better understand the method or way an attacker behaves, including the sequence of steps followed by a threat actor to meet their goals. While it may seem difficult as the threat actors work their way through each phase of the Cyber Kill Chain, the cybersecurity team can analyze the **indicators of compromise** (**IoCs**) and the procedures and tools used by the attacker in each phase to profile a threat actor group or threat actor who is responsible for a real cyber-attack. Through this method, ethical hackers adopt similar tools and procedures to test an organization's security controls.

> **Note**
>
> The following provides an example of the procedures used during the Triton Safety Instrumented System Attack by a real-world threat actor: `https://attack.mitre.org/campaigns/C0030/`.

Furthermore, organizations can leverage the procedures to profile a threat actor group or threat actor who is responsible for a real cyber-attack. While it may seem difficult as the threat actors work their way through each phase of the Cyber Kill Chain to meet their goals, the cybersecurity team can analyze the IoCs, and the procedures and tools used by the attacker in each phase.

Penetration testing methodologies

While the Cyber Kill Chain and MITRE ATT&CK TTPs are essential in the field of ethical hacking and penetration testing, it's also important to consider additional industry-recognized penetration testing methodologies and frameworks. For instance, ethical hackers who are testing web applications are commonly required to use the **Open Worldwide Application Security Project** (**OWASP**) testing guides, whereas testing a payment card system may require the **Payment Card Industry Data Security Standard** (**PCI DSS**) testing guidelines.

The following are common methodologies and frameworks used by ethical hackers, penetration testers, and red teams; however, this is an overview and you're not expected to understand or use all of them:

- **OWASP Web Security Testing Guide (WSTG)** – This OWASP security guide provides ethical hackers with a comprehensive security guide and methodologies for identifying security vulnerabilities within web applications. Developers should be aware of this security testing guideline and incorporate security within their design and **software development life cycle** (**SDLC**).

- **OWASP Mobile Application Security Testing Guide (MASTG)** – This comprehensive security testing guide helps both developers and ethical hackers identify security vulnerabilities within mobile applications.
- **OWASP Firmware Security Testing Methodology** – This firmware security guide provides both developers and cybersecurity professionals with a way to perform security assessments on firmware to identify whether vulnerabilities exist.
- **Penetration Testing Execution Standard (PTES)** – The PTES is a seven-phase penetration testing methodology that provides recommendations on common security testing procedures and tools. The seven phases of the PTES are the following:
    - **Pre-engagement interactions** – During this phase, the ethical hacker focuses on establishing the scope of security testing, the rules of engagement such as the Do's and Don'ts, and better understanding the expectations between the organization (customer/client) and the ethical hacker.
    - **Intelligence gathering** – This phase focuses on collecting information about the target using both active and passive reconnaissance techniques and analyzing the collected information to build a profile and identify any attack vectors of the target.
    - **Threat modeling** – Threat Modeling phase focuses on using the collected intelligence from the previous phase to identify any possible threats and determine the security posture of the target.
    - **Vulnerability analysis** – In this phase, the ethical hacker focuses on identifying vulnerabilities on the target, evaluating and determining the risk of whether those vulnerabilities can be exploited during security testing.
    - **Exploitation** – During the exploitation phase, the ethical hacker will attempt to compromise each identified security vulnerability and gain unauthorized access to the targeted system.
    - **Post exploitation** – This phase focuses on establishing persistent access, performing escalation of privileges, and expanding a foothold on the compromised system and network.
    - **Reporting** – The phase focuses on documenting all the findings of vulnerabilities that were identified, the exploits that worked and recommendations on how to resolve those security vulnerabilities. These findings are written in an executive and technical report that's provided to the client at the end of the security testing.

- **PCI Penetration Testing Guide** – This testing guide provides the instructions for identifying and testing security controls that are aligned to their **Payment Card Industry Data Security Standard** (**PCI DSS**). For instance, it provides guidance on penetration testing components, the qualifications of the person(s) performing penetration testing, recommended penetration testing methodologies, and even reporting guidelines.
- **Penetration Testing Framework (PTF)** – The PTF is another security testing guide that provides recommendations on various security tools for common testing categories. For instance, some testing categories that are covered in the PTF include bluetooth-specific testing, Cisco-specific testing, server-specific testing, and wireless penetration.
- **Technical Guide to Information Security Testing and Assessment (NIST 800-115)** – This testing guideline is developed and maintained by NIST.
- **Open Source Security Testing Methodology Manual (OSSTMM)** – This is an open methodology that's used by cybersecurity professionals within the industry. The OSSTMM outlines what each ethical hacker and penetration tester needs to know, how to perform security analysis, develop operational security metrics, trust analysis, and design an efficient workflow for security testing. The OSSTMM covers the following types of security testing:
    - Human-based security testing
    - Physical security testing
    - Wireless security testing
    - Telecommunication security testing
    - Data network security testing
    - Compliance

Ethical hackers should choose the methodology or frameworks relevant to their client's industry. For instance, if a client is operating within the payment card industry, it would be most suitable to use the PCI DSS testing guide. However, it's also important to understand the client's needs during the pre-assessment phase of ethical hacking or penetration testing to understand the deliverables and meet the client's needs.

Having learned about common methodologies and frameworks used by ethical hackers, now you need to understand why ethical hacking is important and how it's used to help organizations safeguard their assets from cyber-criminals.

Ethical hacking concepts

The term hacking is commonly associated with a criminal activity that involves causing harm or damage to a computing system that's not owned by the person who is performing the action. To put it simply, hacking is the exploitation of security vulnerabilities of a targeted application, system or network. In addition, hacking involves any malicious or intrusive activities that compromise the security controls of a target with the intention to gain unauthorized access into the targeted system or network. Furthermore, due to the intrusive nature, invasion of privacy, and causing harm or damage to an application, system or network that's not owned by the perpetrator, hacking is illegal.

But how does hacking work? There are many computer vendors, hardware manufacturers, and software developers that build and sell their products around the world. Each vendor or developer designs and builds their product to function within the creator's original purpose. For instance, consider the print spooler component of a Microsoft Windows Server, which is used to simply manage the print jobs from client computers to the printer on the network. However, a critical vulnerability identified in the print spooler on Microsoft Windows Server would enable a threat actor to perform **Remote Code Execution** (**RCE**) on the targeted system. This vulnerability is known as *PrintNightmare* and identified as CVE-2021-1675.

> **Note**
>
> To learn more about this vulnerability, please visit: `https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-1675`. Keep in mind, this example is illustrative only and not guidance for exploitation.

By exploiting the vulnerable print spooler on a Microsoft Windows Server, the threat actor will be able to remotely execute their malicious code within the memory of the server. Therefore, the attacker can craft specialized payloads to establish multiple backdoors, set up persistent access, escalate privileges, and more. Hence, the exploit that was either developed or acquired by the threat actor would have the capabilities to modify the software developer's original purpose for the print spooler.

Ethical hacking enables cybersecurity professionals to use similar TTPs as real hackers or threat actors to gain and maintain access to computing systems of an organization, with the intention to help improve the security posture of the company and ensure they do not cause any damage during their security assessments. However, ethical hackers always need to acquire legal permission from the leadership team of the organization prior to simulating any real-world cyber-attack on the organization's applications, systems, or networks.

Since ethical hackers use the same TTPs and tools as real adversaries, they're able to identify hidden security vulnerabilities on targeted systems, test the security controls, and determine how effective the organization's cyber defenses would be if a real cyber-attack were to occur. Keep in mind, ethical hackers also test their targeted systems to determine whether any known vulnerabilities exist and whether those vulnerabilities can be exploited. This helps them determine the severity and potential impact of each vulnerability if it were to be exploited by a real attacker. In addition, the severity and impact score help organizations to prioritize their resources in implementing preventative controls and countermeasures.

There are different types and classes of hackers and we need to understand the common motives and intentions of each. Additionally, you'll discover the common stages of ethical hacking and why each phase is important during security assessments of organizations in this section.

Classes of hackers

The term hacker is commonly used within the mainstream media and news websites to describe a person who uses their computer skills for malicious purposes. However, just a couple of decades ago, the term hacker was used to describe a person who was considered to be a computer wizard (wiz), someone who had a lot of knowledge about computing systems in their era. As times and communication evolved, the term took on a different meaning and became associated with the bad actors in the industry who sought to compromise their targets for malicious reasons.

Nowadays, there are many classes of hackers, and each class can be identified based on their goals or motives behind their malicious actions. The following are common classes of hackers in the industry:

- **Black hat** – The black hat hackers are persons who use their skills for malicious and illegal intent like seeking to install malware within an organization's network. This type of hacker is commonly involved in criminal activities.
- **White hat** – The white hat hackers are commonly known as the good actors in the cybersecurity industry because they use their skills to help organizations improve their security posture. These types of hackers are commonly referred to as ethical hackers, penetration testers and red teamers. These types of hackers have the legal permission of the system owners to perform security assessments to identify and assess security vulnerabilities.
- **Gray hat** – This type of hacker uses their skills for both good and bad reasons. For instance, this individual may be working as an ethical hacker in their day job to help organizations secure their infrastructure. However, the individual may perform

malicious or illegal activities during the night, such as cyberstalking or even hacking into other organizations.

- **Suicide hackers** – This type of hacker focuses on taking down the targeted infrastructure, no matter the cost or consequences of their actions, to support their cause. For instance, this hacker may be planning to obtain top-secret information from a governmental organization and leak the data on the *clear web* (commonly referred to as the internet) and the dark web and is willing to face any consequences of their actions as long as they can meet their objectives or goals.
- **Script kiddies** – These are novice hackers who usually lack the technical skills and knowledge of real hackers but follow the tutorials or instructions of real hackers to achieve their goals. However, since this type of hacker lacks the real skills and knowledge, they can potentially cause more damage to the targeted system or network compared to seasoned hackers.
- **Cyber terrorists** – The cyber terrorists are generally motivated by religious and political beliefs and use their skills to create fear and impose massive disruption of networks around the world.
- **State-sponsored** – These are persons who are sponsored or employed by a government to perform reconnaissance, compromise and even gain unauthorized access to the systems and networks of another government. This class of hackers is usually well-funded with the best-in-class training, tools and technologies to support their objectives and meet their goals. They can leverage cyber warfare weapons to gain an advantage over other nations around the world.
- **Hacktivists** – This type of hacker uses their hacking skills as an act of protest to raise awareness on either social or political agendas. For instance, one of the most popular hacktivist groups known as Anonymous launched a coordinated DDoS attack using the **Low Orbit Ion Cannon** (**LOIC**) tool against the Church of Scientology as a prank.

As an aspiring ethical hacker, it's important to recognize that not all hackers around the world are motivated or driven by the same goals. For example, black hat hackers are motivated by financial gain and/or destruction through their actions, whereas hacktivists are driven to spread their ideology to the public. Script kiddies, who are novices in the industry, are commonly driven by the thrill and excitement of hacking. Nation-state hackers focus on espionage and leveraging intelligence about other nations around the world.

Furthermore, the different classes and types of hackers may use different tools and techniques to pursue their goals; for example, script kiddies may rely heavily on **Graphical User Interface** (**GUI**) and pre-made tools to meet their objectives. Whereas hacktivist groups who focus on

compromising the availability of targeted servers would use a variety of DoS tools and messaging platforms to coordinate their attack.

It's important to learn about the motives of common types of hackers, as it helps you develop an attacker's and a defender's mindset. Entering the field of ethical hacking isn't just about running scripts or tools to collect information; it's also about developing the mindset of an attacker to understand the workflow or thought process of an adversary if they were to target your organization.

Phases of hacking

Breaking into the cybersecurity industry as an ethical hacker is always exciting, and many people are eager to hack their first target while on the job. However, ethical hackers commonly use a systematic approach to ensure they collect relevant and valuable information about the targeted systems, evaluate the vulnerability exposure and exploitability of each vulnerability found, expand their foothold, set up persistence, and cover their tracks as a real adversary would.

Generally, there are five phases in ethical hacking:

- Reconnaissance
- Scanning
- Gaining access
- Maintaining access
- Clearing tracks

Let's discuss each phase of ethical hacking in more detail.

Reconnaissance

Reconnaissance focuses on collecting information about the target that helps the threat actor or ethical hacker build a profile of the target and identify any security vulnerabilities that are exploitable. This phase is very important in hacking as the threat actor or ethical hacker needs to identify at least one vulnerability and attack vector to gain unauthorized access to the target system. There are two types of reconnaissance: active and passive.

In **active reconnaissance**, the threat actor or ethical hacker uses direct interactions with the targeted system using automated or custom-built tools to acquire technical information about the target. For instance, the ethical hacker may create a social engineering campaign and send it to employees of the targeted organization or even directly interact with the targeted organization's website to identify the technologies being used by the web server.

However, **passive reconnaissance** uses an indirect approach for collecting information about the target. For instance, if an ethical hacker were to enumerate the **Domain Name System**

(**DNS**) records of an organization's publicly registered domain, the ethical hacker could query a public DNS server such as Google's DNS servers for the records of the targeted domain name. Using passive reconnaissance reduces the threat detection level of the ethical hacker being known by the target.

Scanning

During the **scanning** phase, the ethical hacker uses automated tools or manual techniques to scan the targeted systems or networks. By performing network scanning, the ethical hacker can identify the number of live hosts on the network and the size of the network architecture. In addition, by scanning each host on a network, the ethical hacker can identify open ports, running services, profile the targeted system to determine the host's operating system and version, and even determine the device's role and function within the network.

However, the scanning phase can be broken down into the following:

- **Pre-attack** - The pre-attack phase focuses on scanning the network to identify specific information about the network architecture, mapping its layout, and determining the number of live hosts on the network.
- **Port scanning** – Port scanning focuses on identifying open ports using network scanners or network mapper tools. Additionally, vulnerability scanners are used to identify security flaws in targeted systems.
- **Extraction** – Extracting sensitive information from the live hosts such as their open ports, port statuses, operating system details and version number, and even the uptime of the targeted system prior to launching an attack.

During the scanning phase, the ethical hacker needs to identify at least one security vulnerability before moving on to the next phase, exploitation. Therefore, ethical hackers need to develop research skills to identify whether a service, operating system, or application contains a vulnerability and conduct tests to determine the exploitability and potential impact of the vulnerability. Once the ethical hacker has identified one or more exploitable security vulnerabilities, they can move on to the next phase of gaining unauthorized access.

Gaining access

In the **gaining access** phase, the ethical hacker either develops their own exploit using common programming languages and exploit development frameworks or acquires one from the various online sources for each vulnerability found during the scanning phase. Since ethical hackers use similar techniques and tools as threat actors, it's important to ensure the exploit is tested on a replica of the targeted system within your lab environment to ensure the exploit is working as expected. If the exploit is not working, the code should be improved and retested until it works as expected.

However, if the exploit is acquired via an online source, you need to ensure the author of the exploit is not trying to compromise you or your target. Furthermore, testing the exploit on a replica system of the target can determine whether the exploit needs improvement or not.

The primary goal of this phase is for ethical hackers to simply gain unauthorized access to the targeted system by exploiting a security vulnerability within its configurations, firmware, applications, or even operating system. Additionally, the ethical hacker can perform password cracking techniques to identify valid user credentials, perform buffer overflow on vulnerable applications, and capture session information over the network, all of which can be used to gain unauthorized access to a system.

Furthermore, ethical hackers can attempt to escalate their privileges to system, administrator or even root, which depends on the compromised system. By escalating privileges, ethical hackers can easily obtain complete control of the compromised system. Once the ethical hacker has gained unauthorized access, it's strategic to move onto the next phase of hacking, maintaining access.

Maintaining access

During this phase, ethical hackers attempt to use similar techniques as the threat actors to retain their foothold on the compromised system. The ethical hacker may use the following techniques to establish persistence on the targeted systems and networks:

- Set up multiple backdoors
- Install **Remote Access Trojans** (**RATs**)
- Create new accounts on the compromised device
- Set up automated reverse shell connection to the attacker's machine

Imagine you've compromised a targeted system such as a Windows Server or even a Linux server but lose access after the device is rebooted. Persistence enables ethical hackers to maintain their access on all compromised systems within a network, even after the device reboots. This enables you to further expand your foothold on the network and pivot attacks through existing compromised devices that have dual/multi network connections to more than one IP network. By pivoting an attack through a compromised host, ethical hackers can reduce their risk of being detected by the target's cybersecurity team.

Clearing tracks

In a real cyber-attack, seasoned threat actors typically clear their tracks, such as removing evidence that can be used to identify the source of the attack and who's responsible. However, since ethical hackers are testing the organization's security controls and incident response plans, they usually clear their tracks to determine whether the cybersecurity team can detect their presence and identify any artifacts that were left behind after the security assessment is

completed. Evidence and artifacts may include customer scripts, tools, and payloads that were transferred onto targeted systems during the ethical hacking process. Clearing of tracks involves the clearing or overwriting of logs on compromised systems and removing any files that were transferred from the ethical hacker's machine to the compromised systems.

However, if the organization's incident response or digital forensics team discovers the event logs and the items that were left behind, they can carefully analyze this information to determine the TTPs during the ethical hacking process and improve their incident detection capabilities. Before starting ethical hacking on an organization, you should be able to recognize the limitations of the assessment. So, let's discuss the importance of scoping the same.

Scoping and limitations

Prior to starting any security assessments with an organization or client, it's important to understand the scope of the assessment, the client's needs and their expectations. This helps the ethical hacker to focus and prioritize on specific targets that are defined by the client to avoid wasted time and efforts. In addition, scoping helps identify critical systems and manage risks to avoid causing system downtime during ethical hacking or penetration testing.

When developing your scoping form and procedures, they should include the following details:

- Include the authorized ethical hacker or penetration tester's details, such as names, contact information, and role during the security assessment.
- Obtain signed authorization for approved security testing on targeted systems within scope. Make sure to specify the IP addresses or address ranges, device hostnames, applications, whether software, mobile application or web application, the physical location of the client's site, any cloud assets or resources, and **Application Programming Interfaces** (**APIs**) if any.
- Specify the testing types and methodology/framework. Some clients may require ethical hacking services to meet compliance and auditing requirements. Therefore, it's important to ask your clients why they are interested in performing a security assessment. For instance, ask them whether it is for compliance, auditing, proactive testing, or other purposes.
- Ensure the scoping specifies any restrictions during the security assessment such as no social engineering or DoS attacks.
- Escalation and communication procedures, such as primary and secondary contact should an anomaly occur, and update frequency.
- Details about reporting and deliverables based on the client's needs and expectations.

These are the additional factors to consider when creating a scoping checklist:

- Signed rules of engagement

- Signed letter of authorization
- In-scope items and out-of-scope items
- Security testing dates and permitted time (include the time zone)
- Permitted techniques and restrictions
- Contact list, including an escalation path if something should go wrong
- Data handling and privacy policy such as **Non-Disclosure Agreement** (**NDA**) and confidentiality statement
- Reporting format
- Re-testing rules
- Insurance and liability clauses

This process also helps the ethical hacker or penetration tester to determine whether there are any limitations or restrictions during the security testing. Sometimes, when getting your first ethical hacking or penetration testing job, you may want to hack everything on the client's network and those actions can create legal issues. Hence, obtaining legal permission from the system owner/client, identifying the scope, and staying within the scope is important. There's a fine line between hacking and ethical hacking, and that's obtaining legal permission from the system owner for the in-scope items.

Now that you understand ethical hacking concepts and its importance for organizations, let's learn the fundamentals of information security controls.

Information security controls

Information security controls are commonly used to prevent or mitigate cyber-risks within an organization. They are used to ensure confidentiality, integrity, and availability of assets are uninterrupted by adverse effects such as cyber-attacks or threats. Without information security controls, cybersecurity professionals won't be able to efficiently defend their organization from both internal and external cyber-risks. By implementing a multi-layered approach of security controls, organizations can reduce the potential impact of a real cyber-attack or threat if it were to occur.

Understanding the fundamentals of information security controls is essential for aspiring cybersecurity professionals, as it helps you to better recommend and apply targeted security controls to protect and safeguard assets from specific cyber-attacks and threats.

The following are various categories of information security controls:

- **Technical** – These are the electronic and digital controls that are implemented by the cybersecurity/network security team to reduce the risk of a cyber-attack. For instance, a

network security engineer would implement and configure next-generation firewalls to efficiently filter inbound and outbound network traffic. Therefore, preventing unwanted traffic from entering the network and ensuring users are not intentionally or unintentionally leaking sensitive data outside the network.

- **Managerial** – These are the administrative controls such as the information security policies, plans, and procedures that are created by the **Chief Information Security Officer** (**CISO**). This type of control focuses on managing both employee behavior and providing governance for implementing security controls on endpoints and the network infrastructure. For instance, the CISO can create a Supplier Relationship policy that's used to manage cyber-risks that are introduced by a third-party supplier/vendor who has access to the organization's network infrastructure.
- **Operational** – Operational controls are people-focused. They are implemented and executed by the employees rather than relying on technical controls to prevent risk within the organization. For instance, an organization can implement and enforce cybersecurity user awareness training for all employees. This training would help employees to better identify cyber-risks and best practices to mitigate those risks. Additionally, organizations can develop **Standard Operating Procedures** (**SOPs**) to ensure employees follow a set of guided procedures to achieve a specific outcome or goal.
- **Physical** – This type of security control is implemented to safeguard physical assets of an organization, such as networking devices, security appliances, servers and anything that has value to the organization. For instance, organizations can install fences to safeguard the compound, doors and locks to ensure unauthorized access is prohibited to secure areas and even employ security guards to patrol and monitor the compound.

Types of security controls are used to describe the goal of implementing them. This includes answering the question of does an information security control protects or detects a cyber-attack against an asset? The following are common types of information security controls:

- **Preventive** – Preventative security control is designed and implemented to proactively stop a cyber-attack or threat from materializing or occurring. Cybersecurity professionals can implement and configure an email security solution to filter inbound emails within an organization with the intent to prevent email-based threats such as phishing campaigns.
- **Deterrent** – A deterrent security control is designed to discourage a threat actor from performing a cyber-attack on a targeted system. This reduces the risk of a cyber-attack or threat. However, it is important to understand that this control does not prevent or mitigate a cyber-attack or threat; rather, it only discourages the threat actor. For

instance, warning messages or banners displaying legal notice about unauthorized access on a system, or warning signs displayed within a building about the presence of surveillance cameras, may discourage someone from performing a malicious activity.

- **Detective** – This type of control is used to identify and alert the cybersecurity team of security incidents on systems and network. For example, an organization can implement a **Security Information and Event Management** (**SIEM**) solution that's configured to analyze incoming security logs from devices on a network and alert the cybersecurity team at pre-defined thresholds.
- **Corrective** – A corrective control is implemented to fix or mitigate cyber-attacks and threats on a system or network. For instance, cybersecurity professionals can configure specific rules on a firewall to prevent outbound network traffic from the entire company to stop malware or a threat actor from exfiltrating data from the network.
- **Compensating** – This type of security control is implemented to help organizations recover from a cybersecurity incident. For instance, data backup and restoration policies and tools can help an organization restore systems back to an acceptable working state after a ransomware attack. This type of control does not prevent a cyber-attack from occurring or mitigate one.
- **Directive** – This type of security control focuses on meeting desired goals such as ensuring an organization is compliant with industry frameworks and regulatory standards. For instance, CISO may be leading an organization to become ISO 27001 certified over the upcoming months.

Each category and type of information security control has its role and function to help an organization to reduce cyber-risks, and protect the confidentiality, integrity, and availability of assets, while improving the overall security posture of the organization. Now that you understand the role of information security controls in managing cyber risk and protecting organizational assets, let's explore the laws, frameworks, and industry standards that help organizations implement and maintain effective security practices.

Information security laws and standards

Information security laws are the legal frameworks created by various governmental organizations around the world that mandate various organizations to ensure their applications, systems, and networks protect data from unauthorized access and disclosure, and systems from harm or damages. In addition, these laws are designed to ensure data compliance with data privacy and protection. The goal of these information security laws is to reduce the risk of cyber-crime, leakage or disclosure of **Personally Identifiable Information** (**PII**), and establish and maintain secure data handling processes. For instance, the **General**

Data Protection Regulation (**GDPR**) is a regulation that defines how organizations must process the personal data of individuals located in the **European Union** (**EU**).

> **Note**
>
> Keep in mind, ethical hackers need to understand these laws, not implement compliance programs.

Information security standards are industry frameworks that provide guidelines for ensuring data privacy and securing applications, systems and networks. These are established guidelines to reduce the risk of unauthorized access, information disclosure, and even threat actors compromising the confidentiality, integrity, and availability of a system. Unlike information security laws created by a governmental body and that are legally binding to citizens and organizations, information security standards and frameworks are developed and maintained by regulatory bodies, industry groups, and even international standards organizations to ensure data is protected and kept safe across multiple sectors.

In the following sub-sections, you'll learn about the importance of various information security laws and standards, and the types of industries they're applicable in.

Payment card industry data security standard (PCI DSS)

PCI DSS is an industry standard that is applicable to any organization that handles payment card data, such as transmitting or storing credit cards, debit cards, prepaid cards, point-of-sale systems, or even **Automated Teller Machines** (**ATMs**). This standard helps organizations to design and maintain a secure network infrastructure for the handling of payment card data, protect the privacy of cardholders' information and their card details, implement access control measures, continuously test for security vulnerabilities and monitor for anomalies, and maintain an information security program.

The PCI DSS provides the following security assurance to organizations:

- Building and maintaining secure network architecture to ensure secure baselines and configurations are applied
- Protecting cardholders' data from unauthorized users and malicious actors
- Maintaining a vulnerability management program to ensure systems and applications are regularly updated
- Implementing strong access control measures to ensure only authorized users or devices can access the data on a need-to-know basis
- Performing regular monitoring and testing of networks
- Maintaining an up-to-date information security policy

For organizations that are required to be PCI DSS compliant, failing to meet the security requirements of this standard can potentially result in fines and even halt the organizations from processing payment cards.

ISO/IEC 27001

The **Organization for Standardization** (**ISO**) and the **International Electrotechnical Commission** (**IEC**) develop and maintain the 27001 standard which outlines how organizations should establish, implement, and maintain an information security management system within their organization. This industry standard is commonly used within many organizations around the world to create their information security governance program for securing their networks and systems, and their goals.

This security standard helps organizations improve their compliance with local and international laws and regulations for information security best practices and security controls. Furthermore, organizations can improve their management of cyber-risks using a cost-effective approach by defining information security management processes, activities, and implementing security controls to prevent and mitigate cyber-attacks and threats.

HIPAA

The **Health Insurance Portability and Accountability Act** (**HIPAA**) is a United States (US) federal act that's developed to safeguard **Protected Health Information** (**PHI**) of a person. This security act specifies the type of physical, technical, and administrative controls needed to safeguard subjects (persons) and businesses that are associated to use the PHI securely.

The following are common examples of PHI:

- The patient's names
- The patient's address
- The patient's date of birth
- Telephone number
- Email address
- Social Security Number (SSN)
- Medical record numbers
- Biometric information

While many healthcare providers use computing systems and networks to improve their processes, it's important to safeguard patients' data from unauthorized access, misuse, cyber-threats, and so on.

SOX

The **Sarbanes-Oxley Act** (**SOX**) is another US federal law that's designed to safeguard anyone in the public from financial fraud by corporations. This law focuses on proving both the reliability and accuracy of financial information and ensures the **Chief Executive Officer** (**CEO**) and/or the **Chief Financial Officer** (**CFO**) are accountable for the financial reports or corporate disclosure of the organization.

These laws protect the investors of an organization by ensuring the financial information is accurate and remains transparent, improving the accountability of the CEO and CFO of the organization, and improving the corporate governance to build trust of the public within the financial industry.

DMCA

The **Digital Millennium Copyright Act** (**DMCA**) is a US copyright law that is designed to prevent the illegal distribution of works by anyone other than the sole owner or those authorized to distribute them. For instance, this law helps protect the work of movies, writings, music, and others that are copyrighted.

FISMA

The **Federal Information Security Management Act** (**FISMA**) is a framework that helps federal agencies implement and maintain effective information security controls. This framework enables federal agencies to perform the following:

- Create standards for categorizing their data and information systems based on business impact
- Set standards for minimum security requirements of systems
- Implement guidelines for choosing the appropriate security controls for systems and networks
- Implement guidelines for testing the effectiveness of security controls
- Implement guidelines for ensuring authorization of the best practices for information systems

With this section, you have learned about communication information security laws and frameworks and have understood how and why they are used in various industries.

Summary

In this chapter, you've learned about the common elements of information security such as confidentiality, integrity, and availability. You were also introduced to ethical hacking and penetration testing methodologies and frameworks such as the Cyber Kill Chain, the role of

TTPs in ethical hacking, OWASP security testing guides, and so on. Furthermore, you learned about the various classes of hackers and their common motives for performing a cyber-attack on a target.

Additionally, the chapter also expanded on the different phases of ethical hacking and the Cyber Kill Chain by providing the essential phases used by ethical hackers and penetration testers during their security assessments on targeted systems and networks. Furthermore, TTPs are commonly used by cybersecurity professionals to better understand how threat actors can compromise their targets, and adopting such techniques and procedures helps ethical hackers to improve their skillsets and identify new and hidden security vulnerabilities.

Lastly, we covered how ethical hackers focus on using real-world tools and techniques of threat actors. It's important to remember that ethical hackers also help organizations resolve security vulnerabilities by providing recommendations on countermeasures to reduce cyber-risks while improving the security posture of organizations.

In the next chapter, you will learn how to set up a virtually isolated environment to safely practice your new skills in this book.

Further reading

The following are additional reading resources to help reinforce your learning:

- NIST CSF 2.0 Framework - `https://www.nist.gov/cyberframework`
- The Cyber Kill Chain - `https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html`
- Mitre ATT&CK framework - `https://attack.mitre.org/`
- OWASP Web Security Testing Guide - `https://owasp.org/www-project-web-security-testing-guide/`
- OWASP Mobile Application Security - `https://owasp.org/www-project-mobile-app-security/`
- OWASP Firmware Security Testing Methodology - `https://github.com/scriptingxss/owasp-fstm`
- Penetration Testing Execution Standard (PTES) - `http://www.pentest-standard.org/index.php/Main_Page`
- PCI Penetration Testing Guide - `https://www.pcisecuritystandards.org/standards/pci-dss/`

- NIST 800-115 Security Testing and Assessment - `https://csrc.nist.gov/pubs/sp/800/115/final`
- Open Source Security Testing Methodology Manual - `https://www.isecom.org/research.html`

2

Setting up an Ethical Hacking Lab

As an aspiring ethical hacker, the feeling of excitement to identify and exploit security vulnerabilities on systems and networks never gets old. However, it's essential to set up a lab environment that operates as a safe place to practice your skills. For instance, if you're to test your skills in an organization's network without legal permission and cause harm or damage to their assets, you'll be legally responsible. Therefore, it's highly recommended to hone your skills on systems and networks you own. Hence, creating your very own lab environment can be both a personal project on learning how to deploy vulnerable systems and setting your ethical hacking lab.

In this chapter, you will learn the fundamentals of building a virtually isolated ethical hacking lab to hone your skills in a safe environment. You will understand the importance of each component, recommended system requirements to ensure everything works well and supports scalability for future expansion. Furthermore, you will learn how to create a virtual network, deploy Kali Linux as an attacker machine, set up vulnerable machines, and a wireless network architecture to support learning.

> **Note**
>
> This ethical hacking lab will be used throughout the book to perform scanning, exploitation, and post-exploitation in a safe and controlled environment.

In this chapter, you will cover the following topics:

- Ethical hacking lab design
- Setting up a virtualized environment

- Setting up attack machines
- Setting up vulnerable machines
- Setting up a vulnerable wireless lab

Technical requirements

To follow along with the exercises in this chapter, please ensure that you have downloaded the following hardware and software requirements:

- Oracle VirtualBox - `https://www.virtualbox.org/`
- Kali Linux - `https://www.kali.org/`
- Vagrant - `https://developer.hashicorp.com/vagrant/install`
- Metasploitable 2 - `https://www.rapid7.com/products/metasploit/metasploitable/`
- Metasploitable 3 - `https://portal.cloud.hashicorp.com/vagrant/discover/rapid7`

Whether you're using a personal laptop or a desktop computer for your lab environment, the following are the recommended system requirements for your host computer:

- 4-core processor or better
- 16 GB RAM or better
- 500 GB of free storage or better

Due to many reasons, such as budget, availability, and liability, there isn't a specific brand or model of laptops or desktop computers that I can recommend. However, if you're looking to purchase a new computer for learning cybersecurity, it's important to choose a model that's within your budget and personal preference.

Before getting started with installing the hypervisor, please ensure you complete the following prerequisites before proceeding:

- Ensure your processor supports **VT-x/AMD-V**. These are hardware-assisted technologies that support virtualization.
- Ensure the virtualization feature is enabled on your processor via the **Unified Extensible Firmware Interface** (**UEFI**). Please use the following link to enable virtualization on a Microsoft Windows device: `https://support.microsoft.com/en-`

`us/windows/enable-virtualization-on-windows-c5578302-6e43-4b4b-a449-8ced115f58e1`.

- There's a software issue that prevents Oracle VirtualBox from powering on any virtual machine if the user installs an upgrade over an older version. Therefore, if you are already using a previous version of Oracle VirtualBox, it's recommended to first uninstall it before installing the current version. In other words, perform a clean installation of the latest version of Oracle VirtualBox.

Note

While this book uses a specific version of Oracle VirtualBox (7.2.4), you can download and install the latest version on your device.

To run the WiFiChallenge virtual environment on your host device, you'll need the following requirements:

- 2 CPU cores
- 4 GB of memory (RAM)
- 1 network adapter with internet connectivity

The setup procedures will be covered throughout this chapter for creating a virtual network, and setting up attack and vulnerable machines within a virtualized lab environment.

Ethical hacking lab design

An ethical hacking lab is a fun playground that enables both aspiring and seasoned ethical hackers to learn and improve their skillset as a cybersecurity professional. There are many resources which you can use to create your own personalized lab environment. For instance, you can use a set of physical systems such as older computers and re-purpose them as targeted/vulnerable machines, networking devices such as routers and switches to interconnect these machines, and even use a dedicated desktop/laptop/mini PC as your attacker machine. With access to the internet and creativity, the possibilities of a lab environment are endless for each person.

Having a physical lab environment enables you to better visualize and interact with physical systems within your lab but it comes with some disadvantages and considerations:

- If you don't have older systems for re-use, then a budget may be needed to acquire used or refurbished devices for personal use
- Physical space for storing each device
- May affect your electrical power consumption

For many aspiring cybersecurity professionals such as ethical hackers, having a single device such as a laptop or a desktop computer is mostly all you need to get started. Hence, this chapter focuses on teaching you how to get started building a virtualized lab environment on your personal computer that supports scalability and enables you to further expand and customize the environment in the future. Having a virtualized lab environment on your personal computer enables customization based on your creativity and the available computing resources on your personal computer. It also enables you to practice and develop your skills in a safe and isolated environment, ensuring you do not accidentally harm systems you do not own.

To create a virtualized lab environment, there's a list of components that all need to work together to achieve the desired outcome, that is, an ethical hacking lab environment with vulnerable machines, attacker machines, and network connectivity.

The following is a list of each component and their role within the lab environment:

- **Hypervisor** – The hypervisor, Oracle VirtualBox, is an application which enables users to create virtual machines. A virtual machine allows a user to allocate computing resources from a physical computer, known as the host, to a virtualized container with an operating system. However, the hypervisor does not pre-install the operating system within the virtual machine; this is done by the user. Additionally, there's VMware Workstation Pro and VMware Fusion Pro as alternative hypervisors.
- **vSwitch** – The vSwitch is a common term for describing a virtual network switch that's within a hypervisor. The vSwitch provides similar functionality as a physical network switch such as interconnecting the virtual machines with the hypervisor application. In Figure 2.1, these are the logical connections between each virtual machine within Oracle VirtualBox.
- **Attacker machines** – The attacker machines are the specific virtual machines that'll be used by you, the ethical hacker to perform security testing on vulnerable machines. In this lab design, Kali Linux will function as the attacker machine.
- **Vulnerable machines** – The vulnerable machine is a pre-built machine that is commonly used for demonstrating and learning about security vulnerabilities on operating systems and networks. Metasploitable 2 and 3 will be set up as vulnerable machines within the lab environment.
- **Internet** – Internet connectivity is needed by Kali Linux for acquiring additional software tools , if needed.
- **vNIC** – Each virtual machine can be allocated one or more **virtual Network Interface Card** (**vNIC**). This enables the virtual machine to interconnect with one or more vSwitches or directly to your physical network.

In a typical attack scenario, the attacker machine (Kali Linux) sends traffic through the virtual network (vSwitch) to the vulnerable machines, simulating how attackers interact with targets in real environments.

Figure 2.1 shows the architecture/topology of the ethical hacker lab environment:

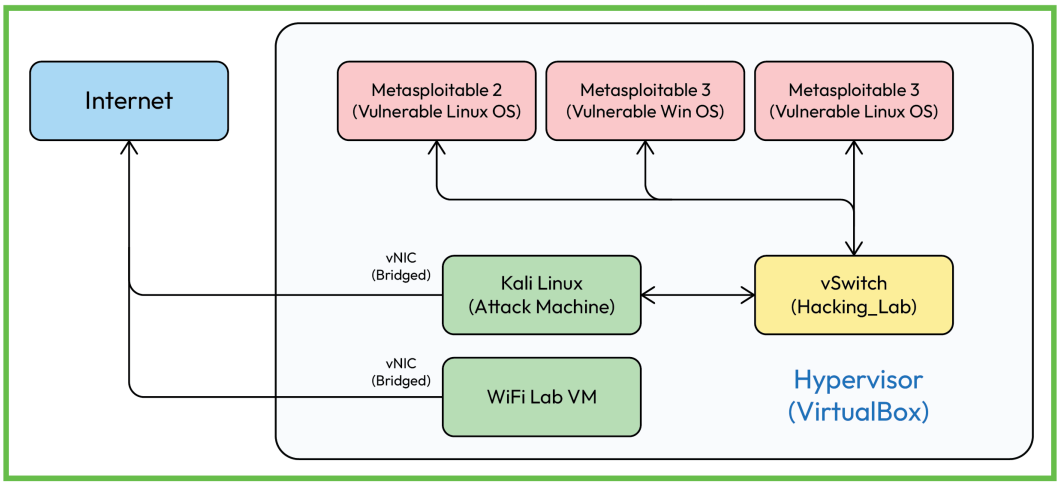


Figure 2.1 – Ethical hacking lab design

As shown in the preceding diagram, the attacker machine will be a **Kali Linux** virtual machine. Kali Linux is the preferred operating system used by many ethical hackers and penetration testers within the cybersecurity industry. It is a Linux-based operating system with hundreds of pre-installed tools to help cybersecurity professionals perform security testing on targeted systems, networks, and applications.

The vulnerable machines will be **Metasploitable 2** and **Metasploitable 3**. The Metasploitable 2/3 virtual machine is a vulnerable system that's intentionally built with security vulnerabilities to help cybersecurity professionals learn how to identify security flaws on systems. Metasploitable 3 enables users to create both Windows- and Linux-based virtual machines.

Additionally, you will set up a specialized virtual machine that will simulate common vulnerable wireless networks. This virtual machine will reduce the need for specialized wireless routers and applying multiple custom configurations to simulate real-world wireless environments.

Lastly, **Oracle VirtualBox** will be the preferred Type 2 hypervisor for the lab environment. Type 2 hypervisors enable you to install them on top of your current operating system as typical software applications. Therefore, you won't need to change any of your current settings or configurations on your personal device.

Note

Oracle VirtualBox is the preferred hypervisor because it's free and easy to use. However, there are other free Type 2 hypervisors such as ***VMware Workstation Pro***. However, at the time of writing, you'll need to create an account on the VMware website using an email address that belongs to a custom registered domain; the website does not support email addresses from free email providers. To download VMware Workstation Pro, please see the following link: `https://www.vmware.com/products/desktop-hypervisor/workstation-and-fusion`.

Figure 2.2 shows the deployment model of a Type 2 hypervisor:

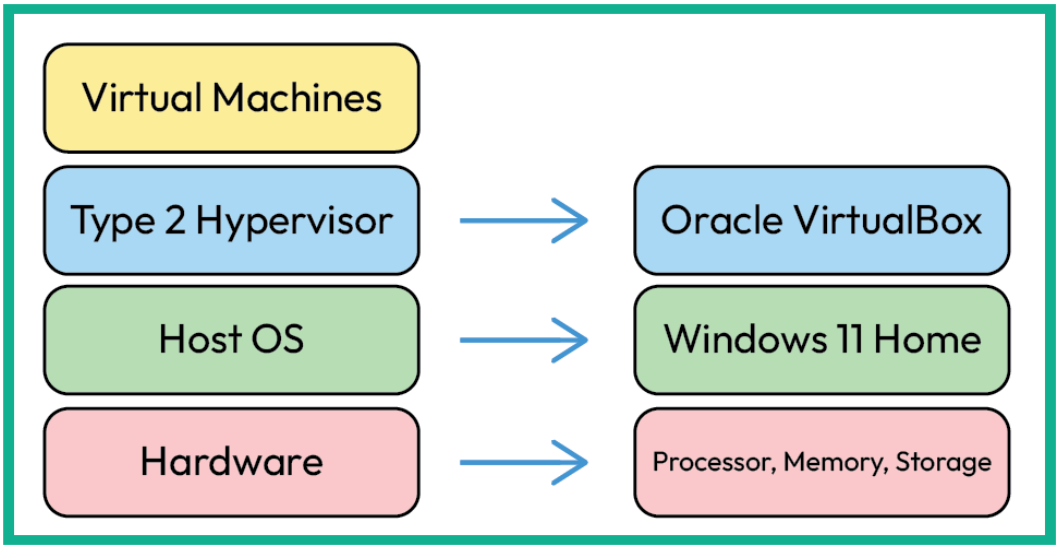


Figure 2.2 – Hypervisor deployment model

As shown in the preceding diagram, the hypervisor operates between the host operating system of a computer and the virtual machines. The host operating system has direct access and precedence in utilizing the hardware components, while the remaining resources are available to the virtual machine via the hypervisor.

If you're using a laptop computer, your entire lab environment will be mobile too. For instance, if you carry your laptop to school, work, or the coffee shop, the lab environment will be with you. This does not mean you need a laptop to build this environment; it's simply an additional benefit for users with laptops. However, the hardware components (such as power

requirements) on a desktop computer are usually more powerful than those on mobile devices such as laptops.

Note

There are Type 1 and Type 2 hypervisors. Type 1 hypervisors are commonly referred to as bare-metal hypervisors as they are installed directly on the storage drive of a server and boot as an operating system. However, Type 2 hypervisors are installed on top of a host operating system, such as installing Oracle VirtualBox onto your existing operating system. Each type of hypervisor has its advantages. To learn more about the various types of hypervisors, please see: `https://www.techtarget.com/searchitoperations/tip/Whats-the-difference-between-Type-1-vs-Type-2-hypervisor`.

Each virtual machine will require its own virtual processor, virtual memory, and virtual storage. During the creation of a virtual machine, the hypervisor enables you to allocate computing resources from your host computer to the virtual machine. Therefore, when a virtual machine is powered on, it will use the allocated computing resources from the host, which enables it to run like a full-fledged computer within an existing computer.

Figure 2.3 shows the Kali Linux operating system running as a virtual machine on a computer with Windows 11 as the host operating system:

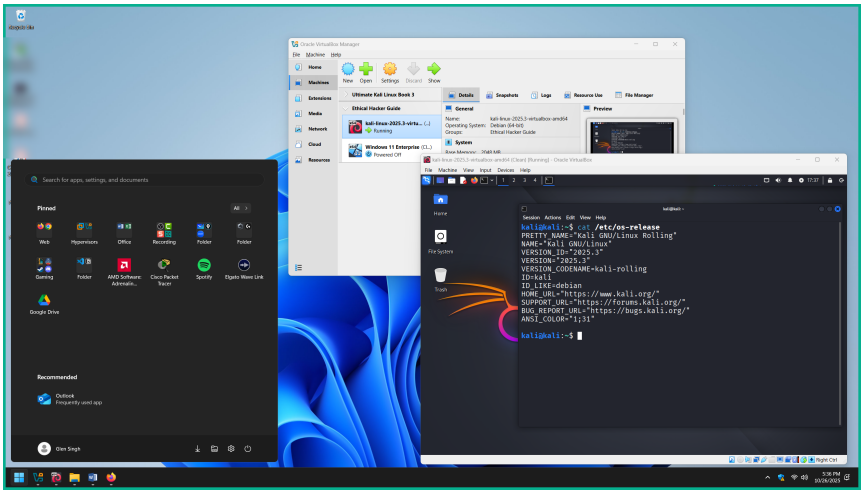


Figure 2.3 – Running a virtual machine on a host computer

As mentioned, the hypervisor enables you to allocate and easily customize the number of cores for the virtual processor and virtual memory for each virtual machine. Therefore, each virtual machine does not need the same allocation of resources; however, if you plan on running multiple virtual machines at the same time, it's important to consider increasing the amount of **Random Access Memory** (**RAM**) on your host computer. Having 16 GB of RAM will be sufficient on your host computer, since no more than 2 virtual machines will be running at the same time.

Now that you understand the lab design, requirements, and the role of each component, the next step is to set up the virtual environment and network using Oracle VirtualBox.

Setting up a virtualized environment

This section will provide you with the procedures and hands-on instructions for installing Oracle VirtualBox as a Type 2 hypervisor on your host operating system. In addition, you will learn how to create a virtual network using the *VBoxManage* component of Oracle VirtualBox. This virtual network will be used for interconnecting the attacker machines to the vulnerable machines within the virtualized environment.

Once you're all set and ready, it's time to install the hypervisor on a Windows-based host device. However, if you're using Ubuntu Desktop as your host operating system, consider jumping to *Part 3 – (Ubuntu users only) Installing Oracle VirtualBox and creating a virtual network.*

Part 1 – Installing the hypervisor

To get started with setting up the hypervisor, please use the following instructions:

1. To download the **Oracle VirutalBox** hypervisor application on a Microsoft Windows host, go to `https://www.virtualbox.org/wiki/Downloads` and select the **VirtualBox Platform Package** for your host operating system:



Figure 2.4 – VirtualBox platform packages

Once you've selected your preferred platform package, the download will begin automatically on your computer. Be sure to note the download location, as the file may be saved to a custom location depending on your browser settings.

2. Next, while you're on the same downloads page for Oracle VirtualBox, download the **VirtualBox Extension Pack**:

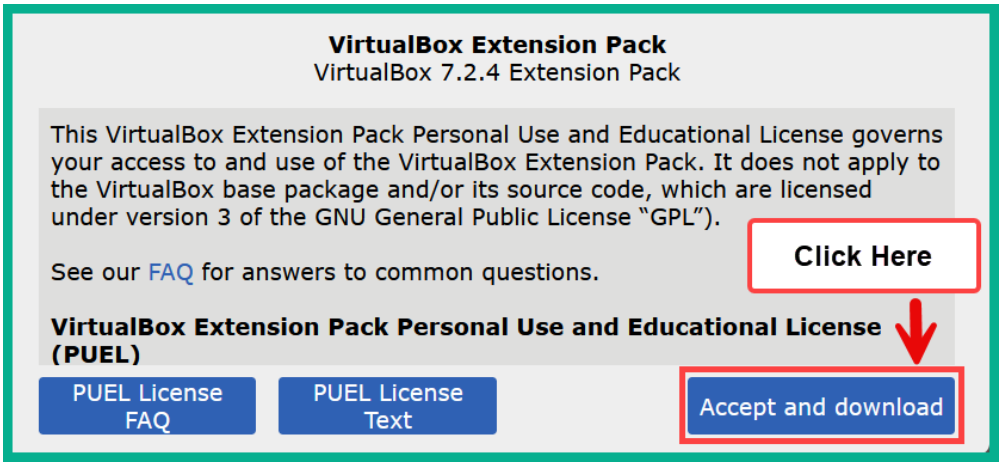


Figure 2.5 – VirtualBox Extension Pack

Once you've clicked on the **Accept and download** button, the download will automatically begin on your device. Ensure you save the file within the same directory (location) as the Oracle VirtualBox installer during *step 1*.

> **Note**
>
> The VirtualBox Extension Pack provides additional functionality to virtual machines within Oracle VirtualBox.

3. To start the installation of **Oracle VirtualBox**, simply double-click to launch the installation wizard. During the installation, ensure the default settings are applied and click on **Next** until the installation is completed. After the Oracle VirtualBox installation is completed, launch the application:



Figure 2.6 – Oracle VirtualBox Manager

4. Next, select **Extensions** and click on **Install** as shown:

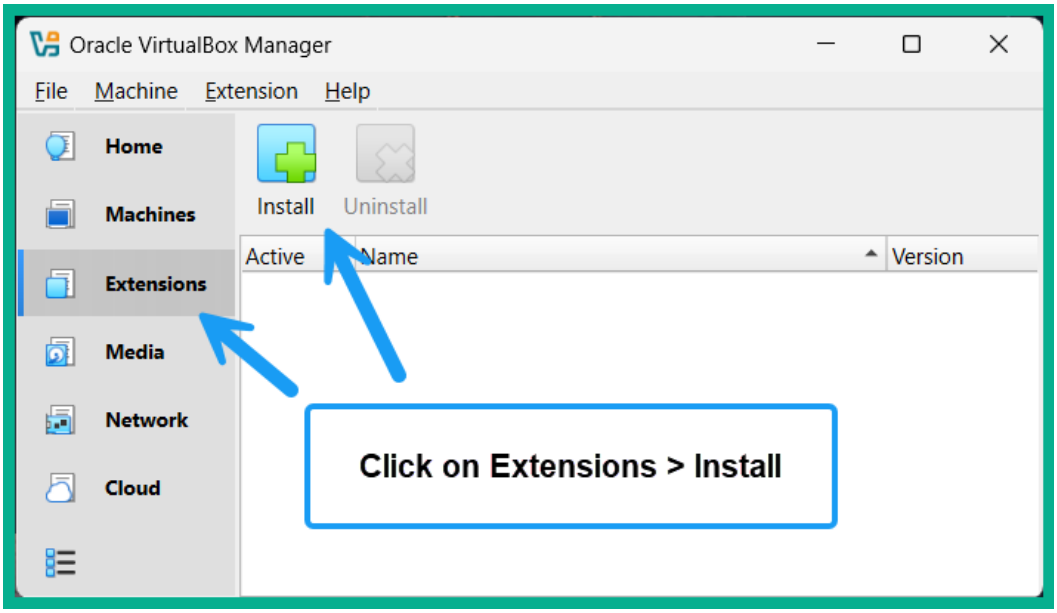


Figure 2.7 – Extensions menu

5. You will then see the **Select an extension package file** window. Navigate to the download directory of the **VirtualBox Extension Pack** file, then select it and click on **Open**:

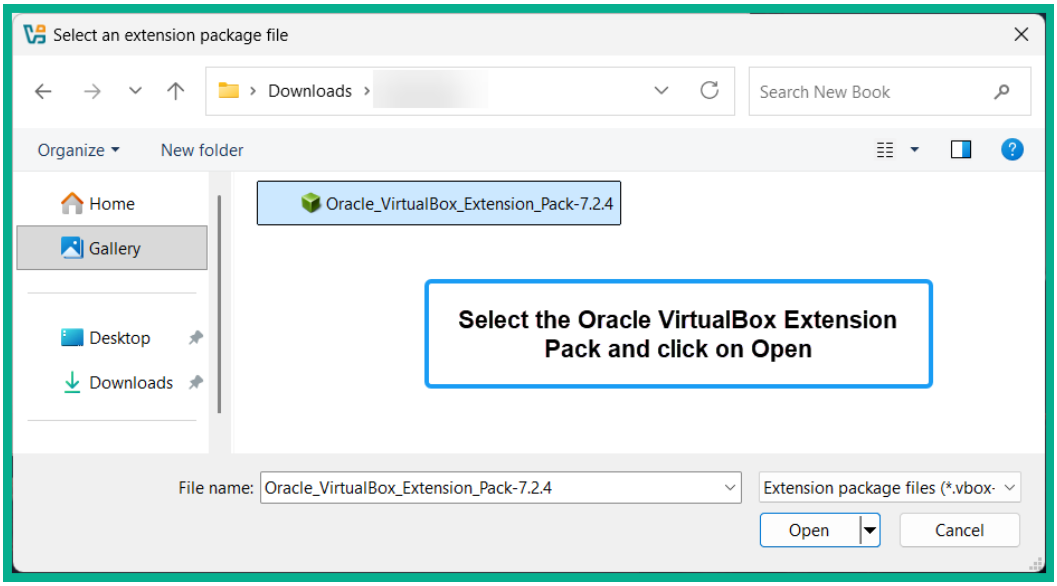


Figure 2.8 – Selection window

6. The **VirtualBox – Question** window will appear next, click on **Install**:

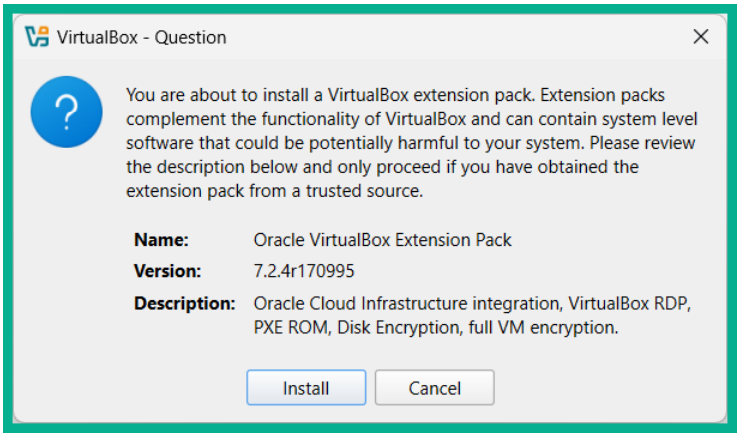


Figure 2.9 – VirtualBox question window

7. Next, the **VirtualBox License** window will appear with the **VirtualBox Extension Pack Personal Use and Educational License (PUEL)**, click on **I Agree** to accept and

proceed with the installation. Once the installation is completed, the **VirtualBox Extension Pack** will be active, as shown:

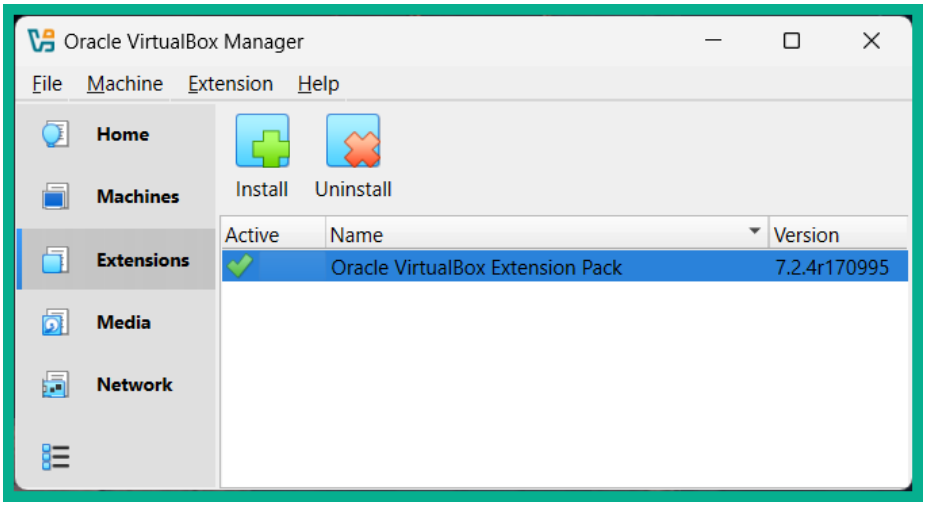


Figure 2.10 – Verifying VirtualBox Extension Pack

Once the installation has completed successfully, close the **VirtualBox Manager** until it's needed during the later sections of this chapter. Next, you will learn to create a virtual network for interconnecting each virtual machine within the lab environment.

Part 2 – Creating a virtual network

The Oracle VirtualBox hypervisor enables users to create virtual, isolated networks for interconnecting virtual machines. This means, all virtual machines that are connected to this virtual network will be able to communicate with each other only and not with systems outside the virtual network.

The following diagram shows that each virtual machine, such as Kali Linux, Windows 11, Metasploitable 2, and Metasploitable 3, is connected via a virtual switch known as vSwitch, as shown:

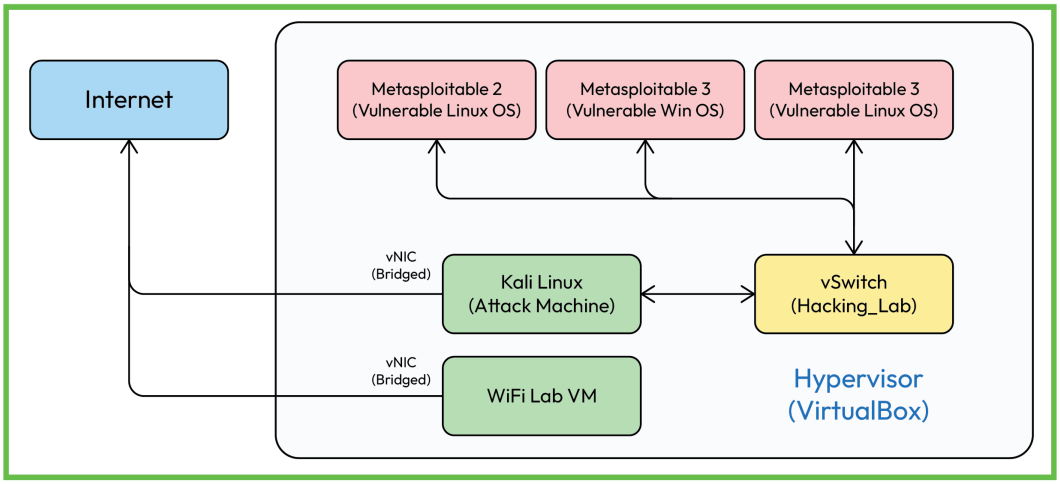


Figure 2.11 – Virtual lab design

To get started with creating a virtual network within the hypervisor, please use the following instructions:

1. On your Windows host computer, open the Command Prompt application.
2. Use the following command to change the present working directory to `C:\Program Files\Oracle\VirtualBox`:

```
cd C:\Program Files\Oracle\VirtualBox
```

3. Use the following command to create a new virtual network with the name `Hacking_Lab`, a virtual Dynamic Host Configuration Server with the IP address `10.10.10.1`, a network range of `10.10.10.20` – `10.10.10.100` and a subnet mask of `255.255.255.0`:

```
VBoxManage dhcpserver add --network=Hacking_Lab --server-ip=10.10.10.1 --
lower-ip=10.10.10.20 --upper-ip=10.10.10.100 --netmask=255.255.255.0 --
enable
```

The preceding configuration creates an isolated internal network with its own IP address range, ensuring all attack traffic stays within the lab environment. The following screenshot shows the execution of the preceding command:

```
Microsoft Windows [Version 10.0.26100.1742]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Glen> cd C:\Program Files\Oracle\VirtualBox

C:\Program Files\Oracle\VirtualBox> VBoxManage dhcpserver add --network=Hacking_Lab --server-ip=10.10.
10.1 --lower-ip=10.10.10.20 --upper-ip=10.10.10.100 --netmask=255.255.255.0 --enable
```

Figure 2.12 – Creating a virtual network

4. Lastly, the `VBoxManage list dhcpservers` command enables you to view all existing virtual networks within Oracle VirtualBox, as shown in *Figure 2.13*:

```
C:\Program Files\Oracle\VirtualBox> VBoxManage list dhcpservers
NetworkName:    Hacking_Lab
Dhcpd IP:       10.10.10.1
LowerIPAddress: 10.10.10.20
UpperIPAddress: 10.10.10.100
NetworkMask:    255.255.255.0
Enabled:        Yes
```

Figure 2.13 – Listing all virtual networks

With the virtual network now in place, you have learned how to use Oracle VirtualBox components to build a Windows-based lab environment.

Part 3 — (Ubuntu users only) Installing Oracle VirtualBox and creating a virtual network

Please use the following instructions to install Oracle VirtualBox as the hypervisor and create a virtual network on Ubuntu Desktop 24.04.3 LTS:

1. On **Ubuntu Desktop**, open the **Terminal** and use the following command to install Oracle VirtualBox with its extension pack:

```
sudo apt update
sudo apt install virtualbox virtualbox-ext-pack
```

2. Next, use the following command to create the `Hacking_Lab` virtual network:

```
cd /usr/bin
VBoxManage dhcpserver add --network=Hacking_Lab --server-ip=10.10.10.1 --
```

```
lower-ip=10.10.10.20 --upper-ip=10.10.10.100 --netmask=255.255.255.0 --
enable
```

3. Lastly, use the `VBoxManage list dhcpservers` command to verify the `Hacking_Lab` virtual network was created:

```
glen@ubuntu:/usr/bin$ VBoxManage list dhcpservers
NetworkName:    Hacking_Lab
Dhcpd IP:       10.10.10.1
LowerIPAddress: 10.10.10.20
UpperIPAddress: 10.10.10.100
NetworkMask:    255.255.255.0
Enabled:        Yes
Global Configuration:
    minLeaseTime:     default
    defaultLeaseTime: default
    maxLeaseTime:     default
    Forced options:   None
    Suppressed opts.: None
        1/legacy: 255.255.255.0
Groups:               None
Individual Configs:   None
glen@ubuntu:/usr/bin$
```

Figure 2.14 – Verifying network within Ubuntu

You are now ready to set up your attack machines within the hypervisor and attach them to the newly created virtual network.

Setting up attack machines

In this section, you will learn how to set up Kali Linux as the attacker machine and configure its virtual network adapters for accessing the internet and the internal `Hacking_Lab` virtual network.

Part 1 – Setting up Kali Linux

To get started with setting up Kali Linux as an attacker machine, please use the following instructions:

1. On your host computer, go to the official Kali Linux website at `https://www.kali.org/get-kali/` and click on **Virtual machines**:

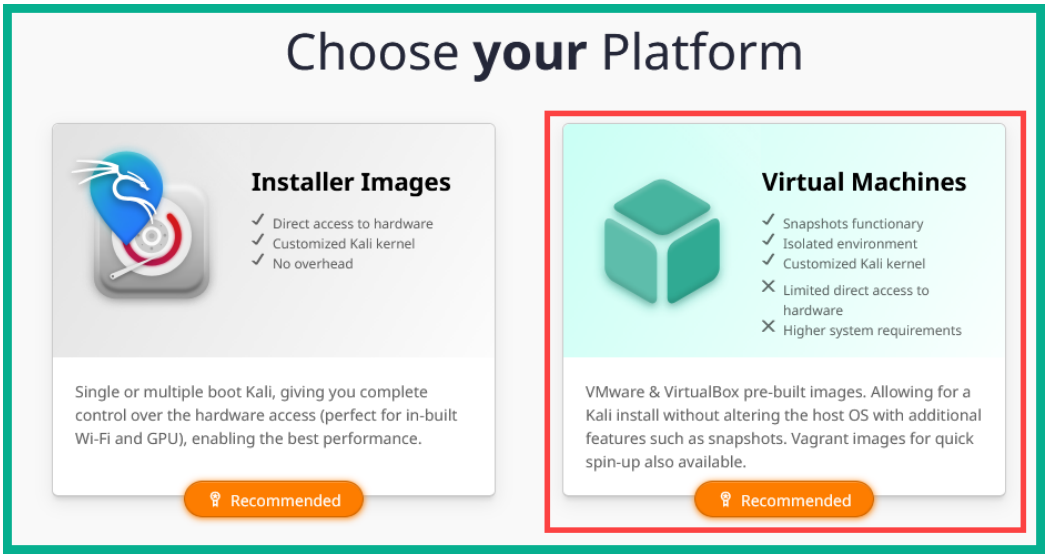


Figure 2.15 – Platform selection

2. Next, check on the **VirtualBox** option to download the latest stable release of the Kali Linux operating system onto your host computer:

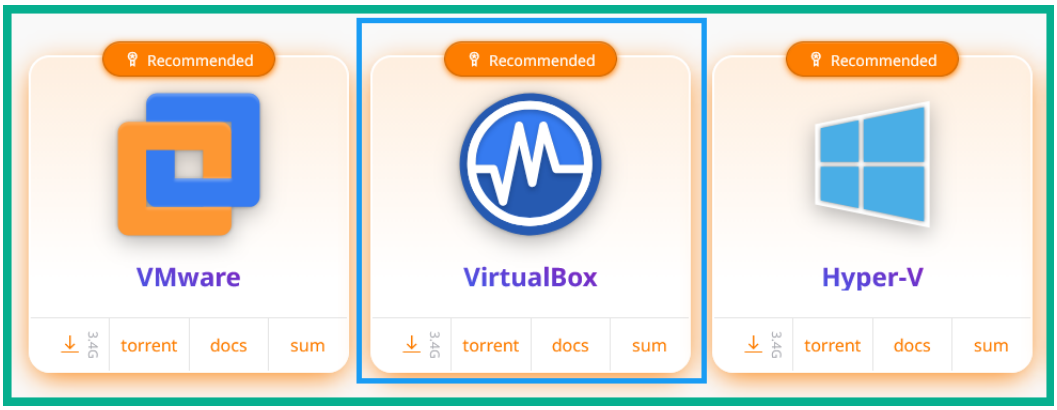


Figure 2.16 – VirtualBox selection

3. Once the download is completed, go to your **Downloads** directory where you will find the compressed file. Select the file, then right-click to open the Windows context menu, and select **Extract All**:

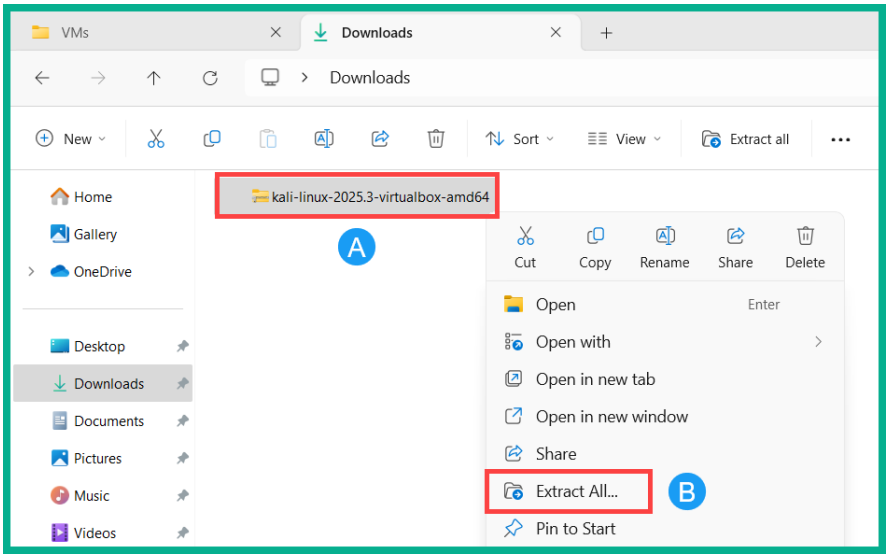


Figure 2.17 – Compressed folder

4. When the **Windows Extract Archive** wizard appears, simply click on the **Extract** button to begin extracting the compressed folder within the present working directory.

> **Tip**
>
> If you're unable to extract the files from the compressed folder, consider downloading, and using 7-Zip, a file compression, and decompression application. The 7-Zip application can be found at: `https://www.7-zip.org/`.

5. To import the pre-built Kali Linux virtual machine, open **Oracle VirtualBox Manager** and select **Open**, as shown in the following figure:



Figure 2.18 – Oracle VirtualBox Manager

6. On the **Select a virtual machine file** window, navigate to the directory with the extracted files and select the **VirtualBox Machine Definition** file and click on **Open**:

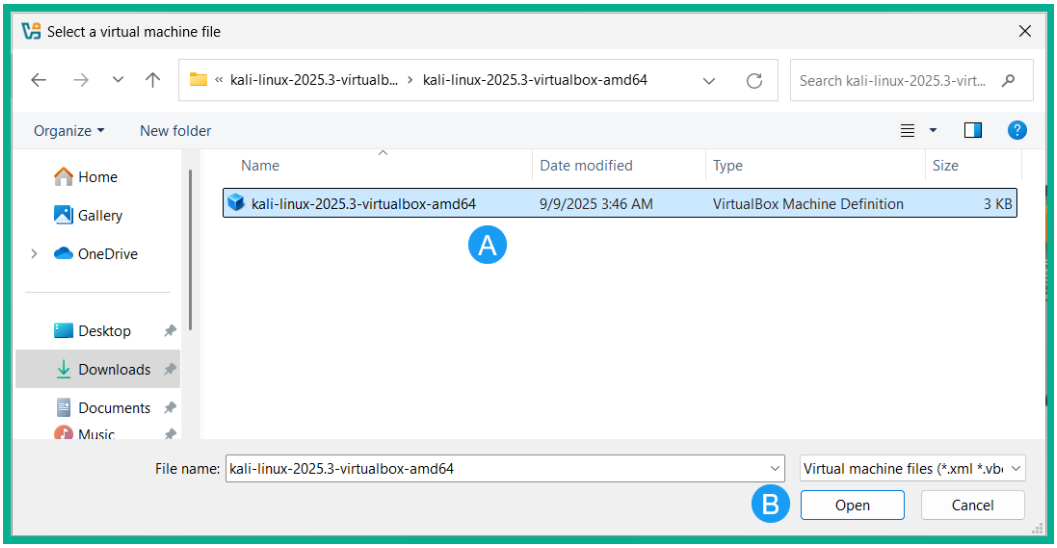


Figure 2.19 – VirtualBox Machine Definition File

7. The pre-built Kali Linux virtual machine will be automatically imported into **Oracle VirtualBox Manager**:

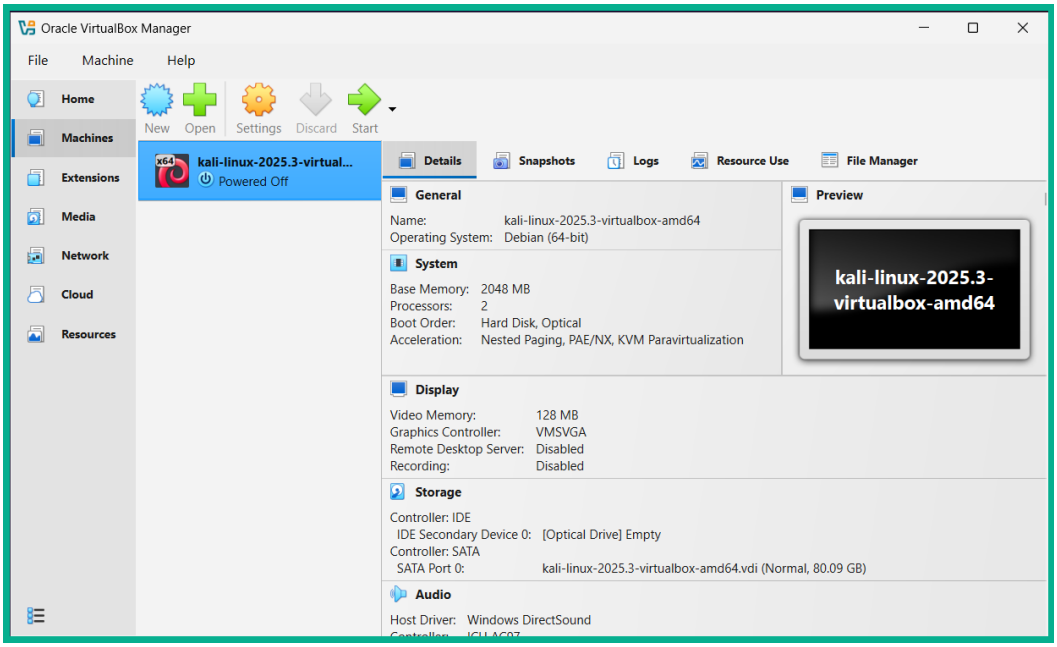


Figure 2.20 – Kali Linux virtual machine

Before powering-on the Kali Linux virtual machine, the virtual network adapters need to be configured correctly for this virtual machine. Therefore, in the next part you will learn how to edit the configurations of a virtual machine.

Part 2 – Configuring the network adapter for Kali Linux

To get started with editing the configurations of the Kali Linux virtual machine, please use the following instructions:

1. On **Oracle VirtualBox Manager**, select the **Kali Linux virtual machine** and click on **Settings**:

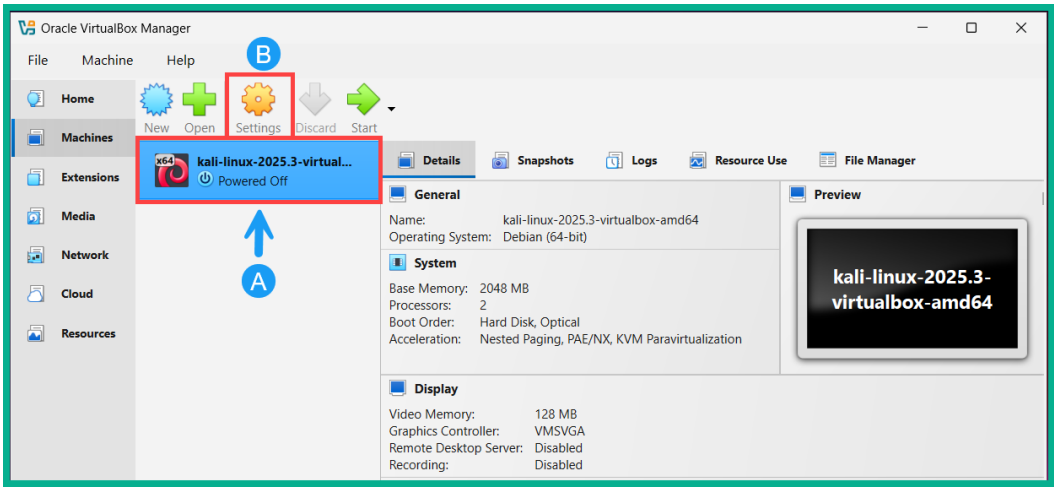


Figure 2.21 – Kali Linux virtual machine settings

2. Next, on the **Settings** window, select **Expert** view, click on **Network**, select **Adapter 1**, and make the following changes:
    - Set **Attached to**: **Bridged Adapter** (This allows Kali Linux to access the internet through the host system, which is necessary for downloading tools and updates.)
    - **Name**: Use the drop-down menu to select your host network adapter that's connected to the internet

    The following screenshot shows an example of the preceding configurations:

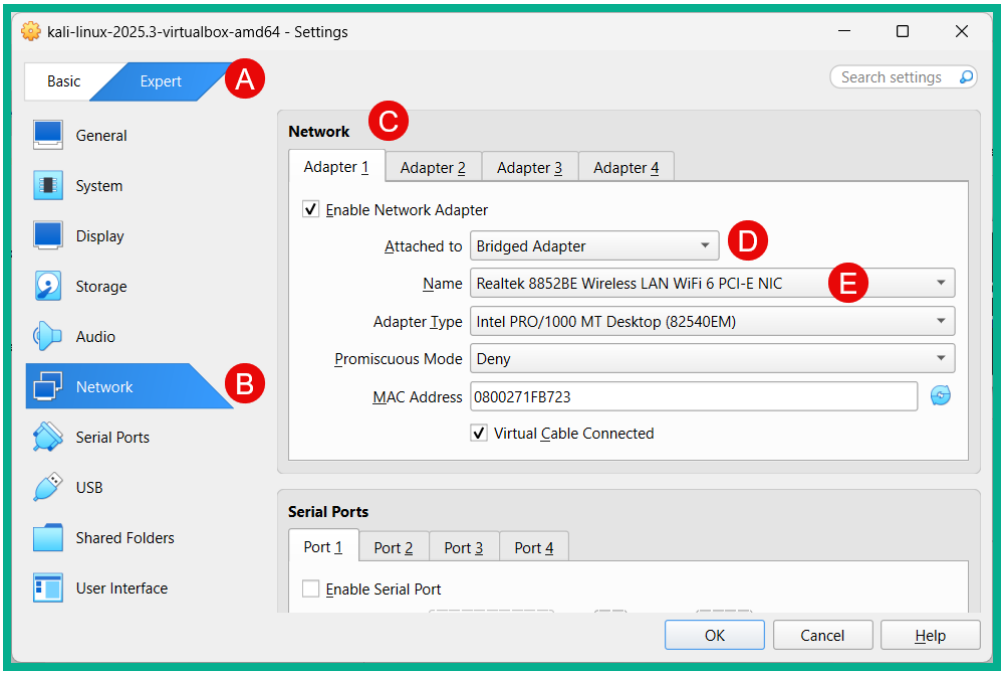


Figure 2.22 – Adapter 1 settings

Tip

If your host device is connected to the internet using a wireless connection, then you should see the name of your wireless adapter within the **Name** drop-down menu. If your host device is using a wired Ethernet connection, then select your physical Ethernet adapter from the drop-down menu.

The preceding settings enable the Kali Linux virtual machine to bridge its vNIC with the physical network adapter on your host computer. This allows the virtual machine to seamlessly access the physical network.

3. Next, to allow Kali Linux to communicate with other virtual machines on the `Hacking_Lab` virtual network, select **Adapter 2** and use the following settings:

    1. Check **Enable Network Adapter**
    2. **Attached to**: **Internal Network** (This ensures attack traffic remains isolated within the lab and does not affect external networks.)
    3. **Name**: Manually enter `Hacking_Lab` within the field

4. **Promiscuous Mode: Allow All**
5. Click **OK** to save the settings of the virtual machine

The following screenshot shows an example of the preceding settings:

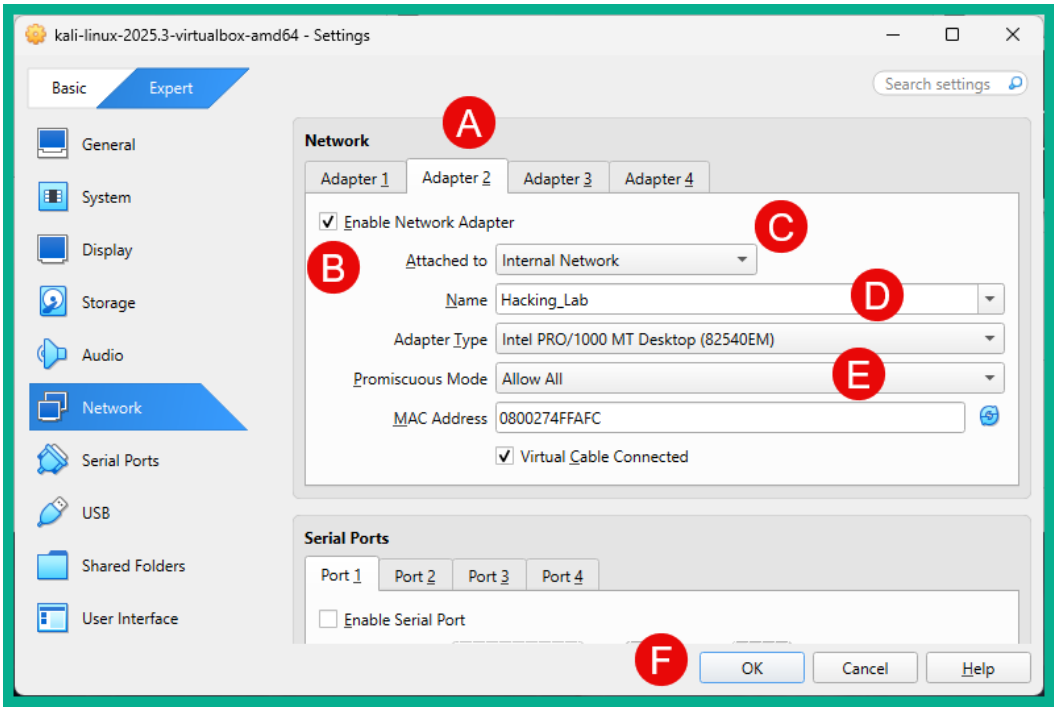


Figure 2.23 – Adapter 2 settings

Once the Kali Linux virtual machine is powered on, **Adapter 1** will automatically receive IP addressing from your physical network for accessing the network. In addition, **Adapter 2** will automatically receive IP addressing for the `Hacking_Lab` virtual network that was created in the previous section, *Setting up a virtualized environment*, *Part 2 – Creating a virtual network*.

Part 3 – Configuring Kali Linux

In this section, you will learn how to perform essential tasks such as changing passwords, enabling **Secure Shell** (**SSH**) for remote access, disabling IPv6 connectivity, refreshing the local list of software packages and upgrading those packages.

To get started, please use the following instructions:

1. Power-on the **Kali Linux** virtual machine and log in using the default username: `kali` and password: `kali`.
2. On the Kali Linux desktop user interface, click on the **Applications** icon to open the applications menu, then click on the **Settings Manager**, as follows:

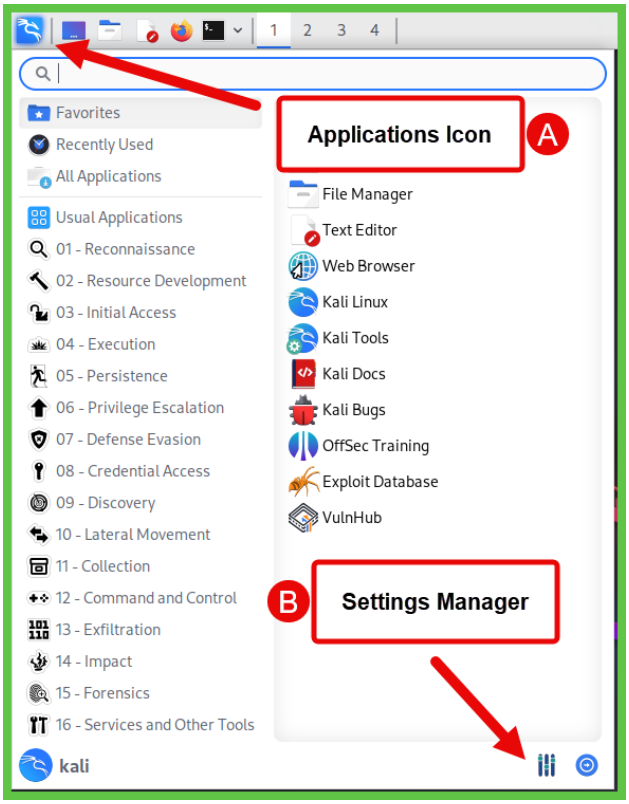


Figure 2.24 – Applications menu

3. When the **Settings Manager** window appears, click on the **Advanced Network Configuration** icon.

4. Next, the **Network Connections** window will appear. Select **Wired connection 1** and click on the **Settings** icon (gear), as shown:

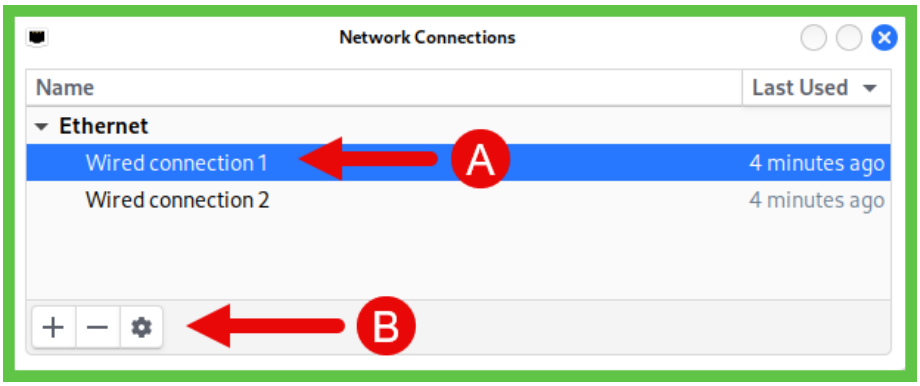


Figure 2.25 – Network Connections

5. Now the **Editing Wired connection 1** window will appear. Select **IPv6 Settings** | change **Method** to **Disabled**, IPv6 is disabled to simplify the lab environment and avoid conflicts, as most beginner-level testing scenarios focus on IPv4:

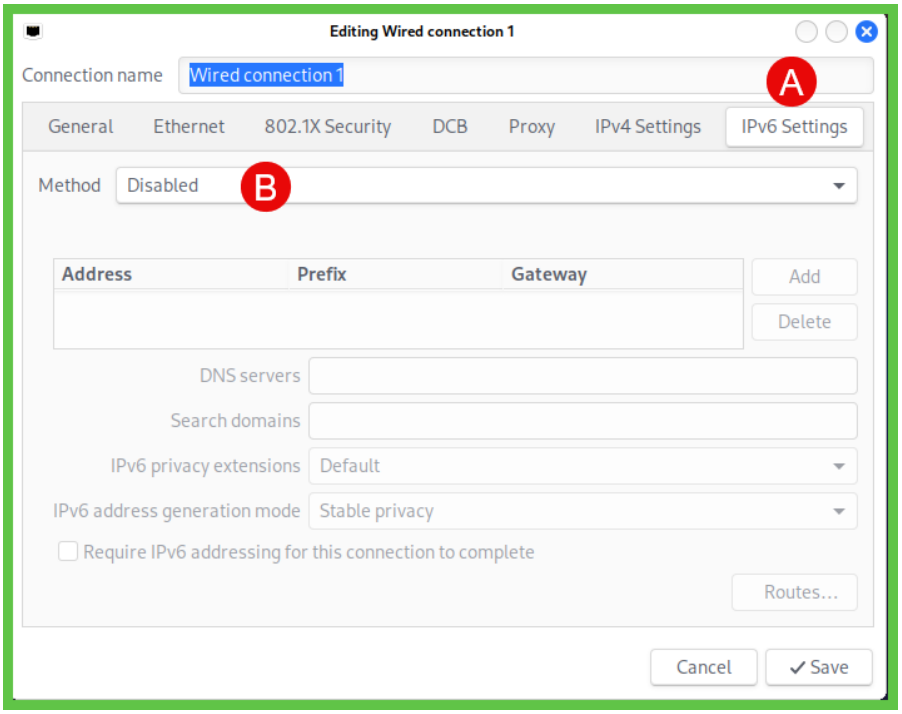


Figure 2.26 – IPv6 settings

6. Select **IPv4 Settings** and enter the following addresses in **Additional DNS Servers**: `8.8.8.8,8.8.4.4` and click **Save**. These public DNS servers ensure reliable name resolution when the virtual machine accesses external resources:

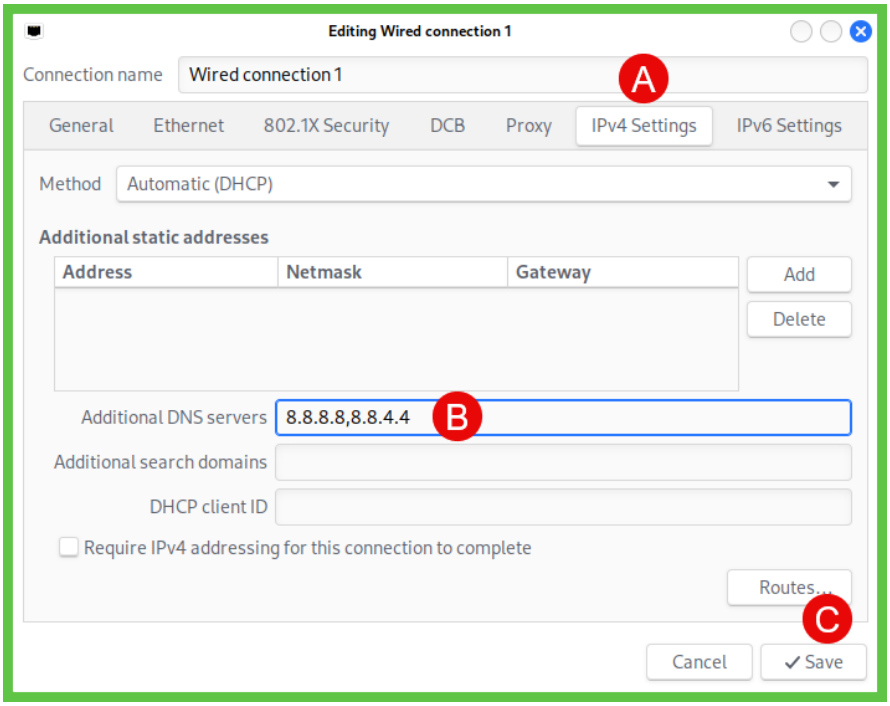


Figure 2.27 – IPv4 settings

7. Close the **Network Connections** window, close the **Settings** window.
8. To verify the IP addresses and network connectivity, open the **Terminal** and use the following command to show the network adapters and IP addresses on Kali Linux:

```
kali@kali:~$ ip address show
```

Figure 2.28 shows each network adapter and the IP address on each interface:

```
kali@kali:~$ ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether                  brd ff:ff:ff:ff:ff:ff
    inet 172.16.17.50/24 brd 172.16.17.255 scope global dynamic noprefixroute eth0
       valid_lft 86228sec preferred_lft 86228sec
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether                  brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.24/24 brd 10.10.10.255 scope global dynamic noprefixroute eth1
       valid_lft 428sec preferred_lft 428sec
    inet6                                scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Figure 2.28 – Network adapters

9. Use the following command to test internet connectivity and ensure **Domain Name System (DNS)** is working as it should:

```
kali@kali:~$ ping www.google.com -c 4
```

In the following figure you can observe that Kali Linux is able to resolve `www.google.com` to an IP address and perform end-to-end connectivity with the destination server:

```
kali@kali:~$ ping www.google.com -c 4
PING www.google.com (142.250.217.4) 56(84) bytes of data.
64 bytes from pnlgaa-as-in-f4.1e100.net (142.250.217.4): icmp_seq=1 ttl=115 time=77.2 ms
64 bytes from pnlgaa-as-in-f4.1e100.net (142.250.217.4): icmp_seq=2 ttl=115 time=76.9 ms
64 bytes from pnlgaa-as-in-f4.1e100.net (142.250.217.4): icmp_seq=3 ttl=115 time=76.8 ms
64 bytes from pnlgaa-as-in-f4.1e100.net (142.250.217.4): icmp_seq=4 ttl=115 time=77.0 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3012ms
rtt min/avg/max/mdev = 76.777/76.954/77.227/0.169 ms
```

Figure 2.29 – Testing connectivity

10. To change the default password on Kali Linux, use the following command:

```
kali@kali:~$ passwd
```

When entering passwords on a Linux terminal, the operating system conceals the password as a layer of security; therefore, it's invisible as you type:

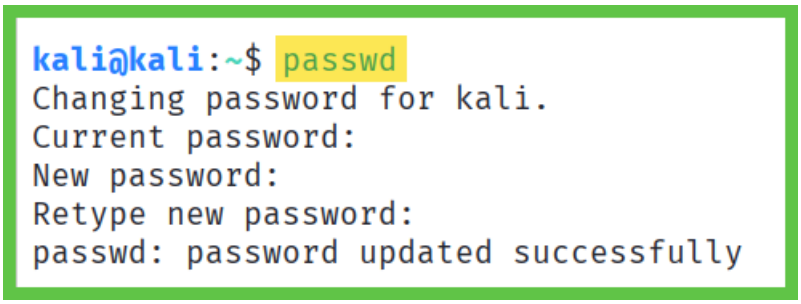


Figure 2.30 – Changing passwords

11. To update the local software repository list and upgrade existing software packages on Kali Linux, use the following commands:

```
kali@kali:~$ sudo apt update
kali@kali:~$ sudo apt upgrade
```

12. Lastly, to configure and enable SSH to auto-start when Kali Linux boots up, use the following commands:

```
kali@kali:~$ sudo systemctl start ssh.service
kali@kali:~$ sudo systemctl enable ssh.service
kali@kali:~$ sudo systemctl restart ssh.service
```

SSH allows remote access to the Kali machine, which will be useful in later scenarios where command-line access is required from other systems

The following section is optional and intended for advanced users who wish to explore AI-assisted automation in ethical hacking. In this section, you will learn how to set up a local AI environment using Ollama and a **large language model** (**LLM**) on Kali Linux to support ethical hacking workflows and automation tasks.

Part 4 – Implementing local AI power with Ollama on Kali Linux

Using a local AI-powered solution on Kali Linux can help ethical hackers streamline workflows and automate repetitive tasks. In this section, you will learn how to configure **Shell-GPT** (**SGPT**) to use the Ollama-hosted Llama 3 LLM on Kali Linux. This setup enables you to interact with the AI directly from the **command-line interface** (**CLI**), allowing you to generate responses and automate tasks without leaving the terminal. To get started, follow these instructions:

1. On Kali Linux, open the **Terminal** and use the following command to download and install Ollama:

```
kali@kali:~$ curl https://ollama.ai/install.sh | sh
```

2. Use the following command to download the Llama3 LLM manifest files onto Kali Linux:

```
kali@kali:~$ ollama pull llama3
```

3. Use the following commands to enable auto-start of the Ollama service during boot time. Start the service presently and verify its version; the current version at the time of writing is version 0.13.0:

```
kali@kali:~$ sudo systemctl enable ollama
kali@kali:~$ sudo systemctl start ollama
kali@kali:~$ ollama --version
```

4. Use the following command to ensure the Ollama server is running and bound to a service port:

```
kali@kali:~$ ollama serve
```

The following screenshot shows the Ollama server is bound to port 11434:

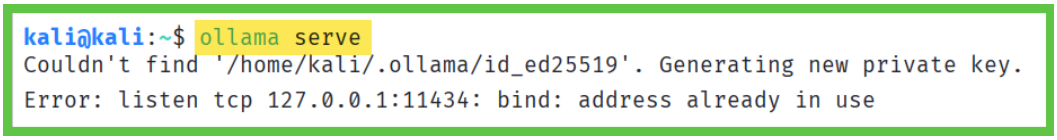


Figure 2.31 – Verifying Ollama port number

5. Next, use the following command to install SGPT:

```
kali@kali:~$ pipx install shell-gpt
```

6. Create a directory for SGPT configuration files:

```
kali@kali:~$ mkdir -p ~/.config/shell_gpt
```

7. Create a default configuration file for SGPT with the following command; ensure to enter `ollama` when prompted to ensure an API key:

```
kali@kali:~$ sgpt --configure
```

The following screenshot shows the execution of the preceding command:

```
kali@kali:~$ pipx install shell-gpt
  installed package shell-gpt 1.4.5, installed using Python 3.13.9
  These apps are now globally available
    - sgpt
done! ✨ 🌟 ✨

kali@kali:~$ mkdir -p ~/.config/shell_gpt

kali@kali:~$ sgpt --configure
Please enter your OpenAI API key:
Usage: sgpt [OPTIONS] [PROMPT]
Try 'sgpt --help' for help.
╭─ Error ──────────────────────────────────────────────────────────────────────╮
│ No such option: --configure                                                  │
╰──────────────────────────────────────────────────────────────────────────────╯
```

Figure 2.32 – SGPT initial setup

8. Next, to configure SGPT to use Ollama Llama 3 LLM in offline mode, use the following command to access and modify the SGPT configuration file:

```
kali@kali:~$ nano ~/.config/shell_gpt/.sgptrc
```

9. When the configuration file appears in the Nano text editor, make the following changes:
    - `DEFAULT_MODE =llama3`

- `API_BASE_URL =http://localhost:11434/v1`
- `OPENAI_KEY =ollama`

Here is the snapshot of the configuration file:

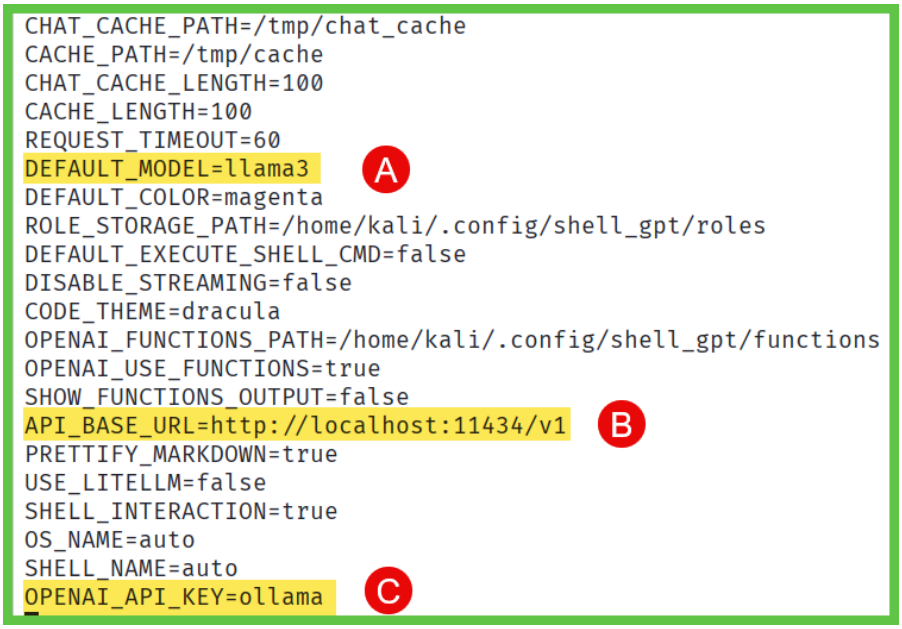


Figure 2.33 – SGPT configuration file

To save the configuration file using the keyboard sequence, press *CRTL + X*, then *Y*, and *Enter*.

10. Next, use SGPT to provide the **Network Mapper** (**Nmap**) command for identifying open ports and their service versions of a host:

```
kali@kali:~$ sgpt "Create an nmap commmand to identify service version of
open ports on a host"
```

Figure 2.34 shows SGPT leveraging the Ollama Llama 3 LLM to provide a response to the query:

```
kali@kali:~$ sgpt "Create an nmap commmand to identify service version of ope
n ports on a host"
To identify the service version of open ports on a host using Nmap, you can
use the following command:

nmap -sV <host_ip>

This command will perform a version scan (-sV) on the specified <host_ip> and
display information about the services running on open ports.

Example:

nmap -sV 192.168.1.100

This command will scan the host at IP address 192.168.1.100 and provide
information about the services running on open ports, including version
numbers.
```

Figure 2.34 – Using the local AI on Kali Linux

11. Use the local AI to automate a task using the Terminal:

```
kali@kali:~$ sgpt --shell "Perform 4 pings on www.google.com"
```

The `sgpt --shell` command enable SGPT to automate and execute processes on the existing terminal, as follows:

```
kali@kali:~$ sgpt --shell "Perform 4 pings on www.google.com"
ping -c 4 www.google.com
[E]xecute, [D]escribe, [A]bort: E
PING www.google.com (142.250.217.4) 56(84) bytes of data.
64 bytes from pnlgaa-as-in-f4.1e100.net (142.250.217.4): icmp_seq=1 ttl=115 t
ime=77.0 ms
64 bytes from pnlgaa-as-in-f4.1e100.net (142.250.217.4): icmp_seq=2 ttl=115 t
ime=76.7 ms
64 bytes from pnlgaa-as-in-f4.1e100.net (142.250.217.4): icmp_seq=3 ttl=115 t
ime=76.8 ms
64 bytes from pnlgaa-as-in-f4.1e100.net (142.250.217.4): icmp_seq=4 ttl=115 t
ime=76.9 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3011ms
rtt min/avg/max/mdev = 76.662/76.866/77.049/0.145 ms
```

Figure 2.35 – Automating tasks on Kali Linux

You've learned how to set up and configure Kali Linux as the attacker machine. Next, you will learn how to set up vulnerable systems.

Setting up vulnerable machines

In this section, you will learn how to build both vulnerable Windows- and Linux-based virtual machines and deploy them within Oracle VirtualBox. Both of these virtual machines will be connected to the `Hacking_Lab` virtual network based on our lab design as shown in *Figure 2.36*:

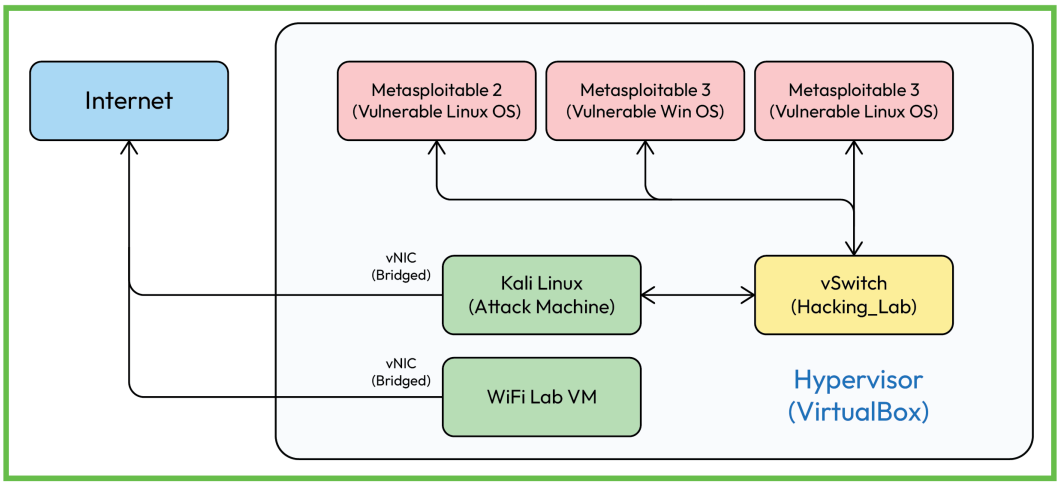


Figure 2.36 – Lab design

Metasploitable 3 is a vulnerable system that's built for the purpose of learning how to identify and exploit security vulnerabilities on a targeted device. The developers of this vulnerable system enable users to build both a Windows- and Linux-based version for Oracle VirtualBox and VMware solutions. These machines contain intentionally vulnerable services and configurations that will be exploited in later chapters to demonstrate real-world attack techniques. Therefore, Metasploitable 3 will be the preferred Windows and Linux vulnerable machines for the ethical hacking lab environment.

Part 1 – Building a vulnerable Windows-based virtual machine

To get started with building the Metasploitable 3 – Windows-based version, please use the following instructions:

1. On your Windows host computer, you'll need to download and install the latest version of **Vagrant**. Vagrant enables users to both create and manage virtual systems. To download Vagrant, go to `https://developer.hashicorp.com/vagrant/install` and click on the **Windows AMD64 Binary** download, as shown in *Figure 2.37*:

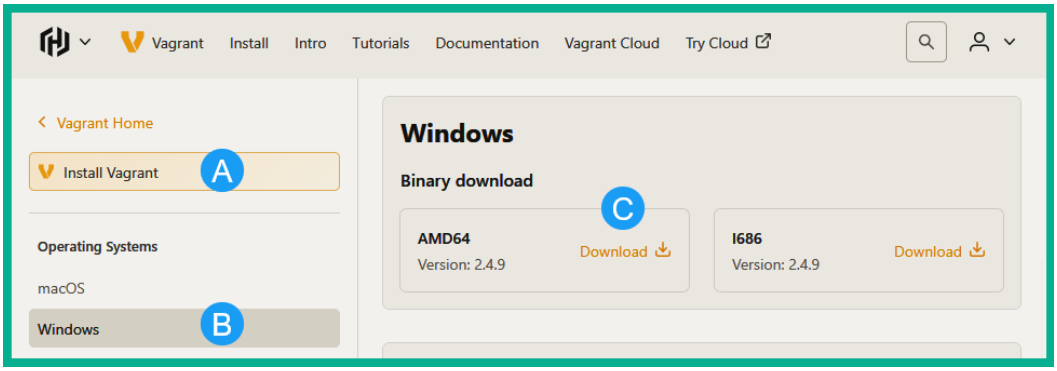


Figure 2.37 – Download Vagrant

2. After downloading the Vagrant installation file, install it. After the installation is completed, you'll be prompted to reboot your host operating system; select **Yes** to restart your device.
3. After your host operating system reboots, open the Command Prompt and execute the following commands to reload and install additional plugins for Vagrant:

```
vagrant plugin install vagrant-reload
vagrant plugin install vagrant-vbguest
```

4. Next, use the following command to initiate the process for setting up the Windows-based version of Metasploitable 3:

```
vagrant box add rapid7/metasploitable3-win2k8
```

5. Type `1` and hit *Enter* to select VirtualBox as the preferred provider, as follows:

```
C:\Users\Glen> vagrant box add rapid7/metasploitable3-win2k8
==> box: Loading metadata for box 'rapid7/metasploitable3-win2k8'
    box: URL: https://vagrantcloud.com/api/v2/vagrant/rapid7/metasploitable3-wi
This box can work with multiple providers! The providers that it
can work with are listed below. Please review the list and choose
the provider you will be working with.

1) virtualbox
2) vmware
3) vmware_desktop

Enter your choice: 1
```

Figure 2.38 – Setting VirtualBox as the preferred provider

Vagrant will attempt to download the required files for building the Windows-based version of Metasploitable 3 (*Figure 2.39*):

```
==> box: Adding box 'rapid7/metasploitable3-win2k8' (v0.1.0-weekly) for prov
ider: virtualbox
    box: Downloading: https://vagrantcloud.com/rapid7/boxes/metasploitable3-
win2k8/versions/0.1.0-weekly/providers/virtualbox/unknown/vagrant.box
    box:
==> box: Successfully added box 'rapid7/metasploitable3-win2k8' (v0.1.0-week
ly) for 'virtualbox'!
```

Figure 2.39 – Downloading required files

6. Use the following command to change the present working directory to the location where the files are downloaded:

```
cd .vagrant.d\boxes
```

7. Use the following command to rename the default folder to `metasploitable3-win2k8`:

```
REN "rapid7-VAGRANTSLASH-metasploitable3-win2k8" "metasploitable3-win2k8"
```

8. Use the following command to build the Windows-based Metasploitable 3 configuration file:

```
vagrant init metasploitable3-win2k8
```

The following screenshot shows the execution of the preceding command:

```
C:\Users\Glen> cd .vagrant.d\boxes

C:\Users\Glen\.vagrant.d\boxes> REN "rapid7-VAGRANTSLASH-metasploitable3-win
2k8" "metasploitable3-win2k8"

C:\Users\Glen\.vagrant.d\boxes> vagrant init metasploitable3-win2k8
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.

C:\Users\Glen\.vagrant.d\boxes>
```

Figure 2.40 – Building the Windows-based configuration file

9. Next, use the following command to build the virtual machine using Vagrant:

```
vagrant up
```

If the `vagrant up` command provides an error, execute the command again. If the following Windows Security prompt appears, select **Allow**:

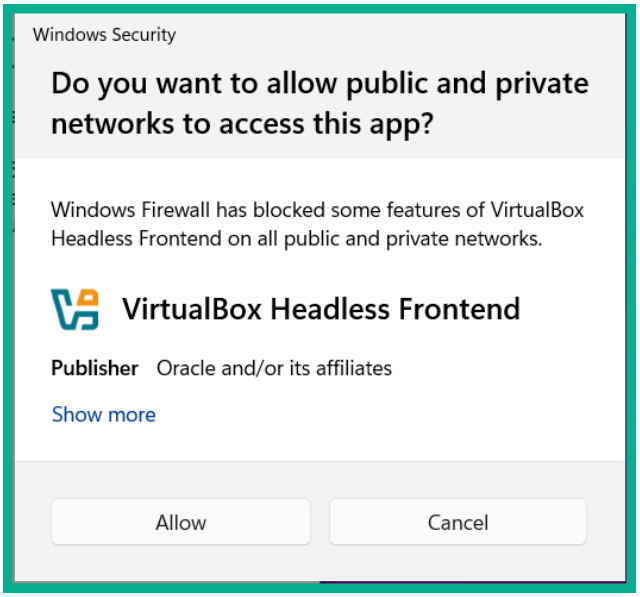


Figure 2.41 – Windows Security prompt

10. After the process is completed, open **Oracle VirtualBox Manager**; there you will find the newly created Windows-based version of Metasploitable 3 running.
11. Select the Windows-based Metasploitable 3 virtual machine and click on **Show** to access its console, as shown in *Figure 2.42*:

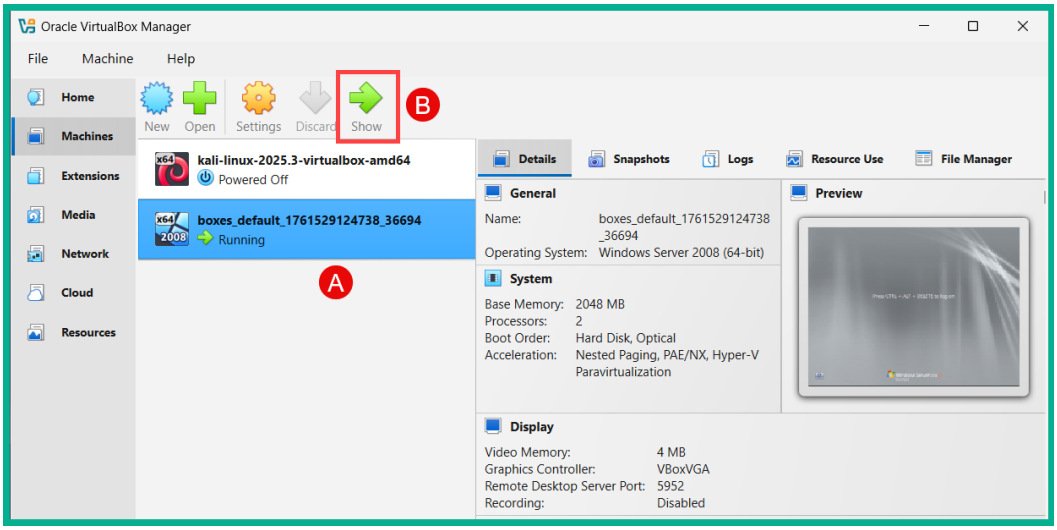


Figure 2.42 – Accessing the virtual machine console view

12. On the virtual machine console view, select **Input | Keyboard | Insert Ctrl-Alt-Del**, as follows:

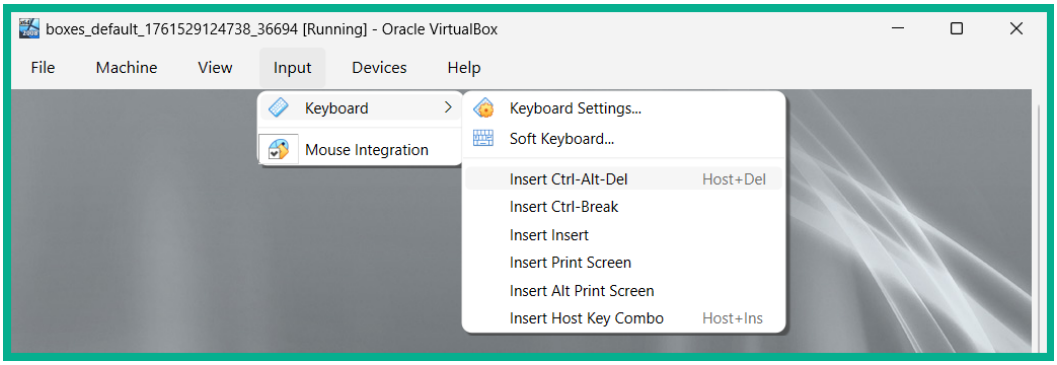


Figure 2.43 – Input menu

13. Select the **Administrator** account and log in with the default password: `vagrant`.
14. Upon logging-in, the operating system may prompt you to choose a reboot option and whether to activate the operating system. Choose **Restart Later** then **Ask me later**:

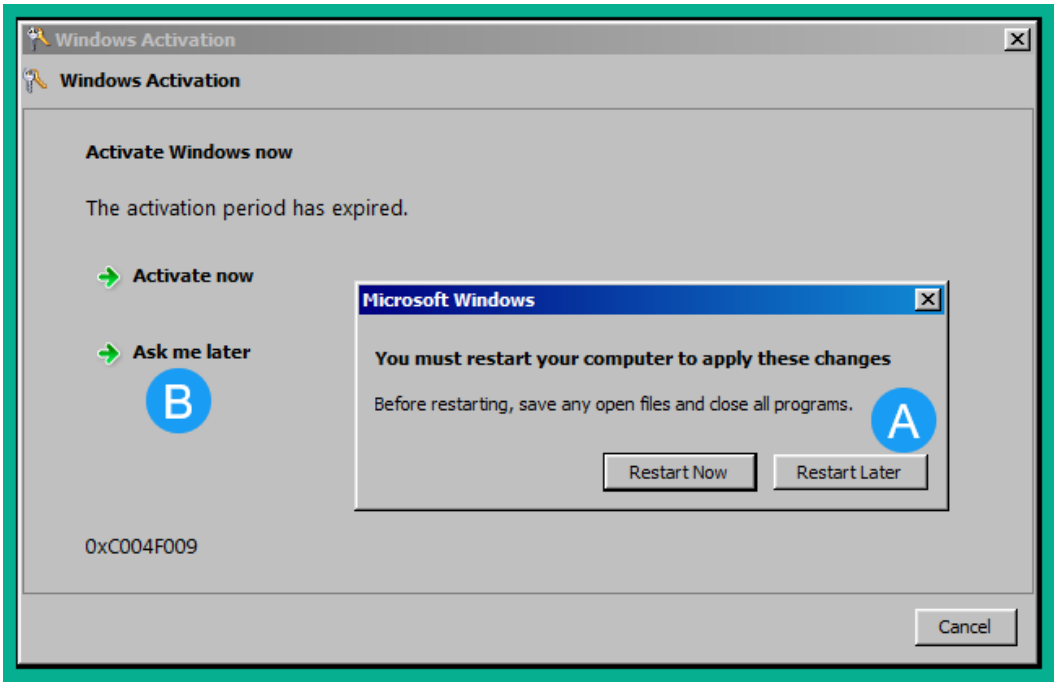


Figure 2.44 – Windows Activation prompts

15. Once logged in to the desktop user interface, shut down the vulnerable Windows-based Metasploitable 3 operating system.
16. On **Oracle VirtualBox Manager**, select the **Windows-based Metasploitable 3 virtual machine** and select **Settings**.
17. Rename the virtual machine by going to **General** | **Identity** | **VM Name** and set it as *Metasploitable 3 – Windows*.

18. Next, click on **Network** | **Adapter 1** and use the following settings:
    - **Attached to**: **Internal Network**
    - **Name**: Manually enter `Hacking_Lab` within the field
    - **Promiscuous Mode**: **Allow All**
    - Click **OK** to save the settings of the virtual machine

The following screenshot shows an example of the preceding settings:

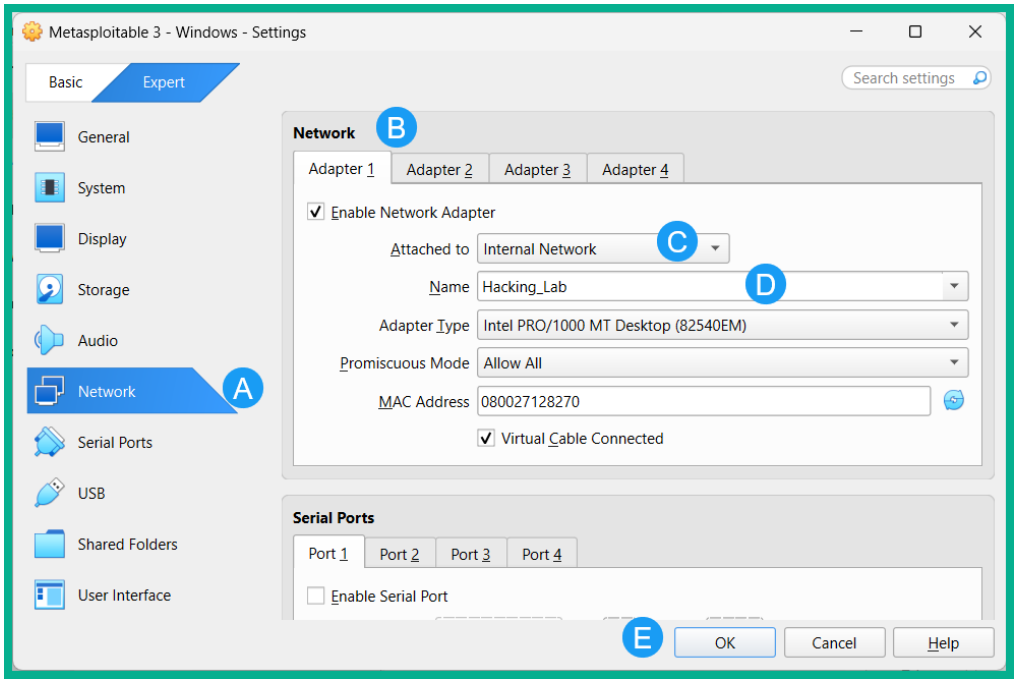


Figure 2.45 – Network settings

Next, you will build and set up the Linux-based version of Metasploitable 3 within the lab environment.

Part 2 – Building a vulnerable Linux-based virtual machine

To get started with building the Linux-based version of Metasploitable 3, please use the following instructions:

1. On the Windows host computer, open the **Command Prompt** and use the following commands to load the Linux-based version of Metasploitable 3:

```
cd .vagrant.d\boxes
vagrant box add rapid7/metasploitable3-ub1404
```

2. Enter `1` at the prompt to select VirtualBox as the preferred provider.
3. Once the repository files have been downloaded, use the following commands to delete the previously created Vagrantfile, rename the repository directory of the Linux-based version of Metasploitable 3 and create a new configuration file:

```
del Vagrantfile
REN "rapid7-VAGRANTSLASH-metasploitable3-ub1404" "metasploitable3-ub1404"
vagrant init metasploitable3-ub1404
```

4. Next, open the **Oracle VirtualBox Manager** application.
5. Open Windows Explorer and go `C:\Users\<username>\.vagrant.d\boxes\metasploitable3-ub1404\0.1.12-weekly\virtualbox`. There you will find the complied files for the virtual machine, right-click on the file named **box** | **Open with** | **VirtualBox Manager**, shown as follows:

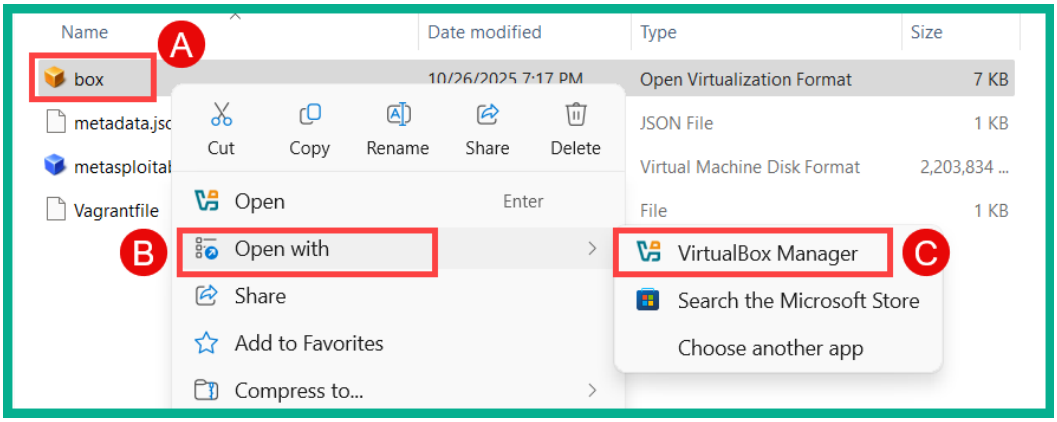


Figure 2.46 – Load new virtual machine into VirtualBox Manager

6. The following **Import Virtual Appliance** window will appear. Click on **Finish**:

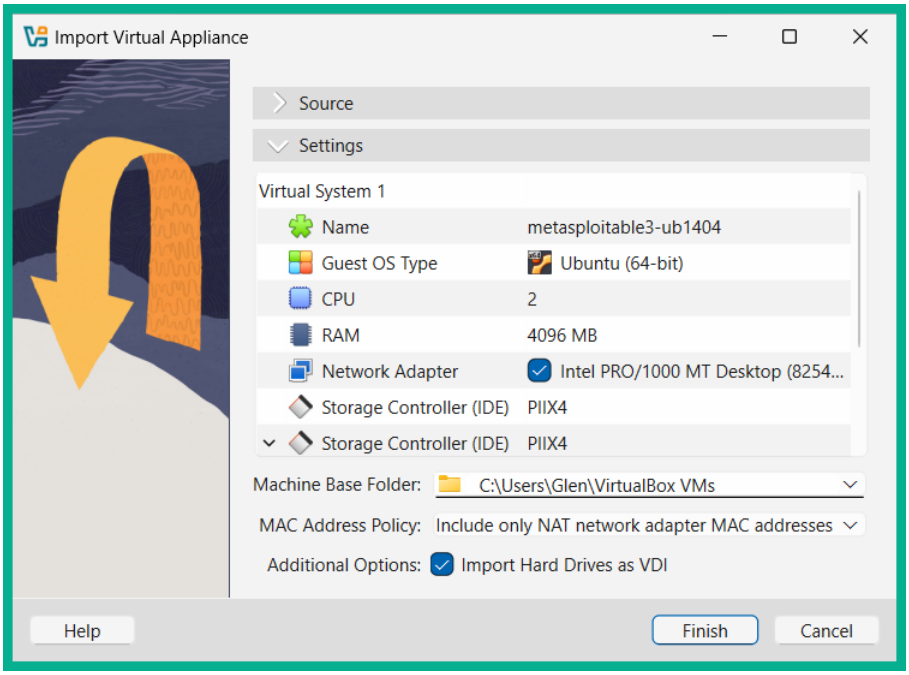


Figure 2.47 – Import window

7. Select the newly created Linux-based version of Metasploitable 3 and click on **Settings**:

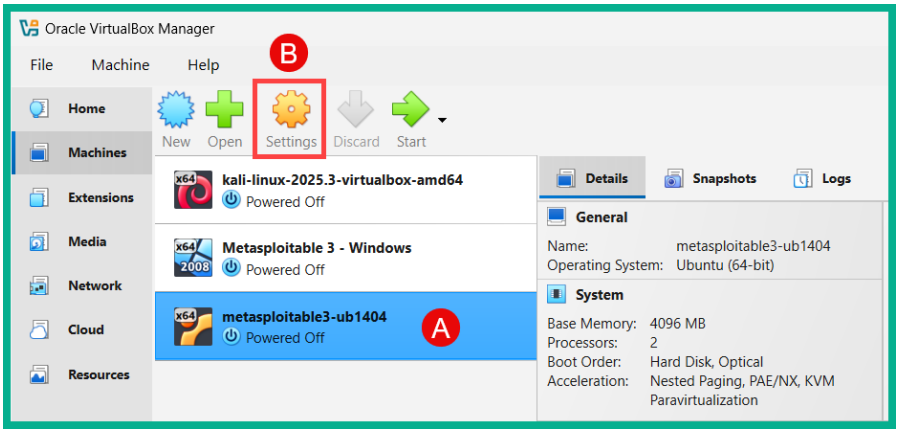


Figure 2.48 – Settings

8. Next, click on **Network** | **Adapter 1** and use the following settings:
    - **Attached to**: **Internal Network**
    - **Name**: Manually enter `Hacking_Lab` within the field
    - **Promiscuous Mode**: **Allow All**
    - Click **OK** to save the settings of the virtual machine

    The following screenshot shows an example of the preceding settings:

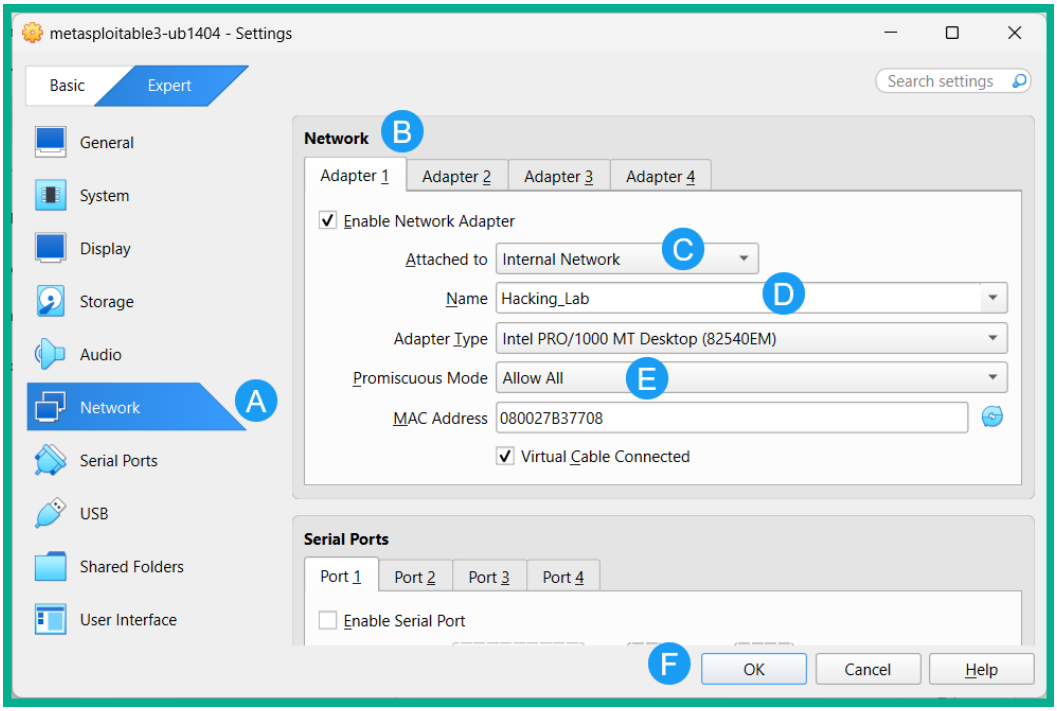


Figure 2.49 – Network settings

The default username: `vagrant` and password: `vagrant` can be used for logging into the Linux-based version of Metasploitable 3. Using the `sudo halt` command enables you to shut down this virtual machine.

You are now ready to set up the Metasploitable 2 virtual machine.

Part 3 – Setting up the Metasploitable 2 virtual machine

To get started with this exercise, please use the following instructions:

1. Firstly, go to `https://sourceforge.net/projects/metasploitable/` to download the Metasploitable 2 zipped file onto your host computer.
2. Unzip the downloaded folder to extract its contents.
3. Open **Oracle VirtualBox Manager** and click on **New**.
4. Next, the **New Virtual Machine** window will appear; use the following information:
    - **VS Name: Metasploitable 2**
    - **OS: Linux**
    - **OS Distribution: Other Linux**
    - **OS Version: Other Linux (64-bit)**

    The following screenshot shows the preceding settings:

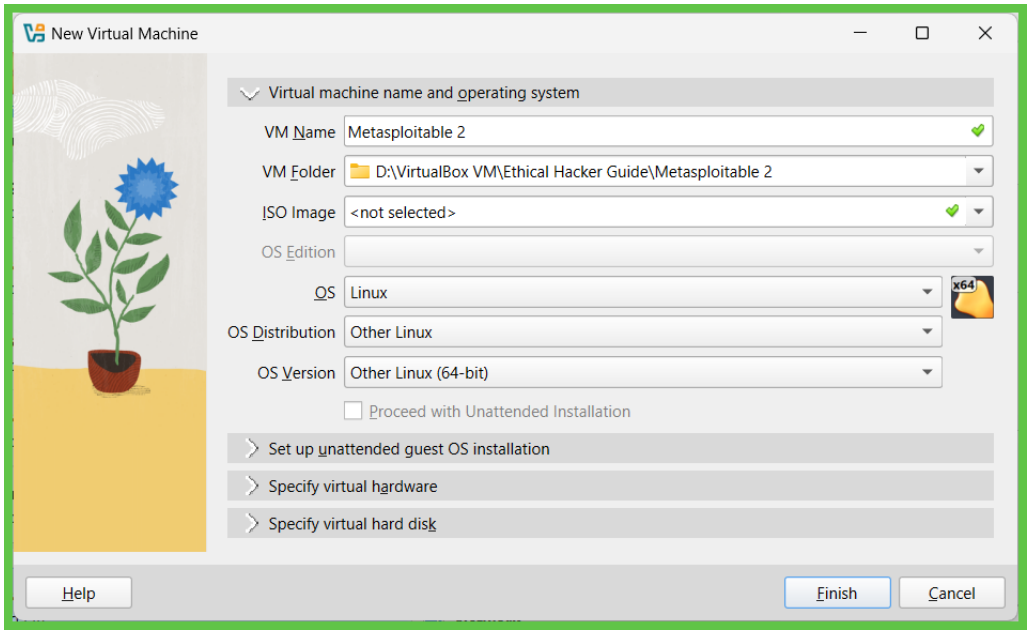


Figure 2.50 – New Virtual Machine window

5. Select the expand the **Specify virtual hard disk** menu, select the **Use an Existing Virtual Hard Disk File** and click on the **Folder icon with the green arrow** to open the **Hard Disk Selector**, shown as follows:

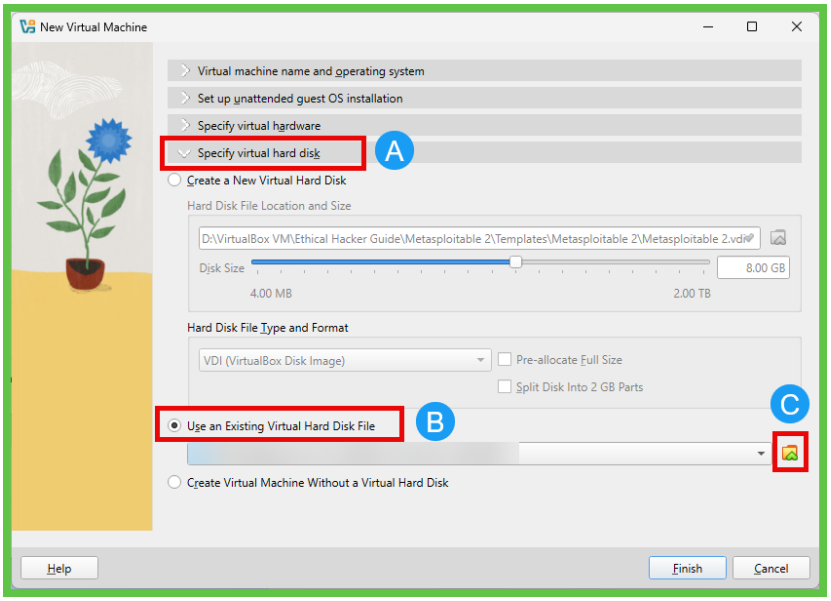


Figure 2.51 – Virtual disk options

6. On the **Hard Disk Selector** window, click on the **Add** to open the virtual disk select window:

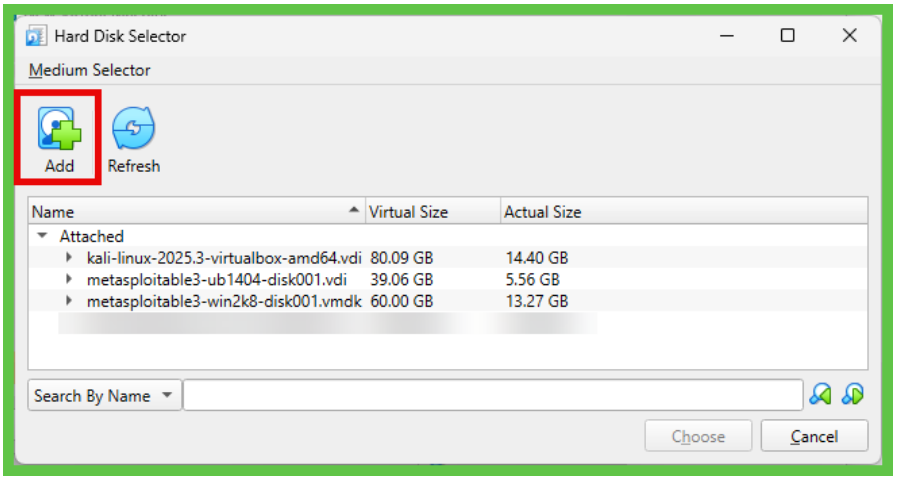


Figure 2.52 – Hard Disk Selector

7. The following window will appear, navigate to the extracted Metasploitable 2 folder, select the **Metasploitable** virtual disk file and click on **Open**:

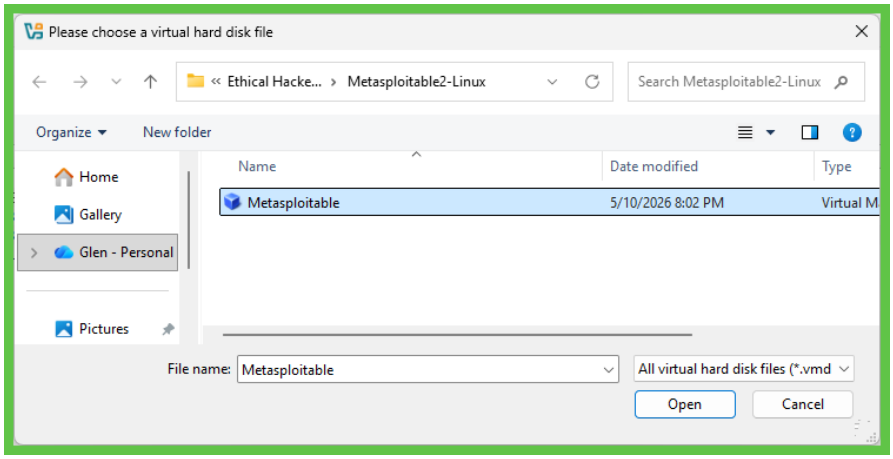


Figure 2.53 – Selecting the virtual machine file

8. The **Metasploitable** virtual disk file will be added to the **Hard Disk Selector** window, select it, and click on **Choose**:

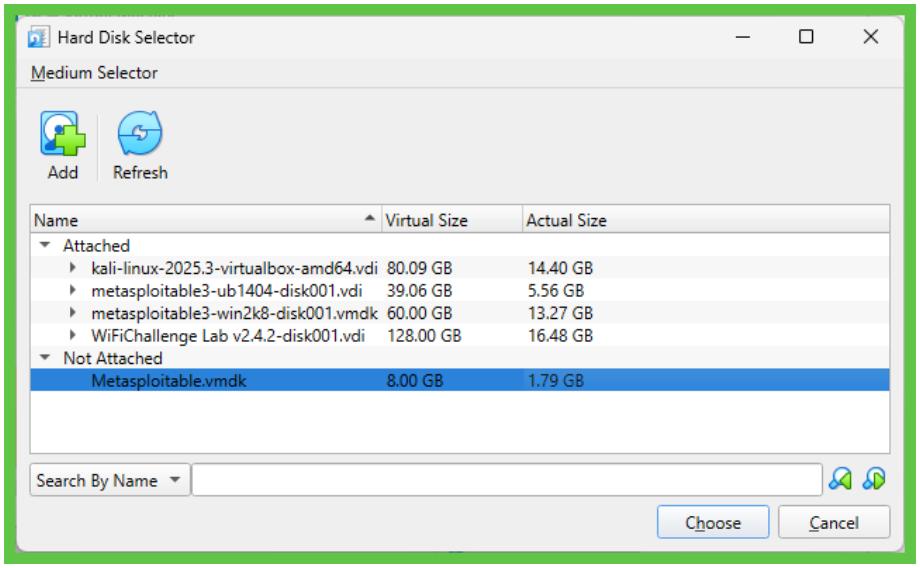


Figure 2.54 – Assigning the virtual machine file

9. You'll return to the **New Virtual Machine** window, and you'll notice the **Metasploitable** virtual disk file is assigned; click on **Finish** to save the virtual machine:

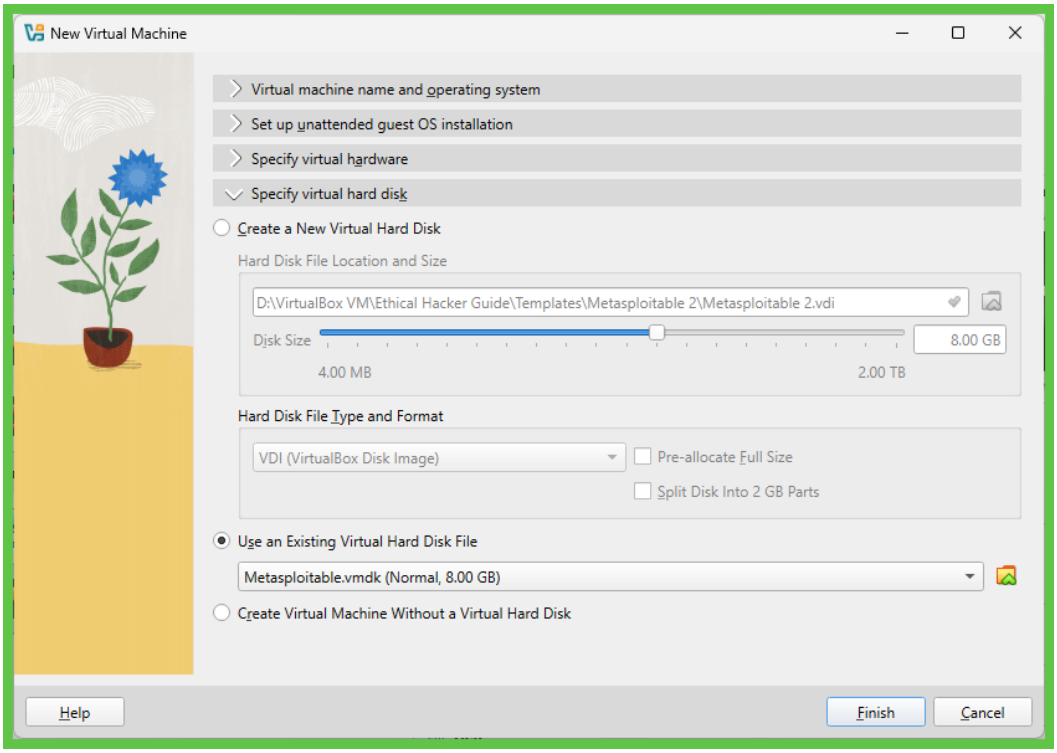


Figure 2.55 – Creating a virtual machine

10. On the **Oracle VirtualBox Manager** user interface, select the newly created **Metasploitable 2** virtual machine and click on **Settings**.
11. On the **Settings** menu, select the **Network** category and apply the following configurations to **Adapter 1**:
    - Check **Enable Network Adapter**
    - **Attached to**: **Internal Network**
    - **Name**: `Hacking_Lab`
    - **Promiscuous Mode**: **Allow All**

    Here is a snapshot of the preceding configuration:

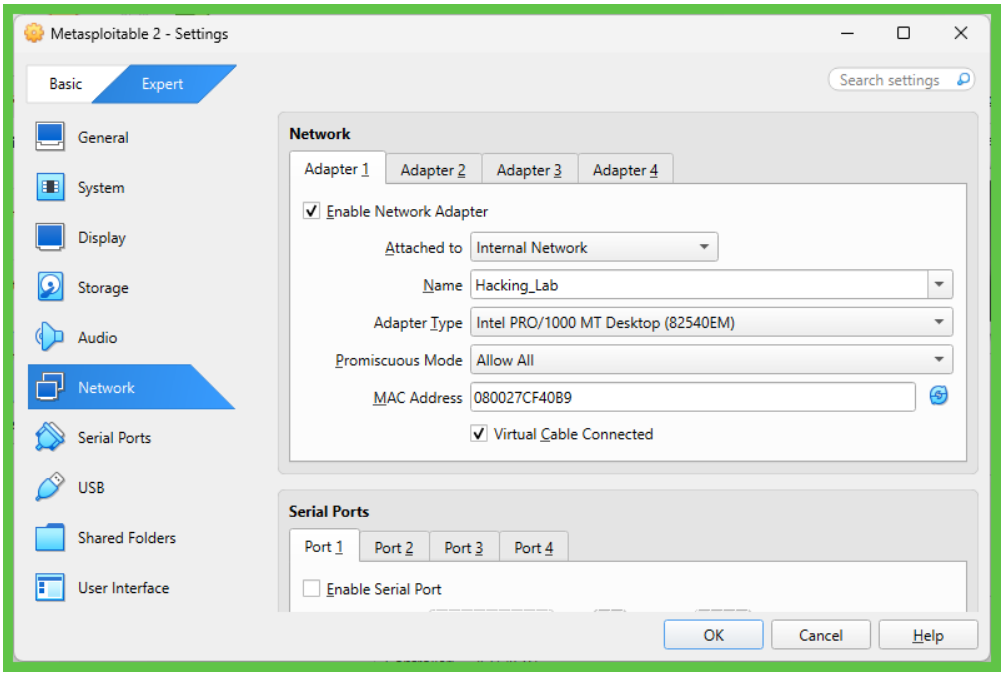


Figure 2.56 – Network configurations

12. After you have saved the configuration, you can power on the virtual machine. Use the following credentials to login:
    - Login: `msfadmin`
    - Password: `msfadmin`
13. To power off this virtual machine, use the `sudo halt` command.

You have now learned how to set up various vulnerable virtual machines within the ethical hacking lab. Next, you'll set up a virtual machine that simulates multiple vulnerable wireless networks.

Setting up a vulnerable wireless lab

In this section, you will learn how to deploy a virtual machine that simulates real-world wireless networks that are vulnerable to various types of cyber-attacks and threats. This lab will be used in later chapters to demonstrate wireless attack techniques such as network discovery and authentication attacks.

Acknowledgements to Raúl Calvo Laorden, a cybersecurity professional in the industry, who created this virtual machine. Raúl made a virtualized lab environment to help others learn and

practice real-world attack simulations on wireless infrastructure in a safe environment while being able to identify security vulnerabilities as an ethical hacker.

To get started with setting up the virtualized wireless environment, please use the following instructions:

1. On your host device, you will need to download the VirtualBox version of the WiFiChallenge Lab by using the following link: `https://drive.proton.me/urls/Q4WPB23W7R#Qk4nxMH8Q4oQ`
2. Select the `.ova` file then click on **Download** to initiate the download process:

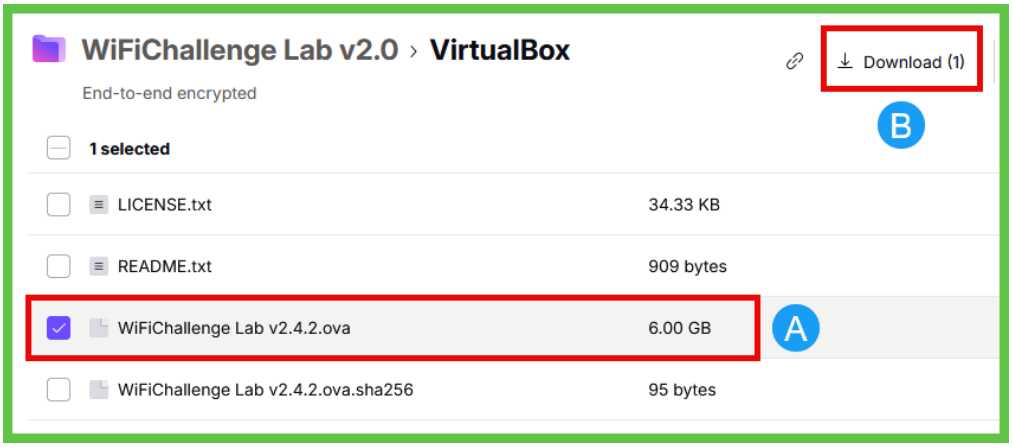


Figure 2.57 – Downloading page

3. Navigate to the download file on your host computer, right-click to choose **Open with VirtualBox Manager**:

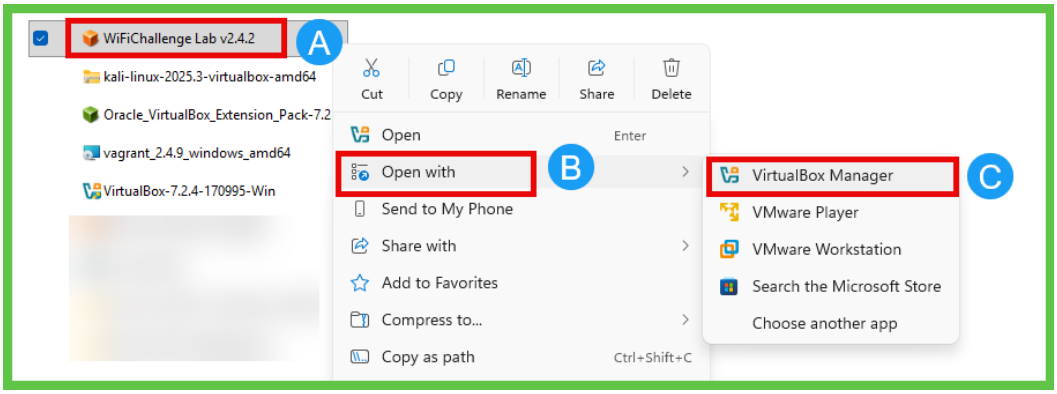


Figure 2.58 – Loading onto VirtualBox Manager

4. The **Import Virtual Appliance** window will appear. Click on **Finish**:

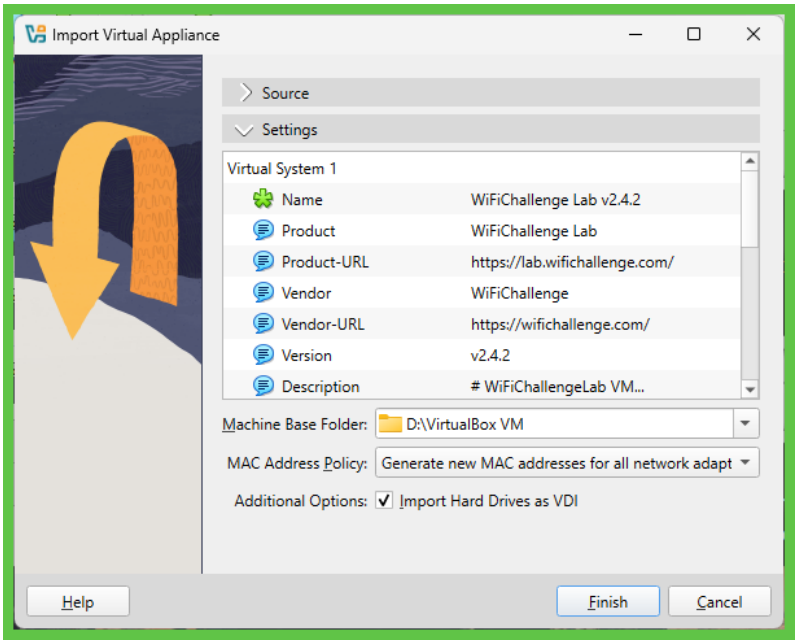


Figure 2.59 – Importing a virtual machine

Ensure to accept the **Software License Agreement**, then the importing process will start and take a few minutes to complete. Once it's finished, the new virtual machine will appear automatically within **Oracle VirtualBox Manager**:

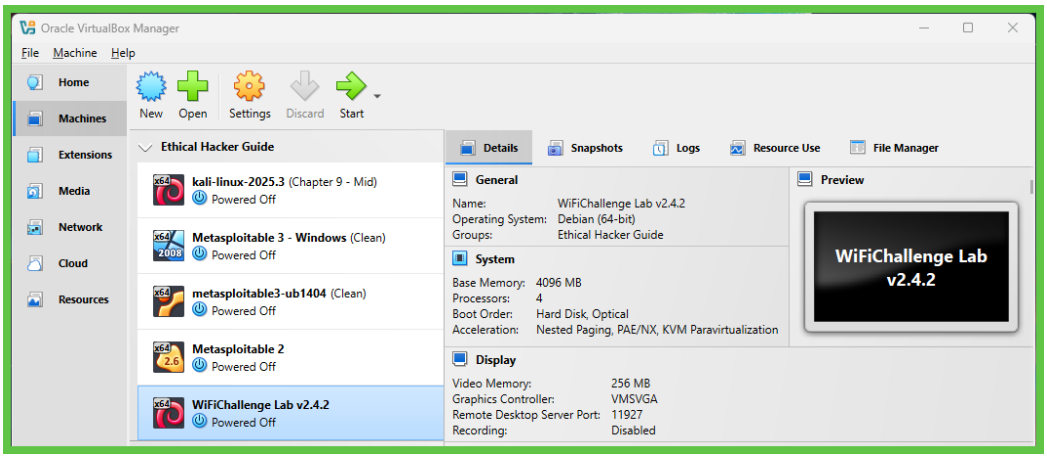


Figure 2.60 – New virtual machine

5. To power-on the **WiFiChallenge** virtual machine, select it and click on **Start**:

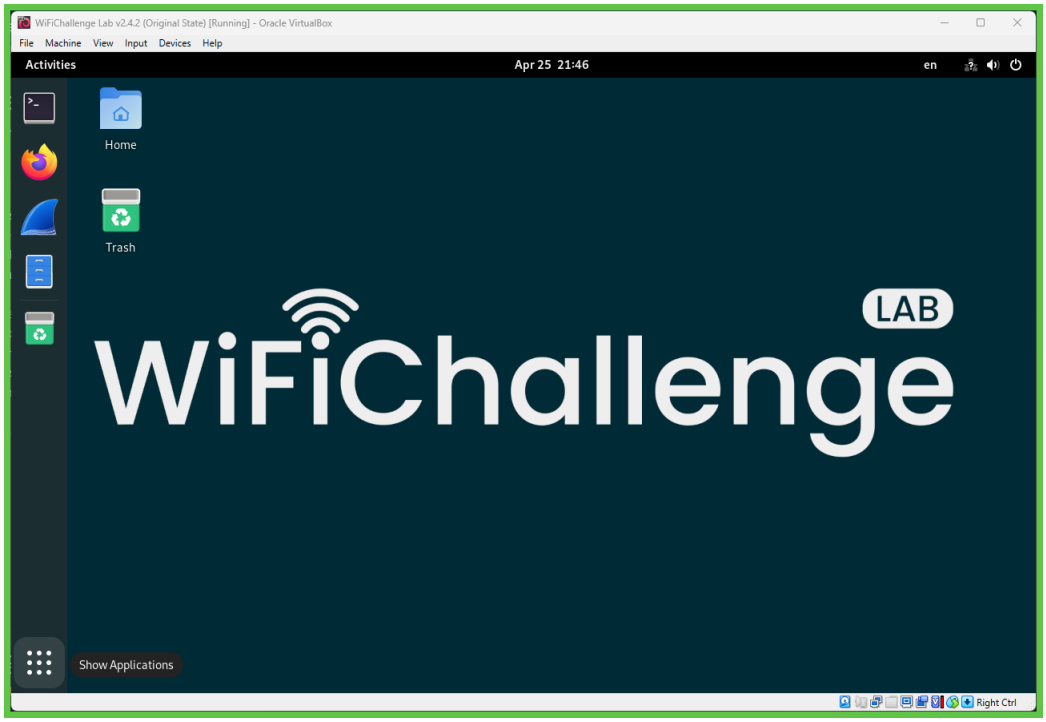


Figure 2.61 – WiFiChallenge

You'll be able to automatically log in to the desktop.

6. Lastly, to power off this virtual machine, click on the power icon on the top-left corner and click the power button on the menu shown:

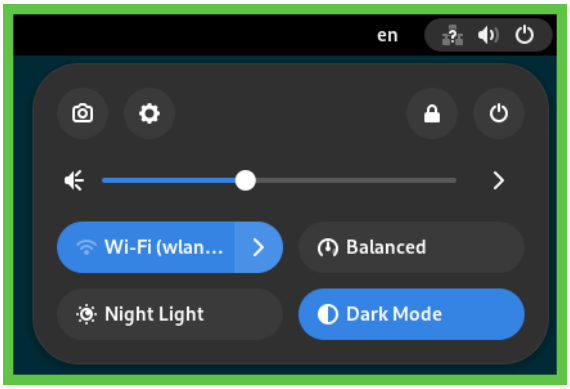


Figure 2.62 – Powering off virtual machine

You will explore common techniques used by ethical hackers to identify security vulnerabilities on wireless networks in *Chapter 11*.

You've now successfully learned how to deploy a virtual wireless lab environment to learn the fundamentals of wireless security.

Summary

Setting up a virtual ethical hacking lab environment enables aspiring cybersecurity professionals to reduce the cost of purchasing dedicated systems to simulate a real-world organization's network. In this chapter, you've learned how to deploy virtual networks, set up and customize Kali Linux as the attacker machine and implement multiple vulnerable machines. This lab now provides a controlled environment where you can safely perform reconnaissance, scanning, and exploitation exercises covered in upcoming chapters. Additionally, you've set up a local AI model on Kali Linux to perform AI-augmented tasks to improve your ethical hacking workflows.

In the next chapter, , you will learn common techniques and tools for performing reconnaissance as an ethical hacker.

Further reading

The following are additional reading resources to help reinforce your learning:

- Kali Linux Documentation - `https://www.kali.org/docs/`
- Oracle VirtualBox User Guide - `https://www.virtualbox.org/manual/`
- Metasploitable 2 Project Page - `https://docs.rapid7.com/metasploit/metasploitable-2/`
- Metasploitable 3 Project Page - `https://github.com/rapid7/metasploitable3`
- R4ucl projects - `https://github.com/r4ulcl`

3

Reconnaissance and Footprinting

During reconnaissance, cybersecurity professionals identify an organization's attack surface. They examine which assets (systems, applications, and networks) are exposed on the internet and have the potential to leak sensitive information to threat actors. Using techniques and procedures similar to threat actors, ethical hackers leverage the information collected during the reconnaissance phase to identify security vulnerabilities, research, and develop exploits, to improve their plan of attack against targets. Hence, ethical hackers can use this information to footprint exposed systems and networks with the intent of helping organizations to determine their exposure factors and reduce their level of cyber-risk.

Reconnaissance is the overall phase, while **footprinting** is a technique used within this phase to gather detailed information. Ethical hackers use this phase to gather as much information on targeted systems, networks, applications, people, and even organizations. Footprinting enables ethical hackers to create a digital map of their target's infrastructure and better understand how systems, networks, and people are interconnected.

In this chapter, you will learn the importance of performing thorough reconnaissance and footprinting techniques on targeted systems, networks, and organizations. You will develop the skills that are commonly used by ethical hackers and threat actors, while learning how to apply those skills in real-world scenarios. Additionally, you'll see how organizations are both intentionally and unintentionally leaking data on the internet which can be leveraged by real cyber-criminals. Lastly, you'll explore common countermeasures and security controls used to prevent and mitigate reconnaissance-based attacks by adversaries.

In this chapter, we will cover the following topics:

- Footprinting concepts and methodology
- Reconnaissance using search engines and web services
- Email and web footprinting

- WHOIS and DNS footprinting
- Reconnaissance countermeasures

> **Note**
>
> **Download the code bundle and the PDF version of this book**
>
> Your purchase includes a DRM-free PDF copy of this book, the code bundle, and a range of exclusive benefits. To unlock everything, follow the *Free benefits with your book* section in the *Preface*.

Technical requirements

You only need Kali Linux to run the practical examples in this chapter. For installation details, please refer to *Chapter 2*.

Footprinting concepts and methodology

As discussed in the introduction, reconnaissance focuses on collecting as much valuable information as possible on a target, enabling the threat actor to better understand and build a profile about the target. It is the first phase of ethical hacking.

With a better understanding of the target, the threat actor can improve their plan of attack. For instance, if a threat actor wants to compromise the online banking portal of a well-known commercial bank, the attacker needs to identify security vulnerabilities that are associated with the targeted online banking platform. This includes identifying the underlying technical infrastructure such as the systems and applications used by the target for hosting their online web application, determining the web components and dependencies, and identifying the operating systems and any other information that's useful in mapping the target's digital infrastructure. Furthermore, a threat actor would be required to perform research to determine whether any component of the infrastructure is vulnerable and can be exploited.

There are two types of reconnaissance that are commonly used by threat actors and seasoned ethical hackers:

- **Passive reconnaissance**
- **Active reconnaissance**

Passive reconnaissance techniques use an indirect approach by the threat actor and ethical hacker to collect information about the target from public data sources. This means neither the cyber-criminals nor ethical hackers establish a direct connection to the targeted system, network, or organization to gather information about them.

You're probably wondering how and where ethical hackers and adversaries collect information if they're not gathering it from the target. The answer is, from the organizations that unintentionally leak sensitive information on the internet without realizing the potential impact and increase in cyber-risk. This enables threat actors to collect such sensitive information, build a profile about the target's infrastructure, and identify any security vulnerabilities. For instance, many organizations publish information about the technologies used within their internal network when posting a technical job vacancy either on their company-owned website or a recruiting website.

Figure 3.1 shows an example of an organization leaking their technical infrastructure on a job post:

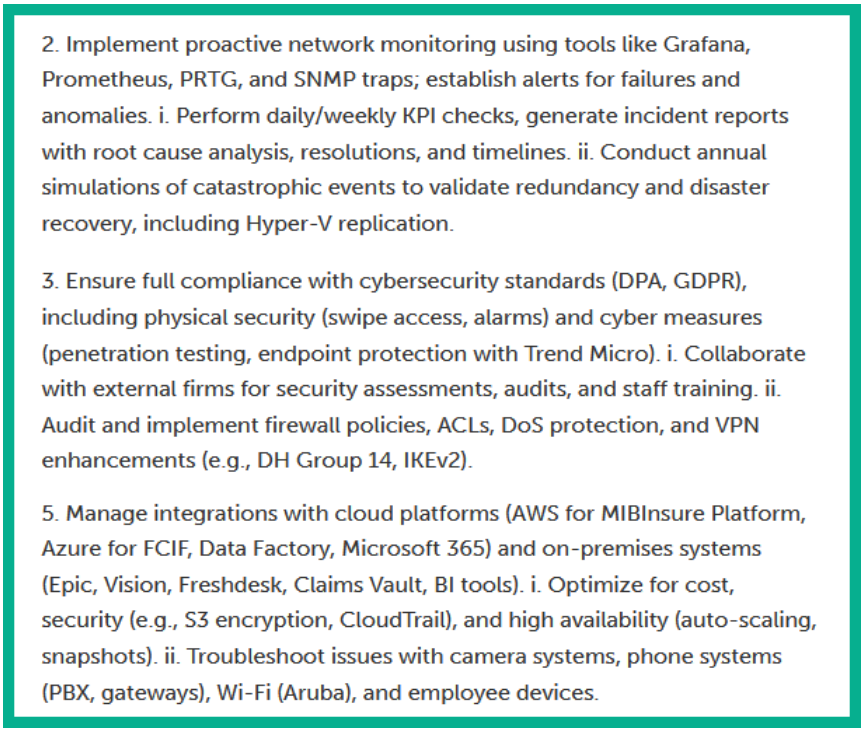


Figure 3.1 – Sensitive data leakage

Here, the organization has shared the following information which can be leveraged by a real threat actor who may be targeting the organization:

- The creator of the job description has leaked one of their network monitoring solutions (**PRTG**) and hypervisor platform for managing virtual machines (**Hyper-V**).

- Additionally, another point indicates their endpoint protection solution. This enables an attacker to create and test special payloads to bypass the malware solution. In addition, they can identify and exploit any known security vulnerabilities that are associated with the solution.
- While this may be least significant, the organization has posted the **Diffie-Hellman (DH)** key exchange group, that is, DH group 14. This enables the attacker to research security vulnerabilities about the brand of firewalls used by the organization and specifically any security flaw with the **Virtual Private Network (VPN)** configuration or package.
- The organization indicates common applications used on their cloud and on-prem environment, enabling an attacker to improve their plan of attack.

As you can imagine, there are many additional sources where employees of an organization share too much sensitive information without realizing the potential impact and damage that can be done if a real threat actor were to leverage such information.

The following are additional data sources for collecting information using passive reconnaissance techniques:

- **Social media platforms** – Many organizations are expanding their online presence and improving their marketing strategy by leveraging social media platforms for sharing information with new and existing customers. In addition, many people share a lot of sensitive information on their personal profiles due to the lack the knowledge on how to improve their account privacy and security. A threat actor can simply identify people who are working for a targeted organization, then identify which users' profiles lack privacy and collect any sensitive information about projects, technologies and any other information associated with the target.
- **Online forums** – Online forums have been around for a very long time and are a common ground for many IT professionals to seek support on issues and help the community. For instance, there are many IT professionals around the world who use online technical forums to seek advice and assistance. However, if a technical employee is experiencing an issue, the person may post about the issue and reveal sensitive details about the infrastructure of the organization to anyone on the internet. Online forums are a common place that's used by malicious actors for obtaining such information.
- **Internet archive** – The Internet Archive is an online digital library that contains past information on the internet. For instance, the Internet Archive takes a snapshot of everything that's on the internet and stores it for the next 20 years. Threat actors can use historical data to identify changes that may have occurred in a targeted

organization. For instance, by analyzing the changes of an organization's website, you can determine how the organization has changed over time such as the type of jobs that were posted and whether there were any technical roles.

- **WHOIS databases** – WHOIS databases are public databases that contain sensitive information about the ownership of a publicly registered domain name. While many domain registrars provide WHOIS privacy for a fee to ensure the domain owner's information is kept private, not everyone pays the extra cost for anonymity. Therefore, threat actors can leverage a person's **Personally Identifiable Information** (**PII**) for identity theft, impersonation, or locating the targeted organization.
- **Geospatial platforms** – There are many geospatial platforms, both free and paid solutions, that are easily accessible over the internet and enable anyone to map and locate physical places, networks, aircraft, and marine vessels. Threat actors can analyze the information collected from the internet, determine the physical address of an organization, and use online mapping solutions to identify the physical surroundings of the targeted organization.

The information collected using passive reconnaissance techniques is commonly referred to as **Open Source Intelligence** (**OSINT**). Ethical hackers and threat actors leverage OSINT to create a profile about their target and improve their attack methodology. However, it's important to always remember that OSINT is not always up-to-date or accurate. For instance, an employee of an organization may post sensitive information about their infrastructure within a forum. However, between the time it was posted and the present, the organization may have resolved all known security issues related to the system, network or application. Therefore, it's essential to perform active reconnaissance on targets.

Active reconnaissance uses a direct approach, whereby the ethical hacker or threat actor directly interacts with the target to retrieve information about the targeted system, network or application. Unlike passive reconnaissance, which relies on OSINT and the likelihood that the information may be outdated, active reconnaissance collects live information from the target. Therefore, the ethical hacker will have the most current version of the information to create a profile of the target.

The following are common techniques used for active reconnaissance:

- Performing **Domain Name System** (**DNS**) interrogation on the target's DNS servers
- Performing various types of social engineering attacks on various groups of employees
- Performing network scans to identify live systems, profiling operating systems and open service ports
- Attempting to enumerate valid user accounts and running services on target systems

As an aspiring ethical hacker, it's essential for you to perform both passive and active reconnaissance on your targets providing it's within the scope and agreement with your client. While this chapter focuses primarily on passive reconnaissance techniques, in the next chapter, we will cover active reconnaissance techniques and tools that's commonly used by ethical hackers. Now that you have learned the general concepts of footprinting, let's see how ethical hackers use organizational information to profile their targets.

Organization information

Organizational information is useful for ethical hackers and threat actors. You can perform active reconnaissance techniques by visiting the target's website to identify and collect any information that's useful to improve your attack methodology.

The following are common information found on organizations' websites:

- Branch locations, addresses, and contact details such as telephone numbers and email addresses. This information is commonly used in social engineering or targeting specific employees.
- Employee details such as names, contact numbers, and email addresses.
- Articles, blog posts and media releases on new and emerging technologies associated with the organization.
- Web links to company resources, intranet, and employee-only portals.
- Partnerships with other organizations.

Details about other branches and their locations are useful as they help identify other locations. Ethical hackers can attempt to compromise a branch location which may have weak security controls and expand their attack to the main office location through the **Site-to-Site VPN** or **Wide Area Network** (**WAN**) connection between the head office and branch locations.

Identifying employees' details on an organization website helps threat actors and ethical hackers easily identify high-profile employees for targeting with social engineering campaigns. In addition, it helps threat actors understand each person's role and responsibility to improve their social engineering attacks. Furthermore, identifying email addresses can help threat actors understand the format the organization uses to create email addresses for employees. They can then leverage such information to further improve their attacks and techniques. The collected email addresses can be used to target either all or specific employees of the organization by sending sophisticated phishing campaigns via email messages. Therefore, threat actors can trick employees into revealing their user credentials or even installing a malicious backdoor onto their systems.

Reading articles, blog posts, and media releases found on an organization's website helps threat actors to understand interests, opportunities, and ventures the target may have an interest in. This information also helps threat actors improve their social engineering techniques to compromise the targeted organization. Threat actors can also attempt to perform a third-party compromise such as a vendor or partner organization that's associated with the target. Third-party exists where a threat actor compromises the vendor or partner organization to steal sensitive information about the target. This method enables the threat actor to bypass any direct security controls implemented by the targeted organization with the intent of exploiting security vulnerabilities found within a vendor's network infrastructure, system, or application.

Having discussed how organizational information can be of advantage to ethical hackers during their security assessments, let's dive deeper into learning how networking information can be useful when creating a digital structure of a target.

Network information

Network information can be obtained using both passive and active reconnaissance techniques on a target. Network information helps ethical hackers and threat actors to gain a better understanding of the network infrastructure and its layout, including live systems and their operating systems, and running services and open ports. Furthermore, ethical hackers and threat actors can better understand the external attack surface, such as the public network blocks used by the targeted organization, IP addresses, public DNS records, and even public-facing systems that are directly connected to the internet.

The following are common network information:

- Domain names, sub-domains, public-facing servers
- IP addresses and network blocks
- WHOIS information about registered domains
- DNS records about the targeted domains

Network information helps ethical hackers to identify both the internal and external attack surface of a targeted organization. Ethical hackers need to identify the vulnerable points of entry into a system, network, or application; this helps ethical hackers to better understand how malicious actors can exploit these vulnerabilities and provide recommendations to help organizations protect their assets.

During the process of registering a public domain name on the internet, the domain registrar commonly provides an option to pay an additional fee to protect the owner's registration information. While many individuals and organizations opt in to pay the additional fees for privacy protection, some do not. For instance, a malicious attacker can leverage public

databases to extract WHOIS details, such as the name, location, organization, and even contact details about the owner of a public domain. Such information could be used to improve the planning of an attack, such as determining the physical location for physical social engineering techniques and contact details for sophisticated phishing campaigns.

The IP addresses and network blocks help ethical hackers to determine the geolocation of publicly available servers and other exposed systems. If an ethical hacker is performing a black-box test, it is important to identify all hidden assets that are within scope and assess them for security vulnerabilities.

Note

You will find more information on white-box, gray-box, and black-box testing here: `https://www.eccouncil.org/cybersecurity-exchange/penetration-testing/black-box-gray-box-and-white-box-penetration-testing-importance-and-uses/`.

Also note, the **Domain Name System** (**DNS**) servers are public servers on the internet that contain records for many domains. Public DNS servers are like large traditional telephone directory books that contain names, addresses, and numbers. Sometimes, IT professionals are working on a project and connect a sub-domain to a test server on the internet but forget to disconnect the test server afterwards. Malicious actors can collect public DNS information to identify hidden systems of a targeted organization. Hence, ethical hackers need to stay ahead to help organizations identify their hidden security vulnerabilities and resolve them as quickly as possible before a real cyber-attack occurs.

Now, let's see how system information comes into play during reconnaissance.

System information

Collecting **system information** is essential during the reconnaissance phase of ethical hacking as it helps ethical hackers to build a profile about the target. For instance, ethical hackers can use system information to determine the type of operating systems and firmware version on a device, running services and open ports, usernames and hidden shares. By leveraging these details, they can research known vulnerabilities on the targeted system, create or acquire exploits, and improve their plan of attack.

This process also helps ethical hackers assess the cyber-risk that is associated with each vulnerability found, its exploitability and potential impact, and the overall severity. Such information helps organizations to prioritize and allocate resources in resolving risks with higher severity levels.

Building our discussion on common types of information that's collected during passive reconnaissance, let's explore common reconnaissance methodologies that are used by ethical hackers.

Reconnaissance methodology

As discussed earlier in the chapter, footprinting helps ethical hackers to better understand the security posture of their target. For instance, you'll be able to determine the placement of network-based security appliances such as firewalls and other security solutions. Using footprinting techniques enables you to identify security vulnerabilities such as misconfigurations, vulnerable running services, unnecessary open ports, and loopholes that can be exploited thereby compromising the confidentiality, integrity, and/or availability of the system.

As you may have realized, ethical hackers need to collect valuable information about their target's network infrastructure and system to identify the attack surface. Therefore, developing a structured footprinting methodology for collecting information about targeted systems, networks, and applications plays an important role in ethical hacking.

Imagine you're performing a black box test on a targeted organization and all that's provided to you is the organization's name. Using proper footprinting techniques enables you to collect valuable information and narrow your focus to specific network blocks, domain and sub-domain names, individual IP addresses of systems, and contact details for employees of the organization. As previously mentioned, such information helps ethical hackers to build a profile about their target and improve their plan of attack.

The following techniques can be used for performing passive reconnaissance:

- Footprinting using internet search engines includes using the advanced Google search parameters (Google hacking/dorking techniques) and leveraging information from search engines that scan and index internet-connected devices such as **Shodan** (`www.shodan.io`).
- Footprinting using internet research services include using the **Internet Archive** (`archive.org`) to receive historical information on organizations, visiting internet forums, blogs, and groups to identify any sensitive information that's leaked by employees, and leveraging dark web search tools with caution to identify any sensitive information that's leaked and can be used to improve your plan of attack.
- Footprinting using social media platforms include platforms that contain many employees' personal and corporate profiles with weak security settings to collect sensitive information. Checking Instagram for photos with sensitive details, Facebook

for posts, pictures and videos, LinkedIn for technical projects and photos, and YouTube for videos on technical projects by employees can also be considered here.

- Performing WHOIS queries by extracting WHOIS details about target organizations and determining physical location of servers and offices by performing geolocation searches

The following techniques can be used for performing active reconnaissance:

- Using DNS interrogation techniques to identify DNS records and hidden systems.
- Performing network and email footprinting by analyzing email headers to trace the source of communication and performing tracerouting to map the target's network infrastructure.
- Impersonating legitimate employees to trick others into revealing sensitive information or performing an action that leads to compromising a system.
- Using automated tools and AI platforms for reconnaissance is optional but can assist when advanced learning is needed. This helps reduce time to efficiently collect sensitive information about targets by using automated tools. You can also develop and improve custom scripts with the help of AI tools to perform active reconnaissance on targeted systems.

You've now learned the importance of performing reconnaissance and footprinting on targeted systems, networks, and applications. Now, you'll be able to further develop your own methodology for collecting valuable information about your targets. In the next section, we will discuss common reconnaissance techniques using common search engines and web services.

Reconnaissance using search engines and web services

Ethical hackers can leverage the information provided by popular and common internet search engines and web services, such as Google, Bing, Yahoo, and others. For instance, ethical hackers can take advantage of using advanced Google search operators to filter and locate specific information about a person, system, network or an organization. The technique of *Google Hacking* is a common misinterpretation, as many folks would think it would involve hacking into Google's infrastructure. However, *Google Hacking* is referred to as *Google Dorking*, such as using advanced Google operations to narrow your searches and results of the Google search engine.

The following syntax is commonly applied when writing an advanced Google search operator:

```
Advanced_search_operator:keyword
```

The following is a list of advanced search operators and their descriptions:

- `site:` This operator enables you filter search results to the website or domain you've specified. For example, using this operator can help ethical hackers to identify sub-domains of a parent domain name.
- `intitle:` This operator enables you to filter results that contain the specific keywords. For instance, ethical hackers can leverage this operator to locate sensitive or exposed systems/directories that's connected to the internet.
- `inurl:` This operator filters the pages that contain the keywords you specified. Using this operator enables you to identify sensitive directories or files that may be exposed on the internet.
- `intext:` This operator helps display the search results that contain the keyword within the body of a webpage. Ethical hackers use this to identify webpages on the internet that contains a specific keyword.
- `cache:` Enables you to display the cache version of a website. This operator is useful when seeking historical information on websites and webpages on the internet.
- `link:` Helps ethical hackers to filter websites that contain a specific link.
- `info:` This operator is commonly used to locate information that is either similar or related to the specified website or webpage.
- `location:` This operator helps ethical hackers to identify information based on a specific geolocation.
- `filetype:` Enables you to filter search results to locate webpages that contain the specific file type.
- `before:` Helps ethical hackers filter search results before the specified date.
- `after:` Enables ethical hackers to filter search results after the specific date.

The following list provides various combinations of using advanced Google search operators:

- Discover webcams - `intitle:"webcamXP" inurl:8080`
- Discover sensitive directory with SSH data - `intitle:"index of" ".ssh"`
- Discover Excel files with passwords - `filetype:xls "password"`

- Discover text files with SSH private keys - `filetype:txt "ssh" "private key"`
- Discover configuration files - `ext:ini | ext:cfg | ext:conf "db_password"`

Ethical hackers can leverage the preceding list of advanced search operators to locate the following on the internet:

- Identify exposed systems and networks
- Locate sensitive directories
- Identify login portals
- Identify software versions and source code
- Locate connected **Internet of Things** (**IoT**) and **Industrial Control Systems** (**ICS**)
- Gain access to unsecured networking and security appliances

Figure 3.2 shows an example of using advanced Google search operators to locate PDF files on webpages that contain the keyword 'cybersecurity' on the CompTIA domain:

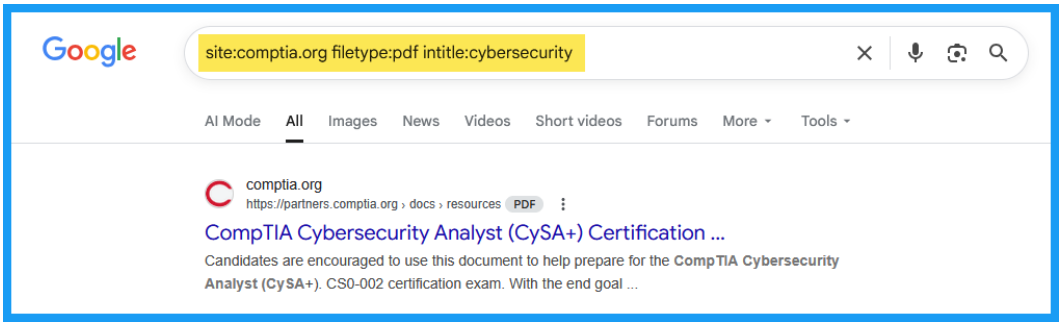


Figure 3.2 – Advanced search operators in action

As shown in the figure, ethical hackers can use a combination of different advanced search operators with keywords to filter their search results. This enables them to focus on discovering sensitive directories and login pages that are associated with their target.

Note

The Exploit Database contains an active repository of advanced Google search operators which can be found here: `https://www.exploit-db.com/google-hacking-database`.

Over the next few sections, we will discuss common techniques and procedures used by ethical hackers to identify sensitive directories, sub-domains, exposed systems, and use geolocation to

find places on the internet. For now, let's dive into the hands-on labs on performing Google dorking techniques to extract sensitive information from the Google Search platform.

Lab: Using Google dorking techniques to locate hidden systems

In this exercise, you will learn common Google dorking techniques for identifying hidden systems and sensitive information that's either intentionally or unintentionally exposed by organizations around the world.

To get started with this exercise, please use the following instructions:

1. Using a computer with internet connectivity, open your preferred web browser and head over to the Google Search website via `https://www.google.com/`.
2. To discover login portals that have the keywords `Admin Login` and use `admin` as part of its **Uniform Resource Locator** (**URL**), use the `intitle:"Admin Login" inurl:admin` syntax, as shown:

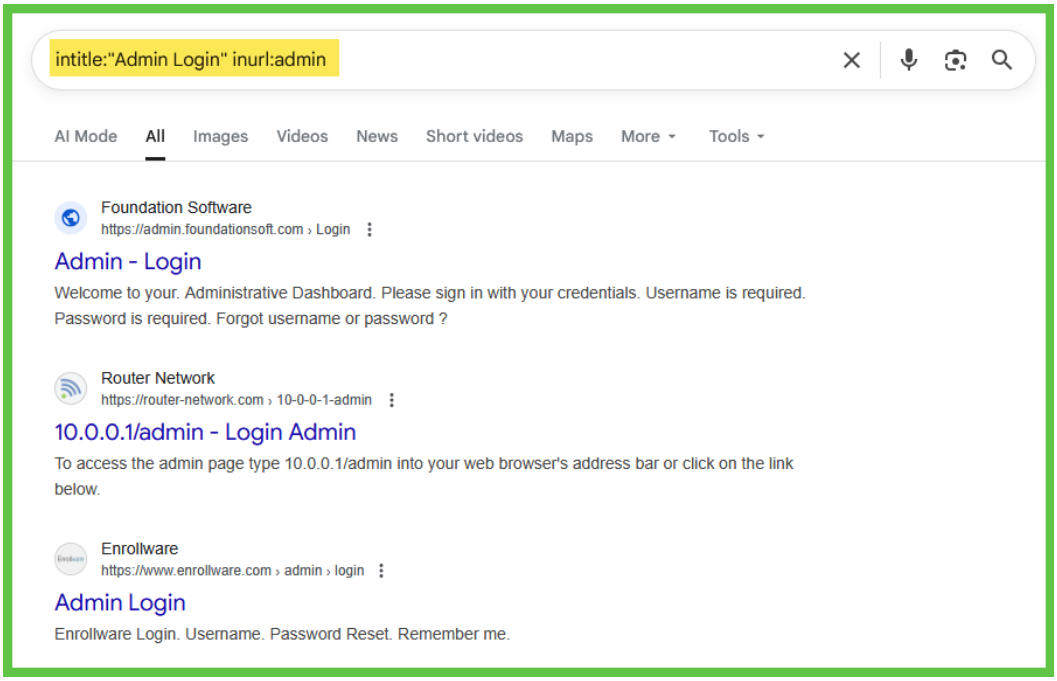


Figure 3.3 – Admin portals

3. Another search operator such as `inurl:/admin/login.jsp` syntax enables ethical hackers to discover exposed login portals, as shown:

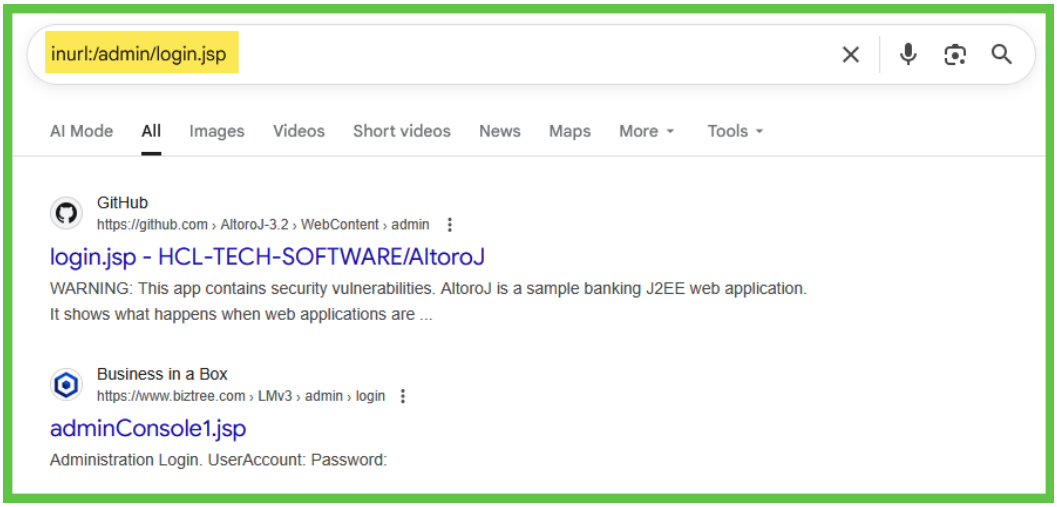


Figure 3.4 – Exposed admin portals

4. To discover exposed databases such as ElasticSearch, use the `intitle:"kibana" inurl:app/kibana` syntax:

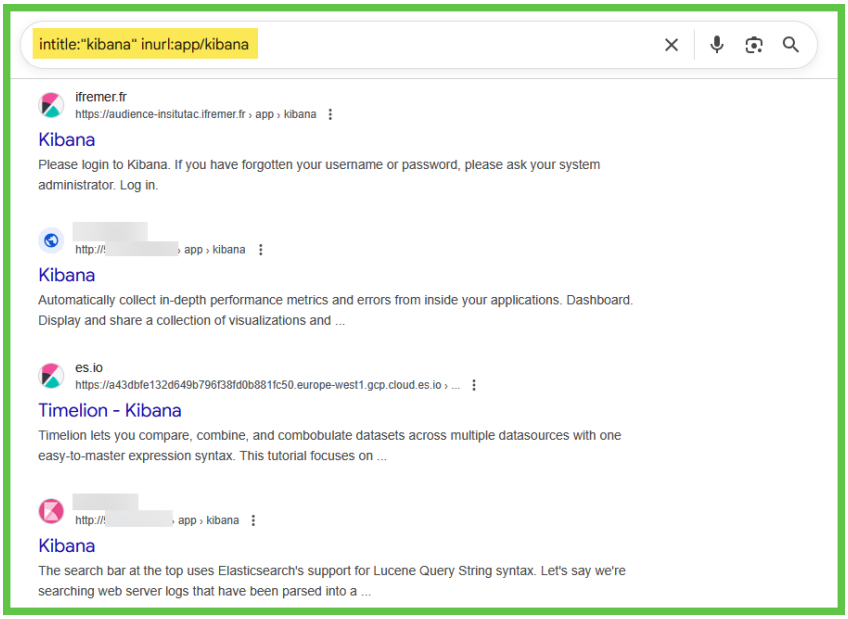


Figure 3.5 – Exposed database

5. To discover exposed home routers with their login pages on the internet, use the `inurl:8080 intext:"router login"` syntax:

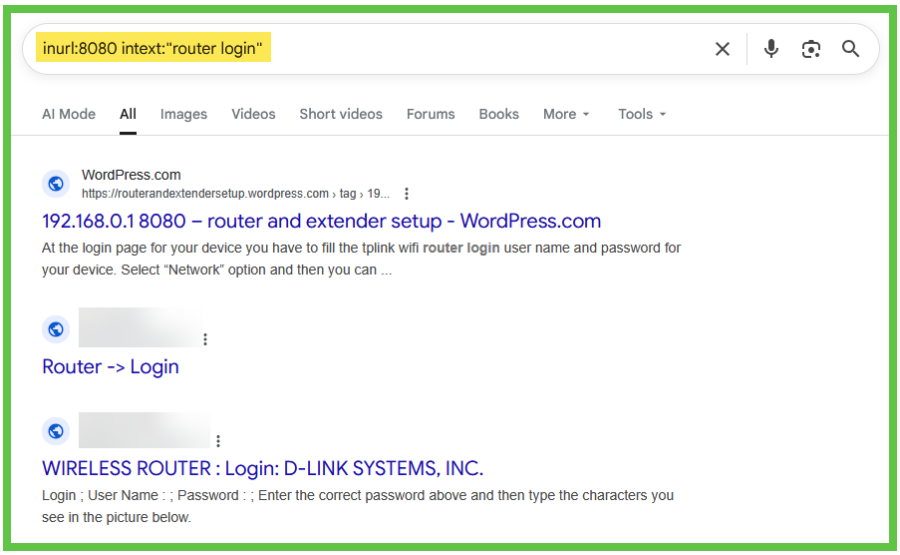


Figure 3.6 – Exposed home routers

6. To discover exposed IoT devices with their login pages, use the `intitle:"IoT" inurl:login` syntax:

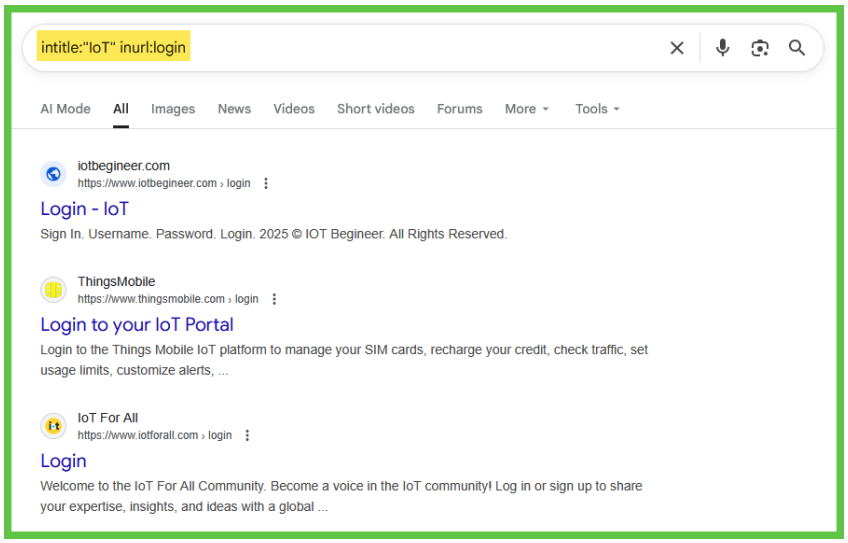


Figure 3.7 – Expose IoT devices

7. To discover exposed **Supervisory Control and Data Acquisition** (**SCADA**) systems that use port `8080` on the internet, use the `intitle:"SCADA" inurl:8080` syntax:

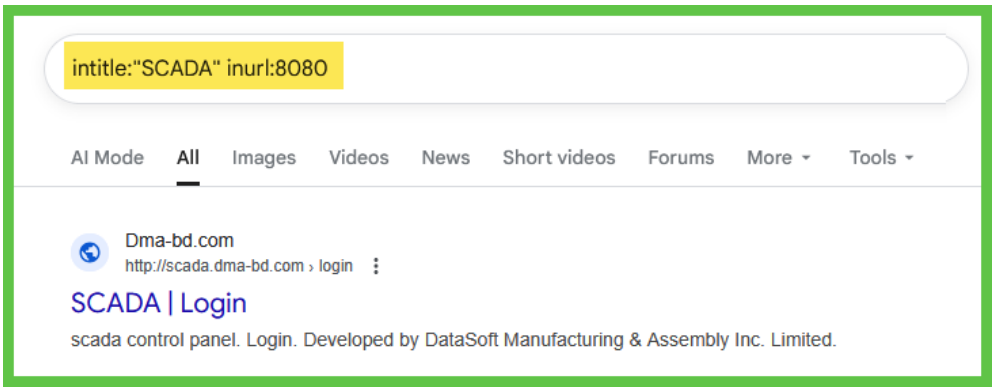


Figure 3.8 – Exposed SCADA systems

8. To discover sensitive files such as data backups on the internet, use the `intitle:"index of" "backup` syntax:

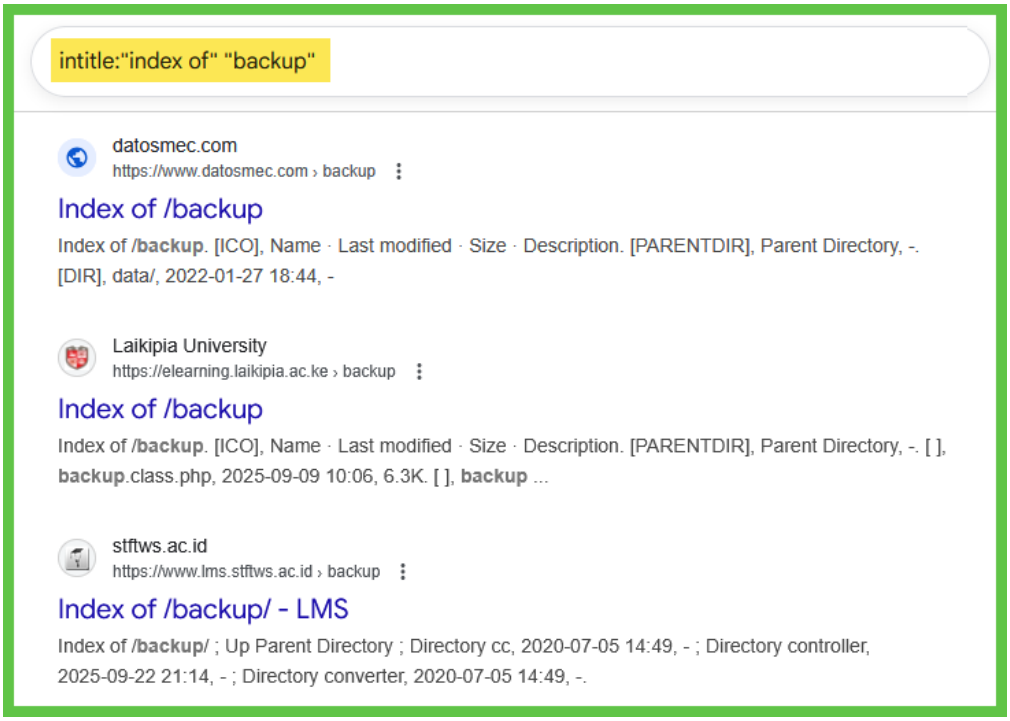


Figure 3.9 – Data backup directories

As you have learned from this lab, using various combinations of Google advanced search operators can help ethical hackers to identify exposed systems, hidden directories, and gain a better understanding of the external attack surface of their target. In the next lab, you will learn how to use common tools and techniques to identify subdomains and exposed assets.

Lab: Identifying organization's subdomains and exposed assets

In this exercise, you will learn how to use common ethical hacking tools to identify subdomains of an organization and locate exposed assets on the internet using a combination of online services and automated reconnaissance tools. First, you will use Netcraft to profile an organization's public infrastructure. Next, you will use Subfinder and DNSx to enumerate and validate subdomains. Finally, you will explore online services that can help automate the subdomain enumeration process.

Part 1: Using Netcraft to profile the public infrastructure of an organization

Netcraft is a popular online website that enables anyone, such as ethical hackers and cybersecurity researchers, to track the underlying technologies that are associated with public websites and better understand the type of data that is being exposed on the internet by the organizations that manage such websites.

Ethical hackers can query any publicly registered domain name using Netcraft's site report tool to determine the background, network information, IP delegation, and additional information about the domain. Such information is useful in creating a profile of a targeted organization as an ethical hacker.

To get started, follow these steps:

1. Using a computer with internet connectivity, whether your host operating system or the Kali Linux virtual machine, open your preferred web browser and go to: `https://sitereport.netcraft.com/`.
2. Choose a targeted domain name of an organization and enter it within the search field and click on **LOOK UP**:



Figure 3.10 – Performing a domain lookup

3. Netcraft will then provide all the information it knows about the domain, such as the network details:

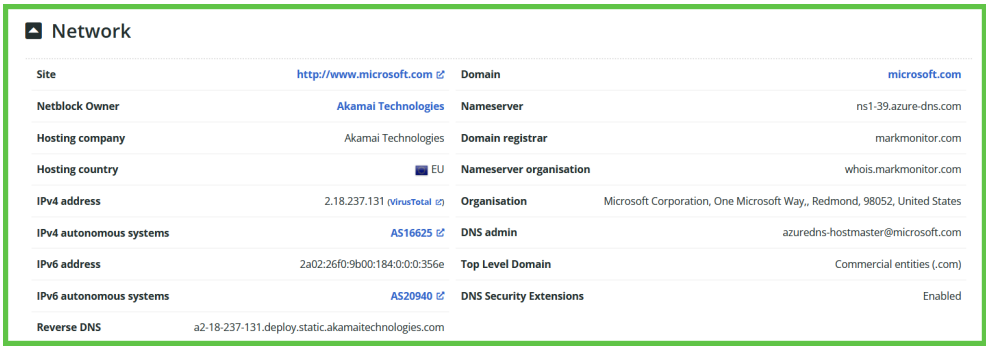


Figure 3.11 – Network information

As shown in the preceding screenshot, you can determine the owner of the network block, that is, the organization responsible for managing the IP address used by the domain or hosting provider. The location of the hosting company and IP addressing information is useful in determining whether you're targeting the accurate website. For instance, during a black box test, the ethical hacker is usually provided with very little information about the target, such as only the organization's name.

Hence, it's quite probable to mistake an incorrect domain name with your target. As you can imagine, it's easy to get into legal issues if you were to begin scanning or attempt to exploit a vulnerable on the wrong domain name. Therefore, the network

infrastructure helps ethical hackers to build a profile about the target and provides accuracy of the target.

Moreover, the network infrastructure details assist ethical hackers with mapping the external attack surface and better understanding what threat actors can view on the internet.

4. Then, Netcraft provides **Secure Sockets Layer/Transport Layer Security (SSL/TLS)** details about the targeted domain:

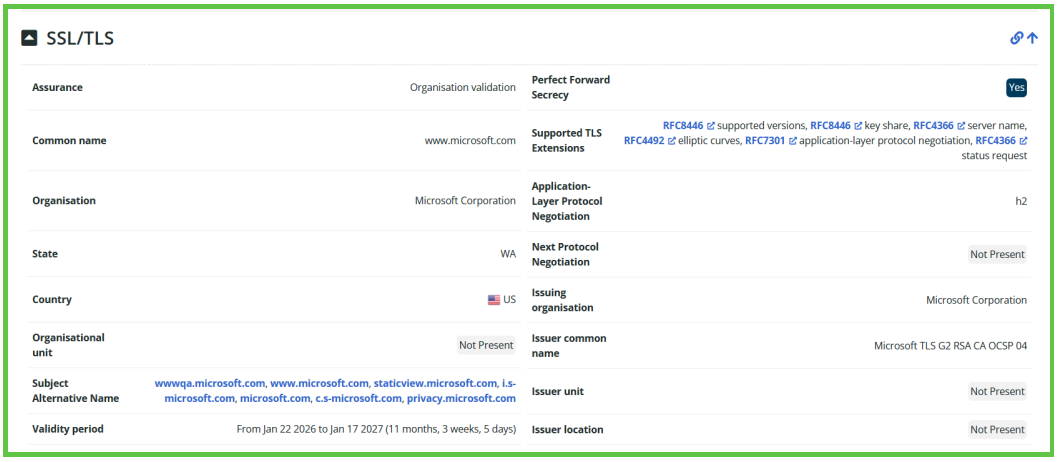


Figure 3.12 – SSL/TLS details

The information found within an SSL/TLS digital certificate can be used to validate the identity of the website and its ownership. For instance, the preceding screenshot shows whether the digital certificate is valid for alternative or additional domains and sub-domains. This is useful to help ethical hackers identify hidden or unknown sub-domains of a target.

Additionally, there are many web administrators who don't update their security configurations and encryption technologies with industry best practices, which leaves their systems and applications vulnerable to cyber-attacks and threats. Hence, the SSL/TLS determines can help ethical hackers to determine whether there are weak encryption algorithms or misconfigurations with the digital certificates and the website and provide recommendations to resolve them before a real threat actor exploits these vulnerabilities.

5. Additionally, Netcraft provides information on additional technologies and components used on the targeted website:

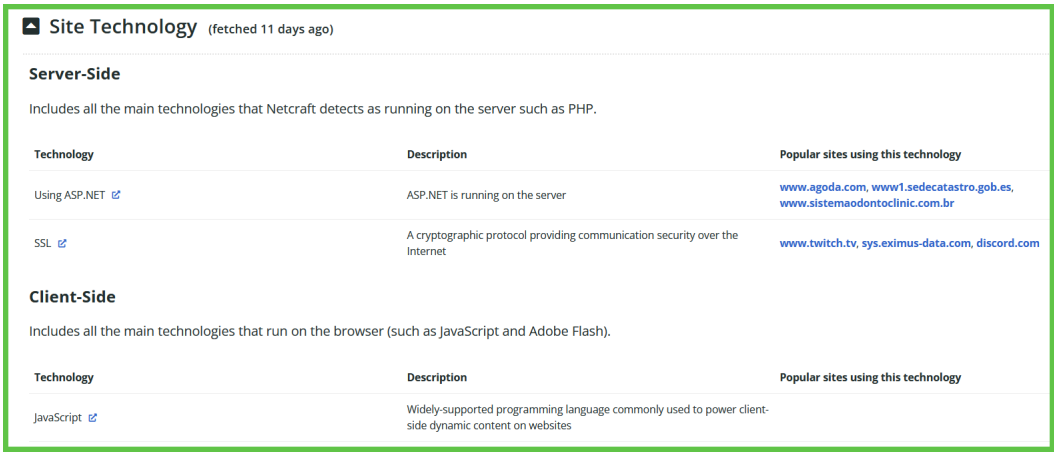


Figure 3.13 – Site technologies

The site technologies help ethical hackers identify whether there are vulnerable components used on the website and evaluate the risk and potential impact should a vulnerability be exploited by a real threat actor.

While it's fun to use Netcraft for performing reconnaissance and footprinting of a target, it's important to remember the various limitations of using passive reconnaissance such as outdated information and false positives of sensitive information.

Part 2: Using Subfinder and DNSx to enumerate subdomains

Subfinder is an open-source reconnaissance tool that enables ethical hackers to perform passive sub-domain enumeration on a targeted organization. However, subfinder provides the hostnames only. Therefore, **dnsX** enables you to automate the process to resolve each hostname to a valid IP address. In this exercise, you will learn to leverage these tools to automate your workflow for enumerating sub-domains.

To get started with this exercise, follow these instructions:

1. Power-on the **Kali Linux** virtual machine and login to the desktop interface. The default user credentials are username = `kali` and password = `kali`. However, if you changed the default password during the previous chapter, you'll need to enter it at this step.
2. Then, open the **Terminal** and use the following commands to install both Subfinder and DNSx:

```
kali@kali:~$ sudo apt install subfinder
kali@kali:~$ sudo apt install dnsx
```

3. Now use Subfinder to passively enumerate the subdomains of a targeted domain and output the results in a file:

```
kali@kali:~$ subfinder -d microsoft.com -o subfinder.txt
```

The following is a breakdown of the preceding commands:

- `subfinder -d <domain>`: enables you to specify a domain
- `-o <filename>`: sends the results to an output file

At this stage, you should see a list of subdomains/IPs indicating successful enumeration. The following screenshot shows the execution of the preceding commands:

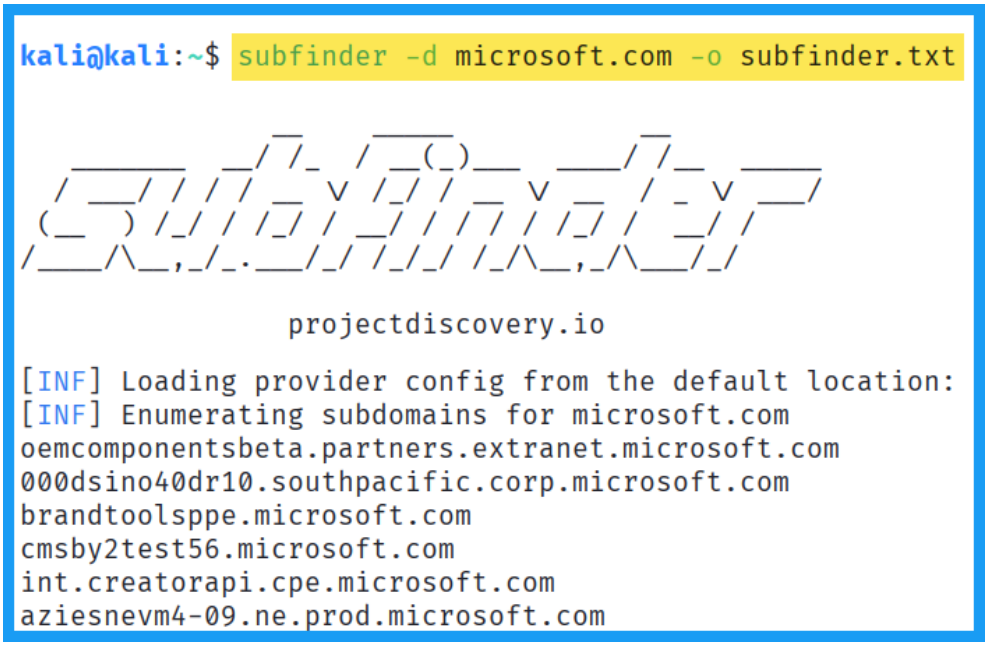


Figure 3.14 – Subfinder

Tip

To view the contents of the `subfinder.txt` file, use the `cat subfinder.txt` command.

4. Use the following command to automate the process for resolving all the hostnames within the `subfinder.txt` file to IP addresses and output the results in a new file:

```
kali@kali:~$ dnsx -l subfinder.txt -resp -o alive.txt
```

The following is a breakdown of the preceding commands:

- `dnsx -l`: specifies a list of domains to resolve from a file
- `-resp`: displays the DNS responses
- `-o`: specifies an output file for the results

The following screenshot shows the execution of the preceding commands:

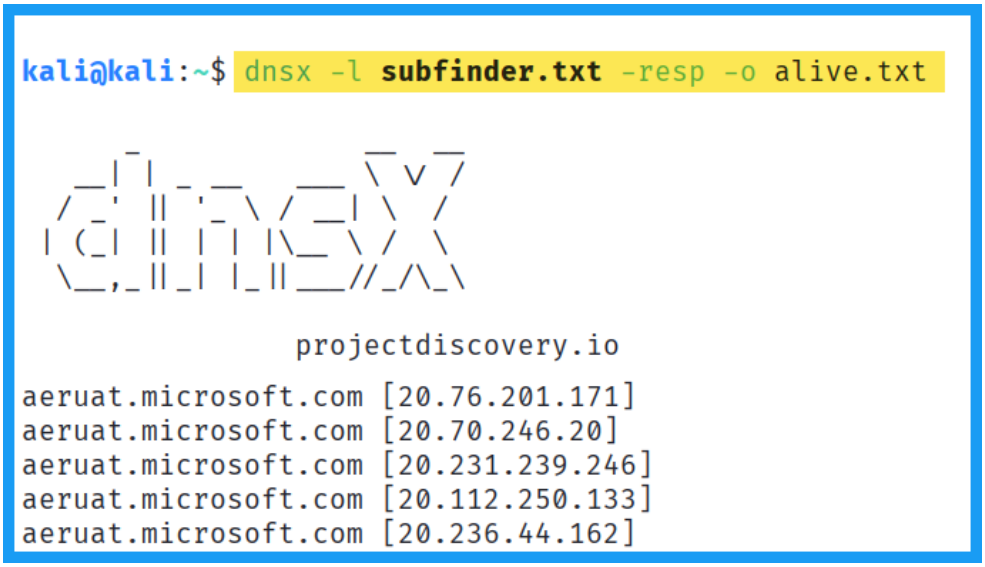


Figure 3.15 – Dnsx

5. Lastly, use the `cat alive.txt` command to display the results of all live sub-domains and their associated IP addresses:

```
kali@kali:~$ cat alive.txt
aeruat.microsoft.com [20.76.201.171]
aeruat.microsoft.com [20.70.246.20]
aeruat.microsoft.com [20.231.239.246]
aeruat.microsoft.com [20.112.250.133]
aeruat.microsoft.com [20.236.44.162]
alerts.microsoft.com [20.76.201.171]
alerts.microsoft.com [20.231.239.246]
aepuat.microsoft.com [52.164.206.56]
aepuat.microsoft.com [104.215.95.187]
```

Figure 3.16 – Live hosts

Part 3: Working with online tools to automate subdomain enumeration

As an aspiring ethical hacker, it's essential to automate many of your workflows and processes to better leverage the time allocated for performing security testing on a target. To do this, you can leverage trusted online services that can help automate the enumeration of identified sub-domains of a targeted organization. **Pentest Tools** is an online service that allows cybersecurity professionals to do just this, automating essential ethical hacking tasks using their platform.

Follow these steps to get started with this exercise:

1. Using a computer with internet connectivity or the Kali Linux virtual machine, open the web browser and go to: `https://pentest-tools.com/information-gathering/find-subdomains-of-domain`.
2. Make sure the **Light scan** tab is selected and enter a publicly registered domain within the **Target** field, then click on **Start scan**:

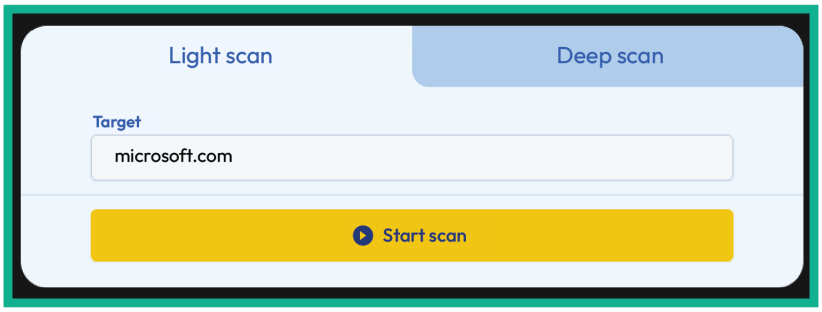


Figure 3.17 – Light scan

The online will run for a few minutes and display the results as shown:

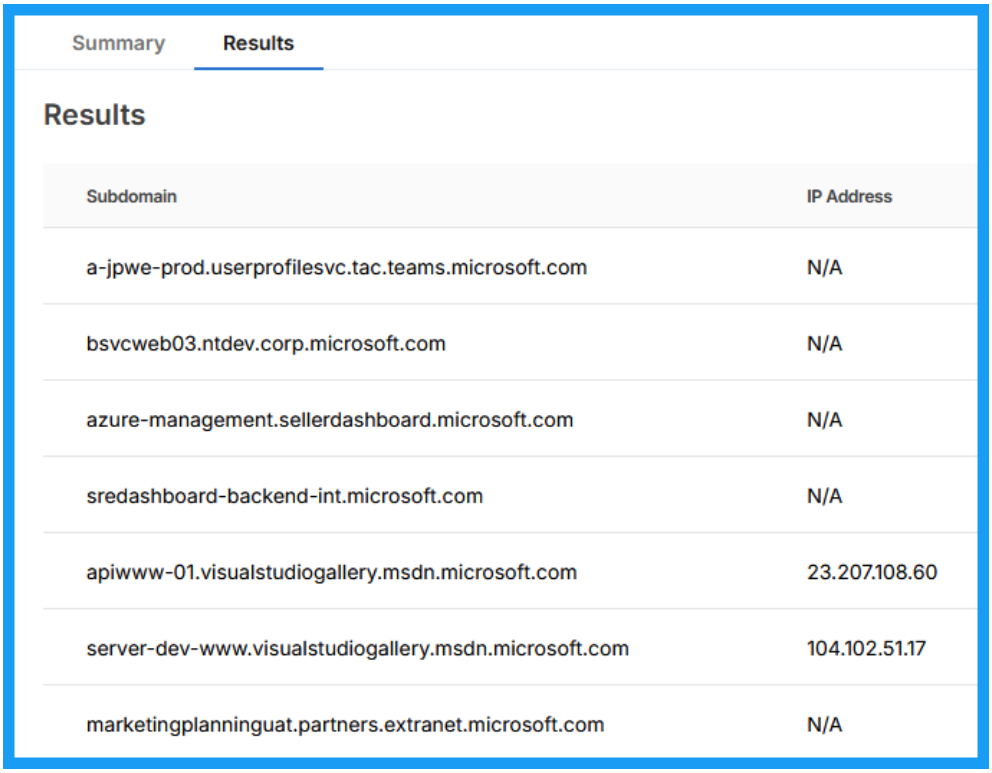


Figure 3.18 – Sub-domain enumeration

As shown in the preceding screenshot, this online tool enables ethical hackers to both automate and passively gather sub-domain information during their reconnaissance phase of ethical hacking. Additionally, this tool attempts to identify the IP address that is associated with a sub-domain. Such information helps ethical hackers to identify the attack surface and exposed assets of a targeted organization.

While some ethical hackers prefer manual techniques to discover sub-domains of a targeted domain, using automated tools reduces the time it takes to identify hidden sub-domains. In the next exercise, you will learn how to leverage geospatial data to locate interesting subjects.

Lab: Using geospatial data to locate an object

Ethical hackers can leverage geospatial data to locate both physical and digital assets of a targeted organization. For instance, geospatial information enables you to gain a better understanding of where systems, devices, and people are in the real world.

Ethical hackers can use geospatial data to locate, but are not limited to, the following:

- Data centers
- Organizations' building
- External camera systems
- Wireless leakage area/zone
- Industrial sites that use **Industrial Control Systems** (**ICS**)

To get started with this exercise, follow the instructions:

1. Power-on the **Kali Linux** virtual machine and log in.
2. Open the **Mozilla Firefox** web browser and go to `https://osint.tools/exif-examples`, click on the **EXIF Example 1** and download the file:

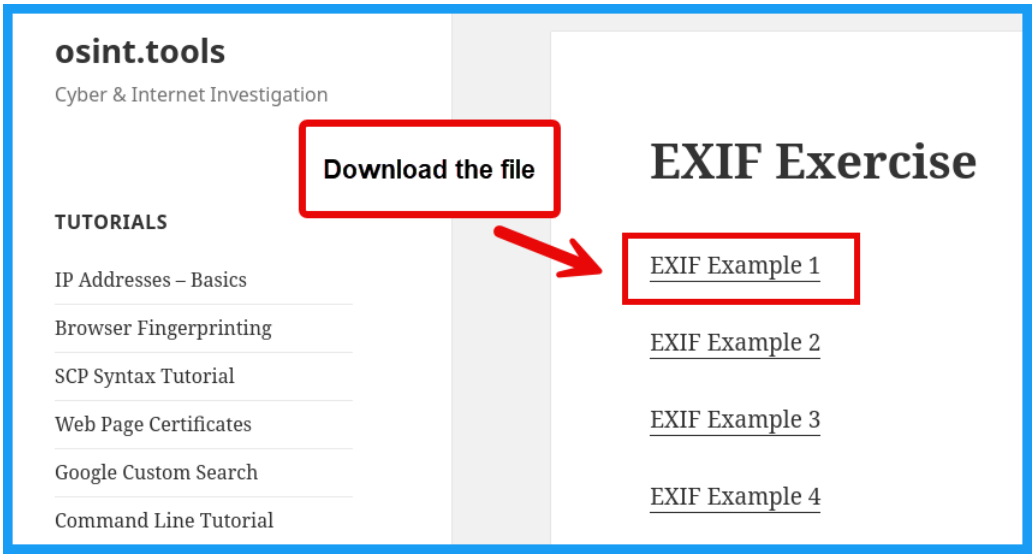


Figure 3.19 – OSINT Tools website

3. Make sure you save the file within the Downloads directory/folder within Kali Linux. Once downloaded, open the file to view the image:



Figure 3.20 – Picture of a storefront

4. Then, open the **Terminal** and use the **ExifTool** to display the meta information found within the image file:

```
kali@kali:~$ exiftool Downloads/exif1.jpg
```

The following screenshot shows the make and model of the camera, the resolution, and timestamps:

```
kali@kali:~$ exiftool Downloads/exif1.jpg
ExifTool Version Number         : 13.25
File Name                       : exif1.jpg
Directory                       : Downloads
File Size                       : 1799 kB
File Modification Date/Time     : 2025:11:23 17:57:19-05:00
File Access Date/Time           : 2025:11:23 17:57:36-05:00
File Inode Change Date/Time     : 2025:11:23 17:57:19-05:00
File Permissions                : -rw-rw-r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Exif Byte Order                 : Big-endian (Motorola, MM)
Make                            : Apple
Camera Model Name               : iPhone XS Max
Orientation                     : Horizontal (normal)
X Resolution                    : 72
Y Resolution                    : 72
```

Figure 3.21 – Metadata from image

5. Scroll to the end of the output and you will see the **Global Positioning System** (**GPS**) data:

```
Create Date                     : 2019:04:07 16:22:36.265
Date/Time Original              : 2019:04:07 16:22:36.265
GPS Altitude                    : 0 m Below Sea Level
GPS Date/Time                   : 2019:04:07 12:22:34.99Z
GPS Latitude                    : 25 deg 16' 6.50" N
GPS Longitude                   : 55 deg 18' 12.99" E
Circle Of Confusion             : 0.005 mm
Field Of View                   : 69.4 deg
Focal Length                    : 4.2 mm (35 mm equivalent: 26.0 mm)
GPS Position                    : 25 deg 16' 6.50" N, 55 deg 18' 12.99" E
Hyperfocal Distance             : 2.04 m
Light Value                     : 12.2
Lens ID                         : iPhone XS Max back dual camera 4.25mm f/1.8
```

Figure 3.22 – GPS data

6. Use the following commands to extract the GPS data and convert it into coordinates:

```
kali@kali:~$ exiftool -n -gpslatitude -gpslongitude Downloads/exif1.jpg
```

The following screenshot shows the extraction and conversion of GPS data into coordinates:

```
kali@kali:~$ exiftool -n -gpslatitude -gpslongitude Downloads/exif1.jpg
GPS Latitude                    : 25.2684722222222
GPS Longitude                   : 55.3036083333333
```

Figure 3.23 – GPS coordinates

7. Append the preceding GPS coordinates to the end of `https://www.google.com/maps?q=` to create this hyperlink for Google Maps: `https://www.google.com/maps?q=25.2684723,55.3036083`.
8. Click on the **Google Maps** link to view the location found in the picture:

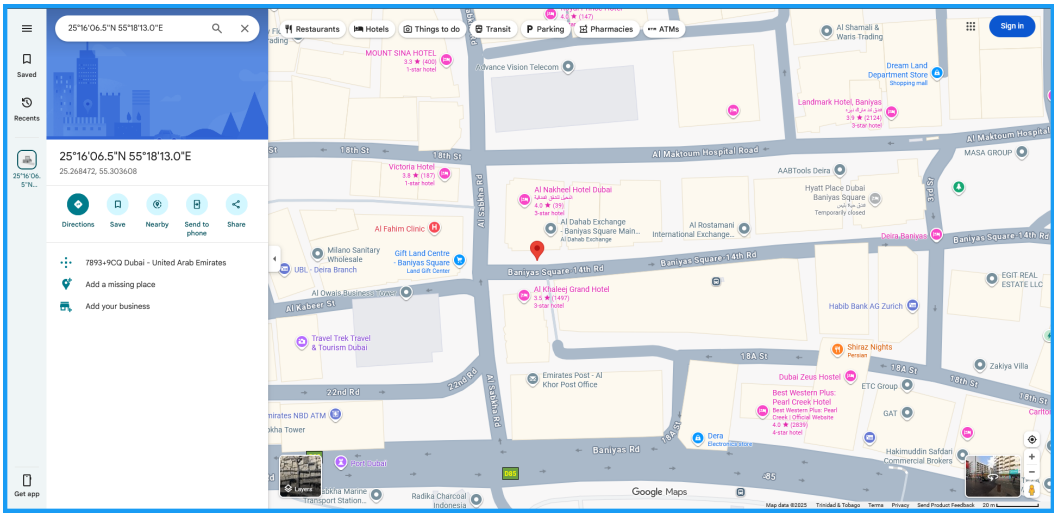


Figure 3.24 – Google Maps

A red pin in the figure marks the location of the picture.

9. On **Google Maps**, enable **Street View** and click closest to the red pin:

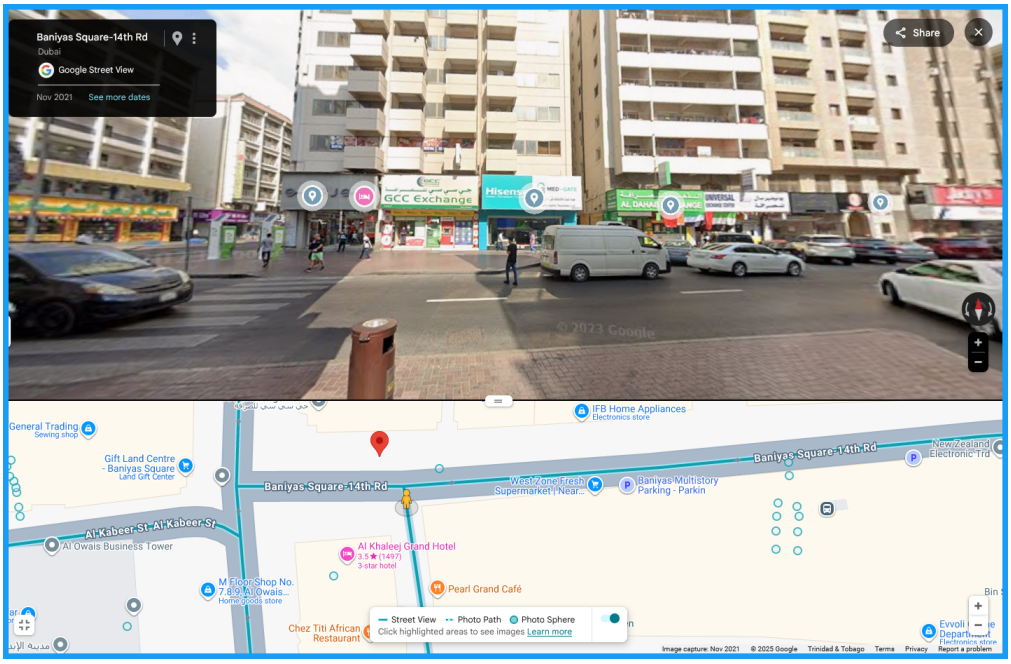


Figure 3.25 – Street View

As shown in the figure, the target has been located by leveraging the imagery solution within Google Maps. Additionally, ethical hackers can navigate around a location using Street View to identify external camera systems, physical points of entry, and gain a better understanding of the vicinity.

Having completed this lab, you now understand how to leverage geospatial data and use public imagery services to locate a target. In the next section, you will learn about websites and email reconnaissance.

Email and web footprinting

Email headers are important for ethical hackers as they contain sensitive information for traceability. For instance, email headers contain hidden technical details that allow ethical hackers to perform footprinting on systems that processed and forwarded the email message. Furthermore, it helps other cybersecurity professionals such as the **Digital Forensics and Incident Response** (**DFIR**) team during investigation to determine the source of an attack or threat as it relates to social engineering.

The following details can be found within an email header:

- `Received:` The sender's email server
- `X-Originating-IP:` The sender's IP address
- `Return-Path:` The route taken by email message

By performing email header analysis, you can reconstruct the timeline of events of an email-based threat. Such as tracking malicious between email mailboxes and relay servers, gaining a better understanding of any compromised email relay paths, and mapping the external attack surface for further investigation.

Popular internet search engines help many people to locate websites on the internet. However, there are specialized search engines that scan for internet-connected systems such as IoT, **Operational Technology** (**OT**), and even exposed systems. Ethical hackers leverage these specialized search engines to identify any exposed systems associated with their targets. In addition, they can determine the services that are exposed on each system, whether sensitive directories and data are being leaked, and the potential impact if a real threat actor were to compromise those devices.

The following are specialized search engines:

- Shodan - `www.shodan.io`
- Censys - `https://censys.com`

Shodan enables ethical hackers to perform passive reconnaissance and gather OSINT on targeted systems, devices, and organizations. Such information is used to build a profile about the target and determine security vulnerabilities, which can later be exploited during a security assessment.

The following screenshot shows the number of exposed systems on the internet that are running the **Remote Desktop Protocol** (**RDP**):

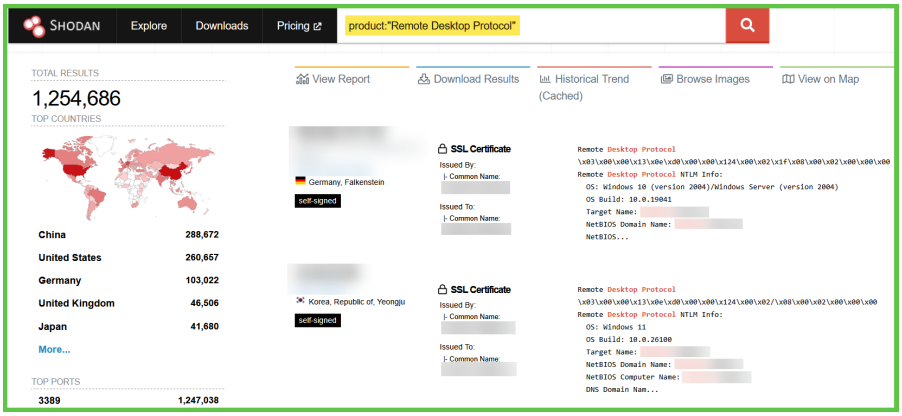


Figure 3.26 – Shodan

Additionally, Censys is a specialized search engine that helps ethical hackers and organizations with gathering intelligence on their exposed systems and devices on the internet. Similarly to Shodan, Censys scans the internet to identify any online devices and collect any data that's found on them. Such information helps cybersecurity professionals such as ethical hackers better understand the external attack surface and security vulnerabilities of an organization.

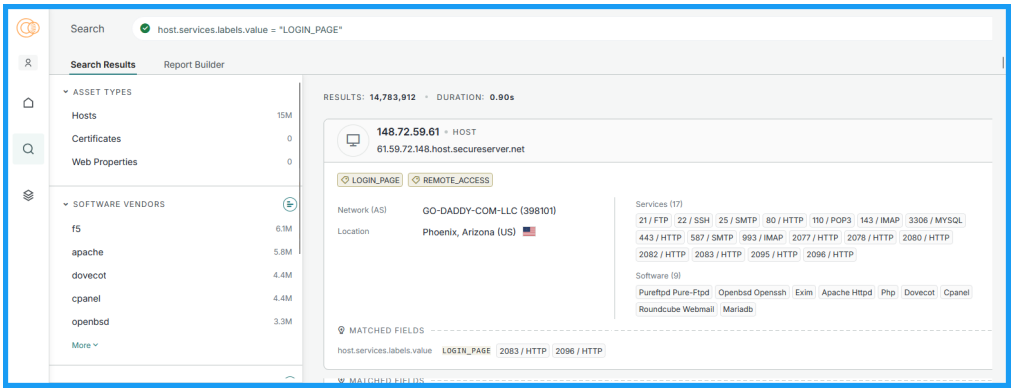


Figure 3.27 – Censys

The figure shows a filtered list of all exposed systems that display a login page on the internet.

Over the next few sections, you will learn how to leverage information found within emails to trace the source of the message and use specialized search engines and tools to collect sensitive information about targets.

Lab: Performing email footprinting

In this exercise, you will learn how to extract the header information found within each email message and trace the source of a message. For this lab, you will need a free email account such as Gmail, Outlook Yahoo Mail or even ProtonMail.

Follow these steps to get started with this exercise:

1. Using a computer with internet connectivity, open your preferred web browser and log in to your free email account.
2. To view the email header on Outlook, select an email message and click on the 3-dots icon (…)| select **View | View message source**:

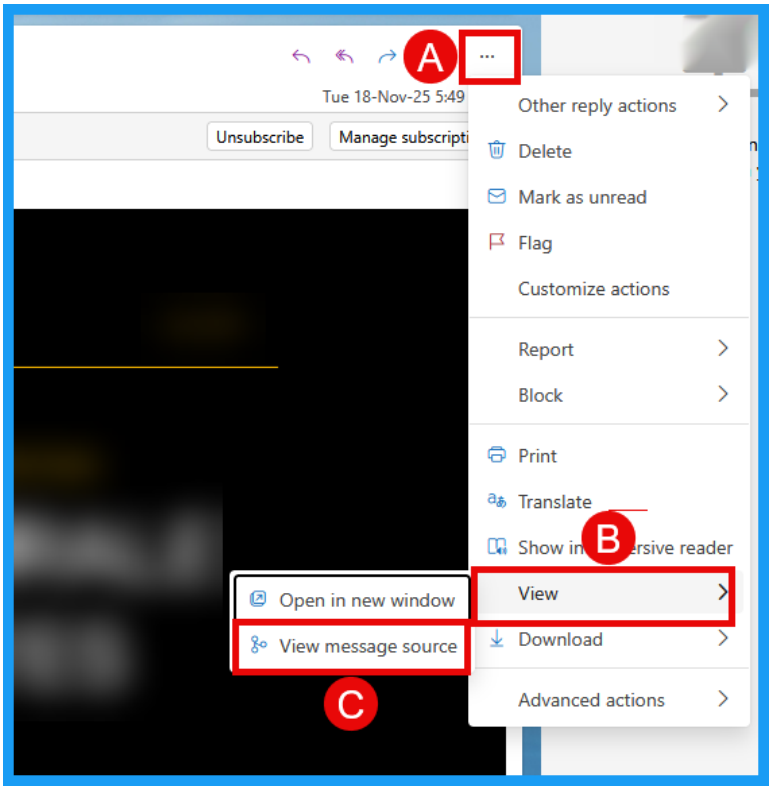
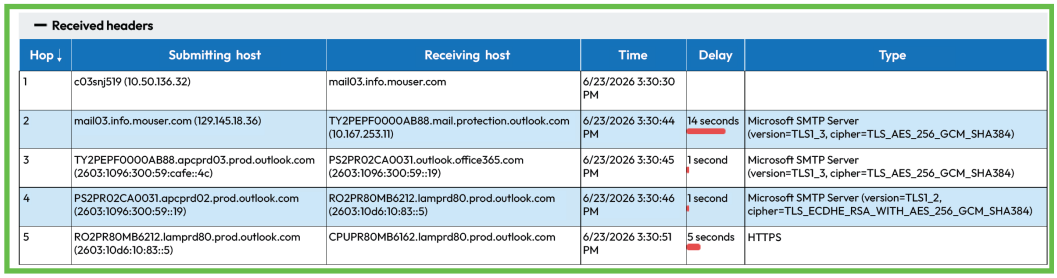


Figure 3.28 – Outlook message menu

3. The **Message source** window will appear with the email header; copy all the information.
4. Go to `https://mha.azurewebsites.net/` and paste the contents, then click **Analyze headers**:



— Received headers

Hop ↓	Submitting host	Receiving host	Time	Delay	Type
1	c03snj519 (10.50.136.32)	mail03.info.mouser.com	6/23/2026 3:30:30 PM		
2	mail03.info.mouser.com (129.145.18.36)	TY2PEPF0000AB88.mail.protection.outlook.com (10.167.253.11)	6/23/2026 3:30:44 PM	14 seconds	Microsoft SMTP Server (version=TLS1_3, cipher=TLS_AES_256_GCM_SHA384)
3	TY2PEPF0000AB88.apcprd03.prod.outlook.com (2603:1096:300:59:cafe::4c)	PS2PR02CA0031.outlook.office365.com (2603:1096:300:59::19)	6/23/2026 3:30:45 PM	1 second	Microsoft SMTP Server (version=TLS1_3, cipher=TLS_AES_256_GCM_SHA384)
4	PS2PR02CA0031.apcprd02.prod.outlook.com (2603:1096:300:59::19)	RO2PR80MB6212.lamprd80.prod.outlook.com (2603:10d6:10:83::5)	6/23/2026 3:30:46 PM	1 second	Microsoft SMTP Server (version=TLS1_2, cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384)
5	RO2PR80MB6212.lamprd80.prod.outlook.com (2603:10d6:10:83::5)	CPUPR80MB6162.lamprd80.prod.outlook.com	6/23/2026 3:30:51 PM	5 seconds	HTTPS

Figure 3.29 – Header analysis

The preceding screenshot shows the path taken by the email message before arriving at its destination.

> **Note**
>
> Nowadays, many email providers have implemented an additional layer of privacy for senders by masking their real source IP address in an email header. Therefore, it's not always possible to identify a sender's real source address unless their email application does not mask their public IP address. Additionally, MXToolBox provides a free Email Header Analyzer tool at this link: `https://mxtoolbox.com/EmailHeaders.aspx`.

With this exercise, you have learned how to extract email headers and automate the analysis phase to quickly identify sensitive information for traceability. In the next lab, you will learn how to use specialized search engines to discover exposed systems on the internet.

Lab: Identifying sensitive systems using deep web search

In this exercise, you will learn how to leverage Shodan to discover exposed systems, running services, and open ports. To get started, please use the following instructions:

1. Using a computer with internet connectivity, open your preferred web browser and go to `https://www.shodan.io/` to sign-up for a free account. Keep in mind, there are many limitations to using a free account on this platform.
2. After your account has been created, log in to the Shodan platform.

3. In the Shodan search bar, enter the `os:"Windows Server" product:"Remote Desktop Protocol"` search criteria to discover all Windows Server operating systems with RDP exposed on the internet, as shown:

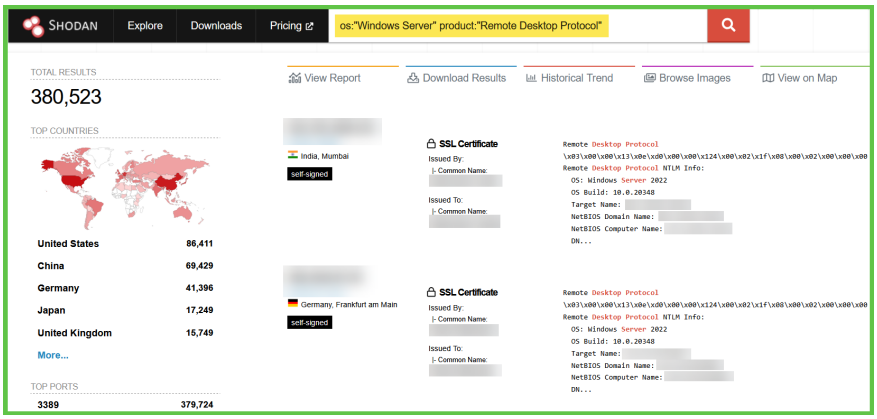


Figure 3.30 – Discovering Windows Servers with RDP

4. Select a device to gain more information such as open ports, running services and even a list of known security vulnerabilities:

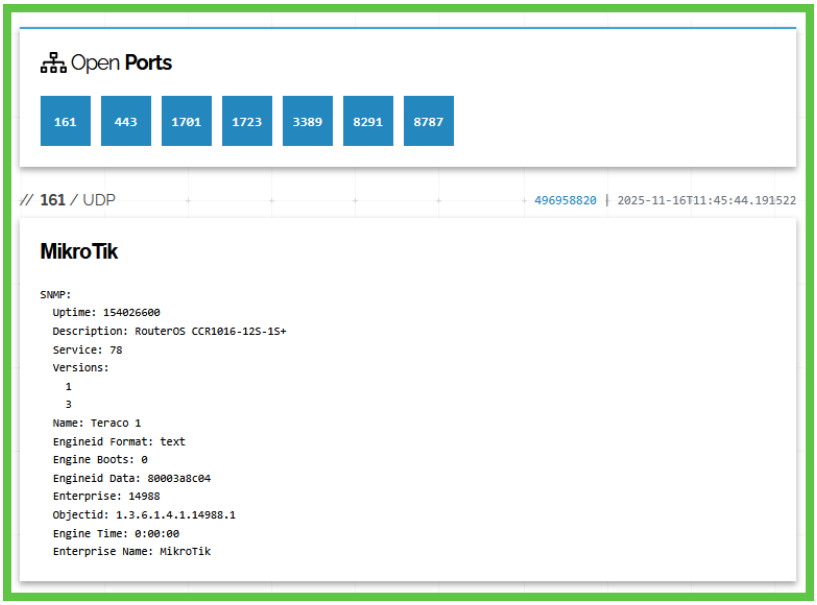


Figure 3.31 – Exposed ports and services

As shown in the figure, ethical hackers can easily identify open ports and determine what's running on a host by leveraging the data collected by Shodan. Some services leak sensitive information that helps ethical hackers to determine the host operating system, service versions, and even security vulnerabilities due to misconfigurations.

Note

Each service or application-layer protocol is usually mapped to a port number. To learn more about service names and their associated port numbers, please see: `https://www.shodan.io/search/filters`.

5. Shodan has the capabilities of identifying security vulnerabilities found in operating systems and software components:

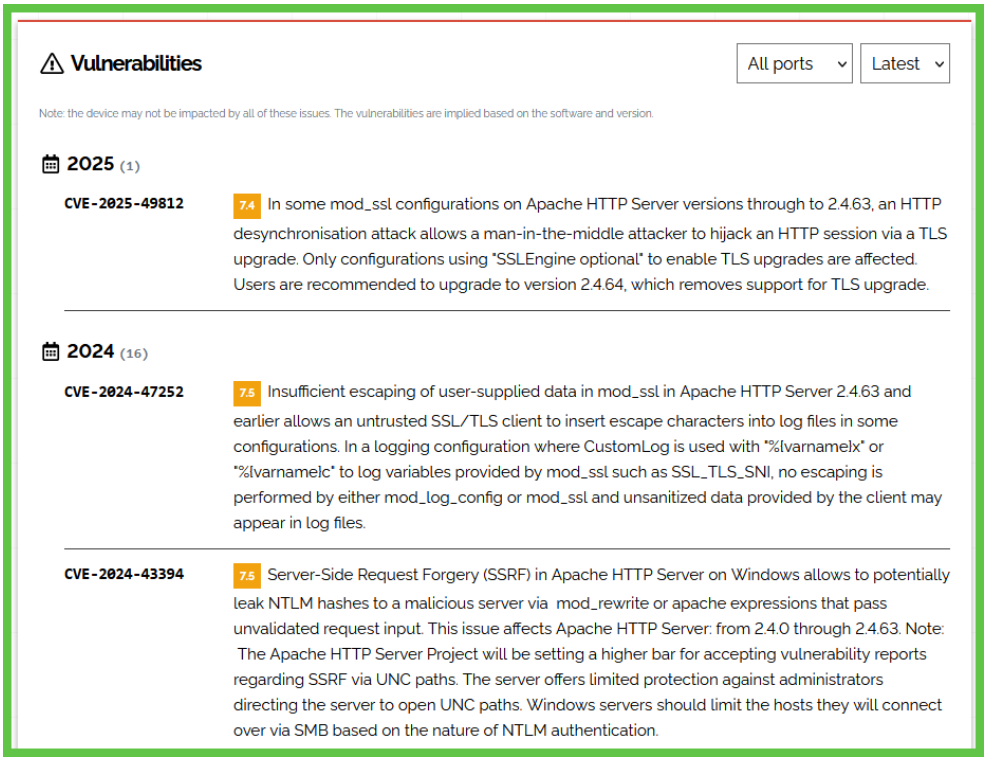


Figure 3.32 – Security vulnerabilities

As shown in the figure, ethical hackers can use Shodan to easily collect OSINT on their targets and identify security vulnerabilities on them. Then, using the information

collected from this platform, they can look into creating or acquiring exploits that can take advantage of any known and identified security vulnerabilities on their targets.

Note

Each known security vulnerability is assigned a **Common Vulnerability and Exposure** (**CVE**) identifier. To get more information on a vulnerability, search for the CVE ID at: `https://www.cve.org/`.

1. To view a point-in-time visual snapshot of systems, go to `https://images.shodan.io` and enter `os:"Windows Server" product:"Remote Desktop Protocol" country:"US"` within the image search field:

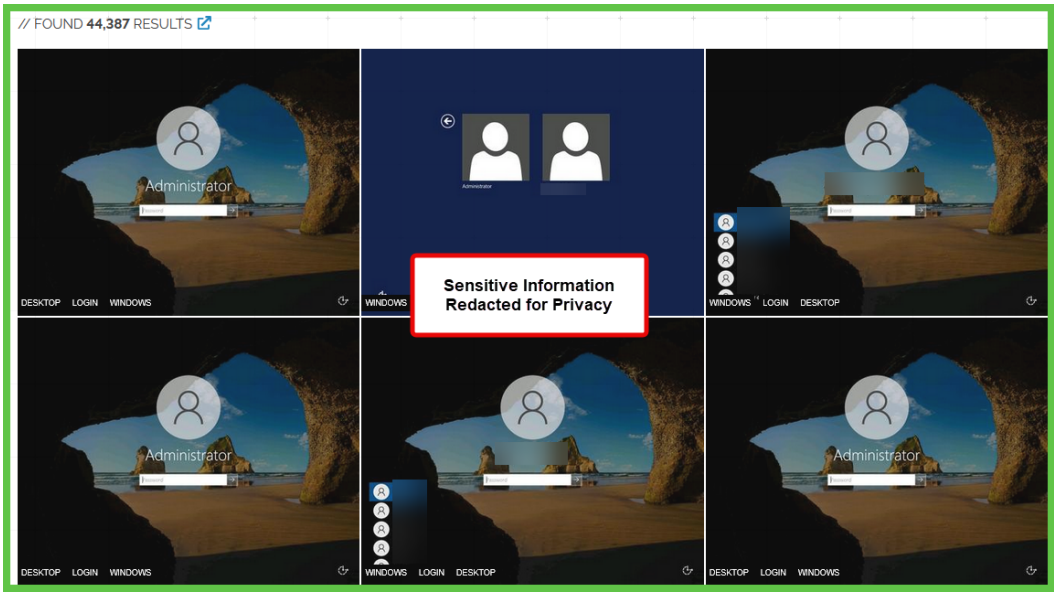


Figure 3.33 – Usernames

As shown in the preceding screenshot, custom search operators enable ethical hackers to easily obtain the usernames of systems that are exposed on the internet. Therefore, this process reduces the time needed to determine valid usernames on targeted systems on the internet.

> **Note**
>
> To learn more about Shodan search operators, please see: `https://www.shodan.io/search/filters`.

1. Let's use Censys to gather internet intelligence on systems. Using your preferred web browser, go to `https://search.censys.io/` and register for a new account.
2. Once your Censys account has been created, ensure to log in to the platform. You may be automatically redirected to: `https://platform.censys.io`.
3. Within the Censys search bar, enter an IP address of a known device on the internet, such as Google's DNS server primary address of `8.8.8.8`:

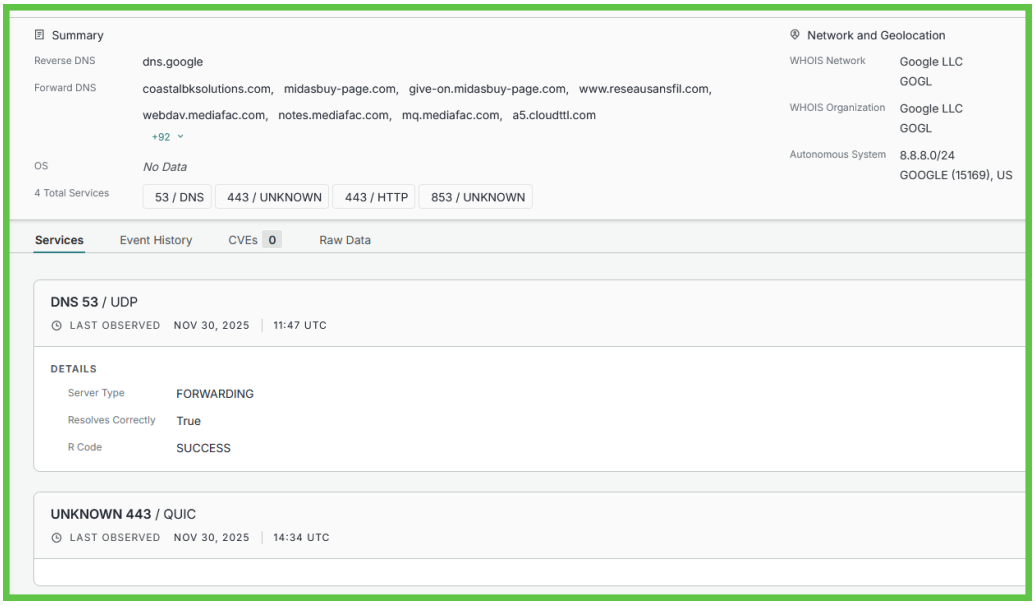


Figure 3.34 – Censys

As shown in the preceding screenshot, the information collected by Censys enables ethical hackers to gather OSINT on targeted systems, networks, and organizations. Censys provides network data and addresses, open ports and running services, and geo-location of systems.

4. Censys provides **Transport Layer Security** (**TLS**) details that enable ethical hackers to identify additional sub-domains and hidden systems of a targeted organization. On the

Censys search bar, enter `web.hostname = "www.amazon.com" and web.port = "443"` as shown:

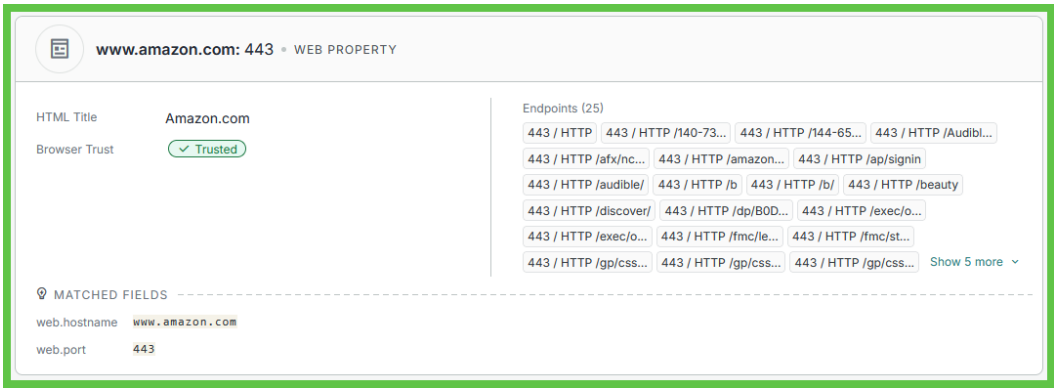


Figure 3.35 – Locating web servers

5. Click on the search result to view the technical details for Amazon, and scroll-down to the **TLS Details** section:

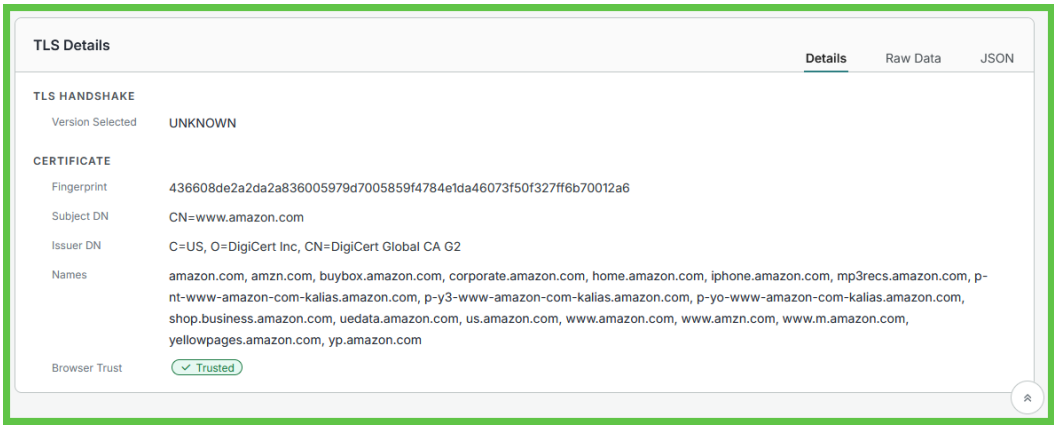


Figure 3.36 – TLS certification details

As shown in the preceding screenshot, ethical hackers can leverage the information found within digital certificates to further identify hidden systems and subdomains of a targeted organization.

Having completed this exercise, you have now learned how to use specialized search engines to collect sensitive information about exposed systems for improving your plan of attack.

Lab: Setting up alerting for a target

Ethical hackers commonly use alerting services to monitor the internet for specific keywords and phrases. Once a keyword or phrase is found, the alerting platform sends an email to notify that an event has occurred. **Google Alerts** is a powerful alerting tool used for this.

Google Alerts enables ethical hackers to stay informed and keep track of new developments in an area, such as new security vulnerabilities and exploits, and changes about a targeted organization. In addition, Google Alerts helps cybersecurity professionals monitor known threat actor activities on the internet and anticipate potential cyber-attacks and threats.

To get started follow the instructions:

1. Using a computer with internet connectivity, open your preferred web browser and go to: `https://www.google.com/alerts`.
2. Enter the keywords or phrases you'd like to monitor and modify the alerting parameters:

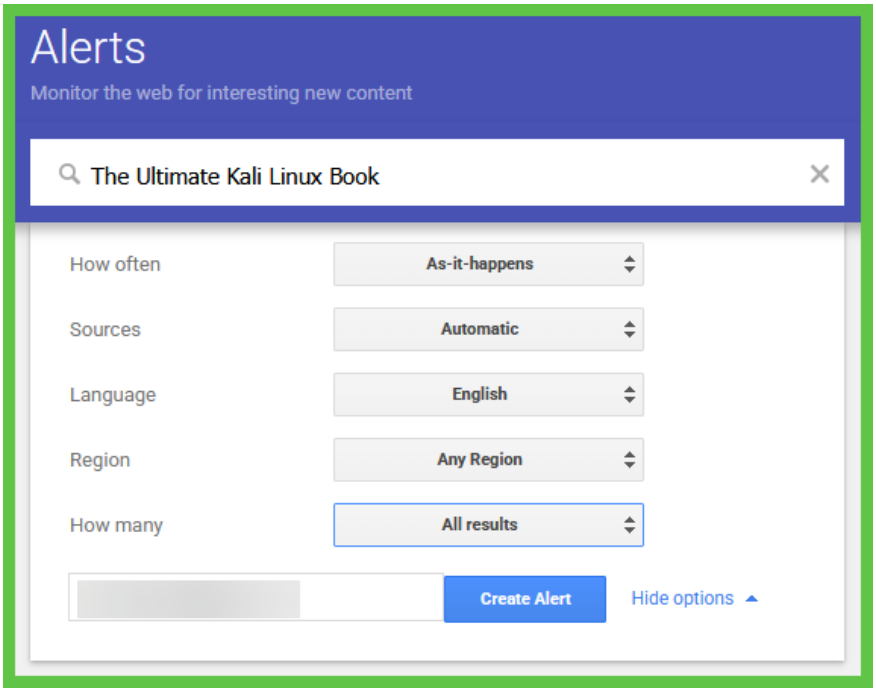


Figure 3.37 – Google Alerts

3. Make sure to set an email address that will be used for receiving alerts.
4. Lastly, click on **Create Alert**. You may be required to sign in using a Google account to manage all your alerting details. Once interesting and new content is found on the internet that matches your alerting parameters, Google will send you an alert email.

Having completed this exercise, you have learned how to set up Google Alerts for alerting new activities on target. It's time for us to set our focus on collecting domain-related information in the next section.

WHOIS and DNS footprinting

During the reconnaissance phase, ethical hackers need to identify the ownership of a publicly registered domain on the internet. **WHOIS** enables ethical hackers to gather information about the organization and people who are behind a domain name. This information helps identify targets for phishing, infrastructure mapping, or weak configurations.

WHOIS database provides the following information:

- **Contact information** – Such information can be used for social engineering
- **Domain registration information** – Determines the history such as age of the domain that can be used for social engineering tactics
- **DNS records** – May reveal sensitive directories and subdomains
- **Hosting provider** – Helps map the external attack surface and infrastructure

There are many techniques used by ethical hackers to gather information from WHOIS databases on the internet. Using the following workflow enables you to efficiently collect public information about a registered domain; additionally, you will learn how to perform these tasks in this section:

1. Identifying your target.
2. Performing WHOIS on the target.
3. Performing a query on the **Top-Level Domain** (**TLD**) server.
4. Identifying domain privacy protection status.
5. Identifying Name Servers of the target.
6. Locating registrar contact details.
7. Identifying IP addresses and **Autonomous System Number** (**ASN**).

Let us not forget the important role DNS footprinting plays during the reconnaissance phase. DNS footprinting enables ethical hackers to gain better insights on a targeted domain infrastructure and identify how various public-facing systems are interconnected. Moreover,

this technique enables you to identify security vulnerabilities and map the external attack surface of a targeted organization.

DNS footprinting helps ethical hackers to take the following actions:

- Map the external infrastructure by analyzing exposed DNS records. Identify public web servers, email servers, sub-domains and other interface-facing servers.
- Identify vulnerable points of entry such as forgotten systems, exposed services, misconfigured systems, and sensitive URLs such as **Virtual Private Network** (**VPN**) portals.
- Identify third-party dependencies such as cloud platforms, email providers, and **Content Delivery Networks** (**CDNs**).
- Verify and evaluate email security configurations to prevent email spoofing, email server exposure, and phishing techniques.
- Identify misconfigurations on DNS servers to determine open zone transfers, missing **DNS Security Extension** (**DNSSEC**) configurations, and inaccurate DNS records.

Using the following DNS footprinting workflow will enable you to efficiently gather DNS intelligence on a targeted domain:

1. Identify the targeted domain.
2. Identify the Name Server (`NS`) records.
3. Identify IP addresses (`A` and `AAAA` records).
4. Identify email servers (`MX` records).
5. Identify any text records (`TXT` records).
6. Enumerate subdomains.
7. Trace the DNS path.

In the following sub-sections, you gain hands-on experience and develop the skills needed to leverage WHOIS and DNS information to identify sub-domains and sensitive information from domain registrars.

Lab: Performing WHOIS footprinting

To get started with WHOIS footprinting, please use the following instructions:

1. Firstly, power on the **Kali Linux** virtual machine and log in.
2. Then, open the **Terminal** and ensure the `whois <domain>` command is present:

```
kali@kali:~$ whois microsoft.com
   Domain Name: MICROSOFT.COM
   Registry Domain ID: 2724960_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.markmonitor.com
   Registrar URL: http://www.markmonitor.com
   Updated Date: 2025-04-01T12:38:29Z
   Creation Date: 1991-05-02T04:00:00Z
   Registry Expiry Date: 2026-05-03T04:00:00Z
   Registrar: MarkMonitor Inc.
   Registrar IANA ID: 292
   Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
   Registrar Abuse Contact Phone: +1.2086851750
   Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
   Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
   Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
   Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
   Name Server: NS1-39.AZURE-DNS.COM
   Name Server: NS2-39.AZURE-DNS.NET
   Name Server: NS3-39.AZURE-DNS.ORG
   Name Server: NS4-39.AZURE-DNS.INFO
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2025-11-29T20:08:13Z <<<
```

Figure 3.38 – WHOIS

As shown in the preceding screenshot, the `whois` tool automated the process of retrieving the registration details of the targeted domain.

3. Use the domain's registrar WHOIS server to query the targeted domain to determine if additional information can be found:

```
kali@kali:~$ whois -h whois.markmonitor.com microsoft.com
Domain Name: microsoft.com
Registry Domain ID: 2724960_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2025-04-01T12:38:29+0000
Creation Date: 1991-05-02T04:00:00+0000
Registrar Registration Expiration Date: 2026-05-03T00:00:00+0000
Registrar: MarkMonitor, Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
```

Figure 3.39 – Querying the registrar's WHOIS server

4. Use the `dig` tool to identify the **Name Server (NS)** records of the domain:

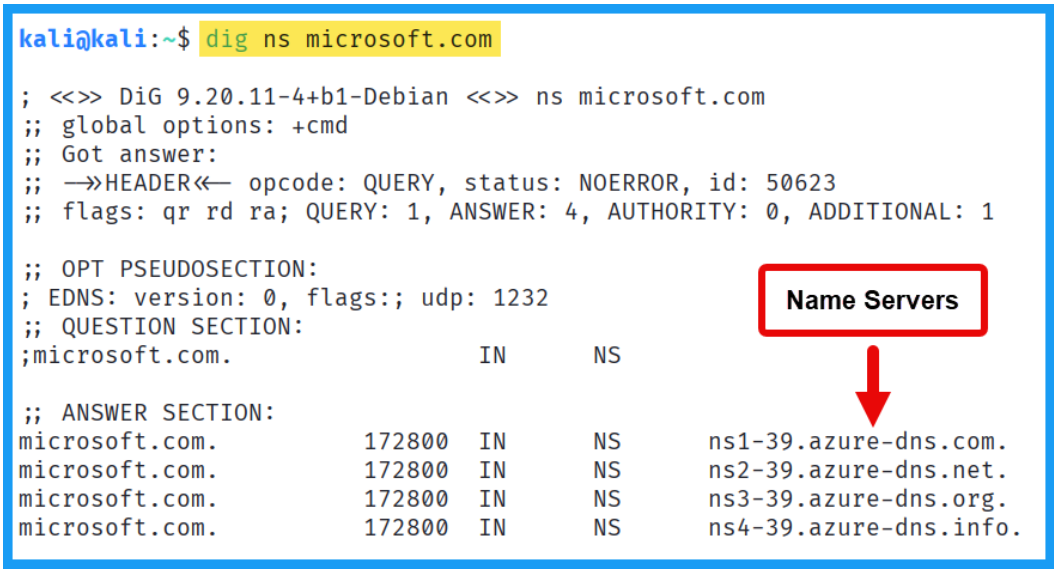


Figure 3.40 – Identifying Name Server records

As shown in the preceding screenshot, the NS records identify which servers are controlling the DNS for the targeted domain.

Tip

The `host -t ns <domain>` command retrieves the NS records of a domain.

5. To automate the WHOIS footprinting process using bash scripting, use the following command to create a new bash file:

```
kali@kali:~$ nano whois.sh
```

6. Now, enter the following coding into the Nano text editor:

```
#!/bin/bash
domain=$1

echo "[*] WHOIS Lookup"
whois $domain> whois_$domain.txt

echo "[*] Resolving IP..."
ip=$(dig $domain +short | tail -n1)
```

```
echo "IP: $ip" | tee -a whois_$domain.txt

echo "[*] IP WHOIS"
whois $ip>> whois_$domain.txt

echo "Output saved to whois_$domain.txt"
```

7. To save the file, press *CTRL + X*, then *Y* and *Enter* on your keyboard.
8. Use the following command to set executable privileges and run the script:

```
kali@kali:~$ chmod +x whois.sh
kali@kali:~$ ./whois.sh <target_domain>
```

The following screenshot shows the execution of the preceding commands:

```
kali@kali:~$ chmod +x whois.sh

kali@kali:~$ ./whois.sh microsoft.com
[*] WHOIS Lookup
[*] Resolving IP ...
IP: 13.107.213.41
[*] IP WHOIS
Output saved to whois_microsoft.com.txt

kali@kali:~$ cat whois_microsoft.com.txt
   Domain Name: MICROSOFT.COM
   Registry Domain ID: 2724960_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.markmonitor.com
   Registrar URL: http://www.markmonitor.com
   Updated Date: 2025-04-01T12:38:29Z
   Creation Date: 1991-05-02T04:00:00Z
   Registry Expiry Date: 2026-05-03T04:00:00Z
   Registrar: MarkMonitor Inc.
```

Figure 3.41 – Creating a Bash script

As shown in the preceding screenshot, the `cat` command enables you to display the contents of a file on the Terminal. Additionally, this automates the process of gathering network information about the target, as shown:

```
NetRange:       13.64.0.0 - 13.107.255.255
CIDR:           13.104.0.0/14, 13.64.0.0/11, 13.96.0.0/13
NetName:        MSFT
NetHandle:      NET-13-64-0-0-1
Parent:         NET13 (NET-13-0-0-0-0)
NetType:        Direct Allocation
OriginAS:
Organization:   Microsoft Corporation (MSFT)
RegDate:        2015-03-26
Updated:        2021-12-14
Ref:            https://rdap.arin.net/registry/ip/13.64.0.0
```

Figure 3.42 – Network information from WHOIS

Such information helps ethical hackers to better understand the attack surface and infrastructure of the targeted domain and organization.

Having completed this exercise, you have learned how to use WHOIS and create a custom bash script to automate WHOIS footprinting. Next, you will learn how to perform DNS footprinting.

Lab: Performing DNS footprinting

To get started with DNS footprinting, follow these instructions:

1. Using **Kali Linux**, open the **Terminal** and use the following commands to automate the process of querying Google's DNS server about a domain:

```
kali@kali:~$ dnsenum --dnsserver 8.8.8.8 microsoft.com
```

2. As shown in the following screenshot, DNSenum was able to enumerate the DNS records of the domain:

```
kali@kali:~$ dnsenum --dnsserver 8.8.8.8 microsoft.com
dnsenum VERSION:1.3.1

-----   microsoft.com   -----


Host's addresses:
__________________

microsoft.com.                           1891     IN    A        13.107.213.40
microsoft.com.                           1891     IN    A        13.107.246.40


Name Servers:
______________

ns4-39.azure-dns.info.                   2359     IN    A        13.107.206.39
ns1-39.azure-dns.com.                    3259     IN    A        150.171.10.39
ns2-39.azure-dns.net.                    2808     IN    A        150.171.16.39
ns3-39.azure-dns.org.                    683      IN    A        13.107.222.39


Mail (MX) Servers:
___________________

microsoft-com.mail.protection.outlook.com. 10       IN    A         52.101.40.0
microsoft-com.mail.protection.outlook.com. 10       IN    A         52.101.9.12
microsoft-com.mail.protection.outlook.com. 10       IN    A         52.101.9.21
microsoft-com.mail.protection.outlook.com. 10       IN    A         52.101.8.34
```

Figure 3.43 – DNSenum

3. To gain a visual representation of the DNS infrastructure of a target, go to `https://dnsdumpster.com` and enter a domain name, as shown:

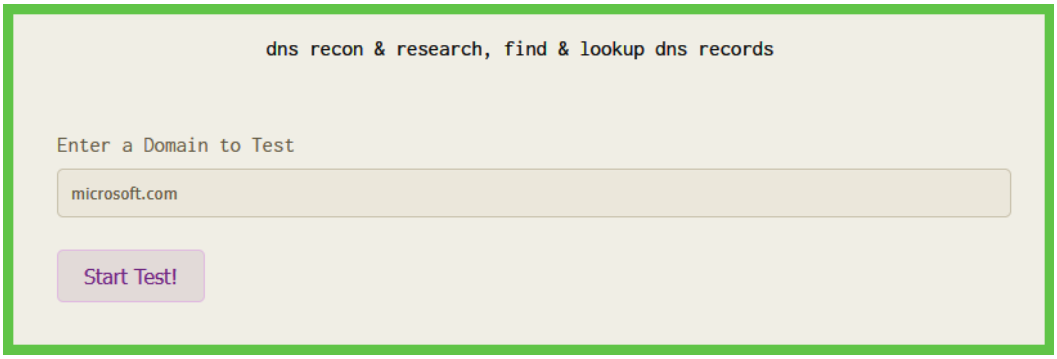


Figure 3.44 – DNS reconnaissance

4. DNSDumpster provides a visualization of the domain's infrastructure with its DNS records:

A Records (subdomains from dataset)

Host	IP	ASN	ASN Name	Open Services (from DB)
064-smtp-in-2a.microsoft.com	157.54.41.37	ASN:3598 157.54.0.0/16	MICROSOFT-CORP-AS - Microsoft Corporation United States	
1esbot.microsoft.com	40.118.129.241	ASN:8075 40.112.0.0/13	MICROSOFT-CORP-MSN-AS-BLOCK - Microsoft Corporation United States	
apps.1js.microsoft.com	20.119.0.21	ASN:8075 20.64.0.0/10	MICROSOFT-CORP-MSN-AS-BLOCK - Microsoft Corporation United States	http: unknown server title: Microsoft Azure Web App - Error 404 https: unknown server title: Microsoft Azure Web App - Error 404 cn: .azurewebsites.net o: Microsoft Corporation
pr.apps.1js.microsoft.com	20.119.0.21	ASN:8075 20.64.0.0/10	MICROSOFT-CORP-MSN-AS-BLOCK - Microsoft Corporation United States	http: unknown server title: Microsoft Azure Web App - Error 404 https: unknown server title: Microsoft Azure Web App - Error 404 cn: .azurewebsites.net o: Microsoft Corporation

Figure 3.45 – DNS infrastructure

Additionally, DNSDumpster provides the following DNS mapping that helps ethical hackers gain a better understanding of how each DNS record is connected to a system:

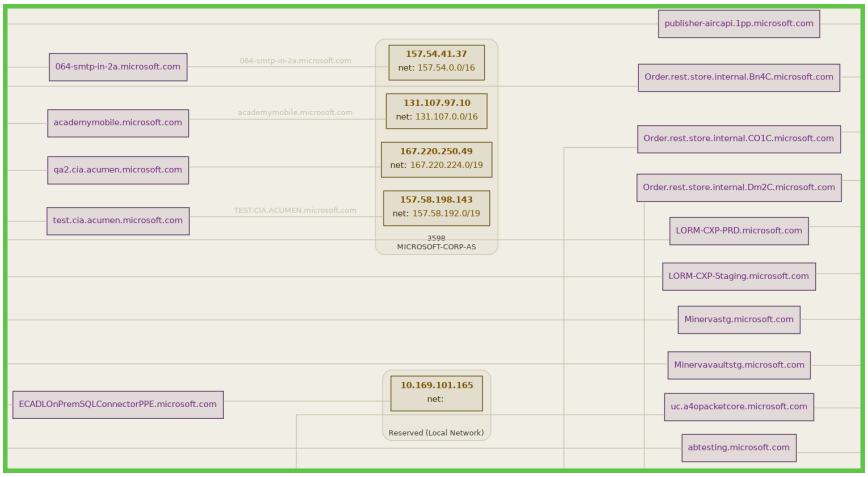


Figure 3.46 – DNS map

5. Lastly, use Shell-GPT on Kali Linux to leverage AI to automate DNS footprinting to retrieve the NS records of a domain:

```
kali@kali:~$ sgpt --shell "Use DNSenum for  DNS footprinting on
microsoft.com using Google's DNS server and resolve the IP addresses of any
hostnames found"
```

The following screenshot shows how the local AI and LLM model work together to automate DNS footprinting on the Terminal:

```
kali@kali:~$ sgpt --shell "Use DNSenum for  DNS footprinting on microsoft.com
 using Google's DNS server and resolve the IP addresses of any hostnames foun
d"
dnsenum -d google-public-dns-ipv4.google.com -i microsoft.com && dig +short @
google-public-dns-ipv4.google.com A $(dnsenum -d google-public-dns-ipv4.googl
e.com -i microsoft.com | grep "has address" | cut -d' ' -f5-)
[E]xecute, [D]escribe, [A]bort: E
dnsenum VERSION:1.3.1
Value "google-public-dns-ipv4.google.com" invalid for option d (number expect
ed)
Unknown option: i

-----   microsoft.com   -----


Host's addresses:
__________________

microsoft.com.                          1827     IN    A        13.107.213.4
1
microsoft.com.                          1827     IN    A        13.107.246.4
1
```

Figure 3.47 – Using local AI and LLM for DNS Footprinting

Having completed this exercise, you have gained the skills needed to perform DNS footprinting to identify sub-domains, DNS records, and even automate the process using a local AI model. In the next section, you will learn about countermeasures to prevent reconnaissance-based attacks and threats. Now that you know how to perform various passive reconnaissance techniques, let us see how ethical hackers can help safeguard organizations from threat actors who are attempting to collect information on them.

Reconnaissance countermeasures

While many organizations have the need to share information with external parties and on the internet, it's important to be mindful of how a real threat actor such as a hacker can exploit such information. Therefore, as an ethical hacker, you are responsible for recommending countermeasures to help organizations prevent and mitigate reconnaissance-based attacks and threats.

The following are common countermeasures for reconnaissance, but are not limited to:

- Implementing strict controls for managing employees' access to social media platforms

- Implementing security controls and auditing to ensure web application servers do not leak sensitive information on the internet
- Performing frequent user awareness training for all employees on data leakage
- Providing guidelines to assist the communication and public relations team on the Do's and Don'ts of sharing too much sensitive information with the public
- Reviewing and limiting the amount of sensitive information that's shared on the organization's website and social media pages
- Performing frequency reconnaissance and footprinting techniques to identify sensitive information that's shared on the internet

As an ethical hacker, it's important to guide organizations to develop and improve their information security governance framework by starting with their information security program that aligns to an industry standard such as ISO 27001 and ISO 27002. This means, provide guidelines and recommendations to create and enforce information security policies that regulate the sharing of information internally and externally of the organization. For instance, a social media usage policy that governs the use of social media platforms and restrictions on accessing company pages and what type of information that should not be shared on both company-owned and personal profiles.

Additionally, since DNS servers commonly share information about domain records with anyone and any device on the internet, ethical hackers should test the security controls on company-owned or managed DNS servers to determine the type of information that is exposed. Internal DNS servers should not be sharing DNS records with users and devices on the internet; they should be restricted to internal use only. A misconfigured DNS server can leak the hostname and IP addresses of all internal systems to threat actors on the internet.

Misconfigured web applications on the internet can expose sensitive files and hidden directories to real hackers, Therefore, it's important to ensure web applications and databases are configured to recommended security controls to prevent a potential cyber-attack or threat. In addition, while it's probably common at present, organizations should continue to protect their privacy during the registration process of public domain name. While it's usually an additional fee, the benefit of privacy plays an important role.

Ensure organizations perform user awareness training frequently. For instance, quarterly training for all staff on various topics and areas in cybersecurity helps people to become more vigilant and aware of threats and attacks. Using a mixture of various modes of delivery such as e-learning (self-paced), online (virtual-live), and even in-person when applicable can meet the needs of various employees within organizations. Furthermore, such training helps to educate employees about various types of social engineering threats and risks. Lastly, developing and

delivering focus-based training to specific groups based on users' roles and responsibilities help employees to better appreciate and understand their role in cybersecurity.

Additional countermeasures to reduce the risk of footprinting include disabling the geo-location tagging features within cameras and ensuring the geo-tagging details are removed from pictures before uploading to social media platforms, as well as training employees not to share their location or traveling details on social media platforms.

Note

Note from the author

During my career as a cybersecurity professional, I have seen many people take self-portrait photos with their smartphones and post them as status updates on social media and messaging platforms, even when the background contains sensitive information. For instance, a new employee in an engineering department once took a self-portrait with an unsuspecting engineer in the background while the engineer was logging in to a security appliance, with configurations visible on his screen. The new employee had not received user awareness training before starting the job and shared the picture on their personal social media accounts. Now imagine the potential damage that could occur if a real threat actor were to leverage the information found in that picture.

Sharing future travel plans on social media platforms can indicate that you will not be at home or in the office for a certain period. Threat actors or anyone with malicious intent can leverage such information and look for opportunities to commit crimes such as burglary or theft. Therefore, it is common practice to avoid sharing travel details, plans, or pictures before or during your travel. Practicing preventive measures can help reduce the risk of adverse outcomes, especially in the field of cybersecurity.

Summary

In this chapter, you've gained a deep dive into understanding the importance of performing passive reconnaissance as an ethical hacker. You've learned how organizational, network, and even system intelligence helps cybersecurity professionals to better understand the external attack surface of an organization. We discussed how certain information leaked knowingly or unknowingly by organizations and employees can be exploited by cyber-criminals and how ethical hackers can leverage this information. Furthermore, you've gained the essential skills needed for the cybersecurity industry to become the next generation of ethical hackers.

In the next chapter, you will learn how to perform active reconnaissance on systems by performing common scanning techniques to profile live systems on a network.

Further reading

The following are additional reading resources to help reinforce your learning:

- MITRE Reconnaissance tactics: `https://attack.mitre.org/tactics/TA0043/`
- Subfinder: `https://github.com/projectdiscovery/subfinder`
- DNSx: `https://github.com/projectdiscovery/dnsx`
- ExifTool: `https://www.kali.org/tools/libimage-exiftool-perl/`
- Email security records: `https://www.cloudflare.com/learning/email-security/dmarc-dkim-spf/`
- DNS records: `https://www.cloudflare.com/learning/dns/dns-records/`

4

Scanning

In the previous chapter, you've explored techniques and tools that are commonly used to perform passive reconnaissance. In this chapter, we will build our discussion on the techniques of active reconnaissance and how they help ethical hackers collect sensitive information from targeted systems.

Passive reconnaissance that leverages **Open Source Intelligence** (**OSINT**) contains information that's publicly available. Such information is commonly found on websites, forums, social media, and any other public data sources on the internet. However, OSINT is not always up-to-date, and ethical hackers can experience challenges when working with outdated information.

As a result of the issues with OSINT, ethical hackers commonly use active reconnaissance techniques to retrieve specific and more accurate information about their targets. Active reconnaissance techniques, such as scanning and enumeration, help ethical hackers collect more accurate information about systems, networks, and applications. Scanning focuses on sending special probes from your attacker's machine like Kali Linux to a target to retrieve details about it. This information can help ethical hackers determine whether a device is online, identify its open ports, and discover the services running on it.

In this chapter, you will learn how to use techniques and procedures similar to those used by real threat actors for ethical hacking. You'll learn the importance of various scanning techniques and how they are used in real-world scenarios, and the skills needed to profile various hosts, identify open ports, and determine running services on targeted systems. Additionally, you'll learn common techniques used by ethical hackers to evade threat detection solutions while scanning systems and networks. Lastly, you'll learn how to implement scanning countermeasures to help organizations prevent and mitigate similar techniques used by real threat actors.

In this chapter, you will cover the following topics:

- Fundamentals of network scanning
- Host discovery concepts and techniques
- Service port and operating system discovery
- Scanning beyond IDS and firewalls
- Scanning countermeasures

Technical requirements

You only need Kali Linux to run the practical examples in this chapter. For installation details, please refer to *Chapter 2*.

Fundamentals of network scanning

Ethical hackers deploy specially crafted network probes to a target network for conducting reconnaissance using various scanning techniques. Scanning enables ethical hackers to identify live hosts, open ports, running services, and the operating systems of devices on a network.

Scanning enables ethical hackers to determine whether live hosts are susceptible to various types of cyber-attacks and threats, or contain known vulnerabilities that can be exploited by a real cyber-criminal.

The following are the core objectives of scanning, and each will be explored in detail throughout this chapter.

- Identify vulnerable communication channels
- Determine live hosts on a targeted network
- Determine open ports and running services
- Profile hosts to determine operating system
- Identify security vulnerabilities on targeted systems

Within many organizations, there are lots of technical teams who are responsible for the implementation, administration, and maintenance of systems, networks, and applications. In the haste of completing projects, implementation teams have to set up new systems for internal and external users as quickly as possible, which can sometimes lead to security aspects being compromised. The technical teams may overlook integrating security controls and practices at the beginning and during the project. For this reason, many technical systems are implemented and left in a vulnerable state until a real cyber-attack is realized. Therefore, many organizations seek to create an internal cybersecurity team that can focus on ethical

hacking, vulnerability management, penetration testing, and incident response and handling. The objective of the internal cybersecurity team is to identify security vulnerabilities and to implement countermeasures to proactively reduce cyber-risk within the organization. In other words, reduce the likelihood of a real cyber-attack or threat.

Port scanning

Port scanning primarily focuses on identifying open ports and running services on targeted systems on a network. This enables ethical hackers to profile a targeted system and identify vulnerable points of entry on that system. Additionally, identifying the running services enables ethical hackers to further enumerate the service versions, which can be used for researching vulnerabilities and improving the plan of attack. Similar to real hackers on the internet, ethical hackers are able to exploit various security vulnerabilities within the most popular network model, that is, **Transmission Control Protocol/Internet Protocol (TCP/IP)**), to extract sensitive information about devices.

For instance, ethical hackers use various software-based tools that enable them to automate the process of crafting and sending specialized network packets to targeted systems. Those packets (**probes**) are designed to check for open ports and identify running services on live hosts on a network. Using this information, you can identify live systems, determine their operating systems, discover running services and their versions, and detect potential misconfigurations.

> **Note**
>
> Probes are a common term used to describe packets that are sent to retrieve sensitive information about a targeted system or network.

Port scanning focuses on identifying open ports on a single system, while network scanning expands this to identify multiple live systems and their exposed services across an entire network.

Network scanning

Network scanning enables ethical hackers to map a network topology. For instance, if you're given a network block such as `10.10.10.0/24`, you would be required to determine the number of live systems, types of devices within the network, how many open ports are on each system, identify any running services for each port per device, and determine the host operating systems of each device. Simply put, network scanning involves the techniques and procedures ethical hackers use to identify live hosts on a network and determine whether to attack each targeted system or evaluate their security posture and cyber-risk on the network.

There are many tools used by ethical hackers to perform port and network scanning, such as the following:

- Nmap – `https://nmap.org/`
- Zenmap - `https://nmap.org/zenmap/`
- Advanced IP Scanner – `https://www.advanced-ip-scanner.com/`
- Angry IP Scanner – `https://angryip.org/`
- Metasploit – `https://www.metasploit.com/`

Vulnerability scanning

Vulnerability scanning is an automated process that identifies known security weaknesses or flaws on a targeted system, network, or application. This process enables ethical hackers to quickly identify security vulnerabilities, determine the cyber-risk and severity level, and determine the countermeasures to prevent or mitigate the risk.

Vulnerability scanners typically look for the following types of security flaws:

- **Default settings** - These are the default configurations that are applied by the vendor before shipping a product or device to a customer. Default configurations include default user credentials, insecure administrative access, and insecure communication protocols.
- **Weak encryption** – Sometimes IT professionals may configure a system to use unsecure or vulnerable encryption technologies, enabling malicious actors to compromise the system or the communication channel.
- **Unsecure protocols** – Using unsecure protocols enables malicious hackers to intercept and capture sensitive information that's being transmitted over a network.
- **Open permissions** – Open permission on a system enables anyone to access files, whether those files are confidential or not. Malicious actors can exploit this type of vulnerability to steal data.
- **Unsecure root/administrator accounts** – By using default or weak passwords for root (Linux) or administrator (Windows), malicious actors can attempt to perform various types of password-based attacks to gain unauthorized access to the targeted system.
- **Missing updates and patch** – Systems with missing updates and patches are vulnerable to various types of cyber-attacks and threats. For instance, a Windows Server 2025 that hasn't been updated in months is highly vulnerable to many types of exploits and cyber-attacks.

As you have learned, there are various types of scans used by ethical hackers during their active reconnaissance phase. However, other techniques enable ethical hackers to profile target

systems in greater detail by identifying open ports, running services, and even their operating systems. In the next section, you will learn the fundamentals of host discovery and common techniques to identify live systems on a network.

Host discovery concepts and techniques

Before an ethical hacker exploits a targeted system or launches an attack, they need to determine whether the targeted system is online, or live on the network, and if it is responding. Hence, performing host discovery is an essential phase in network scanning. Typically, a network scan would attempt to send probes to all devices within the targeted IP network and check whether each port is open on online devices. However, if the objective is to determine live hosts on a network, then it would be better to send probes to determine the status of systems only and not perform a port scan.

The following are common networking protocols that are leveraged by ethical hackers to determine the status of host devices on a network:

- **Internet Control Message Protocol** (**ICMP**) – ICMP is network protocol that's commonly used by systems to send error messages, performing diagnostic checks, and troubleshooting, and provides information about IP communication between hosts over a network
- **Address Resolution Protocol** (**ARP**) – ARP is a network protocol that enables hosts to resolve IP address to **Media Access Control** (**MAC**) address over a local network
- **User Datagram Protocol** (**UDP**) – UDP is a connectionless, transport-layer network protocol that's responsible for delivery of packets and ensuring process-to-process communication between hosts over a network
- **Transmission Control Protocol** (**TCP**) – TCP is a connection-oriented, Transport-Layer protocol that provides guarantee of delivery of network packets, and ensures process-to-process communication between hosts over a network
- **Internet Protocol** (**IP**) – IP is an unreliable network protocol that enables hosts to communicate with other devices over a network

While the preceding list of network communication protocols is common, ethical hackers are able to construct specialized probes to retrieve information from targeted systems. Over the next few sub-sections, you will learn about common scanning techniques, how they work, their use-cases, and benefits.

ICMP scanning concepts

ICMP is a common network protocol that's used by various IT professionals for troubleshooting network-related issues. However, ethical hackers can leverage various ICMP

messages for discovering online/live hosts on a network. The **ICMP Echo Ping** scan is a network discovery scanning technique, whereby the attacker machine sends ***ICMP Echo Request*** packet to a targeted IP address to determine whether the host is online. A system that receives ***the ICMP Echo Request*** (***Type 8***) packets will response with an ***ICMP Echo Reply*** (***Type 0***) message. This response indicates the host is online and reachable.

The following diagram shows how the ICMP Echo Ping scan works:

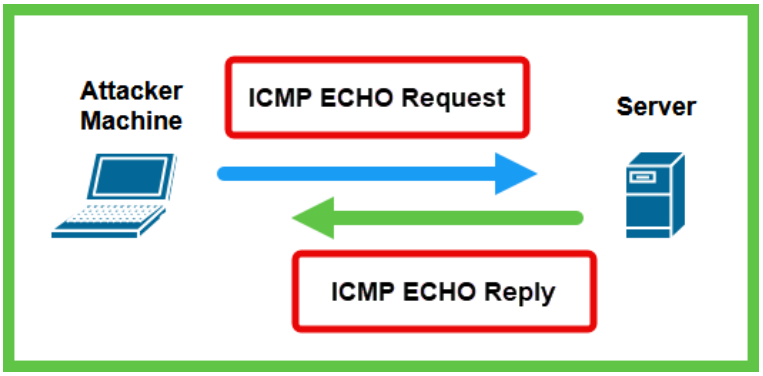


Figure 4.1 – ICMP Echo Ping scan

Another technique is using **ICMP Echo Ping Sweep** for scanning a range of IP addresses to determine whether multiple hosts are online. Unlike the ICMP Echo Ping scan that scans a single targeted host, the ICMP Ping Sweep sends ICMP probes to multiple targeted IP addresses on a network to determine which systems are online and reachable.

The following diagram shows how the ICMP Ping Sweep works:

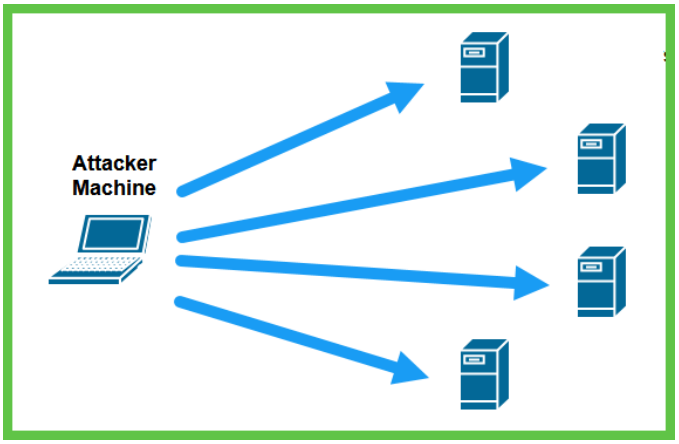


Figure 4.2 – ICMP Ping Sweep

The **ICMP Timestamp Ping** scan is another technique used by ethical hackers to send specially-crafted *ICMP Timestamp Request* (*Type 13*) packets to discover live hosts on a network. Once a targeted host is online, it responds with an *ICMP Timestamp Reply* (*Type 14*) message. Therefore, indicating it's online and reachable. This technique is useful when a targeted host is online, but it's configured to block standard ICMP ping messages.

The following diagram shows how the ICMP Timestamp Ping scan works:

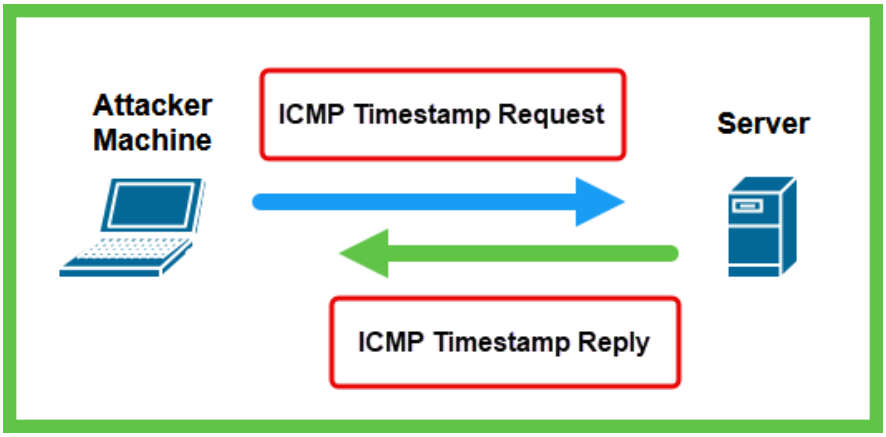


Figure 4.3 – ICMP Timestamp Ping scan

Next, you will learn how ethical hackers are able to leverage ARP to discover online hosts on a network.

ARP scanning concepts

Ethical hackers can use specialized network scanning tools such as Nmap to perform an **ARP Ping Scan**. This type of scan is commonly used to quickly identify online/live devices on a local network whether the attacker's device sends *ARP Request* messages on the network. Online devices that receive the *ARP Request* message will response with an *ARP Reply*, including this MAC address within the response, indicating the online status on the network and their associated MAC and IP addresses.

The following diagram shows how the ARP Ping Scan works on a network:

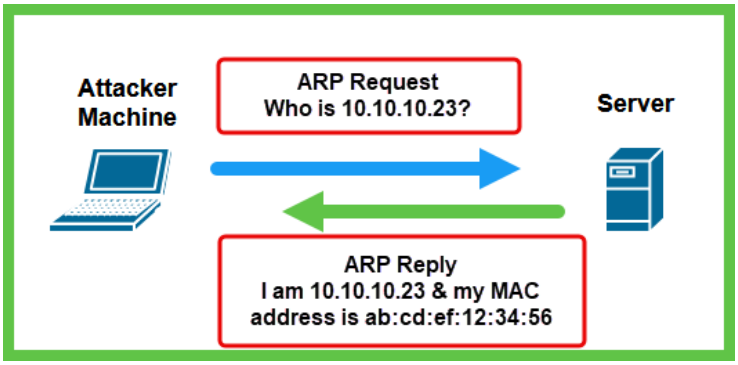


Figure 4.4 – ARP Ping Scan

However, if an ARP Request goes to a targeted IP address on the network and no response is provided to the attacker machine, then the target is considered to be down or offline. The following diagram shows a visual representation of this concept:

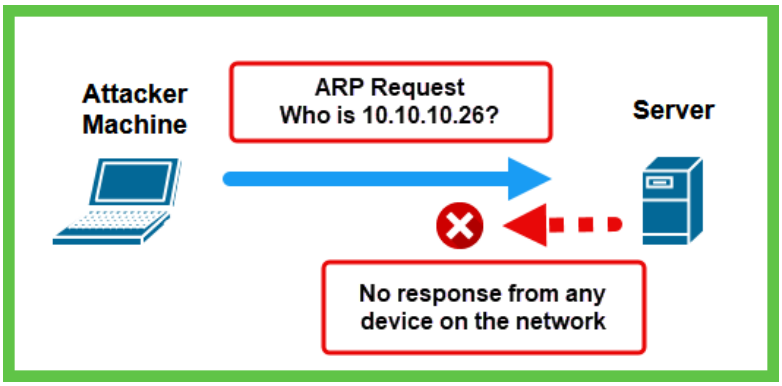


Figure 4.5 – Offline host

Due to the design of TCP/IP, host devices commonly respond to ARP Request messages; hence, this scanning technique is considered to be more efficient and provides better accuracy on host discovery. Next, you will learn how ethical hackers are able to leverage a common Transport-layer protocol such as UDP for discovering live hosts on a network.

UDP scanning concepts

UDP Ping Scan is another technique used by ethical hackers to circumvent network-based or host-based firewalls that are configured to filter incoming TCP traffic. Ethical hackers use UDP Ping Scans to determine the status of host devices. During this scanning technique, the attacker machine sends UDP packets to specific ports on the targeted host. If the targeted host responds with an *ICMP Type 3 Code 3* (*Port Unreachable*) message, it means the targeted port is closed but the host is online/live.

The following diagram shows a UDP Ping scan detecting an online host:

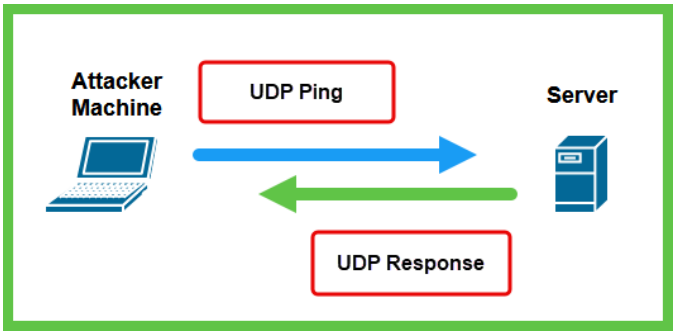


Figure 4.6 – Online host

However, if the attacker machine does not receive any ICMP responses, this indicates the targeted port on the host may be open or filtered. If an *ICMP Type 3 Code 1* (*Host Unreachable*) or a **Time-To-Live** (**TTL**) **Exceeded** response is returned, then the targeted host is offline. The main advantage of this scanning technique is to detect online/live systems behind network-based firewalls that are configured to restrict TCP traffic but permit UDP traffic through the network.

Next, you will learn common scanning techniques that leverage TCP for discovering whether targeted hosts are online on a network.

TCP scanning concepts

TCP is a common Transport-layer protocol that provides guaranteed delivery of network packets between source and destination hosts over a network. A common technique that's used by ethical hackers is the **TCP Synchronization (SYN) Ping scan**, which sends specially-crafted probes to different ports on the targeted system to determine whether the host is online and the port is open. The TCP SYN Ping scan is commonly referred to as a ***stealth scan*** or a ***half-open scan*** because it does not complete the full **TCP-3 Way Handshake**.

The following diagram shows the TCP 3-way handshake, where the attacker machine responds with a **TCP Acknowledgement** (**ACK**) packet:

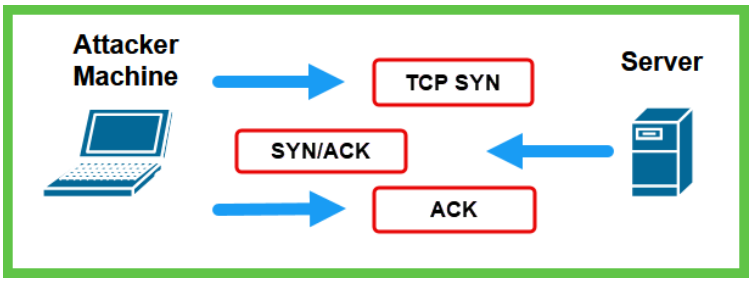


Figure 4.7 – TCP 3-way handshake

The following diagram shows the TCP SYN Ping scan, where the attacker machine sends a **TCP Reset (RST)** packet to reset the connection:

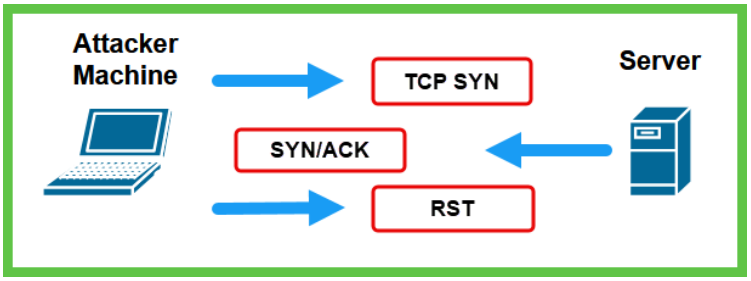


Figure 4.8 – TCP SYN Ping scan

The following is a breakdown of the TCP SYN Ping scan as shown in the preceding diagram:

1. The attacker machine sends a TCP SYN packet to probe whether the targeted host is online and there's an open port.
2. The targeted host responds with a TCP SYN/ACK packet, indicating the host is online and the port is open.
3. The attacker machine responds with a TCP RST packet that instructs the targeted system to reset/terminate the network connection.

The TCP SYN Ping scan is commonly used by ethical hackers to determine which systems are online without completing a network connection between the attacker machine and the targeted systems. Additionally, this technique can be used to scan a range of hosts over a network without getting timeout errors while awaiting the responses from targeted systems.

The **TCP ACK Ping scan** is another technique that leverages TCP packets for host discovery. In a TCP ACK Ping scan, the attacker's machine sends TCP ACK probes the target hosts. Since the attacker's machine did not establish a network connection to the targeted host, the target

responds with a TCP RST packet to terminate the request. This indicates the targeted host is online and reachable.

The following diagram shows how the TCP ACK Ping scan works with an online host:

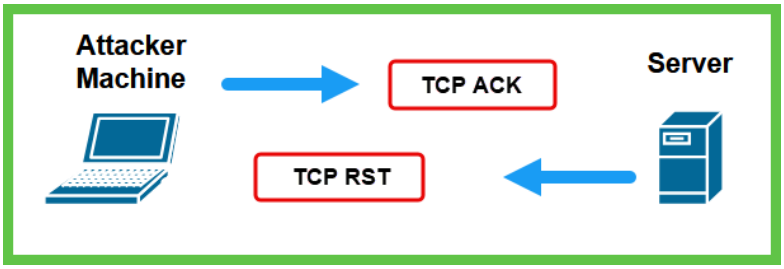


Figure 4.9 – TCP ACK Ping scan

The TCP ACK Ping scan helps ethical hackers to bypass firewalls that are configured to restrict TCP SYN packets. Now you will learn how ethical hackers use IP to assist with discovering hosts on a network.

IP scanning concepts

IP Protocol Ping scan is a common technique used by ethical hackers to discover online/live hosts on a network. In this technique, the attacker machine sends IP Ping packets with an IP Header containing various protocol numbers, such as ICMP, IP, and even **Internet Group Management Protocol** (**IGMP**). The purpose of using these common protocol numbers is to bypass firewall rules/policies that block ICMP messages.

The following diagram shows how the IP Protocol Ping scan works with an online host:

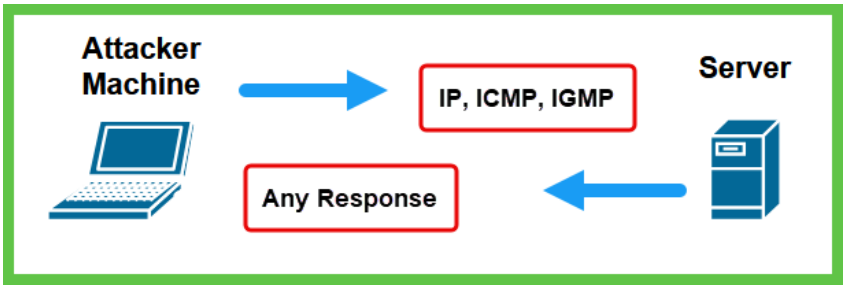


Figure 4.10 – IP Protocol Ping Scan

As shown in the preceding diagram, the following is a deeper dive into how it works:

1. The attacker machine sends IP packets with different protocol numbers such as:
    - ICMP = protocol number 1

- IGMP = protocol number 2
- IP = protocol number 4

2. The response from the targeted host is analyzed to identify specific response types such as unreachable or no response. If the targeted host provides any response, it's an indication the target is online and reachable.

Having completed this section, you've learned the fundamentals of various scanning techniques and use-cases. In the following two labs, you will perform various scanning techniques using Kali Linux and Nmap. You'll start by learning how to discover live systems on a network using the Nmap tool from the command line, followed by Zenmap, a graphical user interface tool for identifying live systems.

Lab: Identifying live hosts using Nmap

In this exercise, you will learn how to use a common workflow as an ethical hacker to identify live systems on a network and open ports. This technique is useful when ICMP is blocked by firewalls.

To get started with this exercise, please use the following instructions:

1. Open **Oracle VM VirtualBox Manager** and power on both the **Kali Linux** and **Metasploitable 3 (Windows-based)** virtual machines.
2. Login to **Kali Linux** and open the **Terminal**.
3. As an aspiring ethical hacker, it's important to verify whether your Kali Linux machine is receiving an IP address from the targeted network. Therefore, use the following command to verify the IP address on the `eth1` network adapter:

```
kali@kali:~$ ip address show eth1
```

4. The following screenshot shows `eth1` has an IP address on the `10.10.10.0/24` network that's connected to our `Hacking_Lab` virtual network:

```
kali@kali:~$ ip address show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
    link/ether 08:00:27:9c:e6:4e brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.24/24 brd 10.10.10.255 scope global dynamic noprefixroute eth1
       valid_lft 600sec preferred_lft 600sec
    inet6 fe80::50b8:23fd:5bef:53b3/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Figure 4.11 – Network adapter

> **Tip**
>
> When performing a network scan to identify live systems on a network, it's recommended to exclude the IP address that's assigned to your attacker machine, which is your Kali Linux machine.

5. To determine the network address and its ranges, go to `https://www.calculator.net/ip-subnet-calculator.html` and enter the IP address of Kali Linux and its subnet mask within the fields and click on **Calculate**, as shown:

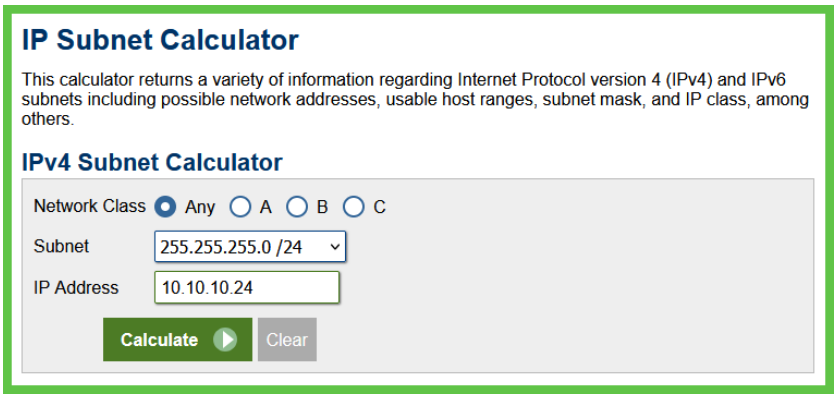


Figure 4.12 – Subnet calculator

6. Once completed, the online IP subnet calculator identifies the network address and its range, as shown:

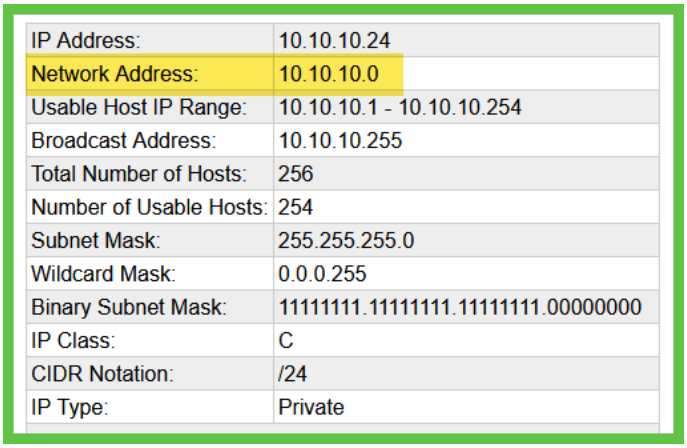


Figure 4.13 – Network address and range

7. Use Nmap to automate the process of identifying live systems on the `10.10.10.0/24` network using a *ping scan*. Do not identify open ports on live hosts and use the `--exclude` command to exclude both the IP address of Kali Linux and the default gateway (`10.10.10.1`):

```
kali@kali:~$ nmap -sn 10.10.10.0/24 --exclude 10.10.10.1,10.10.10.24
```

The following screenshot shows Nmap was able to identify one live host on the network:

```
kali@kali:~$ nmap -sn 10.10.10.0/24 --exclude 10.10.10.1,10.10.10.24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 08:07 EST
Nmap scan report for 10.10.10.23
Host is up (0.00014s latency).
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 254 IP addresses (1 host up) scanned in 2.30 seconds
```

Figure 4.14 – Ping scan

> **Note**
>
> Performing a ping scan is excellent for quickly identifying live hosts on a subnet. However, both network-based and host-based firewalls may be configured to block **Internet Control Message Protocol** (**ICMP**) probes. As a result, hosts may appear to be offline or down.

8. Hosts may have ICMP response disabled on their configurations and won't respond to Ping requests. Therefore, to perform an **Address Resolution Request** (**ARP**) **Request** scan, use the following command:

```
kali@kali:~$ nmap -sn -PR 10.10.10.23
```

The following screenshot shows that the ARP Request scan was able to identify the host is online on the network:

```
kali@kali:~$ nmap -sn -PR 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 08:09 EST
Nmap scan report for 10.10.10.23
Host is up (0.00015s latency).
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
```

Figure 4.15 – ARP Request scan

> **Note**
>
> ARP scans are limited to the local Ethernet network only. If you attempt to scan beyond your local subnet or a layer 3 device such as a router, it will not work. Since ARP is limited to a local subnet, it's considered to be faster than scanning techniques that use TCP and ICMP for discovery.

9. Nmap enables ethical hackers to perform a ping scan using the **User Datagram Protocol (UDP)** packets to identify live hosts on a network while bypassing traditional firewalls that filter ICMP messages. Use the following command to perform a ping scan by sending specially-crafted UDP packets:

```
kali@kali:~$ nmap -sn -PU 10.10.10.23
```

The following screenshot shows Nmap is able to identify the live hosts on a network when using the ping scan with UDP probes:

```
kali@kali:~$ nmap -sn -PU 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 08:19 EST
Nmap scan report for 10.10.10.23
Host is up (0.00015s latency).
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
```

Figure 4.16 – Ping scan using UDP probes

> **Note**
>
> The UDP scan is often considered to be less reliable because probes can be dropped during the scanning process, therefore indicating that a host may be offline or down.

10. Next, to perform a ping scan by sending ICMP ECHO Request messages to the target using a more lightweight and direct method, use the following command:

```
kali@kali:~$ nmap -sn -PE 10.10.10.23
```

The following screenshot shows Nmap was able to identify the live host on the network:

```
kali@kali:~$ nmap -sn -PE 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 08:25 EST
Nmap scan report for 10.10.10.23
Host is up (0.00015s latency).
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
```

Figure 4.17 – ICMP ECHO scan

> **Note**
>
> Keep in mind that many firewalls may be configured to block ICMP; therefore, it may appear that hosts are either offline or down on a network.

11. To perform an ICMP Timestamp scan on a target while bypassing a firewall that filters ICMP ECHO request messages, use the following command:

```
kali@kali:~$ nmap -sn -PP 10.10.10.23
```

The following screenshot shows Nmap was able to perform an ICMP Timestamp scan on a target:

```
kali@kali:~$ nmap -sn -PP 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 08:27 EST
Nmap scan report for 10.10.10.23
Host is up (0.00017s latency).
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
```

Figure 4.18 – ICMP Timestamp scan

> **Note**
>
> While the ICMP Timestamp scan was effective in older systems, modern systems and firewalls ignore ICMP Timestamp requests. Therefore, this type of scan may be unreliable.

12. To perform a ping scan using TCP SYN packets to establish a half-open connection, use the following command:

```
kali@kali:~$ nmap -sn -PS 10.10.10.23
```

As shown in the following screenshot, Nmap was able to detect the live host:

```
kali@kali:~$ nmap -sn -PS 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 08:29 EST
Nmap scan report for 10.10.10.23
Host is up (0.00016s latency).
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.08 seconds
```

Figure 4.19 – Ping scan using TCP SYN packets

Note

The ping scan using TCP SYN packets is effective if the firewall is configured to filter or restrict ICMP messages, as TCP SYN packets are usually allowed on the network.

13. To perform a ping scan using TCP ACK probes to bypass a stateful packet-filtering firewall, use the following command:

```
kali@kali:~$ nmap -sn -PA 10.10.10.23
```

As shown in the following screenshot, Nmap was able to determine the status of the targeted device:

```
kali@kali:~$ nmap -sn -PA 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 08:30 EST
Nmap scan report for 10.10.10.23
Host is up (0.00016s latency).
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
```

Figure 4.20 – Ping scan using TCP ACK probes

Note

Many firewalls are configured to block or restrict ICMP packets; they may permit TCP ACK packets over the network. Therefore, a ping scan using TCP ACK probes may be effective in such scenarios.

14. To perform a ping scan using IP probes using Nmap that bypasses a firewall which filters ICMP, TCP, and UDP packets, use the following command:

```
kali@kali:~$ nmap -sn -PO 10.10.10.23
```

The following screenshot shows the execution of the preceding command:

```
kali@kali:~$ nmap -sn -PO 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 08:31 EST
Nmap scan report for 10.10.10.23
Host is up (0.00018s latency).
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
```

Figure 4.21 – Ping scan using IP probes

> **Note**
>
> Using a ping scan that leverages IP probes is usually effective when scanning a network that has a firewall that's configured to block/restrict ICMP packets, since this scan uses various IP protocols within the header of the packet.

15. (Optional) To use the local AI model to automate the process of performing a network scan using Nmap, use the following command:

```
kali@kali:~$
sgpt --shell "Perform an nmap scan to identify live systems on the
10.10.10.0/24 network, exclude this host, do not scan for open ports"
```

The following screenshot shows the expected outcome of using the local AI mode:

```
kali@kali:~$ sgpt --shell "Perform an nmap scan to identify live systems on t
he 10.10.10.0/24 network, exclude this host, do not scan for open ports"
nmap -sn 10.10.10.0/24 -exclude 10.10.10.1
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 08:35 EST
Nmap scan report for 10.10.10.23
Host is up (0.00013s latency).
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
Nmap scan report for 10.10.10.24
Host is up.
Nmap done: 255 IP addresses (2 hosts up) scanned in 2.61 seconds
```

Figure 4.22 – Using SGPT to automate network scanning

As shown in the preceding screenshot, ethical hackers can simply use **SGPT** to automate a lot of manual tasks such as network scanning. In addition, the AI tool helps ethical hackers to leverage common tools and techniques and ensure their commands are accurate.

> **Tip**
>
> To learn more about Nmap and its scanning capabilities, use the `man nmap` command on Kali Linux.

Having completed this exercise, you have learned common scanning techniques to identify live hosts on a network and the use cases of each technique. Next, you will learn how to use Zenmap to perform host discovery.

Lab: Working with Zenmap for host discovery

While the command-line interface may be overwhelming or fun, it's essential to learn how to use Zenmap, the Graphical User Interface (GUI) version of Nmap to perform scanning. To get started with Zenmap, please use the following instructions:

1. To use **Zenmap** for network scanning, on **Kali Linux**, click on the **Applications** icon (top-left corner) | **01 – Reconnaissance** | **zenmap**:

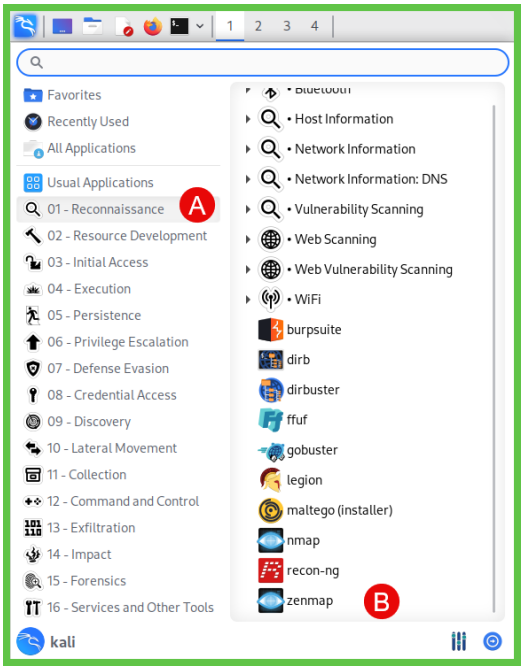


Figure 4.23 – Open zenmap

When **Zenmap** appears, enter the targeted network address within the **Target** field, select the **Ping scan** within the **Profile** menu, and click on **Scan**:

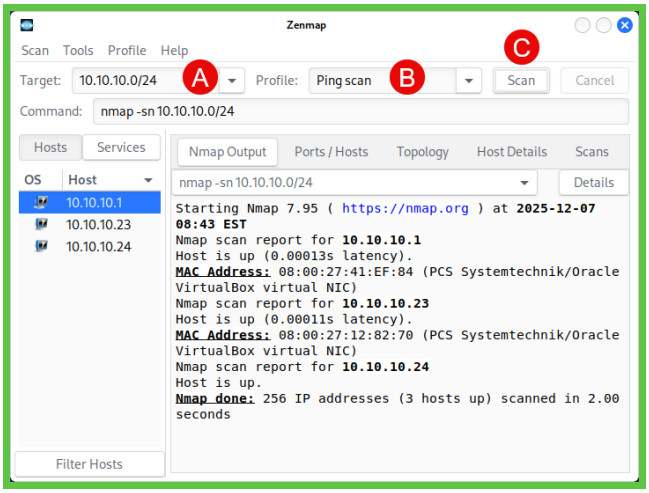


Figure 4.24 – Perform a ping scan using Zenmap

As shown in the preceding screenshot, Zenmap displays the results and lists each IP address it has identified as live/online.

2. Next, click on the **Topology** tab to view a visual representation of the network topology based on the scan results:

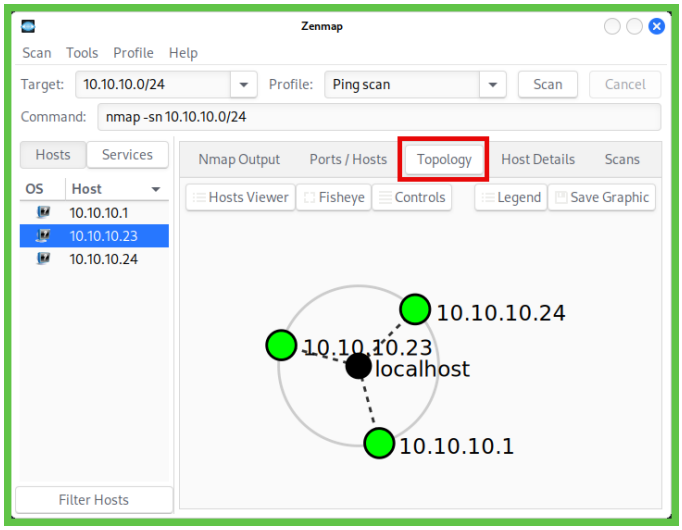


Figure 4.25 – Zenmap topology view

Zenmap enables ethical hackers to easily work with a GUI for convenience. Additionally, you're able to customize the Nmap command within the **Command** field. Having completed this lab, you've gained an introduction to how to use Zenmap for network scanning. In the next section, you'll learn about the importance of identifying open ports and services on targeted systems.

Service port and operating system discovery

The TCP/IP networking model is installed on modern systems with network connectivity. It specifies there are 65,535 service ports on any device. Once a network service is running on a device, the host opens a port to listen for incoming network connections for that service. For instance, a network/system administrator might implement a **Dynamic Host Configuration Protocol** (**DHCP**) server on the network. Once the DHCP service is running on the server, port 67 will be open to listen for any incoming DHCP Discover and DHCP Request messages from clients who are connecting to the network or renewing their address leases. For each unique network service or application-layer protocol, there's a service port associated with it.

It's important for ethical hackers to perform port scanning against their targeted systems and network ranges to identify vulnerable hosts. Sometimes, IT professionals are unaware that they are unnecessarily running services and have unused open ports on their systems. Unnecessary services are any services on a device that are neither essential nor required for the device to either communicate with other systems or are needed by the user.

Furthermore, ethical hackers can fingerprint target systems to identify their operating systems and service versions. Quite often, IT teams fail to maintain and enforce a patch management policy that ensures systems remain up to date, increasing their attack surface.

Therefore, if a real malicious actor can fingerprint targeted systems, they can develop and acquire a working exploit to compromise the target. If ethical hackers identify vulnerabilities within the operating systems and services before malicious actors, they can provide countermeasures to prevent cyber-attacks and threats from affecting those hosts on the network. Therefore, the next step after performing host discovery is to identify open ports, running services, and operating systems on each live host on the network.

> **Note**
>
> For a list of service names and TCP/UDP ports, please see: `https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml`.

We will now look at the three common port scanning techniques in detail: TCP Connect scan, TCP Stealth scan, TCP Inverse scan (with FIN/Xmas/Null), and how TTL values help profile an

operating system or device. These scan types are most commonly used by seasoned professionals to efficiently discover both live and hidden systems on a network.

TCP Connect scan

The **TCP Connect** scan establishes a TCP 3-Way Handshake between the attacker machine and the targeted port number on the target. Once targeted port is open, the TCP 3-way handshake is established. Once the port is open, the attacker machine sends a TCP RST packet to terminate the connection. However, the handshake will not be established if the port is closed.

The following diagram shows the TCP Connect scan establishing the TCP 3-way handshake on an open port on the target:

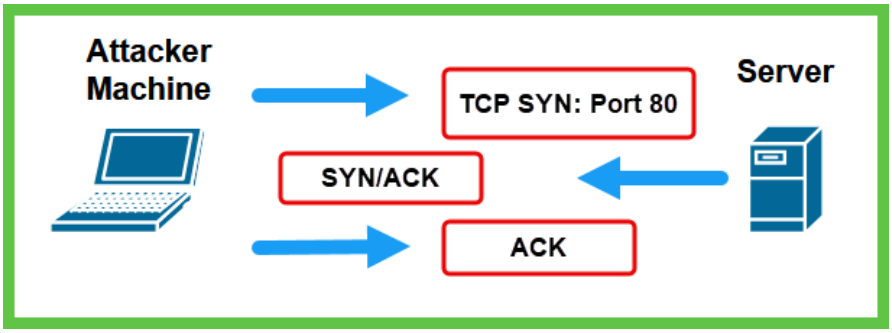


Figure 4.26 – TCP Connect scan

The following diagram shows the TCP Connection scan when a targeted port is closed on a remote host:

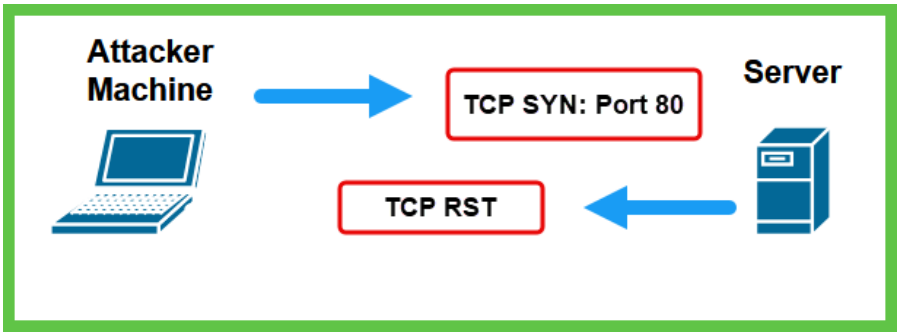


Figure 4.27 – Closed port

TCP stealth scan

The **TCP Stealth** or **Half-Open** scan terminates the TCP 3-Way handshake before it's completed. Therefore, leaving the connection half-open on the target's side. Once the targeted

port responds with a TCP SYN/ACK packet, the port is open and the attacker machine sends a TCP RST packet to terminate the session. However, if the targeted port responds with a TCP RST, the port is considered to be closed.

The following diagram shows the TCP stealth scan:

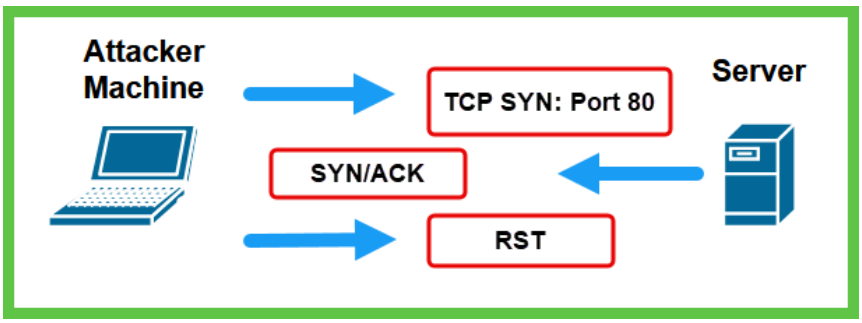


Figure 4.28 – TCP Stealth scan

The following diagram shows a closed port during a TCP stealth scan:

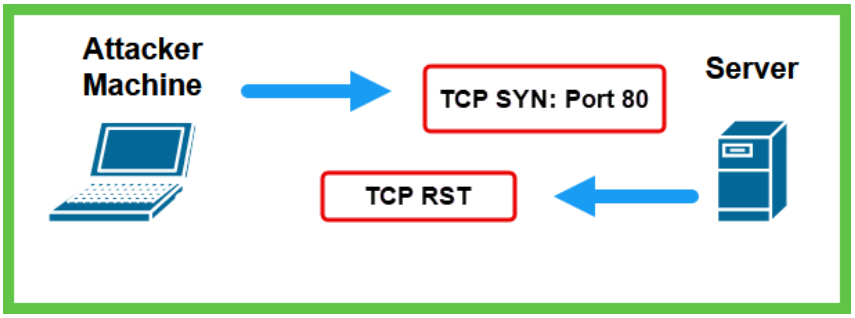


Figure 4.29 – Closed port

The following screenshot shows a Wireshark packet capture of a TCP Stealth scan. The attacker machine (`10.10.10.23`) sent multiple TCP SYN probes to a targeted system (`10.10.10.24`) to identify which of the top 1000 ports are open:

No.	Time	Destination	Source	Protocol	Length	Info
1	0.000000000	ff:ff:ff:ff:…	08:00:27:9c:e…	ARP	42	Who has 10.10.10.23? Tell 10.10.10.24
2	0.000145973	08:00:27:9c:…	08:00:27:12:8…	ARP	60	10.10.10.23 is at 08:00:27:12:82:70
3	0.104104748	10.10.10.23	10.10.10.24	TCP	58	49997 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4	0.104121362	10.10.10.23	10.10.10.24	TCP	58	49997 → 8080 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5	0.104128176	10.10.10.23	10.10.10.24	TCP	58	49997 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
6	0.104132650	10.10.10.23	10.10.10.24	TCP	58	49997 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7	0.104137605	10.10.10.23	10.10.10.24	TCP	58	49997 → 1723 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
8	0.104142275	10.10.10.23	10.10.10.24	TCP	58	49997 → 1720 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
9	0.104148400	10.10.10.23	10.10.10.24	TCP	58	49997 → 143 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
10	0.104154503	10.10.10.23	10.10.10.24	TCP	58	49997 → 587 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
11	0.104159239	10.10.10.23	10.10.10.24	TCP	58	49997 → 53 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
12	0.104167880	10.10.10.23	10.10.10.24	TCP	58	49997 → 111 [SYN] Seq=0 Win=1024 Len=0 MSS=1460

Figure 4.30 – Wireshark packet capture of a Stealth scan

TCP Inverse scan (with FIN/Xmas/Null inside)

The **TCP Inverse** scan is another scanning technique used by ethical hackers to identify open ports on a targeted system. This scan provides advantages such as evading threat detection systems on a network such as basic firewalls. This type of scan works by sending probes with specific TCP flags set, such as FIN, URG, PSH and no flags set (NULL).

The following provides a brief description of each type of TCP Inverse scan:

- FIN scan – The TCP FIN flag is set on the probe
- Xmas scan – The TCP FIN, URG and PSH flags are set on the probe
- NULL scan – No TCP flags are set on the probe

If the targeted system provides no response or no TCP RST packet, then the targeted port is considered to be open, as shown in the following diagram:

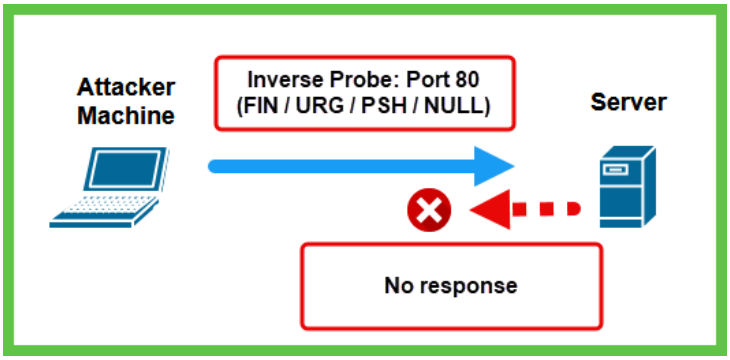


Figure 4.31 – Open ports using a TCP inverse scan

However, if the targeted system responds with a TCP RST/ACK packet, then the targeted port is closed, as shown in the following diagram:

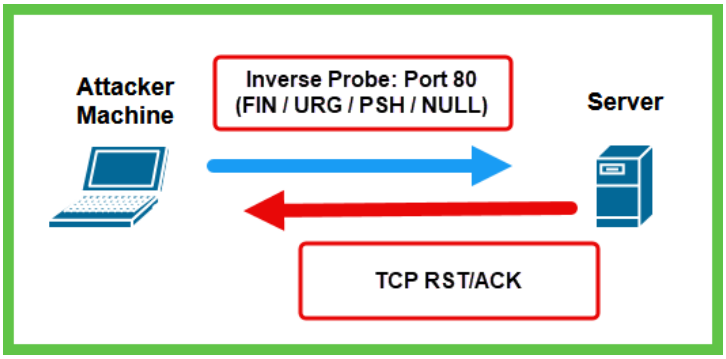


Figure 4.32 – Closed port using a TCP inverse scan

TTL values

While many beginners enjoy identifying open ports on various types of systems, it's also important to profile a targeted host to determine its operating system and running services. As an aspiring ethical hacker, there are various techniques for identifying a host's operating system. One such technique is analyzing the **Time-To-Live** (**TTL**) of the IP header within the TCP response of a targeted system.

The following are common TTL values that are associated with various operating systems:

- Linux = 64
- Windows = 128
- Networking devices = 255

The following diagram shows the TTL responses from each type of device:

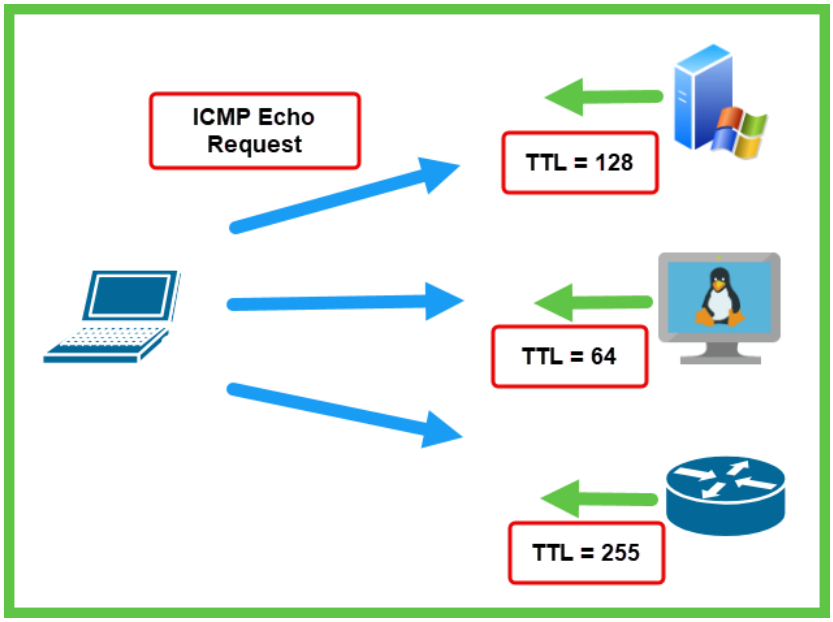


Figure 4.33 – TTL responses

For instance, when an attacker machine sends a TCP SYN probe to a targeted system, the targeted host commonly responds with a TCP SYN/ACK packet. Within the IP header of this TCP SYN/ACK packet is the TTL field and a value, as shown in the following Wireshark packet capture:

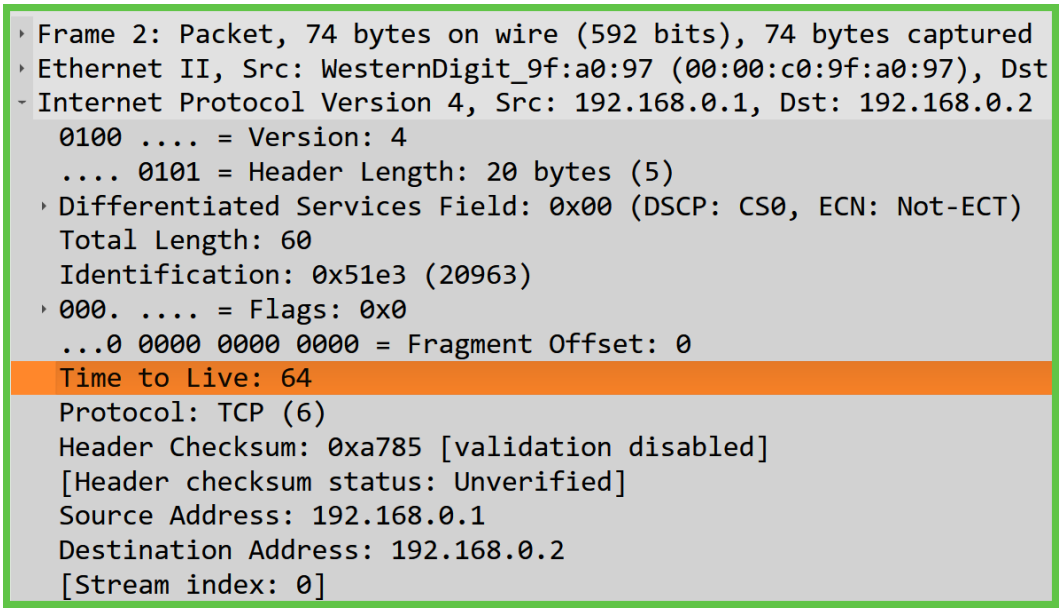


Figure 4.34 – TTL Field

As shown in the preceding screenshot, the TTL value is 64. Therefore, the source device is possibly running a Linux-based operating system.

Please note that it is important to always get a second opinion when profiling the operating system of a targeted system. This process can be automated using Nmap to perform operating system fingerprinting. For instance, Nmap will send special probes to the targeted host, such as ICMP, TCP, and UDP probes that are designed to trigger unique responses for different operating systems. When Nmap gets a response, it analyzes the information from the returned packets and cross-references the data with its fingerprinting database that contains different signatures for known operating systems. Once a match is found, Nmap reports the information and its build version to the user.

Identifying the operating system and its version helps ethical hackers research known security vulnerabilities, acquire or develop working exploits, test for hidden vulnerabilities, and determine the risk and impact level for each vulnerability on the target. Overall, fingerprinting the running operating system helps with improving your plan of attack and creating a topology of the target's network architecture.

Lastly, ethical hackers should be able to identify the service versions of running services found on open ports on devices. Such information helps ethical hackers to further research and identify security vulnerabilities, and determine how they can be exploited. Sometimes, when IT professionals implement new devices on a network and leave unnecessary services running,

if those systems are unpatched and run outdated services, a threat actor can potentially compromise the device. Therefore, it's essential for ethical hackers to identify this type of vulnerability and provide recommendations to prevent or mitigate it before a real cyber-attack occurs.

Having discussed the importance of TCP connect, TCP stealth, and TCP inverse scans, let's dive deeper into learning how to perform these techniques using Kali Linux.

Lab: Using TCP scans to identify open ports

To get started with this exercise, please use the following instructions:

1. Power-on both your **Kali Linux** and **Metasploitable 3** (**Windows-based**) virtual machine via the Oracle VirtualBox Manager.
2. Login to **Kali Linux**, open the **Terminal** and use the following command to identify the IP address of the Kali Linux virtual machine:

```
kali@kali:~$ ip address show eth1
```

The following screenshot shows `eth1` has an IP address on the `10.10.10.0/24` network that's connected to our `Hacking_Lab` virtual network:

```
kali@kali:~$ ip address show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
    link/ether 08:00:27:9c:e6:4e brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.24/24 brd 10.10.10.255 scope global dynamic noprefixroute eth1
       valid_lft 600sec preferred_lft 600sec
    inet6 fe80::50b8:23fd:5bef:53b3/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Figure 4.35 – Network adapter

> **Tip**
>
> It is important to identify your attacker machine's IP address before performing a network scan, as it enables you to remove the IP address from future scans and your results.

3. To perform a *TCP Connect* (*Full Open*) *scan* on Metasploitable 3, use the following command:

```
kali@kali:~$ nmap -sT 10.10.10.23
```

As shown in the following screenshot, the `-sT` syntax enables Nmap to establish a TCP connection to the top 1000 ports of the targeted system and identify which of those ports are open:

```
kali@kali:~$ nmap -sT 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 17:22 EST
Nmap scan report for 10.10.10.23
Host is up (0.00012s latency).
Not shown: 979 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
3389/tcp  open  ms-wbt-server
4848/tcp  open  appserv-http
5985/tcp  open  wsman
7676/tcp  open  imqbrokerd
8009/tcp  open  ajp13
8080/tcp  open  http-proxy
8181/tcp  open  intermapper
8383/tcp  open  m2mservices
```

Figure 4.36 – TCP Connect scan

4. To perform a *Stealth Scan* (*Half-open scan*) using Nmap, use the following command:

```
kali@kali:~$ nmap -sS 10.10.10.23
```

As shown in the following screenshot, Nmap was able to perform a stealth scan on the targeted system:

```
kali@kali:~$ nmap -sS 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 17:24 EST
Nmap scan report for 10.10.10.23
Host is up (0.000093s latency).
Not shown: 979 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
3389/tcp  open  ms-wbt-server
4848/tcp  open  appserv-http
```

Figure 4.37 – Stealth scan

1. To perform a *TCP FIN Scan* on a targeted system, use the following Nmap command:

```
kali@kali:~$ nmap -sF 10.10.10.23
```

As shown in the following screenshot, the targeted system responded with TCP RST packets, which indicates the ports are closed:

```
kali@kali:~$ nmap -sF 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 17:36 EST
Nmap scan report for 10.10.10.23
Host is up (0.000090s latency).
All 1000 scanned ports on 10.10.10.23 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox
```

Figure 4.38 – TCP FIN scan

The Windows operating system is known to ignore **TCP FIN** packets; therefore, it's unreliable to leverage this type of scan on Windows-based systems. However, it's effective on Linux-based systems on a network.

2. Next, to perform a *TCP Null Scan* on a targeted system using Nmap, use the following command:

```
kali@kali:~$ nmap -sN 10.10.10.23
```

As shown in the following figure, the targeted system responded with **TCP RST** messages, therefore indicating the top 1000 ports are closed:

```
kali@kali:~$ nmap -sN 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 17:38 EST
Nmap scan report for 10.10.10.23
Host is up (0.00025s latency).
All 1000 scanned ports on 10.10.10.23 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
```

Figure 4.39 – TCP Null scan

Since the TCP Null probes do not contain any TCP flags, this type of probe may be ignored on Windows-based and other types of operating systems on a network.

3. To perform a TCP Xmas scan on a targeted system, use the following Nmap command:

```
kali@kali:~$ nmap -sX 10.10.10.23
```

As shown in the following screenshot, the `-sX` syntax enables Nmap to set the TCP FIN, URG, and PSH flags on the probes:

```
kali@kali:~$ nmap -sX 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 17:39 EST
Nmap scan report for 10.10.10.23
Host is up (0.000081s latency).
All 1000 scanned ports on 10.10.10.23 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
```

Figure 4.40 – TCP Xmas scan

4. To perform a *TCP Maimon Scan* on a targeted system, use the following Nmap scan:

```
kali@kali:~$ nmap -sM 10.10.10.23
```

The following screenshot shows the results of the TCP Maimon scan against a Windows-based system:

```
kali@kali:~$ nmap -sM 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 17:41 EST
Nmap scan report for 10.10.10.23
Host is up (0.00010s latency).
All 1000 scanned ports on 10.10.10.23 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
```

Figure 4.41 – TCP FIN/ACK scan

Similar to the previous TCP FIN, Null, and Xmas, reset messages were delivered from the targeted system to Kali Linux and Nmap interpreted the ports as closed.

5. To perform a TCP ACK scan using Nmap on a targeted system, use the following command:

```
kali@kali:~$ nmap -sA 10.10.10.23
```

The following screenshot shows the targeted system responding with reset messages that indicate the ports are closed:

```
kali@kali:~$ nmap -sA 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 17:43 EST
Nmap scan report for 10.10.10.23
Host is up (0.000075s latency).
All 1000 scanned ports on 10.10.10.23 are in ignored states.
Not shown: 1000 unfiltered tcp ports (reset)
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
```

Figure 4.42 – TCP ACK scan

Similar to the previous TCP scans, Windows-based systems usually ignore the messages and reset the connection. Therefore, this type of scan is effective on Windows-based devices.

Tip

To perform a *TTL-based TCP ACK scan* on a targeted system, use the `nmap -sA -ttl <time> <target>` command. Where `time` is the TTL value and `target` is the IP address of the targeted system.

Having completed this exercise, you have learned how to use common TCP scanning techniques to identify open ports on a targeted system. Next, you will learn how to scan specific ports on a targeted system.

Lab: Identifying specific open ports

In this exercise, you will learn how to identify whether specific ports or ranges of ports are open on a targeted system. To get started with this exercise, please use the following instructions.

1. Power on both your **Kali Linux** and **Metasploitable 3** (**Windows-based**) virtual machine via the Oracle VirtualBox Manager.
2. Login to **Kali Linux**, open the **Terminal**, and use the following Nmap command to identify whether port 25 is open:

```
kali@kali:~$ nmap -p 25 10.10.10.23
```

As shown in the following screenshot, the `-p` operator enables ethical hackers to scan a specific port number:

```
kali@kali:~$ nmap -p 25 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-20 15:25 EST
Nmap scan report for 10.10.10.23
Host is up (0.00014s latency).

PORT   STATE  SERVICE
25/tcp closed smtp
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)

Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
```

Figure 4.43 – Scanning a specific port

3. Use the following command to scan multiple ports such as ports 25 and 80 on a targeted machine:

```
kali@kali:~$ nmap -p 25,80 10.10.10.23
```

As shown in the following screenshot, the `-p 25,80` operator enables Nmap to scan only those specific ports on the target and port 25 is closed while port 80 is open:

```
kali@kali:~$ nmap -p 25,80 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-20 15:26 EST
Nmap scan report for 10.10.10.23
Host is up (0.0011s latency).

PORT   STATE  SERVICE
25/tcp closed smtp
80/tcp open   http
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)

Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
```

Figure 4.44 – Scanning multiple ports

4. To scan a range of ports, from 25–150 on a targeted system, use the following command:

```
kali@kali:~$ nmap -p 25-150 10.10.10.23
```

As shown in the following screenshot, the `-p 25-150` operator enables Nmap to scan the port range of 25 – 150:

```
kali@kali:~$ nmap -p 25-150 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-20 15:40 EST
Nmap scan report for 10.10.10.23
Host is up (0.000083s latency).
Not shown: 123 closed tcp ports (reset)
PORT    STATE SERVICE
80/tcp  open  http
135/tcp open  msrpc
139/tcp open  netbios-ssn
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)

Nmap done: 1 IP address (1 host up) scanned in 2.12 seconds
```

Figure 4.45 – Port range

Tip

To scan all ports on a targeted system, use the `-p-` operator in Nmap. For instance, the `nmap -p- <target>` command will perform a TCP scan on ports 1–65,535 on the targeted device.

Having completed this exercise, you've learned how to scan specific ports and ranges using Nmap. In the next exercise, you'll learn how to identify running services and profile the operating system of a targeted system.

Lab: Identifying running services and operating systems

In this exercise, you will learn common scanning techniques to profile a targeted system to determine its running services and operating system. Such information is useful to ethical hackers as it enables them to research known security vulnerabilities in those running services and determine whether the operating system has any security flaws.

To get started with this exercise, please use the following instructions:

1. Power on both your **Kali Linux** and **Metasploitable 3** (**Windows-based**) virtual machine via the Oracle VirtualBox Manager.
2. Login to Kali Linux, open the **Terminal**, and use the following Nmap command to identify the service versions of each open port:

```
kali@kali:~$ nmap -sV 10.10.10.23
```

The following screenshot shows Nmap was able to identify the running services of the open port:

```
kali@kali:~$ nmap -sV 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 17:59 EST
Nmap scan report for 10.10.10.23
Host is up (0.000069s latency).
Not shown: 979 closed tcp ports (reset)
PORT      STATE SERVICE              VERSION
21/tcp    open  ftp                  Microsoft ftpd
22/tcp    open  ssh                  OpenSSH 7.1 (protocol 2.0)
80/tcp    open  http                 Microsoft IIS httpd 7.5
135/tcp   open  msrpc                Microsoft Windows RPC
139/tcp   open  netbios-ssn          Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds         Microsoft Windows Server 2008 R2 - 2012
microsoft-ds
3306/tcp  open  mysql                MySQL 5.5.20-log
3389/tcp  open  ms-wbt-server?
4848/tcp  open  ssl/http             Oracle GlassFish 4.0 (Servlet 3.1; JSP 2
.3; Java 1.8)
```

Figure 4.46 – Service versions

As shown in the preceding screenshot, the `-sV` syntax enables Nmap to detect the service versions. Ethical hackers can use the service versions to research security vulnerabilities and exploits:

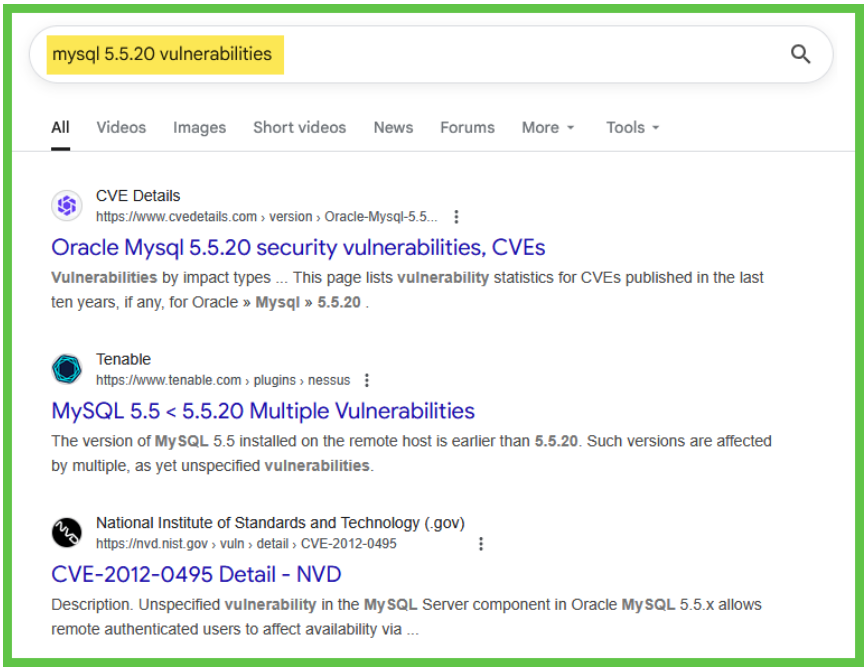


Figure 4.47 – Looking up known vulnerabilities

3. Lastly, to determine the operating system on a targeted device, use the following Nmap command:

```
kali@kali:~$ nmap -O 10.10.10.23
```

As shown in the following screenshot, Nmap was able to identify the target's operating system as Microsoft Windows 2008 R2:

```
Device type: general purpose
Running: Microsoft Windows 2008|7
OS CPE: cpe:/o:microsoft:windows_server_2008:r2 cpe:/o:microsoft:windows_7
OS details: Microsoft Windows Server 2008 R2 SP1 or Windows 7 SP1
Network Distance: 1 hop
```

Figure 4.48 – Identifying operating systems

> **Tip**
>
> The `nmap -A <target>` command enables Nmap to perform a combination of scanning techniques such as operating system detection, service version detection, and script scanning. Additionally, to perform scanning on an IPv6 network, use the `nmap -6 <target>` command.

Next, you will learn how to leverage a local AI model to automate scanning techniques.

Lab: Perform scanning techniques using AI

In this exercise, you will explore how to leverage the local AI model to assist with performing network scanning. To get started with this exercise, please use the following instructions:

1. Power on the **Kali Linux** virtual machine and open the **Terminal**.
2. Use the following **SGPT** command to determine the host address of your Kali Linux machine:

```
kali@kali:~$ sgpt --shell "Identify the host ip address on the eth1
interface of this device"
```

The following screenshot shows SGPT was able to leverage the Llama 3 model to determine the IPv4 address on the `eth1` interface:

```
kali@kali:~$ sgpt --shell "Identify the host ip address on the eth1 interface
 of this device"
ip addr show eth1 | grep "inet" | awk '{print $2}'
[E]xecute, [D]escribe, [A]bort: E
10.10.10.24/24
```

Figure 4.49 – Using SGPT to determine host address

3. Use the following SGPT command to automate a stealth scan on the Metasploitable 3 (Windows-based) virtual machine:

```
kali@kali:~$ sgpt --shell "Perform a stealth port using Nmap on
10.10.10.23"
```

The following screenshot shows SGPT was able to create an Nmap stealth scan to identify which ports are open on the targeted machine:

```
kali@kali:~$ sgpt --shell "Perform a stealth port using Nmap on 10.10.10.23"
nmap -sS -p- 10.10.10.23
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-07 17:55 EST
Stats: 0:00:01 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Sc
an
SYN Stealth Scan Timing: About 0.57% done
Nmap scan report for 10.10.10.23
Host is up (0.00052s latency).
Not shown: 65495 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
1617/tcp  open  nimrod-agent
3306/tcp  open  mysql
3389/tcp  open  ms-wbt-server
3700/tcp  open  lrs-paging
4848/tcp  open  appserv-http
5985/tcp  open  wsman
7676/tcp  open  imqbrokerd
8009/tcp  open  ajp13
8020/tcp  open  intu-ec-svcdisc
8027/tcp  open  papachi-p2p-srv
```

Figure 4.50 – Automating port scanning using AI

With AI-assisted ethical hacking now covered, you are ready to explore techniques for reducing the likelihood of detection while performing network scanning.

Scanning beyond IDS and firewalls

As the cybersecurity industry continues to evolve, network security solutions are becoming better at detecting anomalies and signs of potential intrusions on a network. As a result, it's quite common for network scans to be detected by security monitoring tools and solutions within an organization. Therefore, ethical hackers need to apply similar techniques and procedures as real-world hackers to avoid detection during network scanning.

For instance, security operations teams commonly implement security controls using a Defense-In-Depth approach, that is, multiple layers of security defenses to protect assets from cyber-attacks and threats. However, the security operations team also leverages the monitoring and alerting capabilities of their security solutions. Common security solutions include network-based firewalls for filtering traffic between networks and network-based **intrusion detection systems** (**IDSs**) for detecting and alerting on potential intrusions. Keep in mind, these are simply two of many security solutions that are available in the cybersecurity industry.

The following are some of the common techniques used by ethical hackers to evade both IDS and firewalls on a network:

- Source port manipulation
- Decoys from multiple source IP addresses
- Spoofing the source MAC address

Whenever a network-connected device such as a laptop, server, or even a printer sends a packet on a network, the Transport-header of the packet contains the source and destination port numbers. The source port is usually a random port number between 49,152 and 65,535 (dynamic port range) and is sometimes referred to as an *ephemeral* port because it is opened for a short time on the sender's device. Ethical hackers are able to perform ***source port manipulation***, which enables them to alter the source port number to a well-known port within the range of 1–1,023.

The well-known port range is common within a network. As a result, using source port manipulation as a technique to evade basic or misconfigured firewalls has the potential to allow ethical hackers to perform network scans to discover live hosts and open ports on targeted systems.

The following diagram shows how source port manipulation works:

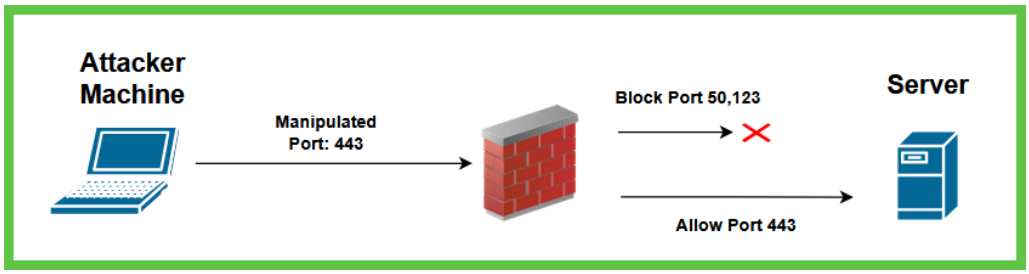


Figure 4.51 – Source port manipulation

Whenever a network-connected device such as a computer sends packets onto a network, its source IP address is automatically inserted within the IP header of the packet. Therefore, network monitoring solutions and the recipient are able to determine the source of the message by inspecting the source address field within the packet. Ethical hackers use the ***IP address decoy*** technique to generate multiple probes with either randomized or specific source IP addresses to evade detection. Therefore, more probes with randomized source addresses should reduce the likelihood that the real sender's address is identified by the security operations teams.

The following diagram shows how the IP address decoy technique works over a network:

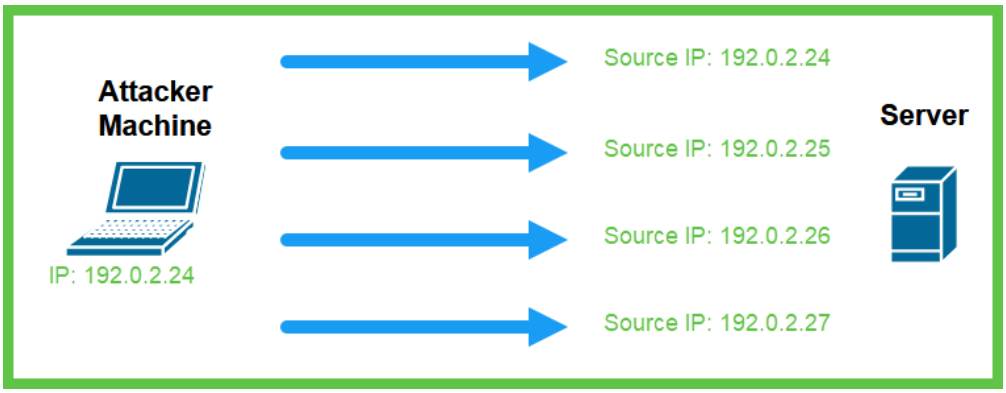


Figure 4.52 – IP address decoys

As previously mentioned, before a device places a message on the network, its source information is inserted. This includes the source **MAC** address of the **Network Interface Card** (**NIC**) of the sender's device. Therefore, security operations teams who are actively monitoring the network can perform packet analysis to determine the type of device based on its MAC address. Therefore, ethical hackers can be inconspicuous by manipulating their source MAC address to either a random or a specific address on their NIC.

The following diagram shows MAC address spoofing, where the fake MAC address is within the probe:

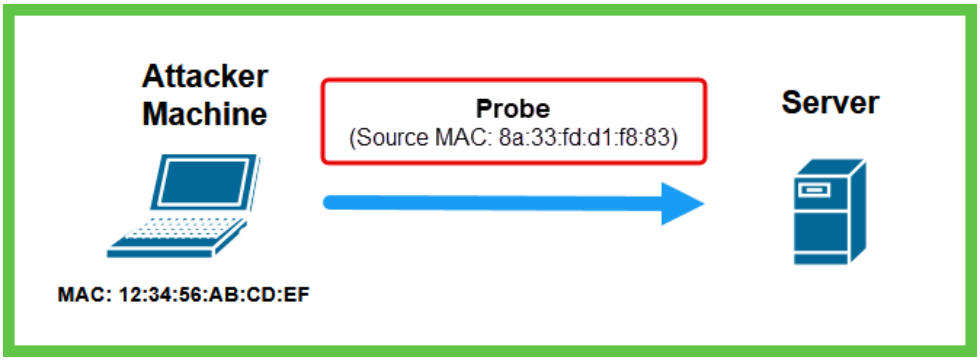


Figure 4.53 – MAC address spoofing

For instance, if you've identified your target network has a specific brand of network printers, then spoofing that vendor MAC address range may be useful and may reduce the risk of being detected on the network. Therefore, this enables ethical hackers to bypass network security solutions that are configured with MAC filtering rules.

Up next, you'll learn how to perform common evasion techniques for network scanning using Nmap.

Lab: Evade threat detection systems with Nmap

In this exercise, you will learn how to use common techniques to reduce your threat level while performing port scanning and host discovery.

To get started with this exercise, please use the following instructions:

1. Power on both your **Kali Linux** and **Metasploitable 3 (Windows-based)** virtual machines via the Oracle VirtualBox Manager.
2. Login to Kali Linux, open the **Terminal**, and use the following command to identify the IP address of the Kali Linux virtual machine:

```
kali@kali:~$ ip address show eth1
```

The following screenshot shows `eth1` has an IP address on the `10.10.10.0/24` network that's connected to our `Hacking_Lab` virtual network:

```
kali@kali:~$ ip address show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
    link/ether 08:00:27:9c:e6:4e brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.24/24 brd 10.10.10.255 scope global dynamic noprefixroute eth1
       valid_lft 600sec preferred_lft 600sec
    inet6 fe80::50b8:23fd:5bef:53b3/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Figure 4.54 – Network adapter

3. To perform a port scan on a targeted system while spoofing your source port, use the following command:

```
kali@kali:~$ nmap --source-port 53 10.10.10.23
```

As shown in the following screenshot, the `--source-port 53` command enables Nmap to use port 53 as the source port:

```
kali@kali:~$ nmap --source-port 53 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-20 16:08 EST
Nmap scan report for 10.10.10.23
Host is up (0.00013s latency).
Not shown: 979 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
3389/tcp  open  ms-wbt-server
4848/tcp  open  appserv-http
```

Figure 4.55 – Spoofing source port with Nmap

4. To use multiple decoys or spoof your source IP address with random addresses, use the following Nmap command:

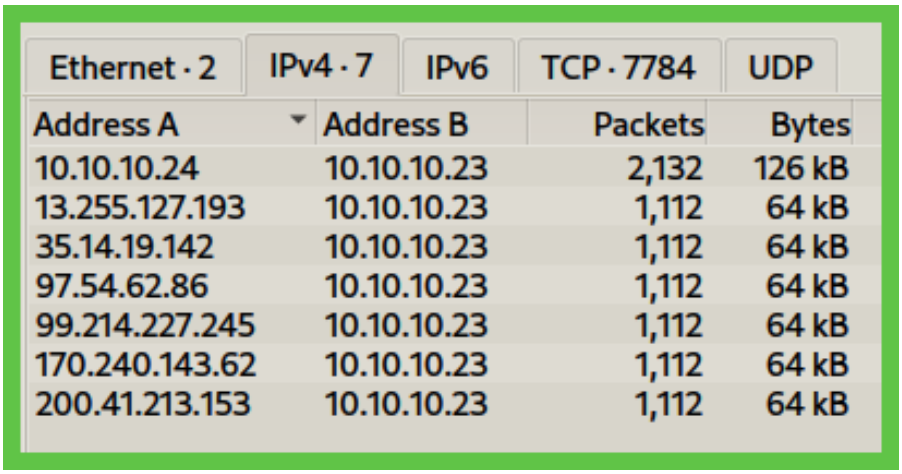
```
kali@kali:~$ nmap -D RND:6 10.10.10.23
```

As shown in the following screenshot, the `-D RND:6` command enables Nmap to use 6 different decoy source addresses when scanning the targeted system:

```
kali@kali:~$ nmap -D RND:6 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-20 16:11 EST
Nmap scan report for 10.10.10.23
Host is up (0.00014s latency).
Not shown: 979 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
3389/tcp  open  ms-wbt-server
4848/tcp  open  appserv-http
```

Figure 4.56 – Decoys with Nmap

Additionally, the following screenshot shows the 6 decoy addresses (column: Address A) that were identified during a packet capture:



Ethernet · 2 | IPv4 · 7 | IPv6 | TCP · 7784 | UDP

Address A	Address B	Packets	Bytes
10.10.10.24	10.10.10.23	2,132	126 kB
13.255.127.193	10.10.10.23	1,112	64 kB
35.14.19.142	10.10.10.23	1,112	64 kB
97.54.62.86	10.10.10.23	1,112	64 kB
99.214.227.245	10.10.10.23	1,112	64 kB
170.240.143.62	10.10.10.23	1,112	64 kB
200.41.213.153	10.10.10.23	1,112	64 kB

Figure 4.57 – Decoy addresses

Security operations teams can still identify which addresses are decoys and which one is the actual source address. For instance, the packet count for each randomized address is the same while the real source address has a higher packet count. Seasoned security operation teams can easily notice such information. However, if an organization is not closely monitoring their network or lacks expertise, this can go unnoticed.

Tip

If you'd prefer to specify the decoy addresses, use the `nmap -D <decoy1>,< decoy2>,< decoy3>,< decoy4> <target>` syntax. For instance, using `nmap -D 10.10.10.100,10.10.10.101,10.10.10.102,10.10.10.104 10.10.10.23` will send decoys from the range `10.10.10.100`–`10.10.10.104`.

5. Lastly, to perform an Nmap scan using the MAC address spoofing technique with a randomized address, use the following command:

```
kali@kali:~$ nmap -Pn -sT --spoof-mac 0 10.10.10.23
```

As shown in the following screenshot, the `--spoof-mac 0` command enables Nmap to use a randomized MAC address:

```
kali@kali:~$ nmap -Pn -sT --spoof-mac 0 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-20 16:22 EST
Spoofing MAC address 04:B1:AA:EF:DB:D3 (No registered vendor)
You have specified some options that require raw socket access.
These options will not be honored for TCP Connect scan.
Nmap scan report for 10.10.10.23
Host is up (0.049s latency).
All 1000 scanned ports on 10.10.10.23 are in ignored states.
Not shown: 990 filtered tcp ports (no-response), 10 filtered tcp ports (host-
unreach)

Nmap done: 1 IP address (1 host up) scanned in 36.19 seconds
```

Figure 4.58 – Spoofing the MAC address

Having learned the fundamentals of evading threat detection systems during network scanning, you are now ready to explore the countermeasures organizations can use to detect scanning activities and improve their security posture.

Scanning countermeasures

As you've learned common techniques and procedures for performing various types of scans, it's also important to understand how to implement countermeasures for preventing and mitigating network scans from unauthorized users on the network.

The following are common countermeasures for ping sweep scans:

- Ensure network-based firewalls are configured to detect and filter inbound ICMP Echo Request probes from both unknown and untrusted sources. While seasoned hackers may be able to bypass this security control, it adds a layer of defense from unskilled malicious actors.
- By implementing rate limiting on firewalls and networking devices, it helps organizations to cap the number of request packets that are being sent over a network segment over a period of time.
- Many next-generation firewalls contain an **Intrusion Prevention System (IPS)** module. IPS are security solutions that are implemented in-line with network traffic to detect network-based intrusions, block them, and alert the security operations team.
- Without alerting, cybersecurity operation teams will not be aware of any potential attacks or threats. Therefore, it's essential to ensure all endpoint and network-based security solutions are configured to send alerts when an anomaly is detected.

The following are common countermeasures for mitigating port scanning on a network:

- Configure network-based firewalls to detect and restrict probes and prevent inbound ICMP packets on a network
- Security professionals should perform regular port scans on their network to detect whether their firewalls are configured using best practices for detecting and preventing port scans
- Ensure end devices, servers, networking devices, and security appliances are regularly patched to ensure their operating systems and firmware are up-to-date
- On host devices, configure the host-based firewall with custom rules to lock down or close unwanted ports
- Configure **Dynamic ARP Inspection** (**DAI**) and port security to prevent ARP poisoning attacks, unauthorized devices from accessing live physical ports on switches and IP spoofing on a network

Summary

Network scanning is essential during ethical hacking, especially prior to the system hacking or exploitation phase. Ethical hackers commonly leverage the information collected during the scanning phase to determine vulnerability points of entry onto a system, determine whether targeted systems are online/live, and gain insights on the network architecture and topology. However, security professionals and teams usually configure their systems to filter common probing techniques such as those which use ICMP.

Hence, as you've learned during this chapter, ethical hackers use a wide variety of scanning techniques to identify live hosts on a network, identify open ports, and determine running services and their versions. Additionally, you've learned how to use Nmap to profile targeted systems to determine their operating system and leverage the local AI model to automate your workflow. Lastly, you've learned common defense mechanisms to help organizations mitigate various scanning techniques.

In the next chapter, you will learn to extract sensitive information from a targeted system and how it can be used to gain unauthorized access.

Further reading

The following are additional reading resources to help reinforce your learning:

- What is network scanning - `https://www.techtarget.com/searchnetworking/definition/network-scanning`
- Host discovery techniques - `https://nmap.org/book/host-discovery-techniques.html`
- Port scanning techniques - `https://nmap.org/book/scan-methods.html`
- TCP RFC 9293 - `https://datatracker.ietf.org/doc/html/rfc9293`
- UDP RFC 768 - `https://www.ietf.org/rfc/rfc768.txt`
- ARP RFC 826 - `https://datatracker.ietf.org/doc/html/rfc826`
- ICMP RFC 792 - `https://datatracker.ietf.org/doc/html/rfc792`

5

Enumerating Systems

Before ethical hackers move onto the system hacking phase, they need to extract sensitive information from their targets by establishing direct communication channels. This enables them to perform direct queries on targeted systems on a network to retrieve information that improves their plan of attack. They can then move on to the exploitation phase.

In this chapter, you will learn what enumeration is and why it's needed by ethical hackers and penetration testers. You will also gain a deeper understanding and develop the hands-on skills needed to retrieve sensitive information from targeted systems and networks, while learning how such details can be used to improve the plan of attack.

In this chapter, you will cover the following topics:

- Enumeration concepts
- Network Basic Input/Output System (NetBIOS) and Simple Network Management Protocol (SNMP) Enumeration
- Simple Mail Transfer Protocol (SMTP) and Domain Name System (DNS) Enumeration
- Common network service enumeration
- Enumeration countermeasures

Technical requirements

To follow along with the exercises in this chapter, please ensure that you have met the following hardware and software requirements:

- Kali Linux - `https://www.kali.org/`

- Metasploitable 2 - `https://www.rapid7.com/products/metasploit/metasploitable/`
- Metasploitable 3 - `https://portal.cloud.hashicorp.com/vagrant/discover/rapid7`

Enumeration concepts

Enumeration is often considered an advanced stage of active reconnaissance, where deeper information is extracted from identified services. It is the set of techniques used by ethical hackers to collect sensitive information by sending specialized queries to targeted systems and devices over a network.

During the enumeration phase, ethical hackers focus on collecting some of the following details:

- Usernames and groups
- Hostnames and IP addresses
- Network resources
- Running services and applications
- Networking details

In the previous chapter, we focused on learning the fundamentals of scanning and developing new skills to perform various types of scans to discover live hosts, open ports, and running services. However, enumeration takes you a step further to retrieve actionable intelligence that helps ethical hackers determine exploitable vulnerabilities on targeted systems.

For instance, discovering open ports helps ethical hackers determine which doorways are open on a system and allows you to identify the services running on the open ports. However, if an open port is running a common remote access protocol such as **Secure Shell** (**SSH**) on port 22, **Telnet** on port 23, or even **Remote Desktop Protocol** (**RDP**) on port 3389, the next step is to determine valid user accounts on the target. An ethical hacker who's unable to retrieve usernames on a targeted system may need to try common pre-built wordlists that contain thousands of possible usernames and passwords. However, if ethical hackers are able to retrieve valid usernames from the target, they can focus on performing online password-based attacks to determine the valid passwords for each account against the remote access services.

IT professionals commonly assign hostnames to their devices, such as routers, switches, access points, and firewalls, that help them easily identify which device they've logged on to and its location. For instance, the IT team of many organizations would either use an existing

hostname convention or develop their own. Nevertheless, many hostnames have the following elements:

- **Location** – This would be an abbreviation of the location, such as *NYC* for New York or *LDN* for London.
- **Environment** – Systems and devices are commonly deployed in many environments such as production, development, testing and quality assurance. Each of these environments can be abbreviated to *PROD*, *DEV*, *TEST* or *QA*.
- **Device type and role** – Each device within a network architecture has a unique role and function, and network professionals commonly include an identifier within the hostname. For instance, the hostnames of firewalls may include *FW*, switches would use *SW*, routers may contain *RTR* and access points use *AP*.
- **Unique identifier** – Within many medium to large organizations, there are multiple devices and systems within the same environment and location. Therefore, appending a unique identifier at the end of the hostname helps IT professionals distinguish one device from another. Hence, using unique IDs such as 01 or even 001 ensures each device has a unique hostname on the network.

Let's combine these hostname conventions. Imagine the IT team wants to deploy a new firewall at the London office within the production network and needs to create a unique hostname for this device. Using the common hostname convention, the structure would be `ldn-prod-fw-002`.

At this point, you're probably wondering why this information is important or relevant for ethical hackers. To put it simply, ethical hackers who are performing a black-box or gray-box security assessment may not have a network topology diagram or network details prior to the reconnaissance phase. However, understanding common hostname conventions can help ethical hackers to create a network topology of their target's infrastructure and determine the location, type, and role of each device on the network. Such information provides meaningful intelligence during the reconnaissance phase.

While many organizations are moving toward the cloud and leveraging the benefits of data centers, there are many organizations with a lot of on-premises, self-hosted solutions on their **Local Area Network** (**LAN**). For instance, some organizations may have many Windows Servers on their LAN that are configured with network resources, such as shared folders and printers. While many IT professionals enforce industry best practices for system configurations on servers, there are many who don't. Ethical hackers commonly seek to enumerate network shares to determine whether they are misconfigured.

The following are common signs of a misconfigured network:

- Default passwords
- Open ports and unnecessary running services
- Permissive firewall rules

If a network share is misconfigured, then ethical hackers will attempt to identify the type of data found and its sensitivity levels. Additionally, misconfigured printer services, and network printers are highly vulnerable to unauthorized access, enabling threat actors to pivot their attacks via the network printer and retrieve valid credentials that are stored within the printer's administrative settings.

If you recall in the previous chapter, Nmap enables ethical hackers to determine the service versions found on each open port on a targeted system. However, ethical hackers can go a step further to determine the running applications and their versions on targeted systems. It's a common practice to get a second opinion in cybersecurity, such as using another tool to ensure you've gotten the same results and have not missed anything. For instance, enumerating running services and their versions can be done during the scanning phase and again during the enumeration phase by using another technique. By determining the applications, services, and their versions, ethical hackers identify exploitable vulnerabilities on the target.

Systems and devices that leak sensitive network information such as **Domain Name System** (**DNS**) details, **routing table**, and **network protocols** provide ethical hackers with better insight into the network topology of their target. For instance, ethical hackers will be able to determine the placement of network devices such as routers and switches, and security appliances like firewalls on the network. Additionally, they will be able to gain a better understanding of how these networks forward traffic to destinations, how firewalls filter traffic between networks and identify common network protocols used within the organization.

In the next section, you'll learn how to perform NetBIOS and SNMP enumeration on a network using common tools and techniques.

NetBIOS and SNMP enumeration

Network Basic Input/Output System (**NetBIOS**) is a legacy network communication protocol that enables devices such as computers on a network to share resources such as printers, files, and information. NetBIOS operates over ports:

- **Port 137** – The **NetBIOS Name Service** (**NetBIOS NS**) is a legacy communication protocol that was used for resolving NetBIOS names to IP addresses on a network. However, systems are vulnerable to **Man-in-the-Middle** (**MITM**) attacks and enable

threat actors to perform operating system fingerprinting to determine the operating system and its version.

- **Port 138** – The **NetBIOS Datagram Service (NetBIOS-DGM)** uses UDP port 138 for network discovery services on a LAN and is commonly used in legacy versions of Microsoft Windows operating systems for sharing files and printers.
- **Port 139** – The **Sever Message Block (SMB)** over NetBIOS is commonly used on Windows- and Linux-based systems for file and printer sharing services on a network. However, this service has known security vulnerabilities that are exploitable. For instance, malicious actors can deliver and execute the EternalBlue exploit on a vulnerable system to gain unauthorized access.

While NetBIOS may not be commonly used on modern networks, there are many networks with legacy systems that contain misconfigurations and default settings. Such systems are susceptible to modern cyberattacks and threats. For instance, NetBIOS does not support encryption which enables threat actors and ethical hackers to intercept and capture sensitive information over a network.

As mentioned throughout this book, organizations hire ethical hackers to identify hidden security vulnerabilities on their systems, determine whether security controls are working as they should and help companies to reduce their cyber-risk. Systems that use NetBIOS allow ethical hackers to retrieve the following details:

- **Hostnames** – Hostnames help ethical hackers to better identify targeted systems. Many organizations use various naming conventions for labelling their devices for asset management. Hostnames help ethical hackers to determine whether their intended targets are online and available.
- **Logged-in users** – Determining logged-in users provides ethical hackers with a list of authorized usernames or accounts that are permitted to access a targeted system. Therefore, ethical hackers can attempt to perform a password-based attack on the username of the permitted account.
- **Domain names** – On a Windows-based environment, system administrators commonly implement **Active Directory Domain Services (AD DS)**, a popular directory service provided by Microsoft Windows Server for managing the users, groups, devices and group policies in a Windows environment. Identifying domain and workgroup names helps ethical hackers determine whether user accounts are managed locally or centrally on the network, and further improve their plan of attack.
- **Shared resources** – Sometimes network shares are misconfigured which enables ethical hackers to access sensitive directories and retrieve confidential information from vulnerable systems.

As an aspiring ethical hacker, always remember that legacy systems still exist within many organizations and networks. Some networks contain older and unpatched versions of Microsoft Windows Server and client systems. Hence, it's essential to understand the type of information that can be retrieved from NetBIOS and how it can be leveraged during the system hacking phase (exploitation).

The **Simple Network Management Protocol** (**SNMP**) operates on port 161. It enables network professionals to remotely manage their networking devices and security appliances on a network. It allows us to retrieve details such as CPU and memory utilization, bandwidth details on each interface, and enables us to monitor the availability of each device on the network. While many IT teams within medium- to large organizations work continuously to ensure maximum availability of their architecture and services, they leverage SNMP to assist with real-time monitoring and alerting.

For instance, network professionals want to know when a device or network segment goes offline in the shortest amount of time, or as quickly as possible. If the networking team of an organization is unaware of an internal network outage, users and services will remain affected, and this affects the overall performance of a team. Therefore, many networking devices and security appliances support SNMP, which enables IT professionals to perform real-time network monitoring using a centralized, network monitoring tool.

SNMP implementation on networks, though widely used, is often accompanied by common misconfigurations such as weak passwords and no support for data encryption; older versions of SNMP do not support authentication. As a result, threat actors gain unauthorized access and leak sensitive information about devices and the network architecture.

When SNMP is misconfigured, systems and devices such as routers, switches, access points, and even firewalls disclose the following information:

- Hostname
- Operating system version
- Networking information such as:
    - Routing table
    - Network interfaces
    - IP addresses
- Running processes and services
- Installed software and applications

A common vulnerability is using default and weak community strings on devices and systems. For instance, many systems are still configured to use `public` as the read-only string and `private` as the read-write string. For instance, if an ethical hacker discovers systems that have

port 161 open, then testing `public` and `private` community strings against the target may allow the ethical hacker to retrieve sensitive information and maybe alter the system configurations.

With a theoretical understanding of NetBIOS, we can move on to the steps taken by ethical hackers to enumerate NetBIOS on vulnerable systems.

Lab: Performing NetBIOS enumeration

To get started with this exercise, please use the following instructions:

1. Power-on the **Kali Linux** and **Metasploitable 3 (Windows-based)** virtual machine. This step is essential for remaining labs in this chapter, therefore ensure both of these virtual machines are powered-on at the start of each lab.
2. Login to **Kali Linux**, open the **Terminal** and use the following Nmap command to identify the Metasploitable 3 virtual machine:

```
kali@kali:~$ nmap -p 137,138,139,445 10.10.10.0/24 --exclude
10.10.10.1,10.10.10.24
```

The preceding Nmap command will enable you to determine whether ports `137`, `138`, `139`, and `445` are open on any devices within the `10.10.10.0/24` network, while excluding `10.10.10.1` (DHCP server) and `10.10.10.24` (Kali Linux) from the scan results, as shown:

```
kali@kali:~$ nmap -p 137,138,139,445 10.10.10.0/24 --exclude 10.10.10.1,10.10
.10.24
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-11 18:08 EST
Nmap scan report for 10.10.10.23
Host is up (0.00015s latency).

PORT    STATE  SERVICE
137/tcp closed netbios-ns
138/tcp closed netbios-dgm
139/tcp open   netbios-ssn
445/tcp open   microsoft-ds
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)

Nmap done: 254 IP addresses (1 host up) scanned in 3.96 seconds
```

Figure 5.1 – Nmap scan for identifying common ports used in NetBIOS

As shown in the preceding screenshot, ports `139` and `445` are open on `10.10.10.23`, the targeted system running Metasploitable 3.

3. Use **NBTscan** to perform a network scan on the entire targeted subnet to determine which systems are susceptible to NetBIOS enumeration:

```
kali@kali:~$ nbtscan 10.10.10.0/24
```

The following screenshot shows NBTscan was able to identify `10.10.10.23` (Metasploitable 3) is responding to NetBIOS enumeration probes:

```
kali@kali:~$ nbtscan 10.10.10.0/24
Doing NBT name scan for addresses from 10.10.10.0/24

IP address       NetBIOS Name     Server    User             MAC address
-------------------------------------------------------------------------------
10.10.10.23      VAGRANT-2008R2   <server>  <unknown>        08:00:27:12:82:7
0
10.10.10.255     Sendto failed: Permission denied
```

Figure 5.2 – NBTscan to identify live systems and hostnames

> **Note**
>
> To learn more about NBTscan, please see: `https://www.kali.org/tools/nbtscan/`.

4. The **Nmap Scripting Engine** (**NSE**) enables ethical hackers to leverage the advanced scanning techniques of Nmap, such as retrieving a targeted device's NetBIOS name and its MAC address. To perform NetBIOS enumeration using Nmap, use the following command:

```
kali@kali:~$ nmap -p 137,138,139,445 --script nbstat 10.10.10.23
```

The following screenshot shows that NSE was able to enumerate the NetBIOS name and MAC address of the targeted device:

```
kali@kali:~$ nmap -p 137,138,139,445 --script nbstat 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-11 18:11 EST
Nmap scan report for 10.10.10.23
Host is up (0.00016s latency).

PORT    STATE  SERVICE
137/tcp closed netbios-ns
138/tcp closed netbios-dgm
139/tcp open   netbios-ssn
445/tcp open   microsoft-ds
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)

Host script results:
| nbstat: NetBIOS name: VAGRANT-2008R2, NetBIOS user: <unknown>, NetBIOS MAC:
 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
| Names:
|   VAGRANT-2008R2<00>   Flags: <unique><active>
|   WORKGROUP<00>        Flags: <group><active>
|_  VAGRANT-2008R2<20>   Flags: <unique><active>

Nmap done: 1 IP address (1 host up) scanned in 1.30 seconds
```

Figure 5.3 – NetBIOS enumeration using Nmap

5. Power off the Metasploitable (Windows-based) virtual machine and power on the Metasploitable 2 virtual machine.
6. On Kali Linux, use the following Nmap command to determine the IP address of the Metasploitable 2 virtual machine:

```
kali@kali:~$ nmap -sn 10.10.10.0/24 --exclude 10.10.10.1,10.10.10.24
```

The following screenshot shows Nmap was able to perform a ping scan and identified the IP address of the Metasploitable 2 virtual machine:

```
kali@kali:~$ nmap -sn 10.10.10.0/24 --exclude 10.10.10.1,10.10.10.24
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-21 19:36 EST
Nmap scan report for 10.10.10.26
Host is up (0.00018s latency).
MAC Address: 08:00:27:65:3E:69 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
Nmap done: 254 IP addresses (1 host up) scanned in 9.75 seconds
```

Figure 5.4 – Ping scan with Nmap

7. Use the **Enum4Linux** tool to perform NetBIOS enumeration on the Metasploitable 2 virtual machine by using the following command:

```
kali@kali:~$ enum4linux 10.10.10.26
```

As shown in the following screenshot, Enum4Linux was able to enumerate network shares:

```
 ========================================( Share Enumeration on 10.10.10.26 )========


        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        tmp             Disk      oh noes!
        opt             Disk
        IPC$            IPC       IPC Service (metasploitable server (Samba 3
        ADMIN$          IPC       IPC Service (metasploitable server (Samba 3
Reconnecting with SMB1 for workgroup listing.

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP

[+] Attempting to map shares on 10.10.10.26

//10.10.10.26/print$    Mapping: DENIED Listing: N/A Writing: N/A
//10.10.10.26/tmp       Mapping: OK Listing: OK Writing: N/A
//10.10.10.26/opt       Mapping: DENIED Listing: N/A Writing: N/A
```

Figure 5.5 – Enumerating network shares to discover exposed directories

Enum4Linux is a powerful tool in ethical hacking as it enables you to enumerate usernames, system names, network shares, password policies and user groups. By leveraging such information, ethical hackers can attempt to perform common system hacking techniques to gain unauthorized access to a targeted system or network.

> **Note**
>
> To learn more about Enum4Linux, please see: `https://www.kali.org/tools/enum4linux/`.

With NetBIOS enumeration complete, let's move on to SNMP enumeration and explore how to gather information from vulnerable systems using this protocol.

Lab: Performing SNMP enumeration

To get started with this exercise, please use the following instructions:

1. Login to **Kali Linux**, open the **Terminal**, and use the following Nmap command to identify the Metasploitable 3 virtual machine:

```
kali@kali:~$ nmap -sU -p 161 10.10.10.0/24 --exclude 10.10.10.1,10.10.10.24
```

The preceding Nmap command perform a UDP scan to determine which systems on the `10.10.10.0/24` network have UDP port `161` open for SNMP, as shown:

```
kali@kali:~$ nmap -sU -p 161 10.10.10.0/24 --exclude 10.10.10.1,10.10.10.24
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-11 18:33 EST
Nmap scan report for 10.10.10.23
Host is up (0.00016s latency).

PORT    STATE SERVICE
161/udp open  snmp
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
```

Figure 5.6 – UDP port scan using Nmap

2. Use **SNMPwalk** with the default community string to determine whether the targeted system (Metasploitable 3) is vulnerable to SNMP enumeration:

```
kali@kali:~$ snmpwalk -v2c -c public 10.10.10.23
```

The following screenshot shows SNMPwalk was able to enumerate system information from the target; therefore the targeted system is vulnerable:

```
kali@kali:~$ snmpwalk -v2c -c public 10.10.10.23
iso.3.6.1.2.1.1.1.0 = STRING: "Hardware: AMD64 Family 25 Model 97 Stepping 2
AT/AT COMPATIBLE - Software: Windows Version 6.1 (Build 7601 Multiprocessor F
ree)"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.311.1.1.3.1.2
iso.3.6.1.2.1.1.3.0 = Timeticks: (321740) 0:53:37.40
iso.3.6.1.2.1.1.4.0 = ""
iso.3.6.1.2.1.1.5.0 = STRING: "vagrant-2008R2"
iso.3.6.1.2.1.1.6.0 = ""
iso.3.6.1.2.1.1.7.0 = INTEGER: 76
iso.3.6.1.2.1.2.1.0 = INTEGER: 18
iso.3.6.1.2.1.2.2.1.1.1 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.1.2 = INTEGER: 2
iso.3.6.1.2.1.2.2.1.1.3 = INTEGER: 3
iso.3.6.1.2.1.2.2.1.1.4 = INTEGER: 4
iso.3.6.1.2.1.2.2.1.1.5 = INTEGER: 5
iso.3.6.1.2.1.2.2.1.1.6 = INTEGER: 6
iso.3.6.1.2.1.2.2.1.1.7 = INTEGER: 7
iso.3.6.1.2.1.2.2.1.1.8 = INTEGER: 8
```

Figure 5.7 – SNMP enumeration to identify running services

3. To retrieve the system description from the target, use the following command:

```
kali@kali:~$ snmpwalk -v2c -c public 10.10.10.23 1.3.6.1.2.1.1.1.0
```

As shown in the following screenshot, SNMPwalk was able to enumerate the system description that helps us determine the hardware and operating system type:

```
kali@kali:~$ snmpwalk -v2c -c public 10.10.10.23 1.3.6.1.2.1.1.1.0
iso.3.6.1.2.1.1.1.0 = STRING: "Hardware: AMD64 Family 25 Model 97 Stepping 2
AT/AT COMPATIBLE - Software: Windows Version 6.1 (Build 7601 Multiprocessor F
ree)"
```

Figure 5.8 – SNMP scan identifying a system's description

4. Use the following SNMPwalk command with the OID to determine the system uptime:

```
kali@kali:~$ snmpwalk -v2c -c public 10.10.10.23 1.3.6.1.2.1.1.3.0
```

As shown in the following screenshot, the uptime helps ethical hackers determine how long a targeted system has been powered on:

```
kali@kali:~$ snmpwalk -v2c -c public 10.10.10.23 1.3.6.1.2.1.1.3.0
iso.3.6.1.2.1.1.3.0 = Timeticks: (6050) 0:01:00.50
```

Figure 5.9 – SNMP update

5. To determine the hostname of a targeted system, use the following SNMPwalk command:

```
kali@kali:~$ snmpwalk -v2c -c public 10.10.10.23 1.3.6.1.2.1.1.5.0
```

As shown in the following screenshot, the hostname helps ethical hackers determine whether they have connected to the actual target:

```
kali@kali:~$ snmpwalk -v2c -c public 10.10.10.23 1.3.6.1.2.1.1.5.0
iso.3.6.1.2.1.1.5.0 = STRING: "vagrant-2008R2"
```

Figure 5.10 – SNMP hostname enumeration

6. To enumerate the running processes on the targeted system, use the following SNMPwalk command:

```
kali@kali:~$ snmpwalk -v2c -c public 10.10.10.23 1.3.6.1.2.1.25.4.2.1.2
```

The running processes help ethical hackers to identify vulnerable services and processes that can be exploited during the system hacking phase:

```
kali@kali:~$ snmpwalk -v2c -c public 10.10.10.23 1.3.6.1.2.1.25.4.2.1.2

iso.3.6.1.2.1.25.4.2.1.2.1 = STRING: "System Idle Process"
iso.3.6.1.2.1.25.4.2.1.2.4 = STRING: "System"
iso.3.6.1.2.1.25.4.2.1.2.252 = STRING: "smss.exe"
iso.3.6.1.2.1.25.4.2.1.2.268 = STRING: "svchost.exe"
iso.3.6.1.2.1.25.4.2.1.2.292 = STRING: "svchost.exe"
iso.3.6.1.2.1.25.4.2.1.2.328 = STRING: "csrss.exe"
iso.3.6.1.2.1.25.4.2.1.2.380 = STRING: "wininit.exe"
iso.3.6.1.2.1.25.4.2.1.2.392 = STRING: "csrss.exe"
iso.3.6.1.2.1.25.4.2.1.2.428 = STRING: "winlogon.exe"
iso.3.6.1.2.1.25.4.2.1.2.472 = STRING: "services.exe"
iso.3.6.1.2.1.25.4.2.1.2.488 = STRING: "lsass.exe"
iso.3.6.1.2.1.25.4.2.1.2.496 = STRING: "lsm.exe"
iso.3.6.1.2.1.25.4.2.1.2.592 = STRING: "svchost.exe"
iso.3.6.1.2.1.25.4.2.1.2.652 = STRING: "VBoxService.exe"
iso.3.6.1.2.1.25.4.2.1.2.712 = STRING: "WmiPrvSE.exe"
```

Figure 5.11 – Identifying gunning processes using SNMP

7. Nmap has the capabilities to enumerate SNMP details such as system processes. Use the following Nmap command to enumerate running processes on the target:

```
kali@kali:~$ nmap -sU -p 161 --script snmp-processes 10.10.10.23
```

As shown in the following screenshot, the Nmap script was able to enumerate running processes on the targeted system:

```
kali@kali:~$ nmap -sU -p 161 --script snmp-processes 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-11 18:51 EST
Nmap scan report for 10.10.10.23
Host is up (0.00018s latency).

PORT    STATE SERVICE
161/udp open  snmp
| snmp-processes:
|   1:
|     Name: System Idle Process
|   4:
|     Name: System
|   252:
|     Name: smss.exe
|     Path: \SystemRoot\System32\
|   268:
|     Name: svchost.exe
|   292:
|     Name: svchost.exe
|   328:
|     Name: csrss.exe
|     Path: %SystemRoot%\system32\
|     Params: ObjectDirectory=\Windows SharedSection=1024,20480,768 Windows=O
n SubSystemType=Windows ServerDll=basesrv,1 ServerDll=winsrv:User
|   380:
|     Name: wininit.exe
```

Figure 5.12 – Nmap enumeration script for detecting running process

8. Lastly, the SNMP Enumerator tool is an easy-to-use automation tool for ethical hackers. Simply specify the IP address of the targeted system and the tool handles everything, as shown below:

```
kali@kali:~$ snmp-check 10.10.10.23
```

As shown in the following screenshot, SNMP-Check was able to enumerate system information, user account details, network information, and running processes:

```
kali@kali:~$ snmp-check 10.10.10.23
snmp-check v1.9 - SNMP enumerator
Copyright (c) 2005-2015 by Matteo Cantoni (www.nothink.org)

[+] Try to connect to 10.10.10.23:161 using SNMPv1 and community 'public'

[*] System information:

  Host IP address               : 10.10.10.23
  Hostname                      : vagrant-2008R2
  Description                   : Hardware: AMD64 Family 25 Model 97 Stepping
 2 AT/AT COMPATIBLE - Software: Windows Version 6.1 (Build 7601 Multiprocesso
r Free)
  Contact                       : -
  Location                      : -
  Uptime snmp                   : 00:08:01.51
  Uptime system                 : 00:07:53.55
  System date                   : 2026-1-11 15:47:02.6
  Domain                        : WORKGROUP

[*] User accounts:

  sshd
  Guest
  greedo
  vagrant
  han_solo
  kylo_ren
```

Figure 5.13 – SNMP-check extracts system information

SNMP enumeration enables ethical hackers to collect system information about vulnerable hosts on a network. Such information enables ethical hackers to identify security vulnerabilities that can be exploited.

> **Note**
>
> To learn more about SNMP-check, please visit: `https://www.kali.org/tools/snmpcheck/`

Having explored the fundamentals of NetBIOS and SNMP enumeration, let's now turn our attention to SMTP and DNS enumeration techniques.

SMTP and DNS enumeration

The **Simple Mail Transfer Protocol (SMTP)** is a common application-layer protocol that is used for sending emails from clients to email servers, and between email servers over a

network. To better understand how SMTP works, the following process occurs when a user sends an email to another email address:

1. An email client such as the native Mail application on Windows, Microsoft Outlook desktop application or Mozilla Thunderbird establishes a communication channel to an email server that's running the SMTP service.
2. The email client and SMTP server perform a handshake and validate the sender (source) and receiver (destination) addresses to ensure they are real addresses and can receive email messages.
3. Once the user has finished composing their email message and clicks on the send button, the email client sends the message using specialized SMTP commands to the SMTP server.
4. Once the SMTP server receives the message from the client, the **Mail Transfer Agent** (**MTA**) that resides on the server checks the destination (recipient) domain and uses **Domain Name System** (**DNS**) to locate the **Mail Exchange** (**MX**) records of the destination's domain name.
5. After sender's SMTP server locates the MX records of the destination's SMTP server, it sends the message.
6. When the recipient's (destination) server receives the message, the **Mail Delivery Agent** (**MDA**) on the server places the incoming messages into the recipient's mailbox (inbox).

Keep in mind, SMTP is used for sending emails only. It's not used for downloading from an email server onto a client or synchronizing emails between a client and server. Additionally, SMTP uses port 25 but sends messages in plaintext, therefore it's an unsecure protocol.

However, **SMTP Secure** (**SMTPS**) uses port 465 or port 587 for sending SMTP messages securely using **Start Transport Layer Security** (**STARTTLS**). Threat actors can attempt to enumerate valid/existing user account details by connecting to SMTP servers. The user account details can be used for social engineering and password-based attacks on known accounts.

The following are common SMTP commands:

- `HELO` – This command is commonly used by email clients to initiate a session between itself and an SMTP server. However, the **Extended HELO** (**EHLO**) is the newer and preferred version as it queries the SMTP server to determine which extensions supported.
- `MAIL FROM` – This command is used to specify the sender's (source) email address.

- `RCPT TO` – This command is used to specify the destination's email address. If there are multiple recipients of an email, this command is applied multiple times.
- `DATA` – This command informs the SMTP server that the email client is ready to send the email message.
- `RSET` – This command resets the current transaction and will discard any stored data.
- `VRFY` – This command queries the SMTP server to verify whether a specific user's email address is valid. Ethical hackers can use this command to enumerate valid usernames and email addresses from a vulnerable email server.
- `EXPN` – This command sends a request to the SMTP server to confirm a mailing list and its membership. Ethical hackers can also use this command to enumerate usernames and valid user accounts.
- `AUTH` – This command enables an email client to login to an SMTP server using a single-factor authentication such as a username and password.

Ethical hackers can attempt to enumerate valid usernames and email addresses from vulnerable SMTP servers. Such information helps ethical hackers to determine valid usernames, which enables them to focus on password-cracking techniques to gain unauthorized access into systems that may permit those users. Additionally, extracting valid email addresses can be used in social engineering attacks such as stealing credentials from potential victims.

The **Domain Name System (DNS)** is an application-layer protocol that enables devices on a network to resolve hostnames to IP addresses. While devices such as computers, servers, networking devices, and security appliances need an IP address to communicate with each other, it's difficult for humans to remember each device's address. Therefore, it would be easier to recall a device's hostname rather than its IP address. In another scenario, many devices on a network are assigned a dynamic IP address which can change at any time without notice to the end user. Hence, if a user is trying to connect to a destination device with the incorrect IP address, the session may not be established if the address is no longer assigned or may connect to another device. To solve this issue, DNS servers contain various records that enable clients on a network to query the records from the DNS server to obtain the IP addresses of hostnames.

The following are common DNS records and their purposes:

- `A` – This record is used to resolve a hostname to an IPv4 address
- `AAAA` – This record is used to resolve a hostname to an IPv6 address
- `MX` – This record specifies the mail exchange servers for a domain
- `PTR` – This record resolves an IP address to a hostname

- `NS` – This record specifies the name servers for a domain
- `TXT` – This record enables you to create a text record on the DNS server

The following process explains how DNS works on a network:

1. The user wants to connect to `www.example.com`, which is a hostname of a server on the internet.
2. The user's device checks its local DNS cache to determine whether it has the record cached and uses the IP address if found.
3. If the DNS record is not found within the local DNS cache of the user's device, the device sends a DNS query to its configured DNS server to retrieve a copy of the `A` record.
4. When the DNS server receives the DNS query, it checks its database to locate the requested record and sends a DNS reply with the information.
5. Once the user's device receives the DNS reply, it uses the IP address from the DNS response as the destination IP address within the IP packet header.
6. Lastly, the user's device sends the packet with the destination IP address onto the network.

The following diagram shows a high-level overview of the DNS process:

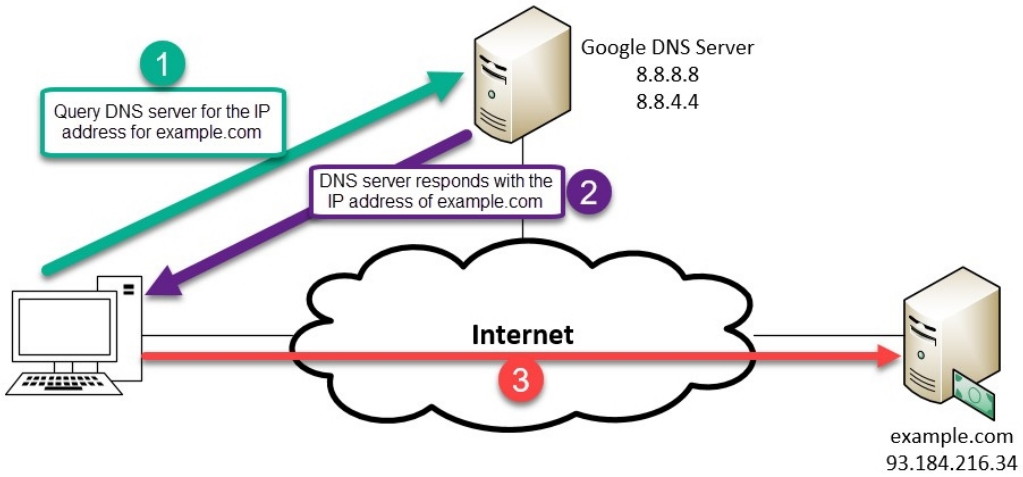


Figure 5.14 – DNS process

Many organizations and internet services rely a lot on the function of DNS and IT professionals work continuously to ensure their DNS servers are working as expected. However, there are various DNS-based attacks that enable threat actors to compromise DNS servers. These are:

- **DNS poisoning** – In this type of attack, the threat actor sends unsolicited DNS reply messages to a victim's device with the intent to modify the contents of the local DNS cache, thereby injecting illegitimate IP addresses that map to common hostnames.
- **DNS hijacking** – This type of attack focuses on compromising the DNS server with the intent to modify the DNS records to redirect users to malicious websites
- **Domain hijacking** – This type of attack focuses on compromising the user account of persons with registered public domain names. This enables the threat actor to modify the DNS records and even transfer ownership of the domain to the attacker.
- **DNS zone transfer** – This attack is successful when a misconfigured DNS server allows an unauthorized device to request a copy of the zone file. This file contains internal addresses and hostnames of devices, and sub-domains. Threat actors use the information to identify targeted systems on an internal network and map an organization's network architecture.

Note

DNS uses the **User Datagram Protocol** (**UDP**) port 53 for client-server communication. For instance, a client such as computer sends DNS Query messages to DNS servers that have port 53 open. However, DNS servers use TCP port 53 for performing DNS Zone Transfer and sharing **DNS Security** (**DNSSEC**) details over a network.

Up next, you will learn how ethical hackers use common tools and techniques to perform SMTP enumeration on a vulnerable email service.

Lab: Performing SMTP enumeration

To get started with the following exercise, please use the following instructions:

1. On **Kali Linux**, open the **Terminal**, and use the following command to identify the IP address of Metasploitable 2 and determine whether SMTP is running on the target:

```
kali@kali:~$ nmap -p 25 10.10.10.0/24 --exclude 10.10.10.1,10.10.10.24
```

As shown in the following screenshot, Nmap was able to determine the IP address of Metasploitable 2, identified that port 25 is open, and that SMTP is running:

```
kali@kali:~$ nmap -p 25 10.10.10.0/24 --exclude 10.10.10.1,10.10.10.24
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-17 18:09 EST
Nmap scan report for 10.10.10.26
Host is up (0.00016s latency).

PORT   STATE SERVICE
25/tcp open  smtp
MAC Address: 08:00:27:65:3E:69 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
```

Figure 5.15 – Identifying SMTP host

2. Use the following Nmap command to determine which SMTP commands are supported on the targeted system (Metasploitable 2):

```
kali@kali:~$ nmap -p 25 --script smtp-commands 10.10.10.26
```

As shown in the following screenshot, the Nmap `smtp-commands` script was able to enumerate various supported SMTP commands, such as `VRFY`:

```
kali@kali:~$ nmap -p 25 --script smtp-commands 10.10.10.26
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-17 18:12 EST
Nmap scan report for 10.10.10.26
Host is up (0.00015s latency).

PORT   STATE SERVICE
25/tcp open  smtp
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY,
 ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
MAC Address: 08:00:27:65:3E:69 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
```

Figure 5.16 – Enumerating supported SMTP commands

3. Use Telnet to establish a connection from Kali Linux to the SMTP service:

```
kali@kali:~$ telnet 10.10.10.26 25
```

The following screenshot shows the connection was successful:

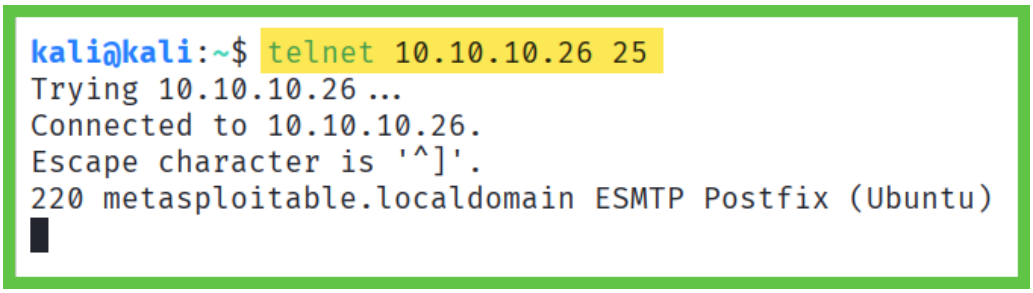


Figure 5.17 – Telnet connection to the targeted host

4. Use the `VRFY <username>` command to test for valid user accounts as shown:

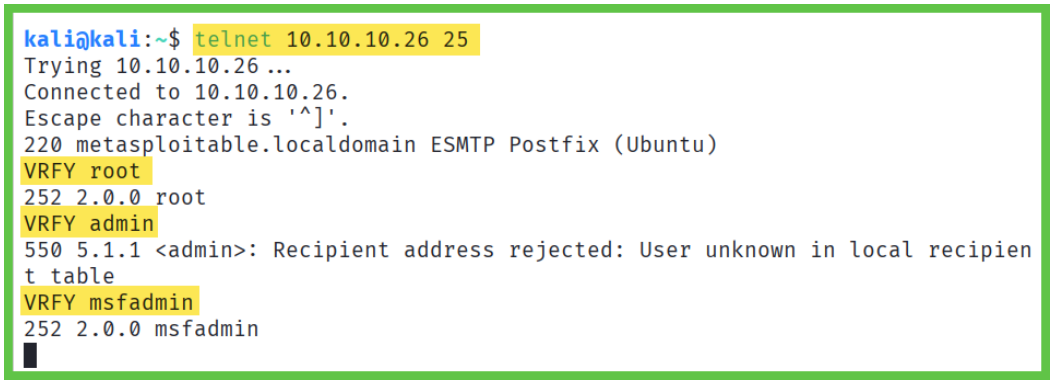


Figure 5.18 – Verifying user accounts

As shown in the preceding screenshot, the `root` and `msfadmin` accounts exist on the targeted system, while the SMTP service sent a rejection notice when the `admin` username was queried.

5. To automate the process of verifying valid usernames with a wordlist, use the following command:

```
kali@kali:~$ smtp-user-enum -M VRFY -U /usr/share/wordlists/metasploit/
unix_users.txt -t 10.10.10.26
```

As shown in the following screenshot, the SMTP-User-Enum tool was able to enumerate a lot of valid usernames on the targeted system:

```
kali@kali:~$ smtp-user-enum -M VRFY -U /usr/share/wordlists/metasploit/unix_u
sers.txt -t 10.10.10.26
Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum
)

 ----------------------------------------------------------
|                   Scan Information                       |
 ----------------------------------------------------------

Mode ..................... VRFY
Worker Processes ......... 5
Usernames file ........... /usr/share/wordlists/metasploit/unix_users.txt
Target count ............. 1
Username count ........... 175
Target TCP port .......... 25
Query timeout ............ 5 secs
Target domain ............

######## Scan started at Sat Jan 17 18:24:44 2026 #########
10.10.10.26: backup exists
10.10.10.26: bin exists
10.10.10.26: daemon exists
10.10.10.26: distccd exists
10.10.10.26: ftp exists
10.10.10.26: games exists
10.10.10.26: gnats exists
10.10.10.26: irc exists
10.10.10.26: libuuid exists
10.10.10.26: list exists
10.10.10.26: lp exists
10.10.10.26: mail exists
10.10.10.26: man exists
10.10.10.26: news exists
10.10.10.26: mysql exists
```

Figure 5.19 – Automating SMTP enumeration

6. To use automated user enumeration using the `RCPT` method on the targeted system, use the following command:

```
kali@kali:~$ smtp-user-enum -M RCPT -U /usr/share/wordlists/metasploit/
unix_users.txt -t 10.10.10.26
```

The following screenshot shows a list of usernames that was enumerated using the `RCPT` command:

```
kali@kali:~$ smtp-user-enum -M RCPT -U /usr/share/wordlists/metasploit/unix_users.txt -t 10.10.10.26
Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )

 ----------------------------------------------------------
|                   Scan Information                       |
 ----------------------------------------------------------

Mode ..................... RCPT
Worker Processes ......... 5
Usernames file ........... /usr/share/wordlists/metasploit/unix_users.txt
Target count ............. 1
Username count ........... 175
Target TCP port .......... 25
Query timeout ............ 5 secs
Target domain ............

######## Scan started at Sat Jan 17 18:26:36 2026 #########
10.10.10.26: backup exists
10.10.10.26: bin exists
10.10.10.26: daemon exists
10.10.10.26: distccd exists
10.10.10.26: games exists
10.10.10.26: ftp exists
10.10.10.26: gnats exists
10.10.10.26: irc exists
10.10.10.26: libuuid exists
10.10.10.26: list exists
10.10.10.26: lp exists
10.10.10.26: mail exists
10.10.10.26: man exists
10.10.10.26: mysql exists
```

Figure 5.20 – RCPT automation

Now that you have learned the fundamentals of SMTP enumeration, let's see how to retrieve sensitive information from vulnerable DNS servers.

Lab: Performing DNS enumeration

During this exercise, you'll learn how to use common DNS enumeration tools and techniques to extract DNS records and identify security flaws on a targeted domain name. To get started with this exercise, please use the following instructions:

1. On Kali Linux, open the **Terminal** and use the following command to ensure the virtual machine has internet connectivity:

```
kali@kali:~$ ping 8.8.8.8 -c 4
```

2. Use the **DNSenum** tool to query Google's DNS server for DNS records of the Microsoft domain:

```
kali@kali:~$ dnsenum --enum --dnsserver 8.8.8.8 microsoft.com
```

The following screenshot shows DNSenum was able to automatically retrieve the various DNS records of the target domain:

```
kali@kali:~$ dnsenum --enum --dnsserver 8.8.8.8 microsoft.com
dnsenum VERSION:1.3.1

-----   microsoft.com   -----


Host's addresses:
__________________

microsoft.com.                           2595     IN    A        13.107.246.4
1
microsoft.com.                           2595     IN    A        13.107.213.4
1


Name Servers:
______________

ns4-39.azure-dns.info.                   1027     IN    A        13.107.206.3
9
ns3-39.azure-dns.org.                    256      IN    A        13.107.222.3
9
ns1-39.azure-dns.com.                    2634     IN    A        150.171.10.3
9
ns2-39.azure-dns.net.                    81       IN    A        150.171.16.3
9


Mail (MX) Servers:
___________________

microsoft-com.mail.protection.outlook.com. 10       IN    A        52.101.10.
5
```

Figure 5.21 – DNSenum tool extracting DNS records

As shown in the preceding screenshot, using a tool such as DNSenum reduces the manual processes and time needed to enumerate DNS records of publicly registered domains on the internet. This allows ethical hackers to improve their time management, especially if they are given limited time to complete a risk assessment within an organization.

Note

To learn more about DNSenum, please see: `https://www.kali.org/tools/dnsenum/`.

Additionally, the nice folks at DigiNinja have set up an intentionally vulnerable domain that enables cybersecurity professionals to learn how DNS zone transfer can be tested. To learn more about their project, please see: `https://digi.ninja/projects/zonetransferme.php`.

3. Let's use DNSenum to identify the Name Servers of a targeted domain and, if vulnerable, perform a DNS zone transfer to extract the records:

```
kali@kali:~$ dnsenum zonetransfer.me
```

As shown in the following screenshot, DNSenum was able to identify the Name Servers of the targeted domain and attempted to perform a DNS zone transfer attack:

```
Host's addresses:
__________________

zonetransfer.me.                         7200     IN    A        5.196.105.14

Name Servers:
______________

nsztm1.digi.ninja.                       10800    IN    A        81.4.108.41
nsztm2.digi.ninja.                       10800    IN    A        5.196.105.10

Mail (MX) Servers:
___________________

ALT1.ASPMX.L.GOOGLE.COM.                 293      IN    A        172.253.116.
27
ASPMX2.GOOGLEMAIL.COM.                   293      IN    A        172.253.116.
27
ASPMX.L.GOOGLE.COM.                      293      IN    A        142.250.31.2
6
ASPMX3.GOOGLEMAIL.COM.                   293      IN    A        192.178.223.
27
ALT2.ASPMX.L.GOOGLE.COM.                 293      IN    A        192.178.223.
27
ASPMX4.GOOGLEMAIL.COM.                   293      IN    A        173.194.76.2
6
```

Figure 5.22 – Identifying Name Servers

The following screenshot shows DNSenum was successful in testing whether the targeted domain is vulnerable to a zone transfer attack and extracted the sensitive records:

```
Trying Zone Transfer for zonetransfer.me on nsztm1.digi.ninja ...
zonetransfer.me.                         7200     IN    SOA                (
zonetransfer.me.                         7200     IN    DNSKEY             (
zonetransfer.me.                         301      IN    TXT                (
zonetransfer.me.                         7200     IN    MX                 0
zonetransfer.me.                         7200     IN    MX                 10
zonetransfer.me.                         7200     IN    MX                 10
zonetransfer.me.                         7200     IN    MX                 20
zonetransfer.me.                         7200     IN    MX                 20
zonetransfer.me.                         7200     IN    MX                 20
zonetransfer.me.                         7200     IN    MX                 20
zonetransfer.me.                         7200     IN    A          5.196.105.14
zonetransfer.me.                         7200     IN    NS         nsztm1.digi.
ninja.
zonetransfer.me.                         7200     IN    NS         nsztm2.digi.
ninja.
zonetransfer.me.                         7200     IN    CERT               (
zonetransfer.me.                         300      IN    HINFO          "Casio
_acme-challenge.zonetransfer.me.         301      IN    TXT                (
_sip._tcp.zonetransfer.me.               14000    IN    SRV                0
14.105.196.5.IN-ADDR.ARPA.zonetransfer.me. 7200     IN    PTR        www.zonetr
ansfer.me.
asfdbauthdns.zonetransfer.me.            7900     IN    AFSDB              1
asfdbbox.zonetransfer.me.                7200     IN    A           127.0.0.1
asfdbvolume.zonetransfer.me.             7800     IN    AFSDB              1
canberra-office.zonetransfer.me.         7200     IN    A          202.14.81.23
0
cmdexec.zonetransfer.me.                 300      IN    TXT                ";
```

Figure 5.23 – DNS zone transfer of vulnerable DNS server

As previously mentioned in this section, misconfigured DNS servers can leak sensitive information to real threat actors, enabling them to gain insights on an organization's network architecture and map the internal/external infrastructure of their network.

With SMTP and DNS enumeration complete, you are ready to explore common techniques for extracting information from Microsoft systems and SSH services to identify security vulnerabilities.

Common network service enumeration

We've discussed how there are many network services within an organization. As an aspiring ethical hacker, it's essential to stay within the scope of your security assessment and within the legal boundaries. Additionally, if you're provided with a testing window, your time will be limited and it's important to focus on the *low-hanging fruits* such as targets and services that are easy. However, it's also important to identify which network services would be most impacted if a real threat actor were to exploit them.

In this section, we'll focus on two common network services:

- Microsoft **Remote Procedure Call (RPC)**
- **Secure Shell (SSH)**

The Microsoft RPC Endpoint Manager enables Windows-based systems to connect with each other and locate common IP services such as **Dynamic Host Configuration Protocol (DHCP)**, DNS and **Windows Internet Name Service (WINS)**. Systems that are running RPC can leak sensitive information such as network information and running services. If a threat actor is able to enumerate such information from systems within a targeted network, they would be able to identify vulnerable services that can be exploited.

Sometimes, a host may have multiple network adapters that are connected to two or more networks. A seasoned hacker may attempt to compromise that host with the intention of gaining access to another network, simply by pivoting attacks through the compromised host.

> **Note**
>
> The Microsoft RPC service operates on port 135 by default.

Additionally, IT professionals commonly set up secure connectivity methods for remotely accessing servers, networking devices, and security appliances over a network. SSH is a secure remote access protocol that enables IT professionals to remotely connect to systems and gain command-line access to the remote device. For instance, if you're a network engineer who's setting up a new router for your network, you'll typically apply your baseline configurations and additional configurations to ensure the device handles the forwarding of packets as intended. However, you'll most likely set up SSH to remotely administer and troubleshoot the device after it's deployed on the network.

> **Note**
>
> SSH operates on port 22 by default.

Threat actors with access to your internal network commonly look for networking devices that are running vulnerable firmware and services. If the intention of a threat actor is to redirect network traffic on the corporate network to an alternate route, the threat actor may seek to compromise the routers on the corporate network. Hence, the threat actor can perform network scans to identify networking devices and remote access services on those devices and then attempt to enumerate information about those remote access services to identify security flaws.

In the upcoming sub-sections, you'll learn how ethical hackers can enumerate RPC and SSH services on a network.

Lab: Enumerating Microsoft RPC services

To get started with the following exercise, please use the following instructions:

1. After powering on the **Kali Linux** and **Metasploitable 3** virtual machines, use the following **Nmap** command to determine whether RPC is running on the targeted system (Metasploitable 3):

```
kali@kali:~$ nmap -p 135,139,445 10.10.10.23
```

The following screenshot shows port 135 is open and running the Microsoft RPC service:

```
kali@kali:~$ nmap -p 135,139,445 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-25 12:25 EST
Nmap scan report for 10.10.10.23
Host is up (0.00015s latency).

PORT    STATE SERVICE
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
```

Figure 5.24 – Checking for RPC

2. Use the `locate` command to identify the location of the `rpcdump.py` script on Kali Linux. This script enables ethical hackers to automate the RPC enumeration process:

```
kali@kali:~$ sudo update
kali@kali:~$ locate rpcdump.py
```

The `updatdb` command enables Kali Linux to create or update the file system database, while the `locate` command is used to identify the location of files:

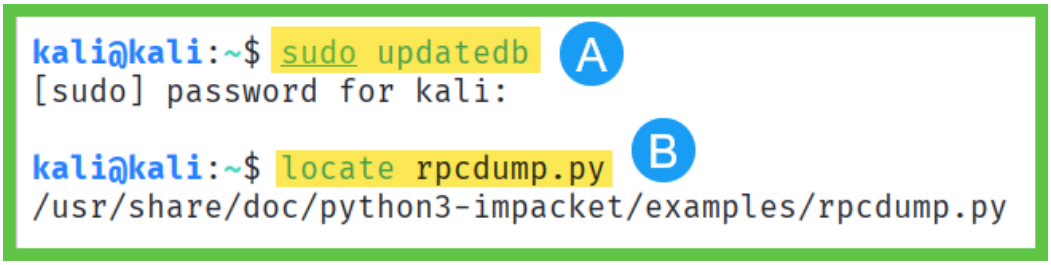


Figure 5.25 – Locating RPCdump script

3. Since the location of `rpcdump.py` script is known, use the following command to automate the RPC enumeration process on the targeted host:

```
kali@kali:~$ /usr/share/doc/python3-impacket/examples/rpcdump.py
10.10.10.23
```

The following screenshot shows the script was able to enumerate the network interfaces and Windows services that are remotely accessible:

```
kali@kali:~$ /usr/share/doc/python3-impacket/examples/rpcdump.py 10.10.10.23
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Retrieving endpoint list from 10.10.10.23
Protocol: [MS-RSP]: Remote Shutdown Protocol
Provider: wininit.exe
UUID    : D95AFE70-A6D5-4259-822E-2C84DA1DDB0D v1.0
Bindings:
          ncacn_ip_tcp:10.10.10.23[49152]
          ncalrpc:[WindowsShutdown]
          ncacn_np:\\VAGRANT-2008R2[\PIPE\InitShutdown]
          ncalrpc:[WMsgKRpc036C40]

Protocol: N/A
Provider: winlogon.exe
UUID    : 76F226C3-EC14-4325-8A99-6A46348418AF v1.0
Bindings:
          ncalrpc:[WindowsShutdown]
          ncacn_np:\\VAGRANT-2008R2[\PIPE\InitShutdown]
          ncalrpc:[WMsgKRpc036C40]
          ncalrpc:[WMsgKRpc0390B1]
```

Figure 5.26 – RPC enumeration

Next, you'll learn how to enumerate the SSH service.

Lab: Enumerating SSH

To get started with this exercise, please use the following instructions:

1. After powering on **Kali Linux** and **Metasploitable 3** (Windows-based), open the **Terminal** on **Kali Linux** and use the following command to determine whether port 22 (SSH) is open on the targeted host (Metasploitable 3):

```
kali@kali:~$ nmap -p 22 -sV 10.10.10.23
```

   The following screenshot shows Nmap was able to identify that port 22 is open and provides the service version of SSH:

```
kali@kali:~$ nmap -p 22 -sV 10.10.10.23
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-25 12:29 EST
Nmap scan report for 10.10.10.23
Host is up (0.00014s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.1 (protocol 2.0)
MAC Address: 08:00:27:12:82:70 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
```

Figure 5.27 – Checking for open ports

2. The **Nmap Scripting Engine** (**NSE**) of Nmap provides additional scanning capabilities such as automating SSH enumeration on a targeted system. Use the following Nmap command to perform SSH enumeration on Metasploitable 3:

```
kali@kali:~$ nmap -p 22 --script ssh-hostkey,ssh2-enum-algos,ssh-auth-
methods 10.10.10.23
```

   The following screenshot shows the various types of encryption algorithms and ciphers that are supported by the targeted system:

```
PORT   STATE SERVICE
22/tcp open  ssh
| ssh-hostkey:
|   2048 fd:08:98:ca:3c:e8:c1:3c:ea:dd:09:1a:2e:89:a5:1f (RSA)
|_  521 7e:57:81:8e:f6:3c:1d:cf:eb:7d:ba:d1:12:31:b5:a8 (ECDSA)
| ssh2-enum-algos:
|   kex_algorithms: (6)
|       curve25519-sha256@libssh.org
|       ecdh-sha2-nistp256
|       ecdh-sha2-nistp384
|       ecdh-sha2-nistp521
|       diffie-hellman-group-exchange-sha256
|       diffie-hellman-group14-sha1
|   server_host_key_algorithms: (2)
|       ssh-rsa
|       ecdsa-sha2-nistp521
|   encryption_algorithms: (6)
|       chacha20-poly1305@openssh.com
|       aes128-ctr
|       aes192-ctr
|       aes256-ctr
|       aes128-gcm@openssh.com
|       aes256-gcm@openssh.com
|   mac_algorithms: (10)
|       umac-64-etm@openssh.com
|       umac-128-etm@openssh.com
|       hmac-sha2-256-etm@openssh.com
```

Figure 5.28 – SSH enumeration

Ethical hackers can use the information found in the preceding screenshot to research known security vulnerabilities. Insecure algorithms are susceptible to various cryptanalysis attacks, such as attempting to reverse the ciphertext (encrypted message) to its original plaintext message or attempting to break the encryption algorithm with the intention of decrypting all messages.

The following screenshot shows the authentication methods that are supported by the SSH service on the targeted host:

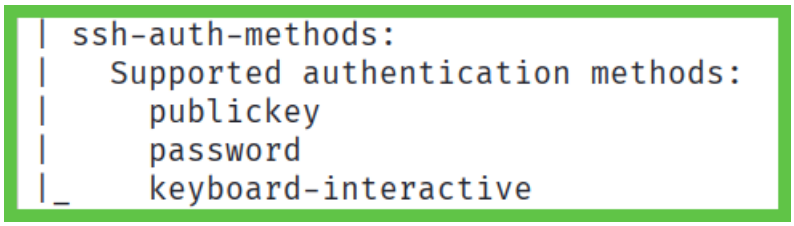


Figure 5.29 – SSH authentication methods

By enumerating the authentication methods can help ethical hackers to improve how they are going to attack the SSH service. For instance, if the targeted host does not support password-based authentication, then an online password-based attack such as brute-force or dictionary-based will be unless and a wastage of time. Therefore, it's essential to determine the authentication methods that's supported by remote access services on targeted systems.

With SSH enumeration complete, let's now examine the countermeasures that can help reduce the risk of enumeration by a threat actor.

Enumeration countermeasures

While it is important to understand how threat actors are able to identify and exploit security vulnerabilities within organizations, it's essential to learn how to prevent these cyberattacks and threats. Keep in mind, when an organization hires an ethical hacker to identify security vulnerabilities, they are also interested in determining how their existing security controls are holding up against real-world cyberattacks, hence the need for ethical hacking services in the industry.

Therefore, ethical hackers should consider the following countermeasures for SNMP enumeration:

- By default, SNMP is not configured on the system and device. However, network professionals commonly configure SNMP to support remote administration and network monitoring. Ensure the default community strings are changed to prevent unauthorized access and enumeration of SNMP-enabled devices. If the default credentials are applied, a threat actor can perform research on the internet to determine what these default credentials are and use them to gain unauthorized access.
- There are 3 versions of SNMP, these are: SNMPv1, SNMPv2, and SNMPv3. Both version 1 and version 2 have security vulnerabilities such as sending the community strings in plaintext over a network and weak authentication mechanisms. Therefore, it's

recommended to disable SNMPv1 and SNMPv2, and use SNMPv3 which supports stronger authentication and data encryption. By using SNMPv3, it will prevent threat actors from viewing sensitive information that's being transmitted over a network.

- Configure network-based **Access Control Lists** (**ACLs**) or firewall policies to permit authorized users to access SNMP-enabled devices. This reduces the risk of unauthorized access.
- Perform regular network traffic auditing to identify suspicious and unauthorized network activities.

The following are common countermeasures for SMTP enumeration:

- To prevent threat actors from retrieving valid usernames, disable both the VRFY and EXPN commands in the email server configuration. These should be restricted to authorized users only.
- Configure the email server to respond with generic responses when handling unknown recipients. For instance, using a generic response such as "Recipient OK" does not provide any error messages or leak sensitive information to threat actors who are attempting to enumerate information.
- Ensure error messages do not reveal sensitive or confidential information.
- Use secure email protocols such as SMTPS, POP3S and IMAP4S.
- Ensure to use SMTPS to encrypt messages over port 587.
- Prevent unauthorized email relays by disabling the open relay function on the email server.
- Implementing email authentication standards such as **Sender Policy Framework** (**SPF**), **DomainKeys Identified Mail** (**DKIM**), and **Domain-based Message Authentication, Reporting and Conformance** (**DMARC**) to apply digital signatures to emails and prevent spoofing.

The following are common countermeasures for RPC and SMB enumeration techniques:

- Configure network-based and host-based firewalls to restrict ports: UDP/TCP 135, UDP 137, UDP 138, TCP 139 and TCP 445 for traffic originating from the internet. This reduces the risk of enumeration from threat actors.
- Disable administrative shares on host devices to prevent access to sensitive directories in a device.
- Configure access controls on the host to permit only authorized users to specific folders and set appropriate permissions. Consider implementing the principle of least privilege for all users.

- Ensure SMBv1 is disabled on all hosts.
- Prevent remote access that uses the local administrator credentials on systems.

The following are common countermeasures for DNS enumeration:

- Configure DNS servers to permit zone transfers to only authorized or trusted secondary DNS servers.
- Disable or remove unnecessary DNS records from the DNS server.
- Ensure the private DNS server is separate from the public DNS server. This prevents data exposure of private network information.
- Consider implementing **DNS Security** (**DNSSEC**) to apply cryptographic signatures to your DNS information and prevent spoofing.
- Consider using **DNS over TLS** (**DoT**) or **DNS over HTTPS** (**DoH**) for encrypting the DNS messages between clients and DNS servers over a network.

The following are common countermeasures for SSH enumeration:

- **Use key-based authentication method** – Disable password-based authentication to prevent online password-based attacks. However, authorized users will need to use a permitted cryptographic key to gain access.
- **Use an uncommon port** – Since SSH uses port 22 by default, it's common to perform a port scan to determine whether port 22 is open or not. However, if system administrators use another port for SSH, a threat actor will need to scan and enumerate all ports on a system to determine which one is running SSH. Therefore, this technique reduces the likelihood of discovering SSH.
- **Limit authorized users** – Create an access control list to permit specific users and systems to access remote devices using SSH. This technique would reduce the risk of SSH sessions originating from untrusted users and devices on a network.
- **Rate limiting** – Using a tool such as Fail2ban enables system administrators to automatically block source IP addresses that show signs of enumeration or SSH-based attacks.
- **Disable outdated algorithms and protocols** – Disabling outdated algorithms can prevent a threat actor from performing various types of cryptographic attacks. Additionally, ensure the SSH service is using the latest version of the protocol and disable outdated versions.

Summary

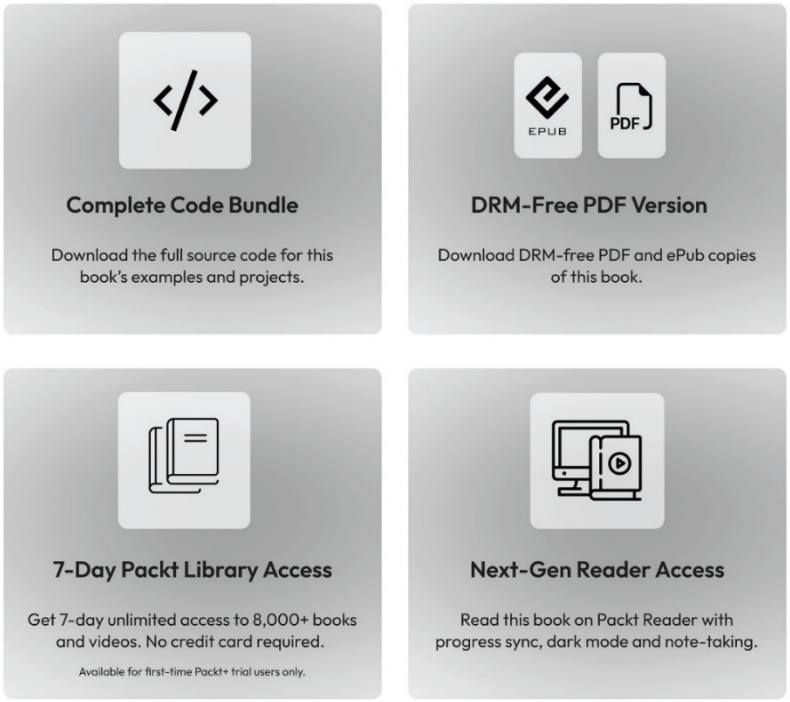
In this chapter, we discussed enumeration. It is a phase during active reconnaissance that helps ethical hackers to gather specific information which can be used to identify and exploit security vulnerabilities. You've gained hands-on skills to perform NetBIOS, SNMP, SMTP, DNS, RPC and SSH enumeration techniques. By enumerating NetBIOS, ethical hackers are able to identify specific information about a device such as its configurations and shared folders which can be exploited.

Additionally, identifying the configurations of other services like SNMP, SMTP, and even DNS allows ethical hackers to test whether misconfigurations exist on the systems and determine how threat actors can attempt to exploit those weaknesses.

6

Unlock the Code Bundle and the PDF Version

Your copy of this book includes the following exclusive benefits:



Follow this guide to unlock them. The process takes only a few minutes and only needs to be completed once.

Unlock this book's free benefits in three easy steps

Step 1

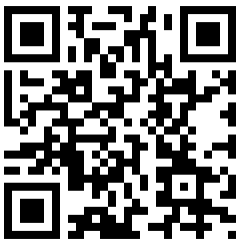
Have your purchase invoice ready for *step 3*. If you have a physical copy, scan it using your phone and save it as a PDF, JPG, or PNG.

For more help on finding your invoice, visit `https://www.packtpub.com/unlock-benefits/help`.

> **Note**
>
> **Note**: If you bought this book directly from the Packt website, no invoice is required. After *step 2*, you can access your exclusive content right away.

Step 2

Scan the QR code or go to `packtpub.com/unlock`.



On the page that opens (similar to *Figure 12.1* on desktop), search for this book by name and select the correct edition.

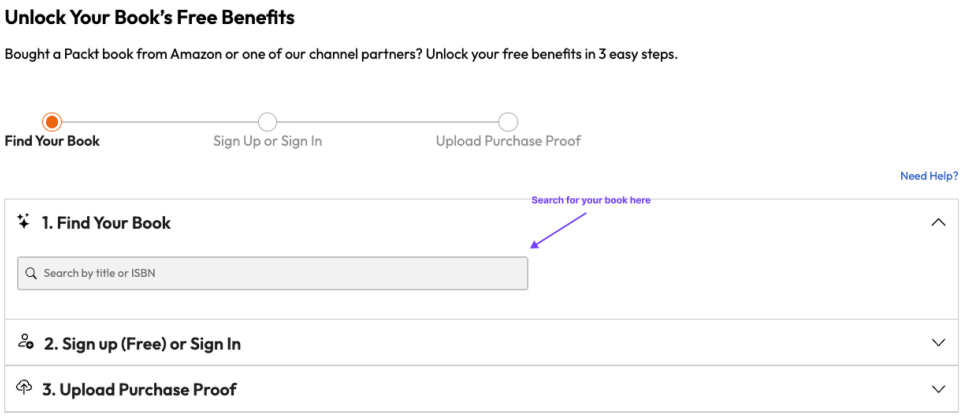


Figure 12.1: Packt unlock landing page on desktop

Step 3

After selecting the book, sign in to your Packt account or create one for free. Then, upload your invoice (PDF, PNG, or JPG, up to 10 MB). Follow the on-screen instructions to finish the process.

Need help?

If you get stuck and need help, visit `https://www.packtpub.com/unlock-benefits/help` for a detailed FAQ on how to find your invoice and more. This QR code will take you to the help page.

Note

Note: If you are still facing issues, reach out to `customercare@packt.com`.



packt.com

Subscribe to our online digital library for full access to over 7,000 books and videos, as well as industry leading tools to help you plan your personal development and advance your career. For more information, please visit our website.

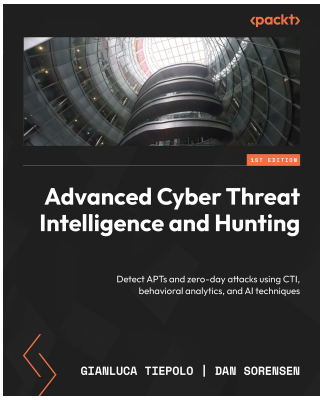
Why subscribe?

- Spend less time learning and more time coding with practical eBooks and Videos from over 4,000 industry professionals
- Improve your learning with Skill Plans built especially for you
- Get a free eBook or video every month
- Fully searchable for easy access to vital information
- Copy and paste, print, and bookmark content

At www.packt.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on Packt books and eBooks.

Other Books You May Enjoy

If you enjoyed this book, you may be interested in these other books by Packt:



Advanced Cyber Threat Intelligence and Hunting

Gianluca Tiepolo, Dan Sorensen

ISBN: 9781806380398

- Build CTI-driven hypotheses for proactive threat hunting
- Detect APT behavior across the Cyber Kill Chain
- Identify zero-day activity using behavioral analytics
- Apply MITRE ATT to map adversary TTPs
- Use machine learning for anomaly-based detection
- Hunt post-exploitation activity and lateral movement
- Investigate threats in cloud and hybrid environments
- Design a scalable, resilient threat hunting program



Automating Security Detection Engineering

Dennis Chow

ISBN: 9781837636419

- Understand the architecture of Detection as Code implementations
- Develop custom test functions using Python and Terraform
- Leverage common tools like GitHub and Python 3.x to create detection-focused CI/CD pipelines
- Integrate cutting-edge technology and operational patterns to further refine program efficacy
- Apply monitoring techniques to continuously assess use case health
- Create, structure, and commit detections to a code repository

Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit `authors.packtpub.com` and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Share your thoughts

Now you've finished *The Ultimate Ethical Hacker's Guide*, we'd love to hear your thoughts! Scan the QR code below to go straight to the Amazon review page for this book and share your feedback or leave a review on the site that you purchased it from.



https://packt.link/r/1806661152

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.

Index

A

ARP scanning concepts **139, 140**

Access Control Lists (ACLs) **211**

Address Resolution Protocol (ARP) **137, 146**

Application Programming Interfaces (APIs) **23**

Autonomous System Number (ASN) **121**

active reconnaissance **20, 87**
- techniques 92

attacker machines **36**

B

black hat hacker **18**

C

CIA Triad **1**

Censys
- reference link 111

Chief Executive Officer (CEO) **29**

Chief Financial Officer (CFO) **29**

Chief Information Security Officer (CISO) **25**

Common Vulnerability and Exposure (CVE) **117**

Content Delivery Networks (CDNs) **122**

Cyber Kill Chain **7**
- Command and Control (C2) 10, 11
- delivery phase 9, 10
- exploitation phase 10
- installation phase 10
- objectives phase 11
- reconnaissance phase 8
- weaponization 9

Cybersecurity Framework (CSF) 2.0 **3**

clear tracks phase **22**

command-line interface (CLI) **60**

cyber terrorists **19**

cyber threat intelligence (CTI) **12**

D

DNS Security Extension (DNSSEC) **122, 197, 212**

DNS enumeration **194 – 196**
- countermeasures 212
- performing 201 – 204

DNS footprinting **121, 122**
- performing 126 – 129

DNS hijacking **197**

DNS over HTTPS (DoH) **212**

DNS over TLS (DoT) **212**

DNS poisoning **197**

DNS zone transfer **197**

DNSenum
- reference link 203

DNSx
- using, to enumerate subdomains 102 – 104

Diffie-Hellman (DH) **86**

Digital Forensics and Incident Response (DFIR) **111**

Digital Millennium Copyright Act (DMCA) **29**

Domain Name System (DNS) **58, 87, 182, 194, 195**

Domain-based Message Authentication, Reporting and Conformance (DMARC) **211**

DomainKeys Identified Mail (DKIM) **211**

Dynamic ARP Inspection (DAI) **176**

Dynamic Host Configuration Protocol (DHCP) **153, 205**

domain hijacking **197**

E

Enum4Linux
reference link 188

ExifTool **108**

email footprinting **111 – 113**
alerting services, setting up 120, 121
for target
performing 113, 114

enumeration
concepts 180 – 182
countermeasures 210

ethical hackers, footprinting
network information 89, 90
organizational information 88, 89
system information 90

ethical hacking **1**
lab design 35 – 39

ethical hacking, concepts **17, 18**
hacker classes 18 – 20
hacker phases 20
scoping and limitations 23, 24

F

Federal Information Security Management Act (FISMA) **29**

firewalls
scanning 169

footprinting **83**
concepts and methodology 84 – 88

G

General Data Protection Regulation (GDPR) **27**

Global Positioning System (GPS) **109**

Google dorking techniques
using, to locate hidden systems 95 – 99

Graphical User Interface (GUI) **19**

gaining access phase **21**

geospatial data
using, to locate object 107 – 111

geospatial platforms **87**

gray hat hacker **18**

H

Health Insurance Portability and Accountability Act (HIPAA) **28**

hacker **18**

hacker classes **18 – 20**
black hat hacker 18
cyber terrorists 19
gray hat hacker 18
hacktivists 19
script kiddies 19
state-sponsored 19
suicide hacker 19
white hat hacker 18

hacker phases
clear tracks phase 22
gaining access phase 21
maintaining access phase 22
reconnaissance phase 20
scanning phase 21

hacktivists **19**

host devices
ARP scanning concepts 139, 140
concepts 137
ICMP scanning concepts 137 – 139
IP scanning concepts 143, 144
TCP scanning concepts 141 – 143
techniques 137
UDP scanning concepts 141

hypervisor **36**

hypervisor types
reference link 39

I

ICMP Echo Ping **138**

ICMP Echo Request **138**

ICMP Timestamp Ping **139**
ICMP scanning concepts **137**
IP address decoy **170**
IP scanning concepts **143, 144**
live hosts, identifying with Nmap 144 – 150
Zenmap, working with 151 – 153
ISO/IEC 27001 **28**
Industrial Control Systems (ICS) **94, 107**
International Electrotechnical Commission (IEC) **28**
Internet Archive **86**
reference link 91
Internet Control Message Protocol (ICMP) **137, 146**
Internet Group Management Protocol (IGMP) **143**
Internet Protocol (IP) **137**
Internet of Things (IoT) **94**
Intrusion Prevention System (IPS) **175**
information security **2, 3**
elements 5, 6
planning 3, 4
information security controls **24**
managerial 25
operational 25
physical 25
technical 24
types 25, 26
information security laws and standards **26, 27**
Digital Millennium Copyright Act (DMCA) 29
Federal Information Security Management Act (FISMA) 29
Health Insurance Portability and Accountability Act (HIPAA) 28
ISO/IEC 27001 28
payment card industry data security standard (PCI DSS) 27
Sarbanes-Oxley Act (SOX) 29
internet **36**
intrusion detection systems (IDSs) **169**
scanning 169

K

Kali Linux **37**
configuring 55 – 59
local AI power, implementing with Ollama 60 – 63
network adapter, configuring 52 – 54
setting up 48 – 51

L

Linux-based virtual machine
building 71 – 73
Local Area Network (LAN) **181**
Low Orbit Ion Cannon (LOIC) **19**
large language model (LLM) **59**

M

MAC address **171**
MITRE ATT&CK framework
tactics 12, 13
Mail Delivery Agent (MDA) **194**
Mail Exchange (MX) **194**
Mail Transfer Agent (MTA) **194**
Man-in-the-Middle (MITM) **182**
Media Access Control (MAC) **137**
Metasploitable 2 **37**
virtual machine, setting up 74 – 78
Metasploitable 3 **37**
Microsoft RPC services
enumerating 206, 207
Multi-Factor Authentication (MFA) **4**
maintaining access phase **22**

N

NBTscan
reference link 186
Name Server (NS) records **123**
National Institute of Standards and Technology (NIST) **3**
NetBIOS Datagram Service (NetBIOS-DGM) **183**
NetBIOS Name Service (NetBIOS NS) **182**
NetBIOS enumeration **183**
performing 185 – 188
Netcraft
using 99 – 102
Network Attached Storage (NAS) **2**
Network Basic Input/ Output System (NetBIOS) **182**
Network Interface Card (NIC) **171**
Network Mapper (Nmap) **62**
threat detection systems, evading 172 – 175
Nmap Scripting Engine (NSE) **186, 208**
network protocols **182**
network scanning **135**
fundamentals 134
network service enumeration **204, 205**
Microsoft RPC services, enumerating 206, 207
Secure Shell (SSH), enumerating 208, 209

O

Open Source Intelligence (OSINT) **8, 87, 133**
Operational Technology (OT) **111**
Oracle VirtualBox **37**
Organization for Standardization (ISO) **28**
online forums **86**

P

Payment Card Industry Data Security Standard (PCI DSS) **16, 27**
Personally Identifiable Information (PII) **26, 87**
Protected Health Information (PHI) **28**
examples 28
passive reconnaissance **20**
techniques 86, 87, 91
payment card industry data security standard (PCI DSS)
security 27
penetration testing methodologies **14 – 16**
port scanning **135**
probes **135**
procedures **14**

R

RPC enumeration
countermeasures 211
Random Access Memory (RAM) **40**
Ransomware-as-a-Service (RaaS) **3**
Remote Access Trojans (RATs) **22**
Remote Code Execution (RCE) **17**
Remote Desktop Protocol (RDP) **112, 180**
Remote Procedure Call (RPC) **205**
reconnaissance **83**
countermeasures 129 – 131
methodologies 91, 92
types 84
reconnaissance phase **20**
active reconnaissance 20
passive reconnaissance 20
reconnaissance techniques
exposed assets, identifying 99 – 106

geospatial data, using to locate object 107 – 111
Google dorking techniques, using to locate hidden systems 95 – 99
organization's subdomains, identifying 99 – 106
with search engines and web services 92 – 94

reconnaissance, types
active reconnaissance 84
passive reconnaissance 84

routing table **182**

S

SMB enumeration
countermeasures 211

SMTP Secure (SMTPS) **194**

SMTP enumeration **193 – 196**
countermeasures 211
performing 197 – 201

SNMP enumeration **184**
performing 189 – 193

SNMP-check
reference link 193

SSH enumeration
countermeasures 212

Sarbanes-Oxley Act (SOX) **29**

Secure Shell (SSH) **55, 180, 205**
enumerating 208, 209

Secure Sockets Layer (SSL) **101**

Security Information and Event Management (SIEM) **26**

Sender Policy Framework (SPF) **211**

Sever Message Block (SMB) **183**

Shodan
reference link 91, 111

Simple Mail Transfer Protocol (SMTP) **193**

Simple Network Management Protocol (SNMP) **184**

Standard Operating Procedures (SOPs) **25**

Start Transport Layer Security (STARTTLS) **194**

Subfinder
using, to enumerate subdomains 102 – 104

Supervisory Control and Data Acquisition (SCADA) **98**

scanning countermeasures
implementing 175, 176

scanning phase **21**
extraction 21
port scanning 21
pre-attack phase 21

script kiddies **19**

search engines
using, for reconnaissance techniques 92 – 94

sensitive systems
identifying, with deep web search 114 – 119

service port **153**
TCP Connect scan 154
TCP Inverse scan 156
TCP Stealth scan 154, 155
Time-To-Live (TTL) 157, 158

software development life cycle (SDLC) **14**

source port manipulation **170**

state-sponsored **19**

suicide hacker **19**

T

TCP Connect scan **154**

TCP Inverse scan **156**
FIN scan 156
NULL scan 156
Xmas scan 156

TCP Stealth scan **154, 155**

TCP Synchronization (SYN) Ping scan **141**

TCP scanning concepts **141 – 143**

Telnet **180**

Time-To-Live (TTL) **141, 157, 158**

open ports, identifying 164, 165
running services and operating systems, identifying 166, 167
scanning techniques, performing with AI 168, 169
TCP scan, using to identify open ports 159 – 163

Top-Level Domain (TLD) server **121**

Transmission Control Protocol (TCP) **137**

Transmission Control Protocol/Internet Protocol (TCP/IP) **135**

Transport Layer Security (TLS) **101, 118**

tactics **12, 13**

tactics, techniques and procedures (TTPs) **12**

techniques **13**

threat actors, methodologies and frameworks **7**
Cyber Kill Chain 7

threat detection systems
evading, with Nmap 172 – 175

U

UDP scanning concepts **141**

Unified Extensible Firmware Interface (UEFI) **34**

Uniform Resource Locator (URL) **95**

User Datagram Protocol (UDP) **137, 147, 197**

V

VMware Workstation Pro **38**
reference link 38

Virtual Private Network (VPN) **86, 122**

vSwitch **36**

virtual Network Interface Card (vNIC) **36**

virtualized environment
hypervisor, installing 40 – 44
Oracle VirtualBox, installing 46, 47
setting up 40
virtual network, creating 44 – 46

vulnerability scanning **136**
security flaw types 136

vulnerable machine **36**
Linux-based virtual machine, building 71 – 73
Metasploitable 2 virtual machine, setting up 74 – 78
setting up 64
Windows-based virtual machine, building 65 – 70

vulnerable wireless lab
setting up 78 – 81

W

WHOIS databases **87**

WHOIS footprinting **121, 122**
performing 122 – 126

Wide Area Network (WAN) **88**

Windows Internet Name Service (WINS) **205**

Windows-based virtual machine
building 65 – 70

web footprinting **111 – 113**

web search
used, for identifying sensitive systems 114 – 119

web services
using, for reconnaissance techniques 92 – 94

white hat hacker **1, 18**