

Identity Attack Vectors

Strategically Designing and Implementing Identity Security

—

Second Edition

—

Morey J. Haber
Darran Rolls

Apress®

Identity Attack Vectors: Strategically Designing and Implementing Identity Security, Second Edition

Morey J. Haber
ORLANDO, FL, USA

Darran Rolls
AUSTIN, TX, USA

ISBN-13 (pbk): 979-8-8688-0232-4
ISBN-13 (electronic): 979-8-8688-0233-1
https://doi.org/10.1007/979-8-8688-0233-1

Copyright © 2024 by Morey J. Haber, Darran Rolls

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Susan McDermott
Development Editor: Laura Berendson
Project Manager: Jessica Vakili

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, New York, NY 10004. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on the Github repository: https://github.com/Apress/Identity-Attack-Vectors-Second-Edition. For more detailed information, please visit https://www.apress.com/gp/services/source-code.

Paper in this product is recyclable

It is easier for a threat actor to log in versus hack in!

Table of Contents

About the Authors xi

About the Technical Reviewer xiii

About the Foreword Author xv

Acknowledgments xvii

Foreword xix

Chapter 1: Introduction: "The Machine" 1

Chapter 2: Introduction: "The Human" 3

Chapter 3: An Identity Crisis 7

Chapter 4: Identity As a Business Function 13

Chapter 5: Identity Access Defined 21

Authentication 22

Authorization 23

Administration 24

Audit 25

Analytics 26

Chapter 6: Understanding Enterprise Identity 27

Personas 29

Physical Persona 30

Electronic Persona 30

Accounts 32

Password 33

Passcode 33
Passkey 34
Certificate 34
Credentials 35
Users 35
Applications 37
Machines 40
Ownership 41
Automation 42
Types of Accounts 43
Local Accounts 44
Centralized Accounts 45
Functional Accounts 46
Managed Accounts 47
Service Accounts 47
Application Accounts 48
Cloud Accounts 49
Entitlements 50
Simple Entitlement 51
Complex Entitlement 51
Roles 52
Business Roles 53
IT Roles 54
Roles and Least Privilege 54
Summary 55

Chapter 7: Identity and Access Management (IAM) 57

Architectures 59
Risks 61
Best Practices 62

Chapter 8: Privileged Access Management (PAM) 65

Chapter 9: Identity Threat Detection and Response (ITDR) 81

ITDR, EDR, and XDR 83

ITDR and Identity Governance and Administration (IGA) 84

ITDR and IAM 84

Privileged Access Management (PAM) and ITDR 85

Chapter 10: Indicators of Compromise 87

Role-Based IoC 88

Hacking Techniques 92

Identity-Based IoCs 105

Chapter 11: Identity Attack Vectors 109

Methods 110

Tactics 111

Implications 114

Privileges 115

Chapter 12: The Identity Cyber Kill Chain 119

Part I: The Cyber Kill Chain 120

Reconnaissance 121

Infiltration 121

Exploitation 122

Exfiltration 123

Part II: Real-World Identity Attack 124

Part III: Identities Under Attack 128

Part IV: Old School 133

Chapter 13: Six Steps to Identity Security 135

1. Identity and Asset Management 136

2. Identity Accountability 138

3. Remote Access 141

4. Implement Least Privilege and Application Control 143
5. Integrate Directory Services 148
6. Identity Security 149

Chapter 14: Evolving Identity Security Threats 153

DevOps (Development Operations) 153
Synthetic Identities 155
Masquerade Attack 158
Operational Technology, IoT, and Nontraditional Endpoints 159
Robotic Process Automation (RPA) 161
Cyber Insurance 162
Artificial Intelligence 167
Secure Remote Access 169
Biometrics 171
Multifactor Authentication 173
Passwordless 176
Ransomware 179

Chapter 15: Complexity Inherent in the IAM System 185

Separate Products and Isolated Infrastructure 185
Doors and Corners 186
Managed Identity Services Platforms 187

Chapter 16: Identity Technical Debt 191

Chapter 17: Identity Digital Transformations 195

Chapter 18: Just-in-Time Access Management 199

Chapter 19: Zero Trust for Identity Security 205

Zero-Trust Details 207
Defining Zero-Trust Architectures 208
Zero-Trust Architectural Models 211
Least Privilege and Zero Trust 214

Privileged Account Control 220
Remote Access 222
Zero-Trust Least Privilege 229
Directory Bridging 234
Measuring Zero Trust 235
Zero-Trust Design Considerations 236

Chapter 20: Identity Obfuscation 239

Chapter 21: Regulatory Compliance 243

Overview 243
Compliance Example 243
US Regulatory Compliance 245
Global Regulatory Compliance 247
Regulatory Compliance Best Practices 248
The Future of Identity-Based Compliance 249

Chapter 22: Key Takeaways 253

Chapter 23: A Final Thought on Vendors 257

Chapter 24: Conclusion 261

Appendix A: Identity Security Sample RFP Questions 263

Appendix B: Department of Defense (DoD) Zero-Trust Framework 273

Index 289

About the Authors



Morey J. Haber is the Chief Security Advisor at BeyondTrust. As the Chief Security Advisor, Morey is the lead technology and identity evangelist at BeyondTrust. He has more than 25 years of IT industry experience and has authored four books: *Privileged Attack Vectors*, *Asset Attack Vectors*, *Identity Attack Vectors*, and *Cloud Attack Vectors*. Morey has previously served as BeyondTrust's Chief Security Officer, Chief Technology Officer, and Vice President of Product Management during his nearly 12-year tenure.

In 2020, Morey was elected to the Identity Defined Security Alliance (IDSA) Executive Advisory Board, assisting the corporate community with identity security best practices. He originally joined BeyondTrust in 2012 as a part of the eEye Digital Security acquisition where he served as a Product Owner and Solutions Engineer since 2004. Prior to eEye, he was Beta Development Manager for Computer Associates, Inc. He began his career as Reliability and Maintainability Engineer for a government contractor building flight and training simulators. Morey earned a Bachelor of Science degree in Electrical Engineering from the State University of New York at Stony Brook.



Darran Rolls has a long history in identity management and security at companies ranging from Tivoli Systems-IBM, Waveset Technologies, Sun Microsystems, and SailPoint. Over the past 20+ years, Darran has helped design, build, and deliver innovative, groundbreaking technology solutions that have shaped the Identity and Access Management industry. For over 12 years, Darran was the Chief Technology Officer at SailPoint, and he spent 4 years as the company's Chief Information Security Officer, leading internal security and compliance through their successful IPO in late 2017. Today, Darran is an investor, advisor, and industry specialist, working with vendors, customers, and financial institutions to understand and take full advantage of the latest identity and security technologies. He works closely with an exclusive portfolio of growth companies to design and deliver the next generation of IAM solutions worldwide. For many years, Darran has been a frequent contributor to IAM standards at OASIS, the W3C, and the IETF. He frequently speaks at industry events and to customers about IAM technologies and security solutions, offering a unique vendor In/Out perspective on designing, delivering, and deploying an identity-centric enterprise security architecture.

About the Technical Reviewer



Derek A. Smith is a cybersecurity expert with a diverse skill set. He currently serves as an IT supervisor in the federal government, is a Lead Professor at the University of North America, and is an Associate Professor of Cybersecurity at the Virginia University of Science and Technology and the Washington University of Science and Technology. Derek also runs a cybersecurity training company. As a prolific author, Derek has completed 13 cybersecurity books. He is a sought-after speaker at cybersecurity events nationwide and conducts webinars as a recognized cyber expert. His career spans roles with IT giants like Computer Sciences Corporation and Booz Allen Hamilton, along with 18 years as a special agent in government and military agencies. Derek's dedication to education is evident through over 35 years of teaching business and IT courses at various universities. His 24-year military service includes the Navy, Air Force, and Army. He holds a doctorate in Strategic Leadership, an MBA, an MS in IT Information Assurance, a master's in IT Project Management, an MS in Digital Forensics, a BS in Education, and multiple associate degrees.

About the Foreword Author



Brandon Haberfeld is the Global Head of Platform Security at Investec Bank Limited, based in Johannesburg, South Africa. He has almost 30 years of experience with commercial IT systems design, architecture, and people management.

Brandon was an undergraduate lecturer in computer science and information systems for 6 years before beginning a corporate career with Investec Bank in 1999 as a systems architect. He has been responsible for large corporate IT implementations, engineering, and deploying technologies to support a range of banking functions, from risk management with distributed and computational analytics to enterprise automation and human resource technologies. He developed and headed the global operating systems teams for Investec for over 10 years and now oversees his organization's journey toward a zero-trust environment. He is passionate about zero-trust architectures, on which he has advised, interviewed, and spoken internationally since 2017. He is a classical pianist and an avid human psychology enthusiast, which gives him a unique perspective on the development of his staff—a perspective which has served him well in his global management roles.

He holds both a Bachelor of Science (Hons) degree in Computer Science in Artificial Intelligence and a Bachelor of Commerce degree in Information Systems from the University of South Africa.

Acknowledgments

An incredibly special thank-you to:

- Matt Miller, Contributing Editor, BeyondTrust
- Laura Bohnert, Technical Editor, BeyondTrust
- Allen Longstreet, Technical Editor, BeyondTrust
- James Maude, Director of Research, BeyondTrust (Contributions on ITDR and Attack Vectors)
- Chris Hills, Chief Security Strategist, BeyondTrust (Contributions on Cyber Insurance)
- Austin Caver, Director of Information Security, BeyondTrust (Attack Vectors, Part II Timeline)

Foreword

How can one ever conceptualize the vast expanse of possible mechanisms exploitable by malicious and nefarious threat actors in today's corporate world?

Having spent a modest amount of time thinking deeply about this, the outcome has had both interesting and surprisingly satisfactory results for me. In recent years, I've had the privilege of serving on the advisory board of a notable US security software company. During one particular session, a customer presented a case study detailing their nightmare end-to-end ransomware breach during the COVID pandemic. I came away not with a mere ethereal impression of what such events look like up close—and in gruesome detail—but with an overstimulated imagination running through the many possibilities of exactly how a fundamental breach of privilege might occur and play out in my own world. Even now, I find myself repeatedly imagining scenarios where a potential threat actor could inflict damage if they were able to masquerade as a stolen identity. And I consider how one might view the protection of corporate assets and identities differently in the days thereafter. I was fundamentally impacted by that board meeting.

A few years ago, I made the critical mistake of debating the nuances of least privilege with an Active Directory administrator whom I highly respect. As the governor of his enterprise's entire identity platform, there was great responsibility that was vested in him and his knowledge. He tried to convince me that host-based security products were arguably not the most important focus for security resources. By that, he meant that deploying strict PASM (Privileged Account and Session Management) controls and least privilege at an operating system level, which is where the abundance of excess administrative privilege usually lives in complex enterprises, might not be as useful as anticipated. General disagreement with his stance would be unsurprising. Rather, he suggested that rights within applications themselves were the most relevant sources of attack, justifying them as the focus of scarce security resources.

The rationale here is that, if a threat actor can successfully gain access to an internal application by masquerading as a legitimate employee who is expected to be using that application, all the threat analytics in the world are not going to prevent damage, financial impact, or otherwise. The application doesn't even have to be a business application, it could be a management plane interface, a source control or deployment

and configuration application, or middleware. One can well imagine that, in a myriad of modern, legacy, homegrown, and third-party applications, he was, in fact, indirectly referring to the protection of identity and the malicious misuse of a single identity. While such an exploitation an identity event should trigger alarms, even in an extremely mature and hygienic environment, it would be after the fact. Artificial intelligence–based threat analytics engines have not been around long enough, nor consequentially gathered enough data to "learn" how to prevent such an event, let alone filter and alert on that risk in real time. This technology is simply not mature yet.

I thought about his assertion a lot, and perhaps there are scenarios where focusing on application protection is more impactful than host-based protection of servers and the platforms they run on. In the financial industry, the former is absolutely the area at which one would expect security policy to be directed poignantly to prevent fraud. However, viewed through the lens of a would-be attacker who is out to "damage and extort" as opposed to "silently exfiltrate," I don't necessarily agree with my friend, who is himself an identity expert. That stance fails to appreciate that, in order to achieve such an identity breach, an entire attack chain of events must already have been significantly completed—unless, of course, the threat actor was either the actual authorized persona "going postal" and abusing their legitimate digital rights (and I have definitely experienced that as an insider threat before), or if it was a machine directed by malware to perform damaging actions within an application or database. It leads me to a profound conclusion:

> *The malicious use of authorized privilege is not an uncommon attack surface in the corporate world, and it is always amazingly unexpected by management that these events occur.*

It does not require a great stretch of the imagination to work out that every expectation of cybersecurity inside the boundary of digital trust can be aligned with four Ws, which simply are: *"Who" can do "What" on "Which" asset, and "When"?* Try to visualize this as a quasi-four-dimensional fabric where a simple predicate applies at the unique intersection of each of the four axes: *allow* or *disallow* access. In this model

- The "When" axis is the temporal aspect governed by tools that can restrict access to privilege based on time.
- The "Where" axis is the entire expanse of an organization's information technology assets—from the humble printer in the corner to the most complex cloud application platforms, comprising thousands of servers on and off premise and everything in between.

- The "What" axis maps directly to the actual privileges granted to an identity. This is where we might attempt to address the potential for misuse of legitimate authority. It includes least privilege and just-in-time access.
- The "Who" axis (which, without all the aforementioned, would be pointless) is all your digital identities, human and machine alike.

The "Who" axis is the entire domain of this book.

Who are the people, processes, and machines trying to perform the actions, commands, and activities (manual and automated) across our digital worlds? Can we know them like we know our clients? Have we constrained their access and capabilities so that only authentic entities can operate in our environment? Can technology track them at every step of the way? And finally, can artificial intelligence fill in any gaps and have reliable conclusions based on the threat? Truthfully, we are not there yet. When it comes to an identity breach, the problem domain is not limited only to a human trying to imitate someone else, such as an attempt to process a fraudulent purchase order, to illegitimately transfer funds between systems, or to modify financial rates in trading systems to cover a short position. It's not limited to fake identities and transactions or to authenticated administrators performing corrective actions on an operating system platform to resolve an issue. There is so much more to consider outside of a business's unique vertical transactions.

An identity breach can also include the vast network of inter-process APIs (application programming interfaces) that glue our systems together that have unvetted hidden privileges. It includes the backbone accounts that our applications rely on to access all the data in their databases. It includes the legitimate privileges we endow our enterprise automation with in order to compute our organizations into tomorrow, day in and day out, and which have all the access rights threat actors dream of for their nefarious missions.

I assert that machine identities are vastly underestimated in their potential to yield damage, and this is what actually disturbs my sleep. Today, machine identities are the most voluminous and difficult to map, track, and secure without breaking the very systems we guaranteed to our clients will continue to operate under all circumstances. The confidence that stable, continuous, and fine-grained security demands from our modern computing environments is a ridiculously tall order to maintain. Any failure therein is always a collective failure.

So, it might just be the case that the entire function of cybersecurity rests on two atomic pathways:

1. First, gaining the right to perform privileged (and if you're matured, also non-privileged) actions
2. Second, understanding what actions a malicious threat actor could perform once those privileges are obtained

This is where Privileged Access Management (PAM) and Privileged Identity Management (PIM) collide and allude to a common security methodology called identity security. They are two sides of the same coin. One does not have a *raison d'etre* (rationale for existence) without the other. Security professionals ought not concentrate on one more than the other because the outcome of the first consequentially determines the impact of the other.

Thus, all roads, in my thoughts, lead back to the same overarching ideology, where I believe identity risk and privileged access meet: the place which we now representationally call *zero trust* and which, for the time being, represents *the ultimate goal of identity security*.

We really cannot trust the sustained authenticity of a human with access privileges at any time, on any asset. Likewise, we cannot trust any machine, API, or network packet at any level of the digital communications stack. To be blunt on my position, I have an unapologetically sterile view on the essence of zero trust and the way it translates into actualized systems in the enterprise. Two axiomatic pillars underpin it: dynamic just-in-time least privilege and single identity management. The former is arguably a classic (but not exclusive) domain of host-based security, and the latter is the entire expanse of the work ahead of you in this book. As a seasoned security professional sharing knowledge, you will explore the nuances of a potential failure within this paradigm. It warrants paying close attention to the detail, because the next abused identity that disrupts a major enterprise and slips through the barriers of electronic surveillance may well be yours. Not only will you be oblivious to it, but it will come from a place in the digital landscape that *you* have never visited nor conceived of as an attack vector.

Finally, it has indeed been a great privilege to listen to, ponder ideas over dinner with, and share the stage with Morey on a number of occasions. Notwithstanding his addiction to *Star Trek*, he is an outspoken and accomplished author of previous works of great magnitude with insights I have come to appreciate. I know you will learn equally as much in the chapters ahead.

CHAPTER 1

Introduction: "The Machine"

"Identity Threat Detection and Response refers to the process of identifying individuals or groups that pose a potential threat to an organization or individual and taking appropriate action to mitigate or neutralize that threat. This can involve a range of activities, including detecting and tracking individuals or groups that may be planning or carrying out malicious activities, analyzing their behavior and patterns of communication, and taking steps to prevent or disrupt their actions.

Effective identity detection and threat response relies on a combination of technology, processes, and human expertise. Automated systems can be used to identify and track individuals or groups based on a variety of factors, including their online behavior, communications, and physical movements. This information can be analyzed by specialized software or teams of analysts to determine the level of threat posed by an individual or group.

There are many different approaches to identity detection and threat response, and the specific methods used will depend on the nature of the threat and the resources available to an organization. Some common methods include:

1. Social media monitoring: This involves monitoring social media platforms for signs of potential threats, such as extremist or violent content, or conversations about planning harmful activities.
2. Behavioral analysis: This involves analyzing an individual's behavior, both online and in the physical world, in order to identify patterns or anomalies that may indicate a potential threat.

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_1

3. Network analysis: This involves analyzing the connections and relationships between individuals or groups in order to identify potential threats.
4. Threat intelligence: This involves gathering and analyzing information about potential threats from a variety of sources, including government agencies, media reports, and open-source intelligence.
5. Physical security measures: These can include measures such as security cameras, access controls, and patrols to prevent or disrupt potential threats.

Effective identity detection and threat response requires a combination of proactive and reactive measures. Proactive measures involve actively seeking out and identifying potential threats, while reactive measures involve responding to threats as they are detected. This can involve a range of activities, including disrupting the plans of individuals or groups, alerting law enforcement agencies, or implementing emergency response procedures.

One of the key challenges in identity detection and threat response is the sheer volume of data that organizations must process in order to identify potential threats. This can make it difficult to distinguish between genuine threats and false alarms and requires organizations to have robust processes in place to ensure that they are able to identify and respond to genuine threats in a timely manner.

Another challenge is the evolving nature of threats, as individuals and groups constantly adapt and change their tactics in order to evade detection. This requires organizations to be agile and adaptable in their approach to identity detection and threat response, and to be able to respond quickly to new and emerging threats.

Overall, identity detection and threat response is a critical aspect of ensuring the safety and security of individuals and organizations. By implementing effective strategies and technologies, organizations can identify and mitigate potential threats and protect against a range of malicious activities.[1]”

[1] https://chat.openai.com/chat

CHAPTER 2

Introduction: "The Human"

As a reader, you may have noticed that the entire previous introduction was enclosed in quotes. The mysterious footnote at the end should help clarify the purpose. The identity that created the introduction on ITDR[1] (Identity Threat Detection and Response) was actually not my penmanship, but rather generative artificial intelligence. This is the reality of the world we live in today.

Machines can behave like humans, but humans have a challenging time behaving as efficiently as machines. Some traits are still absolutely distinguishable between machine and human identities, like failure and truth. Machines fail with error codes, while humans fail with words, pain, and emotions.

Consider the following personal scenario of failure that cannot be attributed to a machine identity (at least not today). It should help you as we explore the differences between human identity management and machine personified identities, and why only humans can truly experience the truth when an attack occurs.

For me, Morey J. Haber, as a human identity (and not John Titor[2] as a fictional identity), I have experienced failure. It can take a long time for someone to admit they failed at a task. In fact, many people never recognize their own failures and do not learn from their mistakes. For some, it is hard to publicly admit failures and even harder to learn what they may have done wrong to cause a failure. However, once you get through the guilt, anger, and insecurity, there may be a lesson that is too often overlooked. For fans of Jean-Luc Picard from *Star Trek: The Next Generation*, a timeless quote emerges from a conversation between him and the android, Data: "It is possible to commit

[1] www.microsoft.com/en-us/security/business/solutions/identity-threat-detection-response

[2] www.documentaries.org/films/john-titor-time-traveler/

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_2

no mistakes and still lose. That is not a weakness; that is life." (*Star Trek: The Next Generation* (1989) Season 2, Episode 21). We see this all the time in medicine. Doctors follow time-tested medical protocols, and for unknown reasons, with everything else known to be right, the patient still experiences severe adverse effects or possibly even dies. The doctor did nothing wrong, but in the end, catastrophe still exists.

As an executive team member, failure can have just as harsh an outcome—including job termination or even collapse of the business due to a simple phishing attack. Outsiders may formulate their own opinions on how or why something failed, but with all facts being equal, some things still happen, well out of anyone's control. And sometimes, there is simply no tangible lesson to be learned. Placing blame never provides a good outcome on an identity or a team, nor does it undo the failure. The lesson, which a machine identity can never learn, is to take the results as they are and self-improve; learn from the mistake, even if the fault was due to something malicious. This is the key to addressing identity attack vectors: learning from your mistakes and failures.

As a cybersecurity professional, I am always trying to learn and improve. I've read countless books and articles on self-improvement, management, and leadership styles, but in the end, they must be compatible with your core personality, and they must allow you to interact with and support others. It is not often that you utterly fail alone, but when you do (at least in your own mind), you can either go down in a blaze of glory or be in absolute denial, searching for answers and someone else to blame.

Today, outside of everything I have read, everything I have heard, and stories from my peers, there is one failure you can never control or truly improve upon for yourself. You can improve yourself, your identity as a person, but the absence of truth is something no identity can manage. Simply put, when someone is lying badly enough, it is hard to discern the truth from fiction. The third party may not even recognize their lie themselves for a number of reasons, including various types of personality disorders. While I started speaking about my biggest failure, and I passionately believe I could have done nothing differently, the underlying truth is what matters most. You can learn to be empathetic, you can strive to be reliable in correspondence, and you can attempt to improve your behavior and happiness through self-help, but you can never undo a lie—or worse, chronic lies. When they are from a peer or colleague, your failure might be completely unjust, and your guilt completely misguided. The facts, control over a situation, and the final results were completely warped by misrepresentation. Your identity suffers, and your reputation may be damaged.

For this simple introduction of my own failure, which is based on the deception of another individual and truly irrelevant to lessons learned, I have absorbed a few basic things. I did the best I could, and I can be honest with myself about it. Others poorly represented themselves and lied in specific situations for reasons I will never comprehend. The lesson I have learned is to educate everyone to be truthful all of the time even if it is not in your best interests. If you are true to yourself and truthful to others, then, even when failure strikes you, you can learn from your mistakes without deflecting blame (maliciously or out of fear of consequences) onto others. As the old cliche states, "honest mistakes" should be forgiven and should become teaching opportunities. If lies are the reason for your failure, and hopefully you yourself never lie, then the lesson learned is to work with people that are truthful, honest, and sincere. Then, as a team, you can learn together and grow stronger together because of your mistakes, and no one teammate will bear the burden of failing alone. That is how teams succeed. Misinformation and lies will always undermine an identity—and a team—but many identity attack vectors can be prevented by working as an honest team who learns from its mistakes (better than any machine identity can) and applies that knowledge to solve identity challenges.

This honest example of a human identity reflecting on experience is something a machine identity still cannot do and questionable if a machine (AI) can honestly ever achieve. It may be able to create the words, but there is no emotion or true experience gained by which human relationships can be formed and interpersonal challenges can be overcome.

This establishes our foundation for *Identity Attack Vectors* into two possible categories of identity that we will be expanding upon throughout this book: human and machine-based (including AI) identities—both of which can be leveraged by threat actors to successfully penetrate an organization. If lies, misinformation, and faux data interact with any environment, the security gaps can lead to your own failure and a breach of the organization. Machines do not demonstrate remorse, and while they get to be the root cause of an attack, no one will ever say the machine lied, was deceptive, and tried to cover an attack. While this is philosophical, it helps us understand what we can correct and what we should be suspicious of when protecting against identity attack vectors.

CHAPTER 3

An Identity Crisis

Is your organization undergoing an identity crisis (or identity security crisis for those who need clarification for the play on words)? Whether you believe it or not, it probably is. While you may not appreciate the simple wordplay to describe the actual problem, you made it to the third sentence of this chapter (only something a human can do), and you know that your identity crisis is not a personal one, but rather describes a problem with how you manage identities and identity security within your organization. If you dare to truly understand this problem, please continue reading the rest of this book. It will take a fair amount of introspection to address this issue, and for some, it will mean facing a deep, dark secret that some businesses would rather deny.

Today, an organization that has an identity crisis has vulnerabilities, process issues, and risks associated with the identity-account relationship within the business. The problems can be deeply rooted in any Identity and Access Management (IAM) discipline, from Privileged Access Management (PAM) all the way through Identity Governance and Administration (IGA), and tooling, like multifactor authentication (MFA) and Single Sign-On (SSO) (these disciplines will be covered in detail later in this book). The risks themselves can be diagnosed by identifying flaws in the joiner, mover, and leaver process and the permissions, rights, privileges, roles, entitlements, authentication, and authorization for identities and accounts at rest and operating during runtime.

What makes this an identity crisis is that the detections and recommendations needed to classify the risks are generally buried deep in the data across multiple IAM disciplines and are not easily available in a Security Information and Event Management (SIEM) or log aggregator. Without a clinical approach to the problem, the symptoms often go unnoticed unless a security professional is specifically threat-hunting for them. Therefore, what is needed is a diagnostic solution that understands identity-based risks and uses policy, rules, and artificial intelligence (AI) to translate the risks into plain English for teams to resolve.

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_3

So, what do identity risks look like? Consider this compact list of detections and recommendations (in a production environment, there could be potentially hundreds more):

- *Recent Activity in a Dormant Account*: The identification of some form of activity is an account that has been stale and inactive for an unacceptable period of time.
- *Activity Found on a Partially Disabled Identity*: The identification of some form of activity on an identity that has been disabled, including the identity record itself or any of its associated accounts.
- *Account Associated with a Personal Email Address*: In the identity-account relationship established within your organization, regardless of whether that is on-premises or in the cloud, a personal email is associated with corporate assets.
- *Account Forwarding All or Most of Its Email to an External Address*: An email account for the business is forwarding email externally, either by a rule, policy, or manually.
- *Dangerously Outdated Browser Used*: An Internet browser that is deemed an elevated risk, either based on the version, vulnerabilities, end-of-life, or lack of support, is being used to surf the Web or authenticate for business applications.
- *An Unusual Number of MFA Failures Occur in Quick Succession*: Having multiple denial-of-service-type MFA attempts occur within a short timeframe is generally associated with an MFA fatigue attack.
- *SSO Admin Login from an Account Without MFA*: As a security best practice, all accounts should use MFA. This is especially true for any administrators of IAM solutions, including SSO.
- *A Privileged Account Was Recently Re-enabled*: Any privileged account that is used for just-in-time access or break-glass access or which has been disabled as part of normal runtime should trigger a security alert, if re-enabled, so activity can be verified as appropriate.

- *Partially Revoked Identity*: An identity or associated account did not properly adhere to an established joiner, mover, and leaver process and has left an identity or its associated accounts on an asset, potentially in an unknown state.
- *High-Privileged Account Not Protected by MFA*: A privileged account, like root or administrator, is only using single-factor authentication and not MFA.

In later chapters, we will explore the Cyber Attack Chain in detail; however, as an introduction, an identity crisis can occur when poor hygiene for identities and accounts can lead to a successful security incident. This is reflective of how vulnerabilities, exploits, zero days, and unpatched assets led to exploitation of resources in years past. Figure 3-1 helps illustrate the initial penetration phase of an identity-based attack.

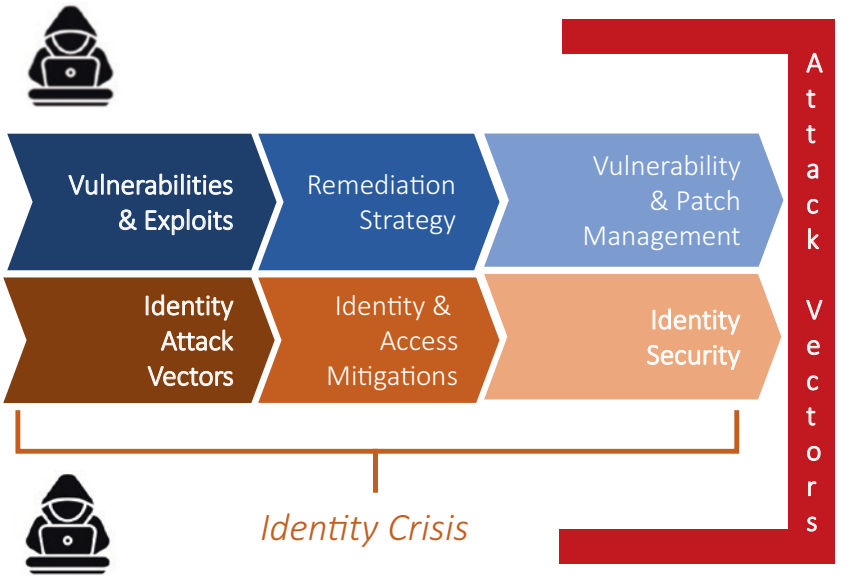


Figure 3-1. *Initial steps for an identity-based attack that creates an identity crisis compared to vulnerabilities and exploit*

For the more experienced security professionals that have read this far, you may be thinking this sounds kind of like ITDR (Identity Threat Detection and Response) to resolve an identity security crisis. Well, it kind of is. The difference is ITDR generally deals with the runtime of accounts and identities based on actual usage and potential abuse. An organization's identity crisis can be solved by looking at the entire lifecycle of identity security at rest and during activity. That is the way to raise awareness and

solve the problem. Looking at identity security based on the implementation within an environment, how all the IAM tools are set up and configured, and how users and machine accounts interact with the setup is key to exposing the flaws. When that analysis is performed, security professionals can overcome their identity crisis within their organization. As I said in the beginning of this chapter, this takes a deep look into the business, and you may not like what you find. If ignored, and cybersecurity training is not enforced, threat actors can certainly use these vulnerabilities as attack vectors in the future. And recent cyberattacks against governments, financial organizations, and the gaming industry have proven that it only takes one compromised identity to potentially compromise the entire business.[1] Therefore, for the purpose of this book, we define identity security as the risk management for all of IAM, including solutions and workflows (fabric), while ITDR is only for the products and solutions themselves.

For those readers looking to explore a sample of real-world identity attack vectors that have led to significant breaches despite regulatory and compliance requirements (covered in detail in Chapter 21), consider these recent events:

- Bangladesh[2] Bank Cyber Heist: Incident Analysis
- Microsoft[3] Actions Following Attack by Nation State Actor Midnight Blizzard
- A 'Worst Nightmare' Cyberattack: The Untold Story of the SolarWinds[4] Hack
- BeyondTrust Discovers Breach of Okta[5] Support Unit

[1] www.darkreading.com/application-security/okta-flaw-involved-mgm-resorts-breach-attackers-claim

[2] https://repository.gatech.edu/server/api/core/bitstreams/df3d1c19-47be-4738-a855-9c049b709d52/content#:~:text=In%20one%20of%20the%20largest,Mazumder%20%26%20Sobhan%2C%202021.

[3] https://msrc.microsoft.com/blog/2024/01/microsoft-actions-following-attack-by-nation-state-actor-midnight-blizzard/

[4] www.npr.org/2021/04/16/985439655/a-worst-nightmare-cyberattack-the-untold-story-of-the-solarwinds-hack

[5] www.beyondtrust.com/blog/entry/okta-support-unit-breach

- 23andMe[6] Claims it Wasn't Breach Despite Stolen Data
- New Hampshire investigating fake Biden robocall[7] meant to discourage voters

[6] https://tech.co/news/23andme-breach-stolen-data

[7] https://apnews.com/article/new-hampshire-primary-biden-ai-deepfake-robocall-f3469ceb6dd613079092287994663db5

CHAPTER 4

Identity As a Business Function

The foundation of cybersecurity defense has been muddied by point solutions, false promises, and "bolt-on" solutions that extend the value of a given technology based on a specific need. After all, if we count how many security solutions we have implemented, all the way from antivirus to firewalls, you typically find dozens of vendors and solutions that have been implemented throughout the organization. The average user or executive is not even aware of most of their cybersecurity technology stack, even though they may interact with them daily—from VPN clients to multifactor authentication.

If we step back and try to group all these solutions at a macro level, we will find each one falls into one of three logical groups, regardless of whether they are on-premises or in the cloud. This is illustrated in Figure 4-1 as a Venn diagram, with identity being one of the core components for role-based access into assets based on privileges.

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_4

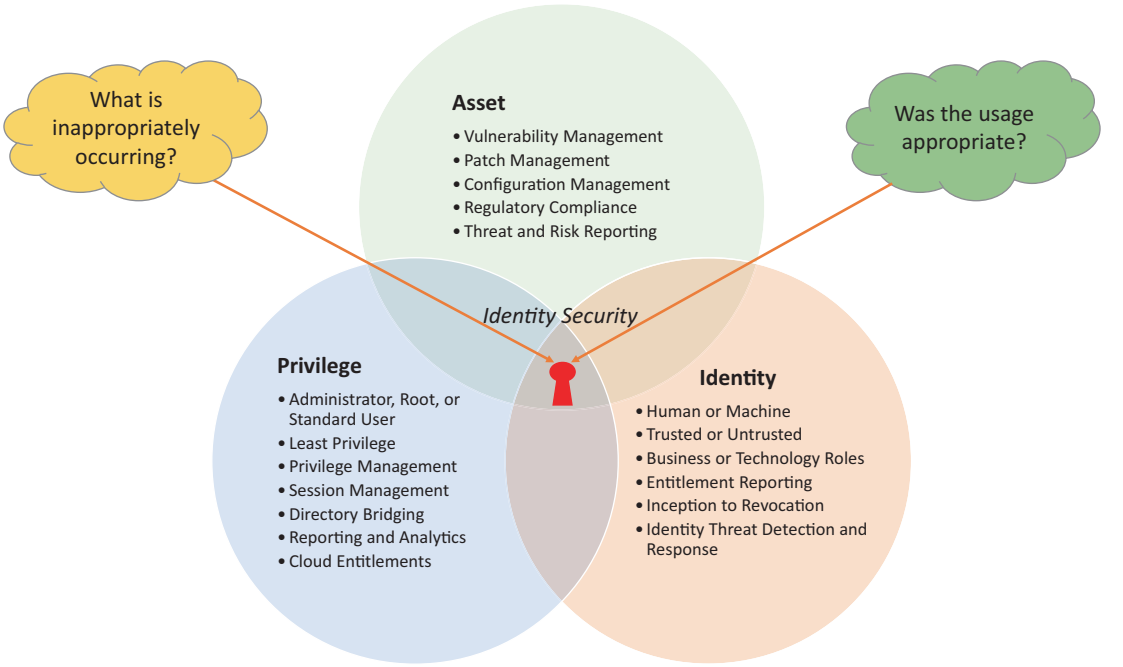


Figure 4-1. *Identity as a Venn diagram with asset and privilege interaction*

These Venn diagram components can be described as

- *Identity*: The protection of a user's identity, account, and credentials from inappropriate access
- *Privilege*: The protection of the rights, privileges, and access control of an identity or account
- *Asset*: The protection of a resource used by an identity, either directly or as a service

While some cybersecurity solutions may be supersets of all overlapping circles, their goal is to unify the information from each in the form of correlation or analytics. Take, for example, a security information and event manager (SIEM). It is designed to imbibe security data from solutions that reside in each group and then correlate the data to inform advanced threat detection and adaptive response. Or consider an endpoint privilege management (EPM) solution that is designed to protect application privileges based on a user on a designated asset. Correlation of common traits across security disciplines enables a more holistic view of an environment based on all three traits. In addition, time and date parameters typically form the foundation (ephemeral or

just-in-time), and an identity accessing an asset with privileges is a rudimentary way of looking at how the circles support the entire cybersecurity foundation of your company. Let us look at a simple correlation:

- Who is the user and what account are they authenticating with (identity)?
- What do they have access to, based on their role (privileges)?
- What did they access (asset and resource)?
- Is that access secured (role-based access)?
- Is that asset secured (configuration and vulnerabilities)?
- Was the access in accordance with the user's responsibilities (governance)?

This answers the question, "What is ***inappropriately*** happening across my environment that I should be concerned about?" Finding the answer to this question is a prime goal of every security team and is the basis for any incident. A robust security program should provide coverage across all three overlapping circles and should identify solutions that provide meaningful data across each boundary (overlap in the Venn diagram) as much as possible.

Having this level of oversight and control helps answer the following questions:

- Are my assets and data secured?
- Are the privileges appropriate?
- Was the access appropriate for the role of the identity?

This leads us to the prioritization of the Seven Layers of Cybersecurity,[1] loosely derived from the OSI Model[2] for network communications that all organizations should appropriately plan for:

1. *Human Security*: Securing the identity/account relationship and the creation, deletion, and modification (joiner, mover, and leaver) process used for authentication and authorization based on identity confidence

[1] https://bilginc.com/en/blog/7-layers-of-cyber-security-you-should-know-5933/
[2] https://aws.amazon.com/what-is/osi-model/

2. *Data Security*: Securing the storage and transmission of data from inappropriate access and exfiltration
3. *Application Security*: Securing software from inappropriate usage, exploitation, and inappropriate installation and manipulation on a network (on-premises, cloud, or remote) and processing information
4. *Endpoint Security*: Protecting all authorized devices, on the network or operating remotely, from inappropriate identity access, misconfigurations, malware, and the exploitation of vulnerabilities
5. *Perimeter Security*: Controlling access to all zones, enclaves, and on-premises networks through secure gateways, bastion hosts, firewalls, joint points, virtual private networks, etc.
6. *Network Security*: Protecting the routing and data flow of information through only appropriate networks and data paths
7. *Physical Security*: Protecting the physical tampering, modification, or installation of unauthorized hardware on corporate assets and infrastructure

Figure 4-2 illustrates this model in a convention that is easy to model after for most organizations. It is important to note that if you search the Internet for the "Seven Layers of Cybersecurity," you will find a wide variety of diagrams customized by vendors to support their solutions and educational sources. There is no right or wrong diagram, but they all consider the "Human" or "Identity" element at the top as potentially the weakest link.

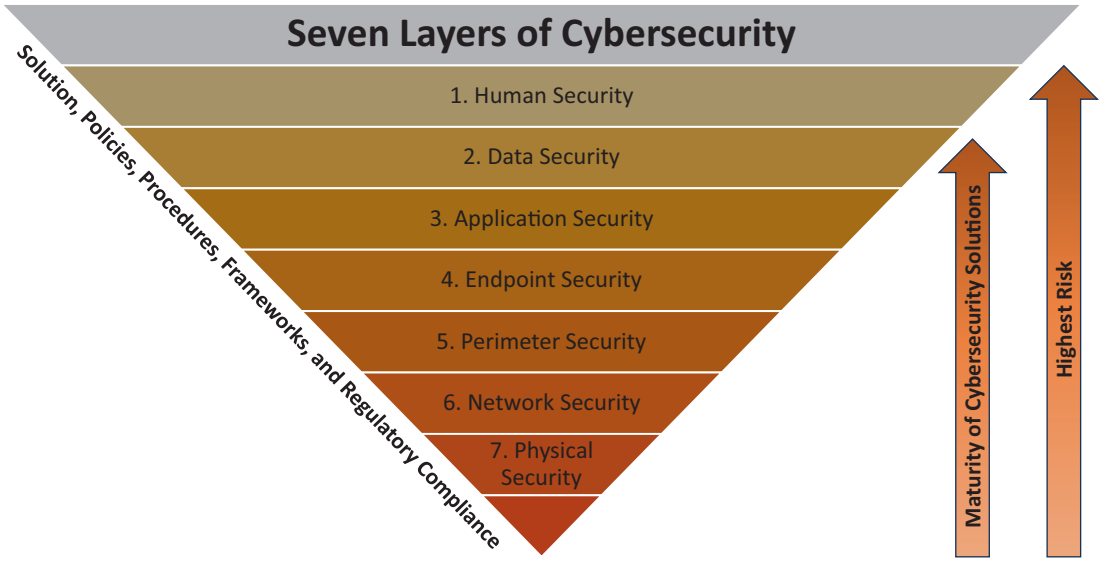


Figure 4-2. *Seven Layers of Cybersecurity*

For most security vendors and their customers, the integration of these three circles and seven concepts is particularly important. If security solutions are isolated, do not share information, or only operate in their own silo (one or two overlapping spheres), their protection capabilities and the data they can report are limited in scope. For example, if an advanced threat protection solution or antivirus technology cannot share user information or report on the context of the identity, then it is like riding a unicycle. The balance of information from the threat is not equally distributed when processing threat information as a log, event, or an alert. You need to have data in all three circles to be truly meaningful when assessing modern threats.

If the unicycle analogy does not resonate with you, imagine not tracking privileged access to sensitive assets. You would never know if an identity were inappropriately accessing sensitive data. Moreover, you would never know if a compromised account were accessing sensitive data and on what asset. That is how threat actors are breaching environments every week—they are actively exploiting situations where we cannot trace indicators of compromise back to these three disciplines.

Therefore, when you look at new security or information technology solutions, ask yourself what circle (or circles) they occupy and how they can support the other circles you trust and rely on every day. If they must operate in a single circle, make sure you understand why and what their relevance will be in the future. To this point, what is an example of a security solution that operates only in one circle? Answer: One that

does not support any integrations nor operate between the three circles. In many new deployments, this may sound like an Internet of Things (IoT) device or a traditional antivirus solution that can report on an infection on an asset, but which has no knowledge of the identity (account or user) or the privileges that the malware tried to use to infect the asset.

To that end, an IoT door lock or camera that provides physical protection for assets based on a static identity that cannot share access logs or integrate with current identity solutions is a bad choice for any organization. A stand-alone antivirus (AV) solution that has no central reporting on status, signature updates, or faults is another poor choice. There is no way of knowing if the AV is operating correctly, whether or not there is a problem, or even if it is doing an exceptionally good job blocking malware. Why would you pick a consumer-grade antivirus solution for your enterprise that does not integrate with your ecosystem or have centralized visibility? Unfortunately, this happens all the time. We end up with the "bolt-on" approach to solving the problem, and when the security tool does alert, it fails to collect the required information to properly mitigate the threat.

As we stabilize our cybersecurity best practices and focus on basic security hygiene, consider the long-term goals of your business. If you choose a vendor that does not operate across these three overlapping circles, has no integration strategy, or is an odd point solution, be aware of the risks.

Everything we choose as a security solution should fall in the center of the Venn diagram; if they do not, then ask a lot of questions. For example, why would you choose a particular camera system without centralized management capabilities? If it falls into the asset protection circle, can monitor physical access by an identity, but doesn't offer centralized capabilities and management, it is a stand-alone silo that is not supporting your foundation. It needs to support all three circles to be an effective security solution that will, ultimately, provide valuable information for correlation, analytics, and adaptive response.

Some may argue that there could be four (or even five) circles for a sound cybersecurity defense, adding a circle for the education, partners, etc., you may need to support your foundation. We prefer to think of all tools, resources, and solutions as falling within these three categories. Why? A three-legged stool never wobbles! And each of these has documented attack vectors that can be managed as integrated solutions. Those are the basis for my other books, *Privileged Attack Vectors*, *Asset Attack Vectors*, and *Cloud Attack Vectors*.

While it is no secret that identifying and correcting network security holes is critical to protecting any business from harmful attacks, the processes of Privileged Access Management, vulnerability assessment, and configuration management often get overlooked. They each, however, represent critical components for sound security practices affecting assets. This is basic cybersecurity hygiene.

Vulnerability management should be an ongoing process, but too often organizations become lazy or complacent when maintaining a proper vulnerability workflow and only react when disaster strikes, and they are forced to inspect the process in detail. Even then, some businesses fail to learn the lesson of proactive vulnerability assessment and remediation and fall behind in managing all three circles. You cannot protect an identity when the asset itself can be exploited due to poor configurations or missing security patches.

Finally, many organizations look at vulnerability management in isolation. Take a step back and look at the wealth of asset and risk information that is captured in a vulnerability scan. This spans everything from vulnerabilities to accounts to collecting detailed information for asset and identity management. Examine how this data can not only help prioritize patches and mobilize IT resources, but also be applied to strengthen other security investments across the organization, including asset management, patch management, application control, analytics, and threat detection (to name a few based on the raw diversity of the data itself). This information can even help strengthen your identity posture by locating the presence of appropriate and inappropriate (rogue) accounts and groups across your organization. It is yet another tool that helps you with the challenges and strategies outlined further in this book for managing identity attack vectors.

CHAPTER 5

Identity Access Defined

There are a myriad of frameworks to help you define, organize, implement, and improve security. Initiatives like the Control Objectives for Information and Related Technology (COBIT[1]), the US National Institute of Standards and Technology (NIST[2]) Cyber Security Framework, and the International Standards Organization (ISO[3]) 27K provide a framework to guide security program thinking. They are considered frameworks because they provide extensive guidance on everything from funding to security incident response readiness.

Identity management is part of most, if not all, of the "official" security frameworks, often as a passing topic, but rarely as the focus for security. To simplify things for this publication, rather than choosing a single framework and pulling out identity, let us focus on a concept called "The Five A's of Enterprise IAM." Figure 5-1 shows an overview of this approach. It offers a basic definition and scope for how to envision identity management as a universal set of principles that apply to all frameworks and just about all enterprise scenarios. The five A's cover authentication, authorization, administration, audit, and analytics. These are explained in detail in the following sections and form a coherent model for how identities should function in a business. Once you gain a mastery of the five A's, you will be ready to deliver identity management operational controls in just about any scenario for virtually any type of organization or vertical. If done correctly, this will provide an effective approach to mitigating identity attack vectors.

[1] www.techtarget.com/searchsecurity/definition/COBIT#:~:text=COBIT%20is%20the%20acronym%20for,business%20risks%20and%20control%20requirements.

[2] www.nist.gov/

[3] www.iso.org/home.html

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_5

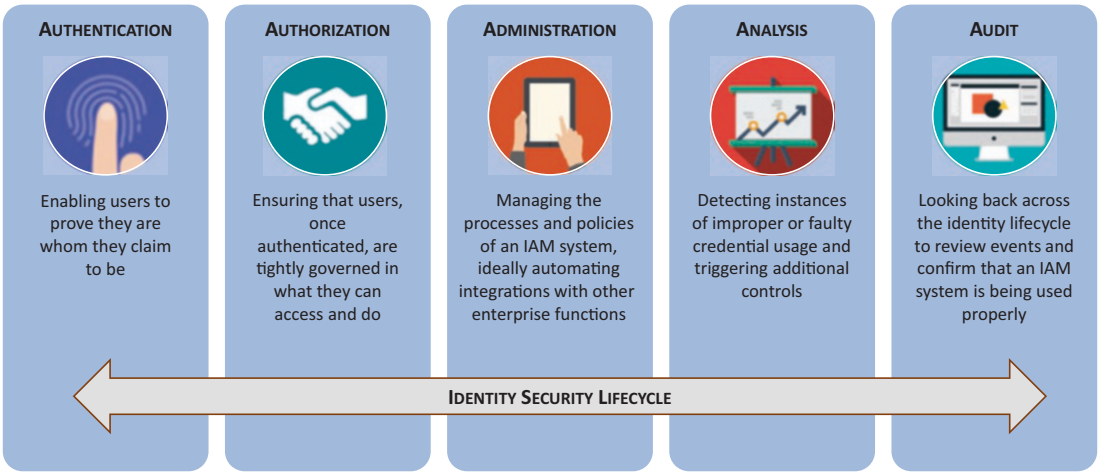


Figure 5-1. *The five A's of identity management*

Authentication

Authentication is the process of verifying the identity of a user, system, or device to ensure that they are who they claim to be before granting access to resources or information. It is a crucial component of cybersecurity measures to protect against unauthorized access and enhance overall security. Authentication typically involves the use of credentials, digital certificates, biometric data (fingerprint, facial scan, etc.), or multifactor authentication. The goal is to establish a reliable and secure means of confirming an identity with confidence for the entity attempting to access a system or data. Unfortunately, authentication is often confused with authorization, even though they are distinct from each other. In some computing models, authentication and authorization have been blended and have little distinction. Apple iOS, for example, uses biometrics for both authorization and authentication, and the end-user experience is blurred regardless of the validation type. By definition, authentication is a login (username) in addition to some form of secret, typically a password, to provide your identity. It is essentially a validation of who you are.

Authentication of your identity = username + secret (password) + multifactor (optional, but a security best practice)

While there are countless variations of secrets that can be used within a login, such as pin codes, passwords, keys, two-factor authentication, multifactor authentication, etc., the login itself is generally not a secret and is often guessable for an identity. For example, my login could be "jtitor" as an alias and abbreviation of my name. This is associated to my identity via an account. However, a login could also be something more complex like an employee number, which better obfuscates a user's identity. For highly secure environments, this second approach is preferable, especially for administrator or root accounts. Additionally, you should not be able to visually identify the privileges by looking at the account or username. Environments commonly make this mistake by creating accounts with "-a" or "-admin" suffixes or prefixes.

In simple terms, authentication is nothing more than proving your identity credentials with confidence. It does not provide permissions, privileges, entitlements, or even a role. It just provides confirmation that you are who you say you are, and the more attributes you can link using techniques from tools with multifactor authentication, the higher the identity confidence for authentication.

Authorization

Authorization is the next step after authentication. You cannot be authorized to perform a function, have privileges assigned, or even perform tasks in a role without some form of authentication. Even if you are a guest in an application or operating system and have no login and/or password, your authentication is assumed to be a guest, and you are granted all the rights, permissions, and privileges of a guest. The login username and password secret are just nulls in this scenario, but you have been authenticated and have some form of authorization, albeit truly little.

Authorization = privileges (what you are allowed to do) + Authentication

Authorization is, therefore, the right to perform a function based on your authentication credentials. Your identity and its associated account are granted privileges to perform specific functions and may be explicitly denied or not privileged to perform other functions. These privileges can be assigned within an application, operating system, Active Directory, Entra ID (formerly Azure AD), or an identity governance solution. For management at scale, the latter is always recommended in some form or formal identity services solution by using an enterprise IGA (Identity Governance and Administration) product.

When privileges are grouped together, they create the foundation for a role. When a role is assigned to a group of accounts, the role is authorized to perform assigned privilege tasks. As a security best practice, privileges and roles should be separate, but some modern implementations have blurred this concept.

In the case of a mobile Apple device, such as an iPhone with iOS, facial recognition and fingerprint touch are both used for authentication and authorization. They will log you into the device (authentication), and they can also make payments or purchase products (authorization) using the same mechanisms.

In the world of cybersecurity, using the same technique for both authentication and authorization is a bad idea because the controls around both are the same. Authorization and authentication should be discrete techniques. When authorization and authentication are blended together, a breach in one model leads to a breach in the other.

While Apple has designed high levels of security into their solutions to minimize the risk, in general, most computing environments should avoid using the same technology for authentication and authorization. Just because you have been authenticated does not mean you should have authorization. Those privileges and permissions should ultimately be decided upon using another method and should be appropriately assigned as entitlements.

In modern computing environments, Single Sign-On (SSO) solutions are notorious for blurring this line between initial authentication and automatic authorization within managed applications. MFA solutions can help mitigate this risk by requiring a second validation for privileged authorization activity and appropriate authorization into an application, since authentication into the SSO solution itself was performed separately.

Administration

Search the Internet and you will find 100 different definitions for administration. In the context of this book, we are specifically talking about the administration of authentication, authorization, and audit controls. Administration here means configuration management and governance controls over any changes made to authentication, authorization, and audit. When managing identities, this is generally referred to as Identity Governance and Administration (IGA).

Today, authentication and authorization mechanisms are inherently distributed, painfully diverse, and perpetually changing. After 20+ years of watching this space develop, most "old-timers" in the IAM space are usually quite retrospective about how much things have changed, yet how much they are still the same. Most organizations still struggle to get full administrative control over the systems and data they are responsible for protecting. In our view, it is essential that administration is treated independently from the ever-changing mix of AuthN (**authen**tication) and AuthZ (**auth**oriz**z**ation) technologies. These are abbreviations that most seasoned identity management professionals tend to reference for no more basic reason than simplicity and are the primary function for management in IGA solutions.

IAM administration is a big part of the Identity Governance model. Add to it audit and analytics (following sections) and you have the product scope of most enterprise-class IGA solutions. With a dedicated heterogeneous management approach, IGA strives to take on the mantle of global administration, audit, and analytics for users and data access. By leveraging the Identity Governance processes, administration provides visibility, controls, and automation for full lifecycle management of all users and their access.

Audit

Facilitating the delivery of repeatable and sustainable user access, the audit process is the responsibility of Identity Governance and monitoring the responsibility of the identity security solutions. Whether it's overlaying specific controls to meet regulatory compliance or creating system-driven lifecycle management functions that are stable and verifiable as control proofs, audit plays a significant part to ensure appropriate authentication models exist for an identity and that there are no security flaws in the access, configuration, usage, or behavior.

For some, the audit of identities means delivering a user access certification program. For others, it defines and implements preventive and detective policies, such as separation of duty (SoD). And finally for others, it is identity security. Identity security is the detection and risk evaluation of identity data based on its static configuration coupled with its current and historical usage patterns. For an organization, it is being able to prove that there are comprehensive administration processes and policies being adhered to for identity governance and proving that any behavior was appropriate.

Analytics

The fifth and final A is the comprehensive analysis of how identities operate during normal business runtime. Analytics means gaining operational and security insights through the ongoing collection and processing of identity-related configuration, assignment, logging, and usage patterns.

Advanced identity security, using analytics, enables a more informed and predictive approach to governance. Using machine learning (ML) and artificial intelligence (AI) techniques, identity analytics tools can provide essential historical context and peer-group analysis information that helps to extend identity audit and administration functions to make them more dynamic and responsive. For example, if an analytics engine discovers suspicious, inappropriate, or unusual access, it can prompt administrators to review access and create management actions to ensure the correct configuration has been implemented or that an indicator of compromise has been tripped. Analytics can provide auto-generated insights and recommendations that allow the line of business to make more informed access decisions that enhance operational identity security and ensure compliance. With the advancements in ML and AI, data can be discovered that is unique to a particular environment and which can provide actionable guidance far beyond what a traditional rules engine can recommend.

In our experience, these five A's, are the cornerstones for every directive, best practice, law, framework, and compliance mandate for identities that has ever been written. If you think of identity as a business function, you can easily see how all the functions of identity interact using the five A's concept.

CHAPTER 6

Understanding Enterprise Identity

The French philosopher Rene Descartes is credited for stating "Cogito Ergo Sum." In English, it translates into "I think, therefore I am." This statement has spawned countless hours of Philosophy 101 classes around discussions on the meaning of life and the existence of a human soul. One thing is certain from this quote; "I think, therefore I am" can also mean you have a human identity.

Barring any philosophical discussions on the meaning of life (a quick shout-out to Monty Python and the machine that goes "bing") and the existence of a human soul, an identity is typically a one-to-one relationship between a human being—a carbon-based life form (one *Star Trek* reference to lighten the topic)—and their digital presence. We will skip machine identities until later in this chapter. A human identity's digital presence, however, can have multiple accounts, credentials, and infinite variations to represent a single identity, depending on how it is represented in an electronic format and the account's role. For example, should an account that is associated with your identity be your first initial and last name or obfuscated by using some pattern of letters and numbers? As mentioned, if my account name (username) is "jtitor," it is easy to link back to a real identity. If it is "THX1138," it has no meaning to anyone outside of the user. For good Identity and Access Management (IAM), this identity/account mapping is critical and must be reliable for any combination your organization chooses, including shared accounts. And if an identity has multiple accounts for specific applications or privileged access, using the person's alias based on their real name can create unnecessary risk, despite the simplicity of determining any existing mappings. In an enterprise, changing an existing paradigm is near impossible, but nonetheless, it is a risk that must be considered for identity security.

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_6

In some environments, an identity may deviate from this human definition and not be a living and breathing organism. An identity can represent a defined resource, asset, or even a physical or software-based robot. This extended definition relies on the fact that complete systems can take on a life of their own, and thus, can be assigned an identity—even though they do not "think" nor breathe nor know a truth from a lie.

While this extended definition for identities becomes more commonplace and shares a different definition for cloud-based identity models, it does represent the direction of technology and the personification of technology we have created. This advanced form of identity should not be confused with accounts or credentials used for service accounts nor for administrator access to a workflow or system. To better explain this, think of it like a robot or android that is possibly using some form of generative AI (artificial intelligence). Perhaps your office has invested in mail robots or other office automation technology. The technology may (or may not) look human. It could even resemble a creepier version of R2-D2 from *Star Wars*.

The assets have an automated mission and some form of reference designator. Therefore, R2-D2 and BB-8 are valid identities, even though they are not flesh and blood. The accounts and credentials they use to operate are linked to their identities and have a human-based owner. If you try to understand the risk, consider how R2-D2 authenticated to the Death Star computer to find Princess Leia without valid credentials. That should have been privileged information, and he may have had access to a stolen account to perform the query. Therefore, was it a plot flaw in the movie or did R2-D2 really perform an identity-based attack? Remember, that was the early 1970s.

Today, real-world discussions of identities are translatable to the masses, and the future cannot ignore that identities assigned to a machine (or resource) are also a potential attack vector. For the purposes of this book, identities can mean either a person (always a one-to-one relationship) or a technology implementation with the potential to be a one-to-many relationship and have a human owner. "I think, therefore I am" is one identity, and a machine (mail robot), even if there are dozens within a building, should have only one identity per machine. Unfortunately, in practical implementations, organizations may find this is not true. As a security best practice, individual machine identities should never be shared, even though their accounts and credentials may need to be shared. Shared credentials are a security taboo as well, but they may be needed for a technology implementation to succeed due to limitations in the technology itself. Thus, machine implementations of identities may have undesirable one-to-many relationships, too.

This leads us to the evolution of identities as an attack vector beyond assets, people, and machines, as well as the privileges being leveraged as attack vectors within any identity.

Personas

A persona is a derivative of an identity. When working with indicators of compromise (IoC), we always try to map the anomaly back to an account, credentials, and an identity. Then we can focus on the root cause and potentially build effective defenses and policies.

What makes a persona different than an identity is a higher level of classification. For example, a generic persona could be classified as an information technology administrator, forensics engineer, or even a compliance auditor. While most organizations do classify internal employees by roles (more on roles at the end of this chapter), potentially by using a job description, vendors and suppliers think of roles as buying personas. Vendors will target personas within an organization to sell a product. For example, a forensics tool is generally targeted at the teams that use the tool for detailed security analysis, while an accounting solution would be targeted at the Chief Financial Officer (CFO) or sales operations manager, based on their role. When managing identities, security professionals think of roles for grouping identities and accounts, while vendors who are marketing and selling solutions think of buying personas.

For an identity as an attack vector, threat actors will target a persona. This could involve targeting teams responsible for accounting or accounts payable with a malicious phishing email for a fraudulent wire transfer. It could involve targeting human resources with a malicious file or fake resume that leverages a vulnerability to exploit the system. Therefore, a persona refers to a logical grouping of people at a high level, without considering their privileges and resources. A persona is just the job description without the technical controls applied to identities and accounts within an organization.

When considering an attack, a persona can be viewed as the groupings of identities targeted for an attack at the highest generic level of their job description. That can include anything from sales to executives to any potential group your organization may have. Identity attacks will typically focus on one group in a targeted act, such as with broad spear phishing campaigns to individuals in that group. This is why personas are so important in relation to identities and the technical controls that are applied via roles. Personas are used for everything from legitimate sales cycles that help vendors and marketing find the right buyers to, targeted attacks that leverage the responsibilities of the persona to perpetrate an attack.

Physical Persona

A physical persona is essentially your job. In the real world, there are traits that make it your persona. For example, these could be the uniforms we wear and the types and colors of badges we use for authorization. Physical identity threats are commonly displayed in movies when actors wear uniforms to breach an environment and impersonate a persona to gain access. If you ever have had a physical pen test performed on your environment, this is one potential attack vector. A janitor's uniform, or even wearing a suit, might be sufficient to circumvent physical defenses and breach an environment.

Electronic Persona

An electronic persona is the digital translation of your physical persona. It is conceptually a superset of a role and differs only in hierarchy. An electronic persona is the many-to-one translation of identities into a digital form, while a role is the high-level grouping of the persona at a security control level.

For example, consider a fictitious employee named Nurse Chapel. She works in a hospital and has the appropriate clothing for her job as a lab technician (physical persona). Her position is a medical lab technician for processing bloodwork, but her role may be restricted to a grouping of technicians for a certain floor, types of tests, or even data that the technician may be able to access. Her electronic persona explains her job (description), but the role is the association of applicable functions she is entitled to, along with her peers (electronically). In many cases, her electronic role may have entitlements unique to her and not assigned to anyone else. Common entitlements should be associated by role. Therefore, Nurse Chapel's electronic persona is a lab technician (job title), and her role provides common work functions among her peers.

This creates an interesting risk as an identity attack vector. The threats against roles with unique entitlements are different from a group of individuals with similar personas defined electronically (i.e., lab access). As an example, if only one person is the database administrator, their role is the only one susceptible to a privileged attack vector. The database and server are susceptible to an asset attack vector, and their electronic persona is simply a database administrator.

If a larger group of people are granted database administrative rights, they all would be susceptible to an attack based on a database administrator role. Performing an analysis of entitlements (certification) is much easier for groups of people vs. individuals. However, assigning broad rights to everyone introduces excessive risk, is undesirable, and stands against the best practice for least privilege assignments. Electronic personas must be defined before any subsets encompassing roles are established, so the high-level definition can be implemented within technology.

Organizations may choose to use job titles to establish this foundation, but keep in mind, not everyone with the title "engineer," for example, should be treated the same, even though that is a part of their electronic persona. If anything, all engineers may have the same laptop image with installed tools, but not the same entitlements based on their department and responsibilities.

To better understand the difference between electronic and physical personas, consider Table 6-1 for a comparison using a criminal perspective. The table explains how physical crimes translate to electronic crimes and why mitigation steps are needed to prevent them.

Table 6-1. *Physical to electronic crime comparison*

Physical Persona Crimes	Electronic Persona Crimes
Burglary: Physical breaking and entering and the theft of material goods	*Hacking*: Electronic intrusion using unauthorized access to steal or alter information
Deceptive, Scams, or Prank Callers: Audible crimes using telephones and cell phones targeting end users directly or organizations like banks	*Phishing, vPhishing, or SMSishing*: Electronic crimes using emails, voice mail, or SMS text messages to socially engineer the end user to expose credentials, passwords, or other sensitive information
Extortion: Illegal use of force and position to obtain funds, property, or other material objects of value	*Data Extortion*: Hacking sensitive information and threatening the release of data unless funds are paid (such as in a ransomware attack). This is typically in the form of bitcoins
Fraud: Dishonest activities used to deceive targeted organizations or people for financial or other monetary gain	*Web-Based Fraud*: A broad category that encompasses websites and other Internet communications used to defraud an individual out of sensitive information or credentials to gain access. This category is typically linked with phishing and other online chat-based attacks

(*continued*)

Table 6-1. (*continued*)

Physical Persona Crimes	Electronic Persona Crimes
Identity Theft: The physical impersonation of another individual to gain access or defraud an individual or organization through false documentation and appearance	*Identity Theft*: The electronic theft of an identity using accounts to impersonate a user and steal information or defraud them of money
Child Exploitation: Criminal victimization of minors for indecent motives	*Child Exploitation*: The electronic distribution of images and information relating to victims as minors

Identity security solutions can help implement safeguards in an electronic world by managing identities, accounts, and privileges and measuring their risks at rest and during operations.

Finally, here's one last note on personas. In many cybersecurity circles, the concept of personas is ignored, and everything is translated directly to roles. While you may agree with the simplification of this approach, a job description is different than the technology implementation to assign privileges. That is why the distinction between roles and personas is so important when building an identity security strategy within your organization. What the person does and how you apply that to specific technology is different per solution and per vendor.

Accounts

An account is an electronic representation of an identity and can have a one-to-many relationship with the identity. A single identity can have multiple (many) accounts. These accounts reference a set of permissions and privileges needed for an application or asset to connect or operate within the confines of a resource. While the definition of an account is obvious for an identity, it can take on a variety of forms when used electronically for services, impersonations, and application-to-application functions.

Accounts can have a complex relationship with identities and can be defined locally, grouped together, nested in groups, or managed via directory services. Accounts can have role-based access applied at the account level, group level, and within a directory,

and these can range from disabled (denied access) to privileged accounts such as root, local administrator, or domain admin. The level of privileges and role-based access is dependent on the security model of the system implementing them and can vary greatly from one implementation to another.

Accounts linked to identities are how we gain access to information technology resources. For technology itself, accounts are a vehicle for authorizing their usage and operational parameters. Excessively provisioned privileges given to any type of account will increase cyber risk and are the basis for a privileged attack vector.

When we consider accounts assigned to machine identities, we should always associate them with a human identity owner. That is, someone responsible for the care and feeding, maintenance, and lifecycle of the machine identity (non-silicon life form like a Horta[1]). This will be discussed later in greater detail.

Password

A password is a secret word or expression used by authorized persons to prove their identity and the right to access information, systems, applications, etc. When used alone with just a username, it is considered single-factor authentication. A password as an expression is a word or other string of characters that uses any localized characters (and even emojis, if supported by the system). It can vary in length, readability, and complexity to maintain confidentiality and secrecy from potential abuse.

Passcode

A passcode is similar to a password, but with one major difference. In general, a passcode does not have an associated username, like "jtitor." A passcode can be used with the same devices as a password, if supported, and provides authentication regardless of the identity or account. The most common passcodes are associated with systems like Wi-Fi, where a passcode is needed to join a wireless network.

[1] www.startrek.com/news/50-years-later-the-devil-is-in-the-details

Passkey

A passkey is an alternative method of account authentication that eliminates the need for usernames and passwords (credentials). In lieu of relying on single-factor methods for authentication that are susceptible to phishing attacks, leaked secrets, keyloggers, and other security flaws, applications (websites and on-premises) can use passkeys to verify a user's identity. Passkeys are only stored on the user's device, so there is no password to be intercepted by threat actors. By definition, a passkey is a digital credential tied to a user's account and a specific website or application. Passkeys allow users to authenticate without having to repeatedly enter credentials, since the system and identity are trusted after the first authentication. Passkeys are a secure alternative to passwords and allow authentication by digital certificate proxy with your fingerprint, face scan, or device screen lock, like a PIN. Passkeys provide the strongest protection against threats like phishing since the trust relationship is stored with the asset and is typically brokered by biometrics or a very complex password that rotates frequently to prevent attacks. Hardware keys (USB, Smart Card, etc.) associated with standards like FIDO2 use passkeys for strong authentication following these methods.

Certificate

A digital certificate is a file or electronic password (or passkey) that proves the authenticity of an asset, application, or identity through the use of cryptography and the public key infrastructure (PKI). The deployment and management of digital certificates helps organizations ensure that only trusted devices and accounts can connect to their networks, applications, and associated resources. For example, if a browser and machine requests contents from a website or API, how do you trust what is returned is genuine, correct, and potentially not malicious? Digital certificates provide the stamp of authenticity by binding the public key (cryptography) with the entity (server or client) that owns it, provided the entity possesses the corresponding private key. Digital certificates are issued by a Certificate Authority (CA), which is typically a company providing these keys as a trusted service. A digital certificate contains the name (company, person, or government entity) of the certificate holder, a serial number, expiration dates, a copy of the certificate holder's public key (used for encrypting messages and digital signatures), and the digital signature of the certificate-issuing authority (company) so that a recipient can verify that the certificate is authentic and has not been spoofed or compromised.

Credentials

A credential is an account with an associated password, passcode, passkey, certificate, or other types of key linked with a username. Credentials can have more than one security mechanism assigned for dual or multifactor authentication or can be an anonymous account for anyone to access. In the case of an anonymous account, the credentials use a null password. This is different than a guest account that can still be associated with a secret for authentication.

Credentials are just a mere representation of the account password combination needed for authentication. They are, nonetheless, the crown jewels for any threat actor to begin an escalation of privileges and a successful identity attack.

When someone indicates that they have "hacked" an account, what they mean is that they have hacked the credentials (username and secret or hash or token) associated with the account. Hacking an account would only yield a username. Both the username and password are needed to successfully compromise a credential and potentially anything it protects.

For simplicity's sake, in the remainder of this book, hacking an account means the same thing as hacking credentials. It is difficult enough to manage privileges in an environment without having to parse the semantics used every day in describing the threat. However, security professionals and the press will probably never change in saying one million accounts were compromised when in fact one million credentials were compromised. See the difference? The account was compromised because the credentials were really breached.

Users

As we have defined, identities are typically associated one-to-one with users (people and their electronic designation with a username). Users can have multiple accounts, credentials, and even personas, but they have only one identity.

If a threat actor can insert themselves into that one-to-many identity account relationship, then we could have an incident. Therefore, the most effective strategy is to protect the relationship and eliminate single points of failure for any user. This means we need a method to link identities to the actual user (human) and all their associated accounts. The linkage for an identity should be a designator with little to no value

(possibly even a GUID[2]) and should not be linked to any other identification system, public or private. For example, an employee number that is unique to the company is a good designator (if properly constructed), but a Social Security number or passport number is not. Therefore, for a successful implementation of this relationship, consider the following:

- Identity designators for a person should never be used for authorization or authentication.
- Identity designators are only for reference and can be alphanumeric, symbolic, electronic, or biometric in representation.
- Storage of the designators should be protected, even if they are public. They are still PII (personally identifiable information) since they can be linked back to an identity and all their accounts.

While a single reference may be an annoyance if compromised, a database of designators can reveal more about the identities and personas and should always be robustly safeguarded. Most modern breaches reveal this information in the simplest form, like an email address. That is, all accounts associated to an identity are linked together by the person's business email address.

In most organizations today, the most common use of an identity's (user's) implementation in a directory service of accounts is in the form of a Security Identifier (SID). In the context of a Microsoft Windows operating system, an SID is a unique, absolute identifier of an identity-to-account reference, user group, and credentials. It's different from a company's employee number, but it does have a relationship back through an account to the same identity.

An account should have a single SID for life (in each Active Directory domain), and all properties of the identity, including its name, account, attributes, etc., are associated with the SID. This is even true for just-in-time access, but the SID is only "alive" for the authorized task or mission in these scenarios. This design allows a person to be renamed (e.g., from "John Titor" to "Morey Haber") without affecting the security attributes of objects that refer to the identity. And the SID alone is not enough to compromise the integrity of the relationship without some other additional factor, like a secret.

[2] https://guid.one/guid

It's important to note that not all technology implementations use the concept of an identity to implement their functions or services. For example, an IoT camera may just have one account that operates as an administrator. The relationship between the user and camera skips the concept of an identity and role and maps straight from the requestor to the camera's administrator account. Not everyone should be a camera administrator. In addition, other employees may know those credentials, making it a one-to-many relationship for any technology implementation. The camera has no concept of an identity unless it has technology implemented for role-based access control (RBAC) that allows it to link back to multiple accounts.

This takes the concept of identity to the highest level: a person (people). The mappings to an employee number, accounts, and SID are all a part of their identity as an electronic persona.

Applications

To be fair, the concept of applications having unique identities is controversial among the information technology community. A payroll system, web application, or even an engineering database should not have an identity. The applications will have various accounts that, in turn, will have owners in order to operate correctly—but an application should not have an identity (as previously defined).

However, this is where an emerging gray area begins. What about a bot? It is an application based on software. It impersonates an identity and has a persona. It may be backed by artificial intelligence (AI) to provide automated responses and may be linked to a true person's identity as an owner. Should it have an identity? There is no simple, universally agreed-upon answer, but in this author's opinion, such an application should have an identity since it is mimicking human traits, and thus, can be classified as another electronic persona.

As an example of this type of outlier, Figure 6-1 illustrates a bot commonly found on the Internet interacting with people. Is it really a person, a bot using artificial intelligence, or just an application responding to you based on a preset list of rules and responses?

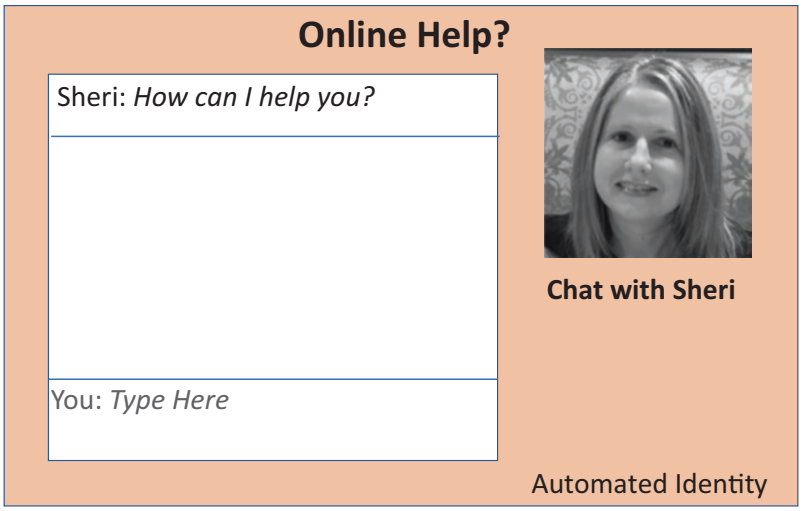


Figure 6-1. *Automated chatbot*

If we consider that this type of application has an identity, its implementation should be governed by the following guidance (in addition to Isaac Asimov's three laws of robotics[3]):

- Application identities should be assigned when the software emulates human behavior, operates as a physical robot or android (machines are covered in the next section), or can interoperate with a person on its own accord.
- All application identities should have human owner(s) assigned since they are not truly sentient beings. Care, feeding, maintenance, and its full operation lifecycle should have a human owner that is responsible.
- Application identities should follow all the same criteria as human identities, except:
  - Their personas should be strictly limited to only the tasks they can perform. For example, there is no reason for a bot to have access to the same systems a human being does, such as local login rights, or access to a location with sensitive equipment. Their privileges must be assigned separately and as appropriate.

[3] www.scientificamerican.com/article/asimovs-laws-wont-stop-robots-from-harming-humans-so-weve-developed-a-better-solution/

- Application logs should be strictly monitored for any privileged access occurring outside of their designated tasks. This activity would represent an indicator of compromise and a valid attack vector based on machine-based identities.
- The relationship of an application to its identity should be one-to-one at the lowest level possible. For example, each fictitious bot used in a chat conversation may not have a unique identity, even though the name and picture are random to the end user. They appear to be separate people. The instantiation of the bots owns the identity in lieu of each chat instance. If it is possible, each chat instance should have an identity since it operates and emulates a person independently. However, this may not be technically possible based on your implementation and underlying technology. Thousands of active chat sessions should not always equate to thousands of temporary identities. Just consider the assignment of an identity at the lowest level possible. This will help digital forensics investigations should they ever be needed. For example, which instance of a chat session is responsible for the investigation using an instance or other designators?

For your environment, assigning identities to applications is truly a business and technical decision. As a best-practice recommendation, consider why you would need to have an identity and when it is appropriate. A good rule of thumb is if it emulates a human, looks like a human during operation, or moves closer to SkyNet (shout-out to the Terminator), it needs an identity of its own. And try to keep the relationship one-to-one. Any forensics or deep dives looking for indicators of compromise will depend on their uniqueness. And for the sake of an identity, make sure one of its attributes is an owner that can map it back to a human identity. We will cover this in more detail in the next sections.

Machines

In a modern office, you may see robots delivering mail or an IoT device brewing coffee or controlling the office thermostat. Machines have a combination of hardware and software that may require an identity to govern their operations, ownership, and autonomy. Deciding whether an identity should be assigned to a machine is more complicated than just a decision of an identity for an application. Our chatbot example illustrates this and makes machine-based identity decisions even trickier based on their real-world interactions.

Consider a smart coffee maker. The word "smart" implies it is network-enabled and an IoT asset. Should it have an identity? Well, that all depends on the coffee maker. An IoT model can brew coffee and notify you if coffee grounds need to be added or if waste needs to be removed or if there is another issue. It may act more like a robot and, therefore, should have an identity. Conversely, if the coffee maker just sets a timer and sends alerts, even though it is an IoT machine, does it really need an identity? Probably not. But if it can be attacked, and its operating system can be owned via an asset attack vector, or if its administrator account can be compromised via a privileged attack vector, it is a risk. Personifying an identity for it becomes a business choice, regardless of whether or not it authenticates on the network to perform its tasks. However, if it does authenticate, it needs an account, and that account should have an identity-based owner. That, in and of itself, should help you make the right decision.

Let us consider a more detailed example by exploring our physical mail robot again. It travels floor to floor to deliver old-school, paper-based mail and packages. For all purposes, it is an autonomous robot, has ownership assigned to a team, and has an identity. This mail robot behaves like a persona with a specific set of tasks: get and deliver mail. This is much like a summer job for many of us "old timers" that actually received mail and went to the office.

In our scenario, the mail robot is an asset of the company, can have physical and software vulnerabilities, and has privileges to perform its job. These privileges could include access to the mail room and to specialized elevators to travel between floors. However, for one reason or another, it does not have privileges to the same elevators and hallways that are used by the human workforce. Sound correct? We have established this asset does have an identity based on its traits. What could its identity attack vectors be?

- Misrouting of mail to threat actors who compromise any type of vulnerability, from asset to identity.
- Physical harm is based on inappropriate movement or physical damage or coercion.
- Inappropriate access by a threat actor to the robot's sensors, including cameras and doors.
- Inappropriate access to the device's logs to determine behavior patterns or alter records.
- Transport illegal, inappropriate, or stolen material in and out, or through, an office.

Any of these attack vectors potentially could be devastating. This is why the mail robot needs an identity. It can be tricked into behaving like a threat actor (insider or external threat) and can be leveraged against the business in ways beyond traditional lateral movement. It can be compromised in ways that impact physical and electronic traits in similar ways that it would impact a human with the same persona.

The preceding scenario leads us to an interesting conclusion. Machines should be assigned identities when their risks can mimic a human persona or role—electronic or physical. These can be quantified in a risk analysis beyond just a compromise or data breach. Identities should not be assigned when the risks are strictly electronic and attack vectors are similar to any other network device or application. The decision between assigning them an identity or not is gray, but an IoT coffee maker is just that, a coffee maker. There is no reason to complicate the situation with over-assignments. However, if something mimics human behavior, it should have an identity. R2-D2 surely did have an identity and, as you will continue to see, an owner, too, and it did have real-world physical interactions. R2-D2 was also a threat actor by harboring the Death Star plans.

Ownership

Every identity requires an owner. If you are a human being, you are your own owner. It sounds silly, but when an identity is assigned to an application or machine, it gets an additional attribute of an owner(s). This is conceptually the same as having a supervisor or manager assigned for human identities; however, for non-carbon-based life forms, there are a few additional traits to consider:

- The owners of a machine or application identity are responsible for its runtime. Owners are responsible for the resource's lifecycle, maintenance, and behavior. This is the same as being a parent or guardian of a potentially mischievous minor.
- Ownership can be a one-to-many relationship and is not explicitly assigned to individuals. Ownership can be assigned to teams, departments, or even other entities outside of the organization, including other identities by role.
- Owners have explicit control over their identities and their associated technology implementation. This includes everything from cradle-to-grave operations for the identity and its privileged lifecycle.

Automation

Identities can operate autonomously or be under strict manual control and supervision. This includes everything from their automated creation to their instrumented revocation. This concept becomes the forefront of any identity discussion when identities are managed using Identity and Access Management (IAM) or Identity Governance and Administration (IGA) solutions.

Autonomous identity management solves many of the challenges with identity creation when the concept of personas is expanded into the concept of roles. Roles are a form of persona groupings that collectively assign identities together based on common technical, electronic, or business functions. An identity can be placed into a role to automate the entitlement assignment based on this grouping. So, if a human identity is required, its creation including accounts, privileges, entitlements, roles, and asset assignments can be automated from end to end. If a machine or application entitlement is required, then the appropriate subsets, including owners, can be automated as well. Once the identity is established, the automation of all the attributes can be controlled in a predictable and reportable fashion, making an audit or attestation of identity governance a simple exercise.

If you had to make these decisions manually, and even decide which machines should have an identity based on their inception, there would be inconsistencies in your implementation. We are humans after all and do make mistakes. Automation simplifies classification in a consistent way and helps enforce best practices when using

any identity management solution. Therefore, automating all the concepts we have previously been discussing is highly encouraged. Consistency in management is key for mitigating risks from identity attack vectors.

Types of Accounts

Identities can be instantiated in a wide variety of accounts across the enterprise. Accounts can be of many distinct types and can use a wide variety of techniques (well outside the scope of this book) to support authentication and authorization.

If an account is compromised, it can be used to compromise the entire identity based on its privileges. For instance, the compromise of an account with root, administrator, or other high-level privileges could be leveraged against the identity to compromise other associated accounts and services. This is a form of account-based lateral movement by a threat actor leveraging one privileged account against another, linked together by an identity.

To reduce the likelihood and impact of attacks that look to abuse privileges or escalate privileges, we want to restrict the privileges to the lowest common denominator for every type of account. This concept is called least privilege management and should, whenever possible, be implemented on each account to avoid an attack from laterally moving back to the identity itself and, therefore, another account. Figure 6-2 illustrates these relationships based on common account types and the credentials that are typically used for authentication.

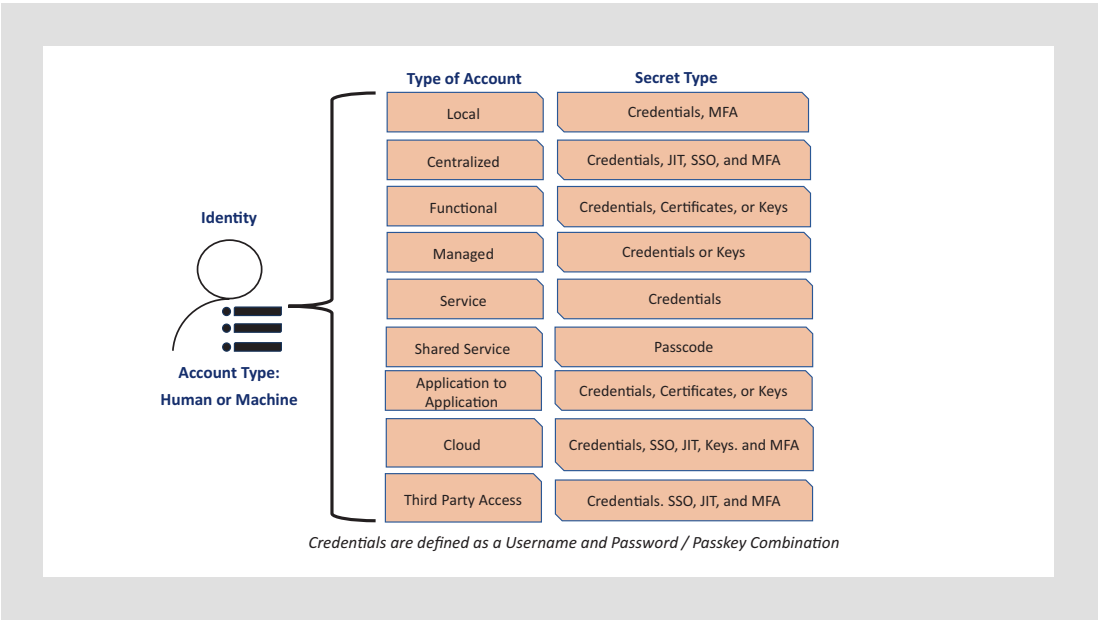


Figure 6-2. *Basic identity, account, and credential relationship*

In the next section, we will explore the most prevalent account types, how they are typically implemented, and how they can be used to compromise an identity.

Local Accounts

Local accounts are just that, local to the resource or asset. They may or may not be centrally managed or referenced in an Identity and Access Management solution. Duplicate accounts with the same credentials may or may not be present on other assets and need to be discovered to certify access or demonstrate the possibility of lateral movement. If duplicate accounts that use the same password do exist, it can lead to a lateral movement based on the same credentials. Therefore, local accounts should be unique by having different credentials—that is, different passwords, since it may not always be possible to uniquely change the usernames. If the usernames can be changed, then they should all be unique, too. This can create a management problem due to the number of unique account usernames and is typically addressed by Identity Governance and Administration (IGA) and Privileged Access Management (PAM) solutions. As a recommendation, identity management and password storage should never be done in a flat file like a spreadsheet.

Next, if the duplicate local accounts are root or administrator, a compromise of the credentials could own the asset. If other resources share those credentials, privileged lateral movement could occur and, consequently, own the identity using the model we explained before. This is a privileged attack vector. Once the identity is owned, all accounts associated with the identity can potentially be compromised as well due to the identity/account relationship we have discussed. Local account creation should, therefore, have the least number of privileges assigned. Each instantiation associated with an identity should have unique credentials to prevent these types of attacks.

To illustrate this further, let us revisit our IoT coffee maker, which has a local administrative account called "admin." Changing the username may not be possible due to technology limitations. If there is more than one coffee maker within your organization, each should at least have a unique password. Managing those passwords is a separate problem and is typically controlled with a Privileged Access Management (PAM) solution. Again, credentials should never be stored in a flat file like a spreadsheet.

Centralized Accounts

Centralized accounts are typically stored in a directory service like Microsoft's Active Directory (AD), Microsoft Entra ID (formerly Azure AD), or a Lightweight Directory Access Protocol (LDAP) implementation. These are commonly referred to as directory-based accounts as a matter of context. The benefit of centralized accounts is that they allow for a single point of management for all subscribing resources that can authenticate against that account. A centralized account may also use permissions based on policies (authorization) assigned to that account or group of accounts.

Centralized accounts allow users to log in to various resources throughout an enterprise and provide access to resources, assets, and data to perform authorized tasks. When implemented using Single Sign-On (SSO), a user can authenticate once and access all associated assets and resources without reentering credentials. The risk here is fairly straightforward to understand but is contradictory to the centralized management benefits that are provided by authenticating once for everything. A compromised central account allows a threat actor to infiltrate and potentially misuse all resources associated with that account. If the user is a standard user with basic privileges, the rights are contained, and lateral movement is restricted just to that account and, normally, just to that user. However, if the compromised account is an administrator or, worse, a domain administrator, then not only is that user at risk but also every other account available

to that user. If SSO is implemented for privileged accounts, then essentially, the entire environment can be at risk. It is important to remember that centralized accounts can be obfuscated via account names, and centralization of accounts and their links to identities, assets, and applications is the primary responsibility of an IGA solution.

This is what makes centralized account management so important for any administrative accounts. As a cybersecurity rule of thumb, if a domain administrator account that includes the domain servers hosting centralized account management is compromised, it is game over. Proper mitigation activities would entail reloading the entire environment from scratch! Therefore, environments should always protect domain administrator accounts with the highest level of security, including multifactor, two-factor, or step-up authentication, whenever possible. This includes minimizing the use of SSO for administrative functions and delegating this work to privileged access management solutions instead. And every domain administrative account should be unique for any identity that requires domain administrative rights. That is, domain accounts should never be shared unless absolutely necessary—and even then, I would ask—why?

Functional Accounts

Functional accounts are used to perform automated account management functions, regardless of being local or centralized. They have elevated privileges and, in many implementations, domain administrator or root privileges across multiple resources and assets. Functional accounts should never be associated with any identity; they are strictly used for automation and should never be used for any daily work—nor should they allow for interactive logins. Ever!

A good functional account architecture limits the reach of each instantiation and prefers multiple functional accounts governing zones, resources, assets, and applications vs. a few that have nearly root-like or domain privileges across the entire environment. These accounts typically also fall outside of any just-in-time management for identity and privileged access management solutions since they must be considered “always-on” to perform automation. While this runs contradictory to security best practices and goals for zero standing privileges (ZSP), in reality, something needs to own all the work that needs to be done. If a functional account is compromised, repercussions are quite pronounced, and every account under the functional account’s control (managed account) is in jeopardy, too.

In our IoT coffee maker scenario, a functional account would manage all the accounts of each and every IoT coffee maker, delegated by an IGA solution, and would potentially manage their passwords to keep them unique using a PAM solution.

Managed Accounts

Managed accounts have a relationship to functional accounts in that the credentials, or their creation and revocation, are managed by a functional account. Managed accounts can be associated with identities, or they can be strictly electronic in nature. It all depends on the use case. The premise, however, is important. The account is not directly managed by an identity; it is managed through another account type. This distinction can help shield the identity from flaws that would result in duplicate credentials, stale or weak passwords, and other authentication problems that could lead to a compromised identity. This also includes accounts for former employees and contractors that should be disabled or removed.

The entire premise of placing an account under management is to avoid the human element that could introduce risk while, providing a vehicle for security to protect these accounts through automation and auditing, reporting, and governance over appropriate usage. A managed account is different than an account under management or centralized management of accounts, and the slight nuance in terminology should be recognized when having conversations or documenting an account governance process.

Service Accounts

A service account is a special type of local or central account created explicitly to provide permissions and privileges for an application, operating under a context within an operating system. The permissions and privileges assigned determine the service account's ability to access local and network resources during its runtime.

Service accounts are a special type of account because they should never have the same entitlements as a person logging in to a system. They should not have interactive user interface privileges nor the capabilities to operate as if they were a real person. Depending on the operating system, this could encompass everything from operating as a batch process to not having a proper shell assigned to them. They normally are not delegated to just-in-time models, nor do they have entitlements assigned like functional accounts. They have privileges assigned unique to their application and service roles.

Service accounts can be managed, and they can be associated with an identity via an owner. To that end, their credentials can be local or centralized, and depending on the combination of managed, local, centralized, and shared, a compromise in a service account could lead to a threat actor owning the process, application, and identity associated with it. Service accounts associated with identities are typically also associated with machines and other applications.

Application Accounts

Application accounts should not be confused with service accounts. Service accounts provide the runtime credentials required for an application to execute, while application accounts provide credentials, secrets, certificates, or tokens (secrets) for applications to interoperate, usually through integrations. These are typically referenced in the industry as application-to-application accounts (A2A) via APIs (application programming interfaces) and can be found everywhere from integrations to scripts to Agile development practices like DevOps.

As another rule of thumb, identities should not be directly associated with application accounts. There are really no good use cases to even consider this; however, it does not diminish the importance of application accounts. With that aside, application accounts should always have an owner attribute. Application accounts exist everywhere in a modern ecosystem for applications to work together, and who is responsible for them is crucial consideration for any business.

Consider if two applications need to share information or interact: they will both need knowledge to authenticate across an application account. This interaction can also be seen everywhere from vendor integrations to RPA (Robotic Process Automation).

Application accounts should be managed in a way that prevents them from becoming a liability and that allows any changes to the secrets to be synchronized between the applications themselves before they are required to authenticate. This is typically performed by linking the credentials in an IGA solution and PAM solution, so the accounts are managed together, and applications can share the same account. This can be a local or centralized account, depending on the application's implementation.

If a threat actor compromises an application account, they can impersonate the runtime communications between applications and potentially monitor activity or extract information. This can happen via spoofing authentication using a man-in-the-middle attack and via rogue calls from addresses outside of access control lists that may be permitted to conduct transactions.

Cloud Accounts

There is technically no accepted definition and terminology to explain cloud accounts. Each cloud provider, SaaS (Software as a Service) vendor, PaaS (Platform as a Service) vendor, and IaaS (Infrastructure as a Service) vendor, uses a different definition to meet their business and technology approaches to solutions and management in the cloud. This even includes the definition for permissions, rights, and entitlements assigned. Ask a dozen security professionals about cloud accounts, and each one will give you a different answer.

Therefore, the use of cloud accounts is strictly a vehicle to describe any account in the cloud that is under the control of an organization. These accounts may or may not have an identity assigned and may or may not have functional, application, or managed account characteristics.

Just remember that, for the cloud, you do not own the operating environment. It is a cloud resource, but it does have accounts and privileges assigned to it, and they may or may not have identities.

To that end, the cloud account may be an administrator, root, or delegated via role-based access, but it is essentially a type of superuser. The cloud owner really is the administrator, and they own and control everything below their access—from the hypervisor to the virtual network. This means your organization can be compromised in two ways, unlike with an on-premises solution. First, you could incur a security compromise from the front—that is all of your cloud account–based access—and second, from the back—everything owned by the cloud provider, including the accounts used to support and manage the service. The latter dynamic is why special considerations need to be made for accounts in the cloud, particularly if they are linked in any way to identities. And if the account is a lower-level account used for containers or instances, the attack vector starts at the bottom of the food chain for the threat actor and potentially allows them to work their way up into an identity that governs more than just a single instance.

As an example, consider a cloud-based solution that stores biometric information for physical access, transactions, law enforcement, or background checks. Identities linked with biometrics warrant special consideration since you cannot change things like your fingerprints (unlike a password). If this data is stolen (and based on recent breaches, we have even seen completely insecure methods of doing so, like the OMB[4] breach) from the cloud, portions of the attack vector may be completely out of your control to manage, irrespective of the threat actor's attack vector. If successful, the threat actor has a method (biometrics) to authenticate an identity that can never be changed. They now have a way to always own your identity. This is another reason why protecting cloud accounts is paramount. It is the lowest hanging fruit for a threat actor to compromise, regardless of any management capabilities you can implement as security controls.

Organizations that trust sensitive data and services in the cloud must be aware of the additional risk, and the accounts that safeguard that information should be treated with exceptional care, potentially using a Privileged Access Management (PAM) solution. If there is any laxity in procedures and policies for passwords, keys, secrets, and vulnerability management, the threats cannot be mitigated with traditional layered security controls.

For any cloud service that you trust with your crown jewels, ask about their security. Ask not only about the security for your access but also how they secure the access from their own employees. Certifications like SOC II/III,[5] SAS 70,[6] SSAE 17,[7] and ISAE 3402[8] help validate their integrity and will provide assurances that the risk is minimal from both the providers and your cloud accounts, regardless of the attack vector. And remember, multifactor authentication is critical protection for every account in the cloud.

Entitlements

Entitlements are any technology implementation that controls access to something we care to manage. Entitlement is a category name used for something that grants, resolves, enforces, revokes, reconciles, and administers fine-grained access, privileges,

[4] https://osec.doc.gov/opog/privacy/Memorandums/OMB_M-17-12.pdf

[5] www.techtarget.com/searchsecurity/definition/Soc-2-Service-Organization-Control-2?Offer=abMeterCharCount_var3

[6] www.colocationamerica.com/data-center-certifications/sas70-compliance

[7] https://ssae-16.com/tag/ssae-17-audit/

[8] https://isae3402.com/ISAE3402_overview.html

access rights, permissions, or rules. An entitlement can stand apart from a role and assignment to an identity and, thus, its associated accounts. Its purpose is to define and execute an information technology access policy, regardless of resource, asset, device, or application and whether it's running in the cloud, on-premises, or anywhere else. Entitlements come in many shapes and sizes and can usefully be categorized as being either simple or complex in nature—this is explained further in the next section. The management of entitlements can be delivered by a range of different technologies and is, by convention, designed to work across platforms, applications, networks, and devices.

Simple Entitlement

Simple entitlements are atomic in nature. What they protect can be infinitely complex and varied, but the entitlement itself is easy to define, provision, audit, and control. A good example of a commonly implemented simple entitlement is a directory group membership being used to control the right to receive IP traffic on a network. "Basic Network Access" is a simple entitlement. It is simple to define and easy to understand, receive, and transmit network traffic. It is easy to provision—a policy manager simply puts your account into a defined AD group that can audit and control its use using a deployed identity management solution. What can or cannot be accessed once on that network? That is a more complex question in most implementations and is covered by a complex entitlement.

Complex Entitlement

Complex entitlements are, well, more complex than simple entitlements. The easiest way to meaningfully differentiate complex entitlements is to highlight their complexity in either their definition, their provisioning, their audit, or their control. A good example of a commonly implemented complex entitlement is an SAP R3 role. By its nature, it is a composition of other entitlements (other roles, T-codes, AuthCodes, etc., each possibly existing as an entitlement in its own right), yet it is assignable as a single unit of access, so it is itself an entitlement. Complex entitlements do not need to be comprised of group entitlements to fit the definition either. A complex entitlement in the cloud to "Create a Virtual Machine" touches multiple components in the cloud—from a hypervisor to a storage medium. The rights, permissions, privileges, etc., are all abstracted from the user, but under the hood, there is a high volume of automated activity that needs to be

performed for the entitlement to be enacted. Therefore, to understand the difference between simple and complex entitlements, think about what must occur in the background for the entitlement to be implemented. If the list grows longer than just a few basic items, like automating a firewall or enabling a network adapter, it is probably complex in nature and has its complexity obfuscated for interactive simplicity.

Roles

Now it is time for a formal definition of roles since it has been touched on in various ways and linked to various concepts needed for a successful identity security strategy. There are several different approaches to enterprise roles, and for the purpose of this book, I will be describing and advocating for the adoption of a basic two-tier role model, as is depicted in Figure 6-3.

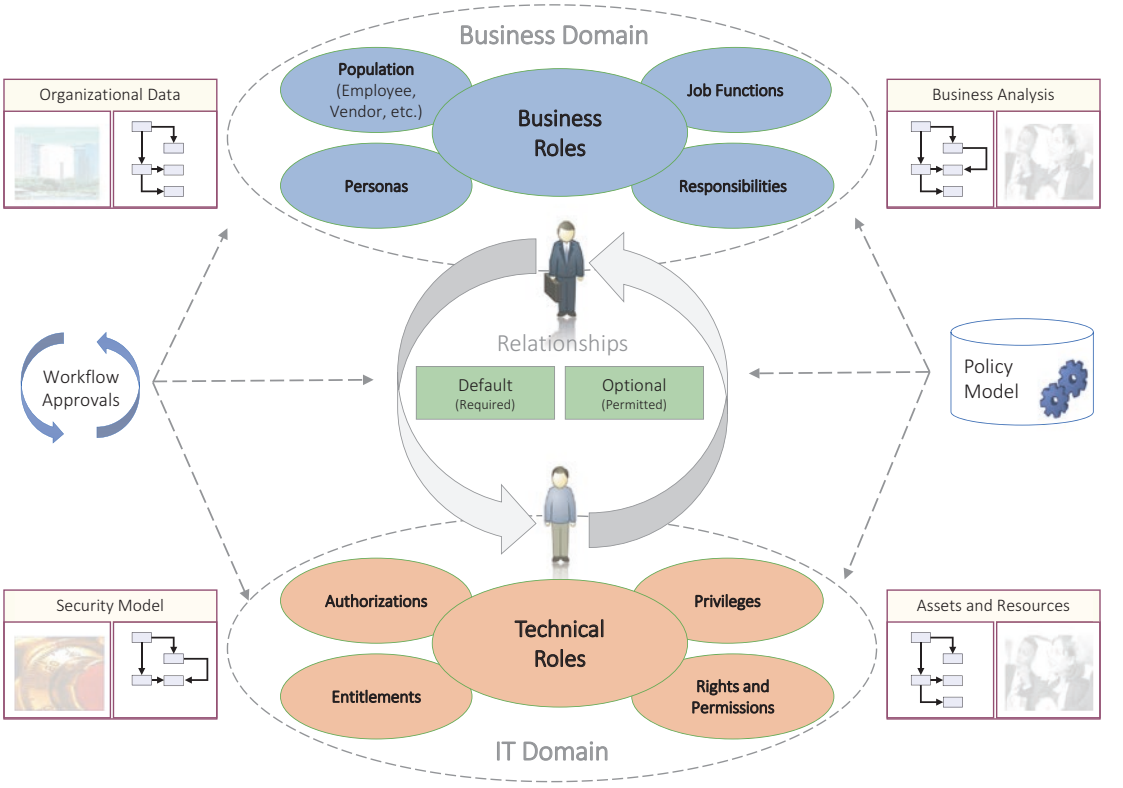


Figure 6-3. *A depiction of a basic two-tier enterprise role model consisting of business roles and IT roles with managed connecting relationships that support least privilege*

At the highest level of abstraction, we define a role as a collection of people or a collection of access, defined and maintained for the purpose of improved manageability, enhanced security controls, and the promotion of auditable governance in the form of security controls. Roles are sometimes used to group like identities that perform similar functions and need the same level of access to assets or resources. These roles are often referred to as business roles and are described in more detail in the next section. Roles are also sometimes used to associate related accounts from different identities and entitlements required to carry out a known set of actions. These are often referred to as IT (information technology) roles.

Business and IT roles are conceptually linked together to form user assignment relationships. It is considered a general identity governance best practice to only connect IT roles to business roles and to use the business role-to-identity relationship to complete the linkage back to the user. By employing different types of "connecting relationships" between business and IT roles, it is possible to further promote the principle of least privilege.

As a matter of relationships, some readers may confuse roles and personas. To assist in the differences, a "role" is primarily associated with access control and defining permissions and privileges within a system, while a "persona" is more focused on understanding and representing user characteristics and behaviors for design and user experience purposes. Roles deal with permissions and security, while personas help in creating user-centric designs by considering the diverse needs of different user types and buying patterns.

Business Roles

In one sense, business roles are simply groupings of people (identities). It is a common best practice to use identity attributes (information about an identity) to filter identities into business role groupings for the purposes of managing the entitlement assignment lifecycle. Using organizational information, a business analyst might define a business role to represent a defined dynamic or static population of people. By convention and through product implementation rules, business roles have defined business owners and serve as a holding place for useful metadata about who, when, and why this business grouping has been defined. Like most group implementations, business roles support inheritance to enable the creation of hierarchical trees of nested roles. Hierarchical business roles provide a useful way to define the generalization and specification required for scalable governance and identity management.

The difference between a business role and a basic directory group, for instance, comes from where it lives in the ecosystem. A business role is a management abstraction used in the identity management layer only, whereas a directory group is an entitlement construct employed by the target application. Business roles are usually the concern of business analysts; therefore, the technology that implements them must be usable and meaningful to the business users that directly own them and manage their lifecycle.

IT Roles

IT (information technology) roles are used to assemble groups of like entitlements. As their name suggests, IT roles live in the domain of IT and are used to define and manage the specifics of a given security configuration. IT roles profile how accounts, entitlements, privileges, rights, and permissions should be set in order to meet a given business purpose. Unlike business roles, IT roles are not defined by identities, but rather by security controls. In addition, account attributes are often used in profiles that make up IT roles, but only to further define a configuration—not to assign themselves to an identity.

IT roles also support hierarchies and nesting to simplify the definition of complex security configuration, and it is common to find complex encapsulation in large, mature deployments. If the nesting tree hierarchy becomes too complex, layered, or recursive, then certifications and identity attack vectors could unintentionally be introduced based on faulty role assignments. Like business roles, IT technical roles also have a defined owner—usually someone close to the resource or configuration in question. IT role owners are responsible for managing the lifecycle of the roles they own and are frequently responsible for approving any change to their definition based on policy or regulatory compliance.

Roles and Least Privilege

Least privilege can be defined as only giving an account the entitlement and privileges needed to perform an intended function, and nothing extra. Steering your entire identity, account, and entitlement lifecycle management process toward least privilege will result in

- *Better System Stability*: By limiting inappropriate or unsanctioned functions
- *Better Overall System Security*: By limiting inappropriate behaviors
- *Lower Overall User Access Risk*: By not giving malware the privileges needed to infect a system or for a threat actor to execute unwanted lateral movement
- *Minimized Risk of Human Mistakes*: By eliminating privileges that can inadvertently affect the runtime of a system

Through the careful management of the relationships between business and IT roles, it is possible to promote goals toward a least privilege environment.

As discussed, it is considered a general identity security best practice to only connect IT roles to business roles and to use the business roles to create an identity relationship to complete the linkage back to an account. By employing different types of connecting relationships between business and IT roles, it is possible to further promote the principle of least privilege. At a minimum, an enterprise-class solution should support mandatory and optional relationships between business and IT roles. A mandatory relationship says, "if you are in this particular business role, you automatically get the entitlements assigned to this set of IT roles." An optional relationship says, "these IT roles are permitted to be associated with, but not by default." Giving out less entitlement by default is the ultimate goal of least privilege.

During an identity security implementation, optional relationships provide implicit "model-based" control relationships. For example, in a self-service and delegated provisioning scenario, this pre-modeled relationship can be used to "preapprove" a user's privilege request, hence improving user experience while supporting least privilege. This is typically performed using endpoint privilege management (EPM) solutions, which are one of the primary subsets of PAM products.

Summary

The implementation of our identities in business has become an essential prerequisite for us to do basic tasks. Everything from walking into an office building and scanning our badge to punching a timeclock at a construction site is all part of identity management, just like their electronic counterparts.

Our digital identities are everywhere, while our physical personas remain rooted in who we are and what we do. The combination is what identity attack vectors prey on.

At the heart of every identity are privileges. If you can compromise an identity, then you assume its privileges. If you can elevate your privileges to root or administrator, you can own an asset. If you own the asset, you can manipulate the data and software to conduct nefarious activities. Therefore, we are back to our three circles and Venn diagram for security. Our implementations must start with strong protections and separation of duty for an identity based on accounts. If designators for our identities are compromised, then the entire model for identity security can collapse.

A simple real-world example of this in the United States pertains to the one-to-one translation of an identity into a Social Security number (SSN). Social Security numbers are considered personally identifiable information[9] (PII), are linked to a person's name, and can be used in a wide variety of identity-based attacks. Opening a credit card or potentially tampering with someone's mortgage or property title is possible if a threat actor knows the victim's SSN. If the threat actors know other tidbits of information, like the victim's address and date of birth, we can see large-scale identity-based attacks using concepts like synthetic identities. Therefore, keeping an SSN safe, secure, and secret is critical to protecting one's identity. By simple deduction, any substantial PII in your identity security model that is compromised could lead to exposure for your employees and business.

However, it's important to point out that not all countries follow the same model, just like many businesses use different identity security models. In many countries, the SSN equivalent is not private, but rather public so that any person can be documented, tracked, or identified by their unique designation. In such instances, possessing the number and a user's name is insufficient for initiating an identity attack.

This leads us to an interesting conclusion. A secure implementation of an identity-based system should not have a single point of failure, such as a Social Security number. Any identity-based system should be resilient to protect a user's identity, even if multiple indicators of compromise have been detected or PII has been leaked. This is a high-level concept we will continue to explore as we provide guidance on how to implement it within your organization.

[9] www.techtarget.com/searchsecurity/definition/Soc-2-Service-Organ

CHAPTER 7

Identity and Access Management (IAM)

Today, Identity and Access Management (IAM) is a core discipline for managing security risks associated with identity accounts, privileges, and their usage in an everyday context. The unique discipline of IAM plays a pivotal role in securing digital assets from inappropriate usage and ensuring the data privacy for information in motion and at rest.

IAM is a multifaceted discipline that encompasses a variety of terms, acronyms, best practices, architectures, and security considerations. This chapter will explore the intricacies of Identity and Access Management, including fundamental IAM concepts like governance, real-time access controls, and privilege management. It will explain the critical role these concepts play in bolstering identity security. Figure 7-1 illustrates IAM at a high level, along with all the potential technology, business, and security disciplines that are included in the framework that we have been discussing. In addition, emerging technologies like artificial intelligence (AI) and its influence on IAM will be discussed separately in Chapter 14 since, as of the writing of this book, no significant impacts have been realized from the adoption of this technology.

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_7

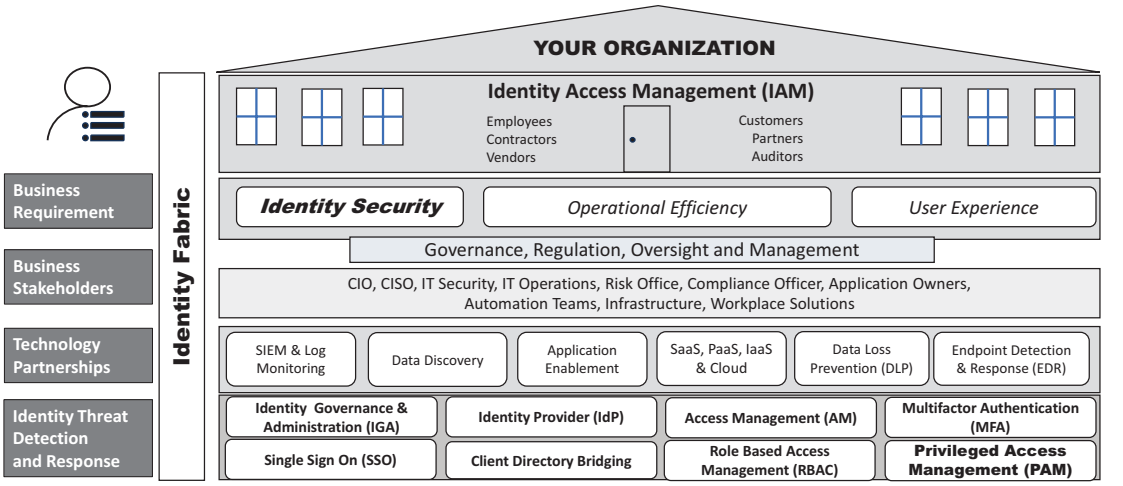


Figure 7-1. *IAM discipline model*

IAM is the overall framework that establishes and manages identities and associated accounts and controls access to resources by personas and roles. In the diagram, think of the house as your organization and everyone that may need access. IAM ensures that the right individuals have the appropriate level of access, all while maintaining security, access, and compliance across an identity fabric (vertical bar), and embraces the (horizontal bars):

- *Business Requirements*: Business requirements refer to the specific needs, objectives, or conditions that must be met to achieve the goals of an IAM project. In our model, the user experience must resist pushback, any implementation must be operationally efficient, and provide security for all aspects of the implementation.
- *Business Stakeholders*: The business stakeholders are the roles and personas within the organization responsible for the technology and business implementation of an IAM solution. This can involve potentially any employee in the organization, from executive team member to entry-level engineer.
- *Technology Partnerships*: Based on discipline, vendor, and solution, the security technology licensed and deployed in an organization that is integrated together and provides some form of automation for the organization.

In addition, each IAM discipline from IGA to PAM (bottom horizontal bar) has Identity Threat Detection and Response (Chapter 9) components that ensure the integrity of their unique IAM discipline (products). As discussed earlier, identity security governs all IAM disciplines and workflows across all horizontal layers and across the workflows and policies that form an identity fabric (vertically) up and down the stack.

Architectures

A typical IAM architecture encompasses a wide range of components and technologies (bottom horizontal bar). These are designed to work in harmony to efficiently manage identities and control access. The following components are fundamental to all IAM systems and can be present as multiple IAM solutions within any given organization:

- *Identity Governance and Administration (IGA)*: IGA is a framework for managing user identities and their access to resources in an organization. It encompasses policies, processes, and technology to ensure compliance, security, and efficient management of identities, accounts, roles, and entitlements used for access control and auditing purposes. One of the primary functions of IGA is the provisioning and deprovisioning of identities and accounts. This involves the automated process of granting and revoking access (joiner, mover, and leaver) to assets and resources based on predefined business processes. It ensures that accounts always have appropriate access and reduces the risk of unauthorized access based on automation, repeatability, and consistency.
- *Identity Provider (IdP)*: This serves as the central storage (database or blockchain) for user identities and their attributes. It typically includes identity, account, and user profiles, access policies, roles, and accounts. Common repositories include Active Directory, Entra ID, LDAP directories, and other cloud-based identity providers.

- *Access Management (AM)*: Access management provides authentication services to validate the identity of accounts during login. This can include multifactor authentication (MFA) to provide confidence in the identity as a security best practice. Popular authentication methods include password-based authentication, biometrics, and smart cards or USB passkeys adhering to standards like FIDO2.
- *Multifactor Authentication (MFA)*: MFA is a security method requiring users to provide at least three authentication factors to access a system. Typically, these factors include something the user knows (like a password), something they have (like a smartphone), and something they are (like a fingerprint or facial recognition) to provide a high confidence for authentication or authorization.
- *Directory Bridging*: Directory bridging is the process of connecting and synchronizing account data and access controls between different directory services, allowing for seamless authentication and access management across multiple operating systems or environments.
- *Role-Based Access Control (RBAC)*: RBAC is a model that assigns permissions to IT roles rather than individual accounts. Users are then assigned IT roles based on their business roles. By using a consistent approach, RBAC helps simplify access management and improves security by reducing the risk of overprivileged accounts. In addition, RBAC are typically defined as business policies. Access control policies define what actions users or systems can perform and what resources they can access. These policies are crucial for enforcing the principle of least privilege, which restricts access to the minimum amount and duration necessary for accounts to perform their tasks.
- *Single Sign-On (SSO)*: SSO is a mechanism that enables users to access multiple applications or services with a single set of credentials, preferably using multifactor authentication. The result streamlines user experience for accessing multiple applications.

- *Privileged Access Management (PAM)*: PAM is a cybersecurity approach and set of tools that focuses on securing and managing access to privileged accounts, such as administrator or root accounts. PAM solutions enforce strict controls, including password rotation, session management and monitoring, and least privileged access, to protect critical systems and data from unauthorized or malicious actions. PAM will be covered in depth in Chapter 13.

Implementing robust IAM practices is essential for maintaining identity security and the integrity of an organization's intellectual property. While these may be available in single vendor solutions, best-of-breed offerings often provide the best identity security for an organization and rely on technology partnerships (third horizontal bar) to implement a seamless solution. This helps minimize identity-based risks based on best-of-breed technology as long as tight integrations exist to enforce a seamless workflow (identity fabric) and do not become a liability in themselves due to vulnerable or poorly implemented integrations.

Risks

Obviously, a failure to maintain a good IAM program can lead to a myriad of security implications, ranging from data breaches to compliance violations. As part of our identity security discussion, consider these common risks:

- *Data Breaches*: Inadequate IAM can result in data breaches when unauthorized users gain access to sensitive information due to an identity compromise or failure to properly implement least privilege (or both). Proper authentication and access controls are essential to preventing this type of breach.
- *Insider Threats*: IAM also addresses the insider threat, where trusted individuals misuse their privileges. IAM in conjunction with PAM solutions can monitor user activities and detect suspicious behavior. These solutions help organizations mitigate insider threats, regardless of whether the access was intentional, or the user was behaving as a mule (compromised identity).

- *Compliance Violations*: Many industries have regulatory requirements related to data privacy and security (e.g., SOX, PCI, GDPR, HIPAA, etc.). IAM plays a crucial role in ensuring compliance by managing user access and certification of roles.
- *Credential Theft*: Weak authentication methods and poor credential management can lead to identity theft, enabling attackers to impersonate legitimate identities and their associated accounts. MFA and enforceable password policies are critical to prevent such theft.
- *Shadow IT*: Without effective IAM, employees may resort to shadow IT. Based on a variety of personal or business needs, they may begin using unauthorized products, solutions, and cloud services that pose security risks when not properly vetted or secured. IAM helps organizations maintain visibility and control over inappropriate solution implementations.
- *Unauthorized Access*: Inconsistent access policies and controls can result in unauthorized access to critical assets and data. The primary function of IAM is to enforce role-based access control and continuous monitoring that can prevent such incidents.
- *Inadequate Identity Lifecycle Management*: Neglecting to manage the entire identity lifecycle, including joiners, movers, and leavers (onboarding and offboarding identities, including changes in roles), can lead to unnecessary identity risks. IGA helps automate these lifecycle steps to ensure timely identity and access changes are maintained.

Best Practices

To maximize the effectiveness of IAM and mitigate identity security risks, organizations should implement best practices:

- *Strong Authentication*: Implement MFA to enhance authentication security and reduce the risk of identity or credential theft. And much like the best science fiction films that involve time loops, I will continue to restate the importance of implementing MFA for every account.

- *Regular Auditing and Monitoring*: Continuously monitor account activities and audit access to detect and respond to suspicious behavior. This means that you are not only logging activity, but you are actively reviewing it on a periodic basis, too!
- *Least Privilege*: Enforce the principle of least privilege to limit access to the minimum amount and duration required for users to perform their tasks. Review any logging that would suggest actions seeking unauthorized privileges.
- *Identity Governance*: Establish clear governance processes to manage the identity lifecycle for all employees, contractors, and vendors.
- *Training and Awareness*: Educate employees about identity and access policies and train on common attack vectors using tools like phishing simulators.
- *Cloud IAM*: Implement a unified IAM solution that can manage both on-premises and cloud environments to ensure consistency and minimize administrative overhead. This will also minimize security risks by eliminating tools that synchronize accounts and passwords between the two environments.
- *Incident Response Plan*: Develop a demonstrable incident response plan that includes identity and access attack vectors and rehearse it using tabletop exercises.

Identity and Access Management is a critical foundation of modern identity security strategies. It addresses the fundamental questions of who can access what, under what circumstances (when), and how does an identity authenticate. By understanding core components, terminology, and security best practices, organizations can effectively safeguard their assets, protect sensitive data, and ensure compliance with regulatory requirements. Embracing best practices in IAM is essential for staying ahead of evolving identity threats.

CHAPTER 8

Privileged Access Management (PAM)

Privileged Access Management[1] (PAM) is a subdiscipline within the Identity and Access Management framework. PAM can be implemented and operate on its own, or it can be integrated into an organization's Identity and Access Management (IAM) program.

Organizations may choose to start with disciplines like Identity Governance and Administration (IGA) or PAM; however, unifying both should be the ultimate goal as the enterprise matures through the IAM lifecycle. In fairness, many organizations will never mature to this point, but the goal should always be to streamline the identity security process.

By definition, PAM is a methodology to secure, control, monitor, and manage privileged activity to assets and resources. The discipline includes multiple components to manage privileged identities, accounts, secrets, and credentials, along with their corresponding passwords, certificates, and keys—regardless of the remote access technology required to complete the workflow. The objective of PAM is to provide complete accountability and control over privileged accounts and access. PAM lowers cyber risk by only providing privileged access to authorized accounts that require administrative or root privileges as needed for completion of a task or mission, and by removing or eliminating privileges for common day-to-day operations. This forms the basis of a least privilege account model.

The resources under PAM management can include anything from an operating system to applications, databases, network devices, scripts, DevOps, IoT, OT, ICS, cloud resources, and so on. The implementation of PAM is performed using dedicated solutions, policies, and procedures that focus on managing privileges and all the locations where they may be present. IGA solutions interface with PAM by managing and

[1] www.beyondtrust.com/products

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_8

certifying the identities associated with PAM accounts and credentials. The connection between IGA and PAM is typically based on a connector technology built into the IGA solution using standards like SCIM[2] (System for Cross-Domain Identity Management).

PAM solutions provide organizations with the secure privileged access tools needed to protect all assets. The traditional focus of PAM has been on protecting the critical resources containing the most sensitive information and infrastructure. However, the ultimate goal is that, if an asset has an administrative account, no matter what it is, it should be managed by PAM and managed by IGA.

In contrast, not all identities and accounts associated with IGA or on your IAM model should be included in PAM. PAM focuses on identities and accounts that are used for privileged access or to gate sensitive information or systems. Whether those are administrators, root, or superusers does not matter. The only time PAM typically includes non-privileged accounts is when the credentials need to be under management for access based on sensitivity, or if they are shared credentials. This is typically the case for technology, like remote access solutions, where credential injection is preferred to obfuscate the credentials from the end user, adding another layer for preventing identity-based attacks.

To understand all the capabilities of PAM, please refer to the bottom of Figure 8-1. This helps illustrate its importance as a subset of IAM.

[2] https://scim.cloud/

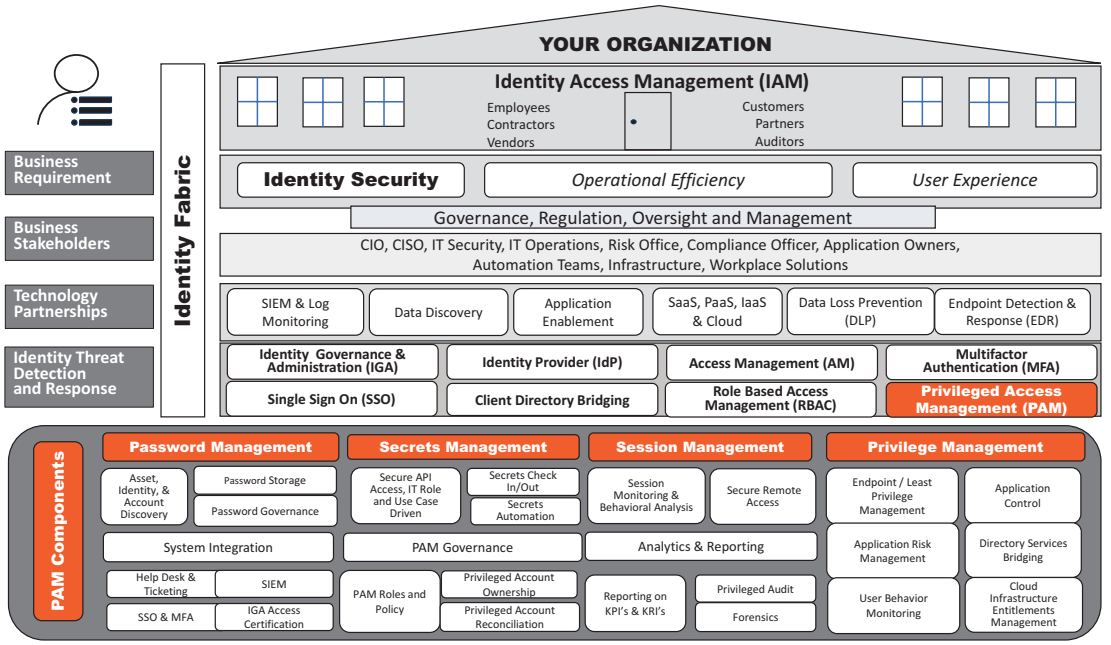


Figure 8-1. *Privileged Access Management (PAM) components*

As a function of any IAM deployment, PAM consists of the following components:

1. *Password Management*: The ability for a solution to securely store or vault credentials and accounts in a password safe for manual, automatic, or programmatic retrieval by an identity.

    - *Asset and Account Discovery*: The capability of a tool or process to identify assets connected to a network, all the accounts (privileged or not) allowed to function on the resource, and their associated privileges and group membership.

    - *Shared Password Storage*: Security best practices dictate that only one identity should have access to an account and its associated password. However, many technology implementations do not support proper role-based access controls (RBAC), and a single account needs to be used by multiple people. For example, a resource may only allow one administrator account, but there are multiple administrators in an organization that need access to the resource. Therefore, shared password storage is a use case expansion of password storage to allow a one-to-many approach for account access.

- *Password Governance*: Any password that is stored should be encrypted against potential theft by a threat actor. In addition, any exposure or transmission over the network of the password should also be encrypted, regardless of its use case and implementation. This includes automatic password injection technology and automatic password population technology, which may provide a seamless user experience and protect the password from human or third-party machine-readable exposure, such as memory-scraping malware. All of these functions should be auditable throughout the lifecycle of a privileged account.

2. *Secrets Management*: The ability to perform management functions on a secret associated with a nonhuman account. This includes secret changes across applications and nonhuman identities, such as service accounts, automation routines, and even scripts, regardless of the IT or business role. This concept within PAM can apply to on-premises technology, cloud implementations, or both, for interoperability.

    - *Secure API Access*: Many organizations have a standard secrets management policy. This can include the complexity of the secret and the required frequency of rotation for persistence. These policies form the basis for automated secrets management and must be electronically translatable into a PAM solution for automated API access.

        - *Role Based*: While secrets management is typically considered an automated process, there are occasions where ad hoc secrets management is required. These can include

            - *Break Glass*: A security or operational exception requiring manual secrets retrieval for business continuity. Please note that this can also be applied to break-glass scenarios for password retrieval since passwords are just another form of secret.

        - *Incident or Breach*: The manual forcing of secret or password changes across potentially large quantities of accounts outside of SOP (standard operating procedures) to mitigate the risks from a security incident.
        - *Employee or Human Resources Request*: The manual changing of secrets associated with a human identity owner due to an employee life event or role change.
    - *Ad Hoc Secrets Management*: Adheres to the policies in your SOP, but allows for them to be enforced or altered to mitigate an out-of-bounds threat or emergency situation. It is important to note that all ad hoc request types should be documented in your SOPs (standard operating procedures), but managed ad hoc since their potential usage and cycle is not predictable.
    - *Secrets Check-In/Out*: The ability to validate an account based on some form of trust (credential, key, certificate, ACL, etc.) for the retrieval of a secret, and subsequently, to check them back in once their usage is complete, for auditability.
    - *Secrets Automation*: The programmatic rotation of secrets, certificates, or keys based on policies implemented in the PAM solution that aligns with the organization's SOP for secrets management. This is typically associated with IT roles, like DevOps, business application integration, and security automation.
3. *Session Management*: The ability of the PAM solution to record user and application interaction with a command or remote session, regardless of connection protocol. The PAM solution should index the activity on screen or through the keyboard or mouse for future searching, retrieval, behavioral analytics, playback, and forensics.
    - *Session Monitoring and Logging*: The ability to document session activity for human auditing in real time or at a later date. In addition, the ability to log activity in machine-readable formats for log consolidation, user behavior analytics, or for event correlation to determine an indicator of compromise (IoC) through human review, policy-based rules, or artificial intelligence.

- *Secure Remote Access*: The ability to automatically launch a session, including by injecting the credentials automatically and managing the connection for duration, contents, data loss, and commands, regardless of protocol.

4. *Privilege Management*: The ability to monitor, control, and terminate any and all privileged activity occurring on an asset, whether by user or application.

    - *Endpoint Least Privilege Management*: The ability to enforce a least privilege user model on any endpoint, regardless of server, workstation, or infrastructure. User and application privileges are minimized to the lowest denominator to perform a task or mission. When higher privileges are required, the application, resource, or operating system function is elevated without the end user or application explicitly entering elevated credentials.

    - *Application Risk Management*: Applications, even from trusted vendors, can have a wide variety of risks based on their configuration and known vulnerabilities. Application risk management assesses the risk of the application before applying privileged access in order to mitigate any known threats. This concept is typically called Reputation-Based Services and includes Internet-enabled inquiries for the application to determine its origin and risk before execution.

    - *User Behavior Monitoring*: A user may execute certain applications and operating system tasks in a repeatable fashion based on their role and job functions. In addition, some applications and commands should never be executed together, like access to sensitive data and a screen-sharing application or file transfer program. User behavior monitoring applies threat-based rules to end-user activity and will send an event or create an alert when suspicious behavior is occurring or when known threat patterns match an identity's electronic behavior.

- *Application Control*: The ability to affect the runtime of an application on a resource based on any criteria the solution can quantify. This includes techniques like allow listing, deny listing, and even gray listing applications to execute (with or without privileges) based on the environment, vendor, geolocation, download source, etc. Typically, organizations will blocklist a specific vendor or class of tools, like bit torrents since they have no licenses and lack legitimate business purposes within an organization.
- *Directory Services Bridging*: Privilege management translates to every platform. It is agnostic. Whether the device is Windows, macOS, Unix, Linux, IoT, DevOps, cloud, virtual, router, switch, or any other infrastructure, the concepts of privilege management apply to all of them. A PAM solution should be able to address every aspect of your organization where privileged access is used by allowing a single identity provider, like Microsoft AD or Entra ID, to provide authentication to any platform. This implies a user can authenticate their account on a non-Windows system using Windows-based credentials in lieu of having multiple directory stores and identity governance solutions.

5. *Cloud Infrastructure Entitlement Management*: Cloud Infrastructure Entitlement Management (CIEM—pronounced "Kim") is the process of discovering and managing permissions and entitlements in the cloud. CIEM solutions are identity-centric cloud security management offerings that aim to streamline management of entitlements and least privilege enforcement across cloud service providers (CSP) and multicloud environments. CIEM solutions provide greater value when leveraged across multicloud environments to reconcile different types of entitlements. Otherwise, organizations would be forced to rely on a patchwork of native toolsets from the various cloud providers they use. CIEM is needed for a variety of reasons and is the latest concept to be added to the PAM stack. First, the cloud

provides a dynamic infrastructure for resources to be constructed and deprovisioned based on demand and workload. Identities created for these use cases are particularly susceptible to being over-provisioned with privilege, creating inordinate risk. Second, while cloud providers offer some native identity management tools, these tools are not portable to the platforms of other cloud service providers. When organizations use multiple providers, the instrumenting policies and runtime needed to manage them become a burden due to the inherent dissimilarities—from terminology to identities and entitlements. Finally, mismanagement of identities in the cloud can lead to excessive cloud security risk. Without a proactive approach to managing cloud identities and their associated entitlements, a damaging incident is bound to happen. This is especially true if an identity is over-entitled. Implementing centralized management **coupled with** the concept of least privilege for these identities can lower risk for the entire environment.

However, absent standardized controls, the complexity of administering access entitlements across multiple clouds is a proven recipe for visibility blind spots, cloud security gaps, compliance anomalies, and a potential breach.

The integration of CIEM within a traditional PAM platform helps ensure no instance of privileged access is overlooked:

- Provides a consolidated and standardized view for cloud identity and entitlement management
- Allows granular monitoring and configuration of permissions and entitlements. Applies the concept of principals to track privilege models between different cloud service providers
- Applies an automated process to ensure all identities are appropriate and appropriately provisioned for each workload
- Enumerates the differences between dissimilar cloud infrastructure platforms and provides a single view with actionable guidance for resolution

Cloud infrastructure entitlement managers should implement the following security best practices:

- *Account and Entitlement Discovery*: Inventory all identities and entitlements and appropriately classify them in real time. This accounts for the dynamic nature of cloud environments and the ephemeral properties of cloud resources.
- *Multicloud Entitlement Reconciliation*: Reconcile accounts and entitlements, and identify which ones are unique per cloud and which ones are shared, using a uniform model to simplify management.
- *Entitlement Enumeration*: Based on discovery information, entitlements can be reported, queried, audited, and managed by the type of entitlement, permissions, and by user. This allows for the pivoting of information to meet objectives and the management of identities and entitlements-based roles.
- *Entitlement Optimization*: Based on the real-time discovery and operational usage of entitlements, the solution classifies over-provisioning and identities. Identities can then be optimized for least privileged access.
- *Entitlement Monitoring*: Real-time discovery also affords the ability to identify any changes in identities and entitlements, thus providing the alerting and detection of inappropriate changes that could be a liability for the environment, processes, and data.
- *Entitlement Remediation*: Based on all the available data, a CIEM solution recommends and, in most cases, fully automates the removal of identities and associated entitlements that either violate established policies or require remediation to enforce least privilege principles.

6. *System Integration*: Every solution licensed and installed in your environment should integrate, in some form, with the rest of your operational and security ecosystem. In fact, any security solution that operates as a silo and does not integrate with other tools is ripe for replacement once it has reached its end of useful life. For PAM and IAM, the following are critical to the success of your implementation, even if the perceived value diminishes:

    - *Help Desk and Ticketing*: Privileged access should follow an established workflow and approval process. The integration into existing ticketing systems, call centers, and help desk solutions allows for a documented workflow and approval process to grant or deny access and verify identities before granting privileged access to resources. In other words, a support ticket must be opened, or dynamically created, and approved (even if self-approved), before privileged activity can occur on any resource.
    - *Single Sign-On (SSO) and Multifactor Authentication (MFA)*: Based on privileged attack vectors and identity attack vectors, privileged authentication should not rely on credentials requiring only a username and password combination. In fact, even systems that are only moderately sensitive, but accessible to the end user, should use SSO and/or MFA to secure access. This will prevent password reuse, credential stuffing, and a variety of other attack vectors that could allow escalation of privileges and result in a threat actor owning an identity.
    - *Security Information and Event Management (SIEM)*: An approach to security management that aggregates security information and event management information into a single platform. The data can be analyzed, filtered, and correlated and can have artificial intelligence engines applied to the results to look for anomalies and other indicators of compromise. All PAM events, from password retrievals and session launches to password changes, should be forwarded to a SIEM for processing as a part of your larger security management program.

- *Certification*: IGA solutions provide certification reports that detail who could have access to an asset. PAM solutions provide certification reports on who accessed an asset and what they did with that access. Most auditors, based on regulatory compliance requirements, will want to see both. This concept applies to all assets, regardless of whether they are in the cloud or on-premise, or if they are an operating system, application, IOT (Internet of Things) device, etc.

7. *Privileged Access Management (PAM) Governance*: While PAM is a discipline based on the management of administrator and root privileges, the governance of PAM focuses on the policies and procedures required by an organization to actually implement it in day-to-day operations.

    - *PAM Roles and Policies*: Outside of the standard operating procedures (SOP) used for password and secrets management covered earlier, PAM roles and policies must also govern who should have access, when they should have access, and what privileges they should have delegated to them. Not every information technology administrator should have administrative rights to every resource, based on roles. Therefore, the governance aspects of PAM detail all the aspects of privileges and when they should be assigned to or revoked from different identities. This granularity goes far beyond a password complexity policy and typically leverages the business and IT role assignments in IGA for proper implementation. In addition, use cases such as break glass, disaster recovery, and incident response should be covered for out-of-bound or emergency PAM account access.

    - *Privileged Account Ownership*: As previously defined, every identity can have multiple accounts. Some identities are human and some are electronic. The ownership of every account should be clearly defined and documented as a part of your processes. No privileged account should ever be created without someone (human) being the responsible contact for its creation, usage, and eventual deletion.

- *Privileged Account Reconciliation*: Organizations change, and so do employees, and so does the ownership of projects, technology, and resources. Privileged account reconciliation is a cumulative process that uses all aspects of PAM—from discovery to privileged application usage, through privileged account ownership—to verify that all aspects are operating according to plan. For example, if an application has a policy to allow users to elevate an application for administrative usage, and the application is no longer being used, the privileged account reconciliation process should identify an obsolete rule. Then, the proper teams can flag it for change control and remove it from the affected policies. This PAM component is typically implemented using reports that are based on live privileged usage data and compared to implemented policies for privileged access within the PAM solution.

8. *Analytics and Reporting*

   - *Reporting on Key Performance Indicators (KPIs) and Key Risk Indictors (KRIs)*: Any information and security solution implemented within an environment should be able to produce reports, alerts, and events to indicate the health of the environment, as well as to quantify performance and risk. It is a self-reporting model. How well are you managing privileges, are there deviations in risk that need attention, and is the overall performance acceptable by end users and applications to meet business objectives?

   - *Privileged Audit*: Regulatory compliance and internal auditors will want reporting on privileged activity, regardless of where it is conducted. A PAM solution should be able to generate a wide variety of audit reports to satisfy these requirements both, in real time and based on historical access.

   - *Forensics*: The data generated from a PAM solution is invaluable for determining indicators of compromise and for forensics investigations. A PAM solution should be able to deliver granular reports for items like an application hash, command-line

switches, and timelines to satisfy these requirements. In addition, session management and monitoring are crucial to any forensics investigation to provide an account of what the user actually did. This is necessary to determine if their behavior was appropriate.

With all these in mind, PAM also has industry-standard acronyms to help group and explain these disciplines. It is important to note, however, that vendors will typically license these disciplines as individual products and may often bundle them together to satisfy an organization's identity security requirements:

- *Account Password Management (APM)*: Provides a technology approach to securely manage privileged credentials, including system accounts, service accounts, cloud accounts, and application accounts that are managed. APM solutions use strong encryption and a hardened password safe for storing passwords, keys, and other privileged credentials.
- *Privileged Account and Session Management (PASM)/Privileged Password Management*: Privileged accounts are protected by storing or vaulting their credentials in a password safe. Access to those accounts is managed by a PAM solution for all resources, including human users, services, and applications. Passwords and other credentials for privileged accounts are placed under management, and passwords can be rotated (changed) at definable intervals or upon occurrence of specific events—such as the end of a session or when an IAM solution triggers a significant personnel change (e.g., a key individual being terminated) or an incident with accounts being compromised.
- *Privileged Session Management (PSM)*: Establishes a remote connection to a resource manually or automatically through credential injection and provides session recording.
- *Session Recording and Monitoring (SRM)*: Provides additional capabilities to PSM tools that offer advanced auditing, monitoring, active management, and review of privileged activities during a privileged session. This includes, but is certainly not limited to
  - Keystroke logging and indexing for searchable content
  - Video session recording and playback at various speeds

    - Screen scraping with indexing, with search capabilities
    - Optical Character Recognition (OCR) translation of graphical screens
- *Application-to-Application Password Management (AAPM)*: This is also referred to as A2A PASM and is the ability for application resources, typically through a REST API, to integrate into a password management and storage solution with a high fidelity of security and encryption. This is also the solution for secrets management.
- *Privilege Elevation and Delegation Management (PEDM)/Endpoint Privilege Management (EPM)*: Specific, context-aware privileges are granted on the managed resource and can be controlled by identity, account, or credentials. The resource can be a server, workstation, mobile device, IoT, or infrastructure. Once a user or application authenticates to the resource, commands and applications are elevated based on the task or mission—without necessarily requiring an explicit input of the privileged account. This technology includes host-based command control (filtering) that can be implemented without the use of local agents or local privileged elevation. This requires a locally installed client to elevate the application, perform a RunAs or Sudo, or uses AAPM (application password management) to retrieve a valid credential.
- *User Behavior Analytics (UBA)*: Uses data analytics to detect threats based on anomalous behavior, established rules, and behavioral profiles. The results are designed to be correlated with other security solutions to determine intent and potential indicators of compromise.

As PAM disciplines, all of these can be managed by an IGA solution that can standardize privileged account entitlements and provide certifications for who or what has privileged rights. PAM then provides the compliance reporting for the actual activity that has occurred and ensures the behavior was appropriate. The goal is to provide true visibility into what an identity is allowed to do and what activities their associated accounts actually performed, whether malicious or legitimate. Appropriate or inappropriate behavior ultimately helps reveal indicators of compromise based on privileged or identity-based attack vectors. Figure 8-2 illustrates this constructive collaboration.

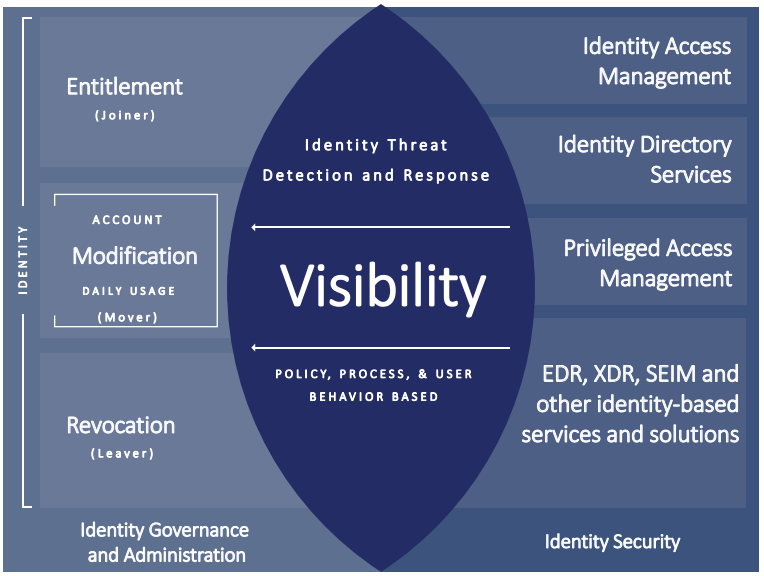


Figure 8-2. *Identity security visibility*

This forms the basis for any threat investigation against an identity and associated account by linking IAM and PAM for forensics and by establishing ownership of the who, what, when, how, and where. This completes the necessary information when determining a threat based on the privileged attack chain and the Cyber Kill Chain.

For those readers looking to embrace Privileged Access Management as a part of their identity security strategy, key RFP (request for proposal) questions can be found in Appendix A.

CHAPTER 9

Identity Threat Detection and Response (ITDR)

Identity Threat Detection and Response (ITDR—not to be confused with IT Disaster Recovery which is the other ITDR) refers to the combination of security tools and processes required to adequately defend identity-based systems. It is a layer in our IAM model from Chapter 7, since every IAM solution needs to be protected per discipline and when integrated together. Gartner[1] defines ITDR as "a security discipline that encompasses threat intelligence, best practices, a knowledge base, tools, and processes to protect identity systems. It works by implementing detection mechanisms, investigating suspicious posture changes and activities, and responding to attacks to restore the integrity of the identity infrastructure."

Despite what some security vendors will tell you, ITDR is a discipline, not a product. ITDR has emerged in recent years in response to the explosion of distributed identities beyond the traditional network perimeter and the resulting massive increase in identity-related vulnerabilities caused by things like unmanaged, misconfigured, or exposed identities.

This chapter explores foundational concepts of identity threat detection and response, why it is needed, and how it aligns to Privileged Access Management (PAM) and identity-based security protocols that are becoming increasingly critical to security mandates like zero trust.

Due to the move to the cloud and increased remote working, we have seen a major shift in the security industry, moving away from perimeter-based security strategies to one that is identity-oriented and response-focused. This has resulted in the identity being the new perimeter as malicious actors no longer need to breach a firewall to enter

[1] www.gartner.com/en/articles/7-top-trends-in-cybersecurity-for-2022

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_9

a network; they only need to compromise an identity. Identity-based attacks are on the rise as threat actors seek to exploit the identity sprawl caused by cloud adoption, the proliferation of nonhuman accounts, and the use of disparate systems to manage identities. These attacks make use of compromised credentials, overprivileged users, and gaps in visibility. Organizations also lack an understanding of which identities represent the most risk, making prioritization of mitigations more difficult. Identity-based attacks may exploit hidden attack paths that are harder to detect than traditional code-based exploits and missing security patches.

In many ways, identity compromise and misuse have become central to almost every modern cyberattack, because distinguishing between how a legitimate user is leveraging an identity and the misuse of that identity by an unauthorized user, is nearly impossible. By compromising an identity, a threat actor can essentially impersonate a user to access resources, compromise systems, move laterally, and further compromise identities to gain higher levels of access and privilege. Attackers have been quick to apply all the well-known "land and expand" techniques to these new environments. They capitalize on the lack of visibility and lack of unified telemetry many organizations experience with regard to cloud/multicloud and hybrid environments. Hopefully, you have seen this explained in detail so far in this book.

Research sponsored by the Identity Defined Security Alliance (IDSA) found that 90%[2] of respondents had an identity-related incident within the past year and that 96% of respondents believed there were missed opportunities to prevent or minimize the business impact of the identity-related incident(s). Businesses recognize that identity-related breaches are indeed preventable, but that does not mean this is an easy task.

The most pervasive factor preventing organizations from effective identity-related risk mitigation is a lack of continuous visibility of identities across all systems, especially on rapidly expanding cloud systems. Even for known identities, there is a lack of understanding around the privileges and entitlements associated with identities. This is a gap in modern IAM solutions. This problem is then compounded by dynamic environments where new users, systems, and integrations are constantly creating new attack paths on top of existing configurations that can mask the activity of threat actors.

Today, we have Identity and Access Management (IAM) tools which provide deep visibility of an account belonging to an identity. However, IAM tools are unable to see the full picture of how this information relates to the disparate systems, access,

[2] www.idsalliance.org/white-paper/2023-trends-in-securing-digital-identities/

and privileges that an identity has during runtime. They generally are responsible for configurations and changes at rest, and not dynamically during operations. Then we have security tools, like SIEM and SOAR, that have a breadth of visibility in events across the environment, but which lack the depth of visibility into identities. They are only tracking auditable events. Attackers increasingly exploit this yawning gap between IAM and security tools to create identity attack vectors.

This is where ITDR comes into play. By combining cyber threat intelligence,[3] detection, investigation, and response actions into a single security discipline, organizations are now much better poised to defend their identity infrastructure. Until recently, organizations tried to compensate for gaps in the visibility of identities—and the resulting increase of identity-based threats—by deploying endpoint security solutions focused on the detection of malicious code and activity. Regardless of the malware or exploit code used, identity compromise, lateral movement, and privilege escalation are at the heart of most attacks, as the attacker abuses the access and relationships between identities. These hidden and often unintended relationships between identities provide exploitable attack paths. Without complete visibility and understanding of identities, it is difficult to comprehend the privileged relationships between users and systems. This, in turn, makes it challenging to fully implement least privilege and just-in-time access models.

Organizations grasp the need to identify, eliminate, and audit attack paths in their dynamic, modern IT environments. These are all keys to reducing the attack surface, uncovering unknown risks, and remediating incidents independently from malicious code detection. Identity threat detection and response is about protecting credentials, privileges, entitlements, and the assets and policies that manage them. As we said earlier, ITDR is a discipline, not a market or product—but we still need to understand where it fits in the broader identity security landscape.

ITDR, EDR, and XDR

Endpoint Detection and Response (EDR) and eXtended Detection and Response (XDR) are complementary products to ITDR and take a similar approach. Both EDR and XDR collect signals across multiple sources, but one focuses on identities and the other on code execution. So while there are some areas of overlap around elevation of privilege,

[3] www.beyondtrust.com/blog/entry/threat-intelligence-an-explainer-guide

these solutions diverge in both detection and response. ITDR solutions effectively sit alongside EDR/XDR solutions and complement each other to offer a greater level of protection by looking at what the identity is doing, rather than just what is being performed on the asset.

ITDR and Identity Governance and Administration (IGA)

Broadly, IGA is a subset of IAM and is focused on account and identity lifecycle management and control and audit process controls. IGA has largely concentrated on improving user security by managing account and identity data and the relationships between people and data using business and IT roles. IGA works under the premise that it knows who your users are, and it understands what applications or systems those users are authorized to access. Provisioning and deprovisioning accounts are now an essential part of any security strategy. In most cases, however, IGA does not address the specific privilege usage and control process typical in a dedicated PAM product. IGA products today also rarely collate and analyze identity activity signals from multiple sources to understand identity or privilege usage. IGA often lacks visibility over accounts and privileges on applications and systems that are granted outside of the IGA solution or outside of its purview. IGA is a deterministic tool that manages only known identities, accounts, and access and reports on the configuration of these items at rest.

ITDR and IAM

ITDR is also differentiated from traditional Identity and Access Management (IAM) solutions, which focus on authentication, authorization, and administration and all the tooling to make both work. ITDR provides a level of visibility into credential abuse, privilege escalation attempts, and entitlement exposure across all systems that goes far beyond authentication and authorization controls implemented by IGA, PAM, or access management.

By focusing on identity signals in real time and understanding the permissions, configurations, and connections between identities and accounts, ITDR can be proactive in reducing the attack surface while also detecting identity threats. This is not a new

concept; tools such as BloodHound[4] already use graph theory to identify weaknesses in Active Directory (AD) configurations that could be exploited by an attacker. The information from this tool can be used to improve identity security by understanding the likely paths of lateral movement and privilege escalation, and shutting them down before a threat actor has the chance to gain a foothold in an environment.

This same concept can be applied at scale beyond AD to understand where the attack surface exists across all assets, resources, and systems. ITDR can illuminate how an attacker might compromise credentials, privileges, and entitlements to move between on-premises systems into cloud containers and infrastructure. This provides an unprecedented level of visibility into the attack paths that present the greatest risk to the business and enables these risks to be proactively mitigated.

This newfound level of insight can also be used to triage and respond to attacks in progress by identifying

- Compromised systems
- Exposed identities
- Where unmanaged identities exist
- When authentication does not match desired policies
- Details about identity and entitlement usage
- Where excessive privileges exist
- When secrets are stale or inappropriately managed
- Where inappropriate identity and account relationships exist

Privileged Access Management (PAM) and ITDR

To effectively protect against identity-based threats, you need both preventative and detective control capabilities. A preventative approach discovers and helps remediate gaps before a threat actor can exploit them. These gaps or vulnerabilities encompass unmanaged, misconfigured, or exposed identities and their impact on access to critical systems or resources. This is complemented by detection capabilities that raise an alert the moment the solution detects an identity or entitlement being compromised or

[4] www.sans.org/blog/bloodhound-sniffing-out-path-through-windows-domains/

misused. The prevention/detection combination allows you to proactively reduce your identity attack surface as well as identify attacks in flight. This is the primary purpose of ITDR. This will all sound familiar to those who have used Privileged Access Management (PAM) solutions, which provide a comprehensive approach to securing accounts and access across all privileged accounts in your organization. In essence, ITDR protects all accounts from identity compromise, whereas PAM mitigates the risk of the most sensitive privileged accounts.

While we have covered many concepts and definitions so far, we have not touched on the fundamental problems of identity attacks and potential abuse. To begin, consider the most basic premise—what could happen if someone compromised an identity and its associated accounts? How much damage could they really do? Could it be a game over event for your business or potentially end with team members incarcerated? Both are real outcomes if the identities within your organization are inadequately protected, and ultimately, compromised. While this should not be perceived as a scare tactic, the reality of the potential impact is quite tangible, regardless of whether your business operates solely using traditional on-premises technology, has begun a digital transformation to the cloud, or operates a hybrid environment. To unpack Identity Threat Detection and Response (ITDR), we must begin with the indicators of compromise that will help your business determine a threat, and if you have an identity security incident.

CHAPTER 10

Indicators of Compromise

There are plenty of solutions that can help provide indicators of compromise (IoCs). Some will highlight the IP address of an asset, the malware detected, or even unusual patterns in user behavior and password authentication attempts. All of these can be mapped back to the Venn diagram of cybersecurity discussed earlier in the book. The goal of any IoC is to identify when something is inappropriate in an environment, what evidence supports the anomaly, and potentially the root cause—from malware to insider threat. With this in mind, there are four characteristics that can create an identity IoC:

1. The entitlements (including privileges) assigned to an identity are misused or hijacked.
2. The inappropriate assignment of entitlements (including privileges) and their potential abuse.
3. Any attempted behavior that was inappropriate, is potentially malicious in nature, or drives undesirable results for the business.
4. Misuse of the IAM management control plane where the IGA, PAM, or entire IAM system itself is used to compromise the environment through malicious identity and account changes or usage.

This translates into two forms of IoC analysis. One parses the entitlements for an identity and their associated accounts on potentially every resource, then documents their entitlements for reconciliation. This is done from configuration data, in effect at rest. The second is based on user behavior and the account's runtime usage operation. This entails an active analysis of an identity and its interaction with resources to determine the risk of the behavior. Essentially, was the user acting appropriately for their role or in accordance with any form of known usage baseline? And was it really that human persona, or was their identity compromised and their access rights abused? This creates a linkage between intention and usage.

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_10

When end users embrace these concepts from an academic level, they quickly realize that technology has been created to simplify identity governance best practices, but implementation gaps tend to make the results impractical without adhering to some additional best practices, as we have previously discussed.

Role-Based IoC

As an exercise, let's explore the identity of a typical user. For this example, I will leverage John Titor (if you do not know who he is, just search the name and enjoy the rabbit hole that is maliciously associated with this identity).

Within Active Directory (AD), John has a username associated with him (JTitor). This username allows him to access Windows resources and any other solutions linked via a Single Sign-On (SSO) solution.

John may also have a secondary account in AD to function as an administrator or have elevated rights as a member of the help desk (JTitor-A). This means John now has two accounts associated with his identity that must be reconciled to him for all user activity and entitlement mapping.

If we add macOS, Unix, Linux, and other infrastructure, we can increase the number of local accounts that are linearly associated with the responsibility (business role) and granted entitlements. Tracking an IoC to the correct identity then becomes an art rather than an academic exercise—especially if security best practices are followed to obfuscate the username to something cryptic vs. user identifiable (e.g., THX1138). For example, John's username may be "jtitor" and easily identifiable, but some organizations, based on regulatory compliance or security best practices, may choose another schema that makes the association very difficult without an external reference database. For example, "THX1138" is John Titor's standard user account. That is something an IGA solution can help reconcile and is a best practice to maintain.

To that end, and for our example, John's obfuscated username is "THX1138," and his administrator account could look completely different (TK421). This helps explain the complexity of mapping any discovered entitlements or user behavior to the proper identity (John Titor). IGA tools can discover TK421, but it still must be reconciled back to its real name and used by forensics tools when discovering IoCs. In the most complex environments, this becomes an art form when illustrated in a simple identity mapping diagram.

As a best practice for simplifying the process of mapping the proper identity to: accounts, and for making it easy to discover IoCs, organizations should attempt to

1. Minimize the number of accounts associated with an identity.
2. Use a namespace schema that is similar on all platforms for account creation but is not easily readable by humans and let automation help resolve identity and account linkage.
3. Use a directory bridge to authenticate with one centralized, authoritative source, like AD or Entra ID, across other platforms, like Unix, Linux, and macOS.
4. Avoid sharing authentication credentials (especially administrator or root) with multiple identities.
5. Create unique authentication credentials for every identity and enforce just-in-time access whenever possible.

A second implementation gap is bidirectional in nature. We may understand what an identity is supposed to do (entitlements) and what it actually did (session monitoring), but determining whether or not the overlap was appropriate is definitely another art form for skilled security engineers and specialized software.

For example, if John is a senior database administrator (DBA), he should be allowed to run virtually any command on a database resource for maintenance, upgrades, backups, and to correct problems. In this example, John may have a junior DBA, Larry. As a subordinate to John, Larry should not have the same privileges. If John and Larry, two different identities, share the same account, there is no way to determine who (without some further investigation) actually ran the commands, whether or not they were correct, and most importantly, whether they were appropriate. While from a security perspective, John and Larry should not have the same entitlements, if they share the same accounts, they do. This leads to some additional best practices that organizations should adhere to:

- Every identity should have unique accounts associated with it. Accounts should not be shared with other identities, whenever and wherever possible. This helps to provide a clean audit trail and clear determination if an IoC exists for an identity.

- An identity's entitlements should also map to real user behavior via a privileged access management solution of specialized application behavior analysis software. Session monitoring will help prove if a behavior is an IoC.
- User behavior should be mapped back to entitlements or to identities (bidirectional) to determine if behavior is inappropriate or never seen before (an anomaly).

Using identities as an indicator of compromise can be a complex undertaking. It may sound simple conceptionally, but account implementations within most organizations can complicate the process. A security team must be able to use an IGA solution to map all accounts to an identity so that any analysis creates a clear path to determining inappropriate behavior. If you consider the most common roles within an organization, then IoCs can be developed based on an identity's persona and role. Consider Table 10-1, listing common roles and identity attack vectors.

Table 10-1. *Common account types and attack vectors*

IT Role (Account)	Function	Identity Attack Vector
Super Administrator or Root	Root and superusers have the highest level of access and control over a system, SaaS application, or network service. They are typically used for system maintenance, configuration, and installation of software or updates	Threat actors typically target these accounts through vulnerabilities in operating systems or through social engineering to obtain or exploit root access. Once obtained, threat actors represent a game over event for an organization. These are the most important accounts every organization should protect
Administrator	Administrator accounts are typically found in Windows environments and have extensive access privileges to manage user accounts, install software, and configure system settings. They are the highest form of interactive privileged accounts on Windows	Threat actors may exploit vulnerabilities, use brute force attacks, or engage in privilege escalation to compromise administrator accounts. Once they are compromised, lateral movement and compromise of any resource operating within a Windows domain is possible

(*continued*)

Table 10-1. (*continued*)

IT Role (Account)	Function	Identity Attack Vector
Database Administrator	Database administrators manage and maintain database systems, control access to databases, perform backups, apply security patches and software maintenance, and optimize performance	Threat actors may exploit weak database configurations, perform SQL injection attacks, leverage database application vulnerabilities, or gain privileged access through identity theft
Service	Service accounts are used by applications to access databases, server resources, and other services provided by an operating system or installed applications	Threat actors can compromise service accounts through vulnerabilities in an application owned by the service or poor credential management
Application	Application accounts are used to execute specific applications or in support of service accounts. These have predefined privileges and entitlements to complete mission-dependent tasks	Threat actors can target application vulnerabilities, exploit poorly configured entitlements, or leverage stolen or lead credentials to gain unauthorized access to the application
Third-Party or Vendor Access	Contractors, vendors, and other third parties may require privileged access to provide technical support and other services to an organization. This access can be performed on-premises or remotely	Threat actors can compromise accounts used by "non-employees" through supply chain attacks, social engineering, and weaknesses in the third party's own Identity and Access Management processes
Privileged User	Privileged user accounts are used by employees, administrators, and third parties (if authorized) who need specific elevated privileges to complete tasks. This can be in support of individual roles or tasks that require elevation not permitted by standard users	Threat actors can leverage privileged accounts via social engineering, insider threats, phishing attacks, credential theft, MFA attack vectors, etc., to directly access privileged accounts responsible for critical functions within an organization

(*continued*)

Table 10-1. (*continued*)

IT Role (Account)	Function	Identity Attack Vector
Break Glass or Emergency Access	Emergency access (break glass) accounts are typically kept secret and are only accessible when a trusted user requests access during a crisis to manage a cybersecurity event or system outage. Access to these accounts should be highly restricted and controlled to avoid potential abuse	Threat actors may target emergency accounts through unsecure policies and procedures governing their access, weak password management, social engineering, or faulty secrets management solutions that do not provide enterprise-grade encryption or protection
Daily Drivers	This should be the account everyone uses for daily work. This account should be provisioned with basic access and use the principles of least privilege. It should be configured for general application usage and allow the end user to carry out a general business tasks according to their role. No special privileges or entitlements should be present compared to peers	Here, a threat actor may leverage basic use access entitlements to impersonate and initiate an action as part of some other form of escalation of privileges or lateral movement. For example, gaining access to an email or chat system to start a conversation with help desk for some nefarious purpose

Hacking Techniques

Based on these roles, the following are the most common hacking techniques that can compromise an identity and some of their associated indicators of compromise:

Password Guessing: One of the most popular techniques for password hacking is simply guessing the password. A random guess itself is rarely successful unless it is a common password or based on a dictionary word. Flat out guessing is somewhat of an art, but knowing information about the target identity enhances the process and likelihood of a successful guess by a threat actor. This information can be gathered via social media, direct interaction, deceptive conversation, or even data gleaned and merged or aggregated from prior breaches. The most common variants for passwords that are susceptible to guessing include these common password schemas:

- The word “password” or basic deviations like “passw0rd” not found in typical password dictionaries.
- Derivations of the account owner’s username, including initials. This may also include subtle variations, including numbers and special characters.
- Reformatted or explicit birthdays for the user or their relatives, most commonly, offspring.
- Memorable places or events.
- Relatives’ names and derivations with numbers or special characters when presented together.
- Pets, colors, foods, or other important items to the individual.

Repetitive guessing does not require automation. This method may be more labor-intensive and has mixed success rates. Password guessing attacks also tend to leave evidence in event logs and result in auto-locking of an account after “n” attempts. For a threat actor, getting detailed information of the intended target normally involves advanced surveillance or inside knowledge. For the average person, it may just be a game of trial and error. In addition, if the account holder does not follow best practices and reuses passwords between resources, then the risks of password guessing and lateral movement increase dramatically. Imagine a person that uses only one or two base passwords everywhere for all of their digital presence. Unfortunately, this happens all the time. The best IoC for password guessing is to determine weak passwords within an environment or using a service to determine if credentials are available for sale on the Dark Web and already compromised. Figure 10-1 illustrates this very basic hacking concept.

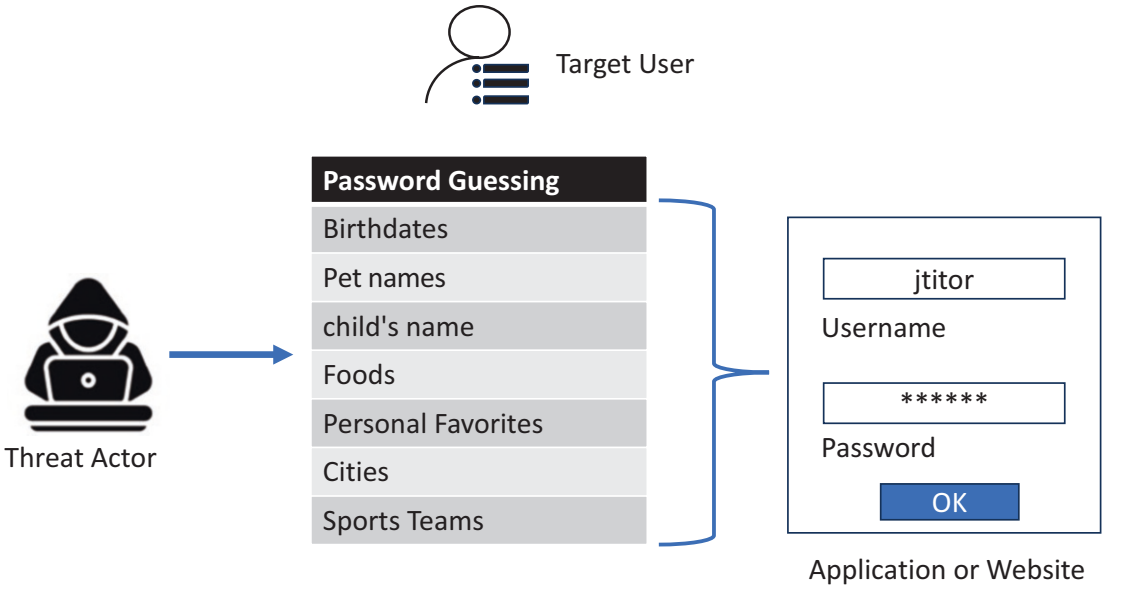


Figure 10-1. *Password guessing based on knowledge about the target user*

Shoulder Surfing: Shoulder surfing enables a threat actor to gain knowledge of credentials through observation. This includes observing passwords, pins, and swipe patterns as they are being entered. This includes even observing a pen scribbling a password on a sticky note. The concept is simple; a threat actor is watching physically, or with an electronic device like a camera, for passwords and reusing them for a later attack. This is why, when using an ATM, it is always recommended to shield the entry of your PIN on a keypad to avoid a threat actor from shoulder surfing your PIN. Shoulder surfing represents one of oldest identity attack vectors and one of the easiest for anyone to leverage. For a threat actor, all they need to do is find a way to watch someone entering their secrets (password, pin, etc.) on a data entry device. Figure 10-2 illustrates the concept of shoulder surfing and threat actors' eye on the prize, the target victim's password.

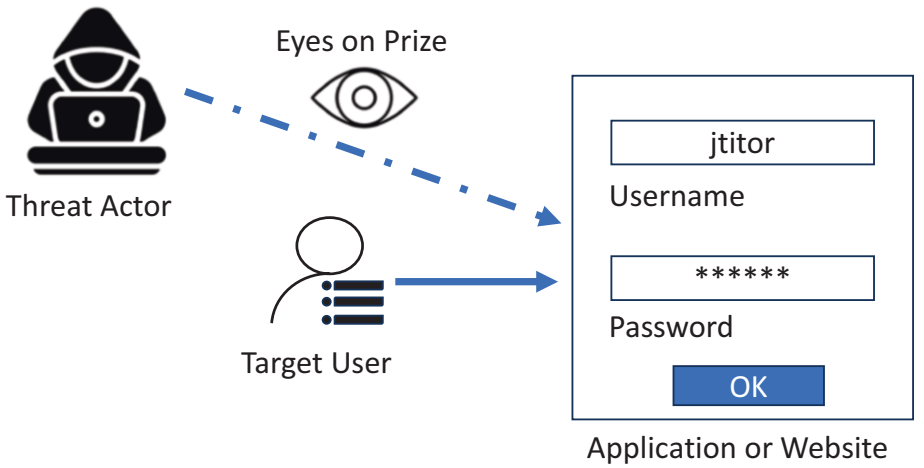


Figure 10-2. *Shoulder surfing a user's password for an eye on the prize*

Dictionary Attacks: These attacks are automated (unlike password guessing), utilizing a list of passwords against a valid account to hack the password. The list itself is a dictionary of words (no definitions, mind you), and basic password crackers use these lists of common single words like "baseball" to guess a password and hack an account. If the threat actor knows the resource they are trying to compromise, like password length and complexity requirements, the dictionary can be customized to target the resource more efficiently. Therefore, more advanced programs often use a dictionary on top of mixing in numbers or common symbols at the beginning or end of the attempt to mimic a real-world password with complexity requirements. An effective dictionary attack tool lets a threat actor do the following:

- Set complexity requirements for length, character requirements, and character set
- Allow for the manual addition of words from names to other personally identifiable words
- Can include common misspellings of frequently used words
- Can operate with dictionaries in multiple languages
- Contain the default credentials used by vendors when devices are initially deployed or reset

A weakness of dictionary attacks is that they rely on real words and derivations supplied by the user of the default dictionary. If the real password is fictitious, uses multiple languages, or uses more than one word or phrase, it will thwart a dictionary attack. There are just too many permutations for it to be successful.

In addition, there are a variety of supplemental attacks based on dictionaries that are available to a threat actor. If the attacker knows the password hashing algorithm used to encrypt passwords for a resource, rainbow tables can allow them to reverse engineer those hashes into passwords, if the password hash tables are exposed. Modern breaches have exposed vast troves of password hashes, but without a basis in the encryption algorithm, rainbow tables and similar techniques are nearly useless without some form of seed information.

Finally, the most common method to mitigate the threats of a dictionary attack is account lockout attempts. That is, after "n" times of wrong attempts, a user's account is automatically locked for a period of time, manually unlocked by an authority like the help desk, or via an automated password reset solution. However, in many environments, especially for nonhuman accounts, account lockout attempts can have undesirable effects to business runtime. Therefore, this setting is sometimes disabled, and if login failures are not being monitored in event logs as an IoC, dictionary attacks are an effective attack vector for a threat actor. This is especially true if privileged accounts do not have the account lockout setting enabled as a mitigation strategy. Figure 10-3 illustrates the methodology used in a dictionary attack.

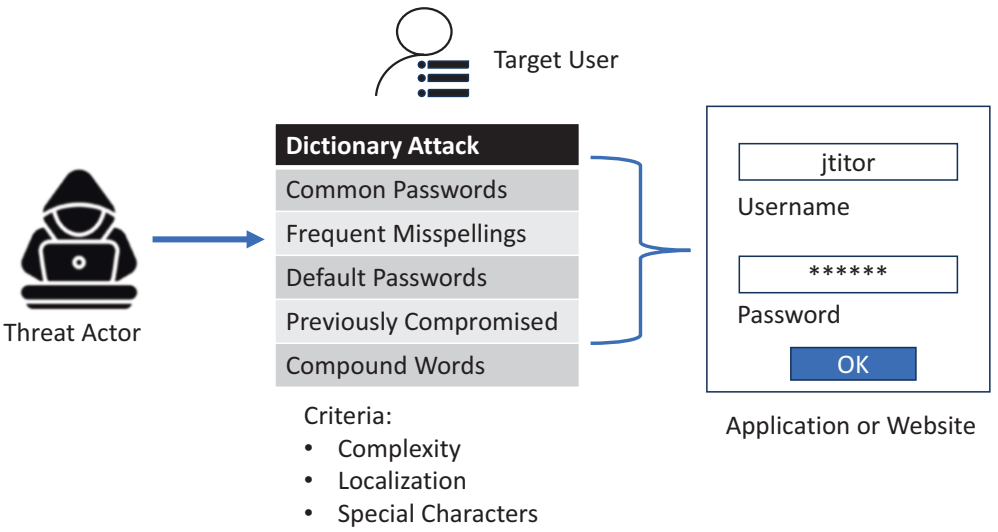


Figure 10-3. *A threat actor's approach to a dictionary attack*

Brute Force: Brute force password attacks are the least efficient method for trying to hack a password. It is generally the last resort based on mathematics. By definition, brute force password attacks utilize a programmatic method to try all the possible combinations for a password. This method is quite efficient for passwords that are short in string (character) length and complexity, but can become infeasible even for the fastest modern systems, with a password of eight characters or more. Therefore, if a password only has alphabetical characters, all in capitals or all in lowercase (not mixed), it would take 267(8,031,810,176) guesses (you have a better chance of winning the lottery!). This also assumes that the threat actor knows the length of the password. Other factors include numbers, case sensitivity, and other special characters in the localized language. The truth is a brute force attack with the proper parameters will always find the password. The problem is the time required may make the brute force test itself a moot point by the time it is done. And the time it takes to perform the attacks is not only based on the speed required to generate all the possible password permutations but also the challenge and response time of a failure on the target system. That last lag time is what really matters when trying to brute force a password, and similar to dictionary attacks, the IoC is looking for repetitive failed login attempts. Figure 10-4 illustrates a threat actor's attempt at brute force attacks.

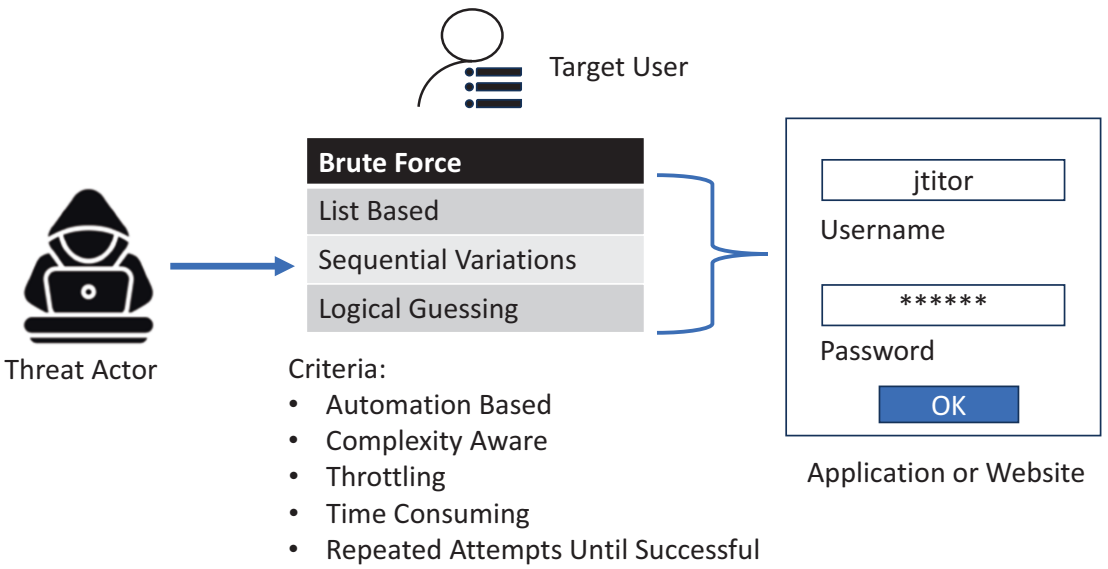


Figure 10-4. *Automation used for a brute force attack based on logical parameters*

Pass-the-Hash: Pass-the-Hash (PtH) is a hacking technique that allows an attacker to authenticate to a resource by using the underlying New Technology LAN Manager (NTLM) hash of a user's password, in lieu of using the account's actual password. After a threat actor obtains a valid username and hash for the password using a variety of techniques like scraping a system's active memory, they then can use the credentials to authenticate to a remote server or service using LM or NTLM authentication. The attack exploits an implementation weakness in the authentication protocol, where the password hash remains static for every session until the password itself is actually changed. PtH can be performed against almost any server or service accepting LM or NTLM authentication, regardless of whether the resource is using Windows, Unix, Linux, or any other operating system. To that end, modern systems can defend against this type of attack in a variety of ways, but based on the weakness itself, changing the password frequently (after every interactive session) is a good defense to keep the hash different between the sessions themselves. Password management solutions that can rotate passwords frequently or customize the security token are a good defense against this technique. Unfortunately, modern malware can contain techniques to scrape memory for hashes, making any active running user, application, service, or process a potential target. Once the hash is obtained, command and control or other automations allow for additional lateral movement or the exfiltration of data. Figure 10-5 illustrates the workflow associated with this attack vector.

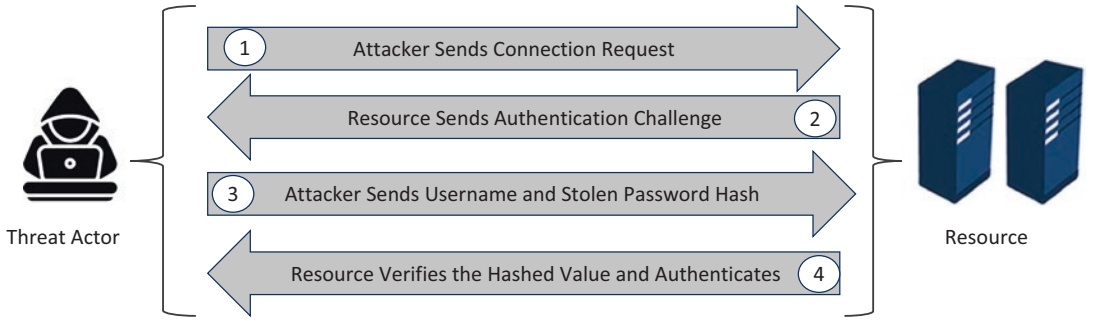


Figure 10-5. *Pass-the-Hash simplified workflow for a stolen hash used for authentication*

Security Questions: A common social technique used by financial institutions and merchants to verify a user against their account is to ask them security questions, challenging them to respond to private and personal information. They are required by many organizations when you set up a new account as a form of two-factor

authentication, and the correct answers are supplied during account creation. The end user is then prompted to respond to the security questions when logging in from a new resource, when you forget your password, or even when you reset your password. Some common security questions are these:

- The city you were born in?
- Your high school mascot?
- Your first car?
- Your favorite food?
- Your mother's maiden name?
- What was your first pet's name?
- Who was your first kiss?

However, these security questions themselves present potentially far-reaching risks. Think about these scenarios:

- How many people would know the answer to any of these questions?
- Are the answers to these publicly available online via social media, biographies, or even school records?
- Have you played any social media games that may have revealed this information?
- Have the security questions, and possibly their answers, been stolen in a previous breach?

The relationship is clear. The more places and people that know your security question answers, the more likely they can be answered by someone else. In addition, if the information is public, then it is really not a legitimate security question at all. When a resource requests that you complete and use security questions, the best recommendation is to use the most obscure questions that no one besides yourself may know, and remember never to share information that is similar online or with another site that uses the same security questions.

The scenario is similar to password reuse and social engineering. Security questions are social facts about yourself, and unfortunately, can be used on multiple sites. If someone invokes "Forget Password" on one resource and already owns your email or text message platform, and your security phrase is the same on multiple sites, the threat

actor can continue to own you through lateral movement between accounts associated with your identity. Making all your passwords different, using different accounts and emails for types of resources (banks, merchants, friends, and spam), and never using the same security questions will help keep a threat actor at bay from compromising your personal systems based on security questions.

Credential Stuffing: Credential stuffing is a type of automated hacking technique that utilizes stolen credentials comprised of lists of usernames (or email addresses) and the corresponding passwords (typically previously stolen from other data breaches) to gain unauthorized access into a system or resource. The technique generally involves large-scale automation to massively submit login requests directed against a web application and to capture successful login attempts for future exploitation. Credential stuffing attacks do not attempt to brute force or guess any passwords, the threat actor simply automates authentication based on previously discovered credentials using standard web automation tools. The result can be millions of attempts to determine where a user potentially reused their credentials on another website or application. Credential stuffing attacks prey on password reuse and are only effective because so many users reuse the same credential combinations across multiple sites. Figure 10-6 illustrates how credential stuffing works with previously stolen credentials and automation.

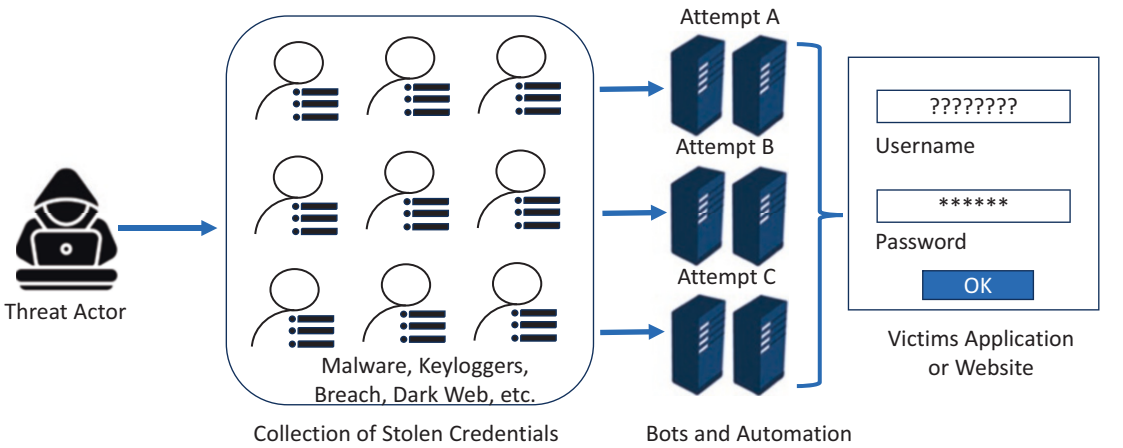


Figure 10-6. *Credential stuffing using bots for automation*

Password Spraying: Password spraying is a credential-based attack that attempts to access a large quantity of accounts by using a few common passwords. This is conceptually the opposite of a brute force password attack, which attempts to gain authorized access to a single account by pumping large quantities of passwords in

over and over. Brute force attempts, as discussed, can quickly result in the targeted account getting locked out. During a password spray attack, the threat actor attempts a single commonly used password (such as "12345678" or "Passw0rd") against many accounts before moving on to attempt a second password. Essentially, the threat actor tries every user account in their list with the same password before resetting the list and trying the next password. This technique allows the threat actor to remain undetected, avoid account lockouts, and avoid hacking detection on a single account due to the time between attempts. If poor password hygiene has been used by any one user or on any one account—human or nonhuman—then the threat actor has succeeded in authenticating the resource. The attack is compounded even further if any of these accounts are privileged. Figure 10-7 provides a simplified version of a spray attack against accounts that may be publicly accessible.

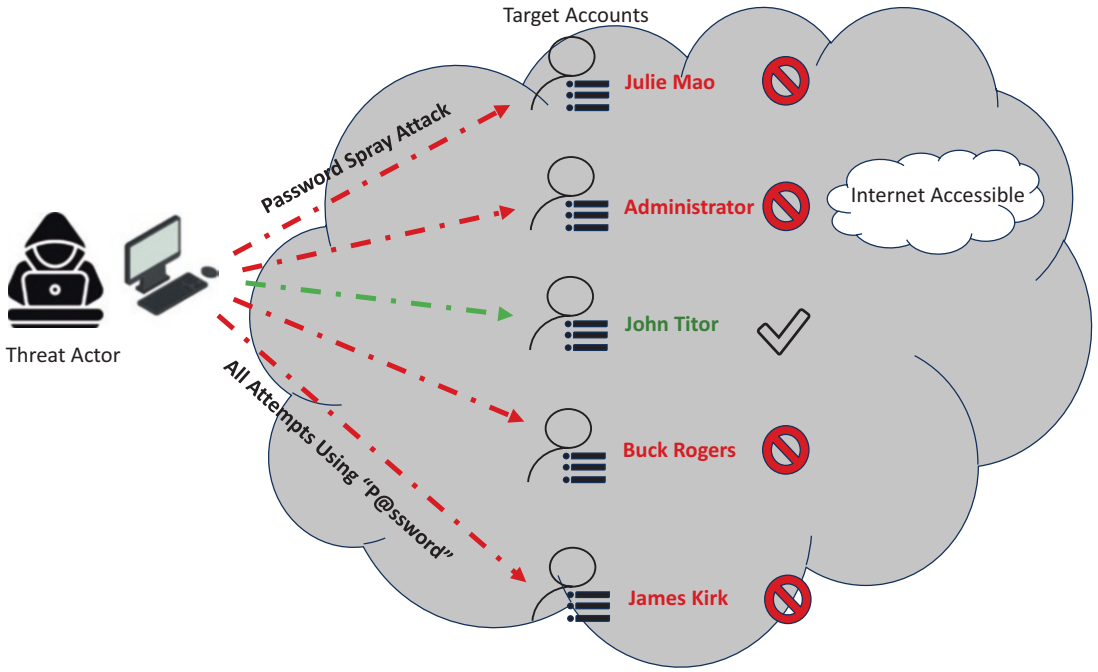


Figure 10-7. *Spray attack conducted on the Internet*

In the real world, password spray attacks typically are successful against cloud-based applications that are not monitored for failed login attempts. The best mitigation for these attacks is to enforce password complexity and multifactor authentication to every web-based resource. This is true for Single Sign-On (SSO) as well. SSO should never be implemented with only single-factor authentication.

Password Resets: How often do you change (not reset) your passwords? Every 30 or 90 days (about 3 months) when prompted to at work? How about at home? How often do you rotate passwords for your bank account or social media? Probably not often enough or never, and surprisingly, that might be acceptable. Without a password manager, keeping all of one's passwords unique, complex, and rotated frequently is a daunting task, even for the most seasoned security professional. One mental schema commonly used involves using the month, year, initials, and a few special characters with each password change so the pattern can be memorized. If the pattern is unique and not shared, the risk can be minimized, but it still allows for guessing since it is a repetitive pattern especially if the syntax is exposed.

Unfortunately, there is a common risk in resetting (not to be confused with changing) passwords that makes them targets for threat actors. Resetting a password is the act of a forced change of the password by someone else, not a change initiated by the user themselves. These risks include:

- Pattern-based passwords (as described earlier) when reset
- Passwords that are reset via email or text message and kept by the end user
- Passwords reset by the help desk that are reused every time a password reset is requested
- Automated password resets that are blindly given due to account lockouts
- Passwords that are verbally communicated and can be heard aloud

Anytime a password is reset, there is a silent acknowledgment that the old password is at risk and needs to be changed. Perhaps it was forgotten, expired, or triggered a lockout due to numerous failed attempts. The reset, transmission, and storage of the new password are a risk until the password is changed again by the end user, or worse, not changed by the end user at all. The password itself resides in the "ether" and the security of which is unknown. A threat actor can request a password reset once an identity has been compromised and then create their own credentials for the account. Anytime a user requests a password reset, the following best practices should always be implemented:

- The password should be truly random and meet the complexity requirements per business policy.
- The password should be changed by the end user after the first usage and require, if implemented, two-factor or multifactor authentication to validate.
- Password reset requests should always come from a secure location. Public websites for businesses (not personal) should never have Forgot Password links.
- Password resets via email assume the end user still has access to email to access the new password. If the email password itself requires resetting, another vehicle needs to be established, the preferred method being verbally on the telephone.
- Do not use SMS text messages, because they are not secure for sending password reset information.
- If possible, password resets should be ephemeral. That is, the password reset should only be active for a predefined duration. If the end user has not accessed the account again within the predefined amount of time, an account lockout will occur.

While changing passwords frequently is a security best practice for privileged accounts, resetting passwords and transmitting them through unsecure mediums is not. Performing frequent resets, and for a large number of users, represents a risk in itself since the initial reset password has been communicated using potentially unsecure techniques. For the individual, a simple password reset can be the difference between a threat actor trying to own your account and a legitimate reason the password needs to be reset. Businesses must be able to distinguish the threat from the legitimate need. And for standard end users without privileged access assigned to their account, the latest NIST[1] applewebdata://60701b47-7499-4f02-9ed5-0f544488089c/guidance does not require periodic password changes unless an indicator of compromise has been triggered.

[1] https://pages.nist.gov/800-63-3/

Token Hijacking: Token hijacking is often referred to as "token theft" or "session hijacking" and is an identity and session attack vector in which an unauthorized individual gains access to a user's authentication token or session identifier. These tokens are commonly used in web applications and APIs to authenticate and maintain user sessions to the cloud (SaaS, IaaS, or PaaS) or web-based on-premise applications. In a token hijacking attack, the attacker intercepts or steals the token, allowing them to impersonate the victim without needing to know their username or password. This can occur through various means, such as eavesdropping on insecure network communications, exploiting vulnerabilities in the application, stealing an HTTP Archive File (HAR), or tricking the user into revealing their token. Once in possession of the token, the attacker can access the victim's account and potentially perform malicious actions, including stealing sensitive data, initiating unauthorized transactions, and impersonate the account's parent identity. Protecting against token hijacking involves implementing robust security measures, like using secure communication protocols (e.g., HTTPS), regularly rotating tokens, filtering sensitive files that contain tokens, and employing strong authentication mechanisms. Figure 10-8 provides a simple token hijacking scenario for reference.

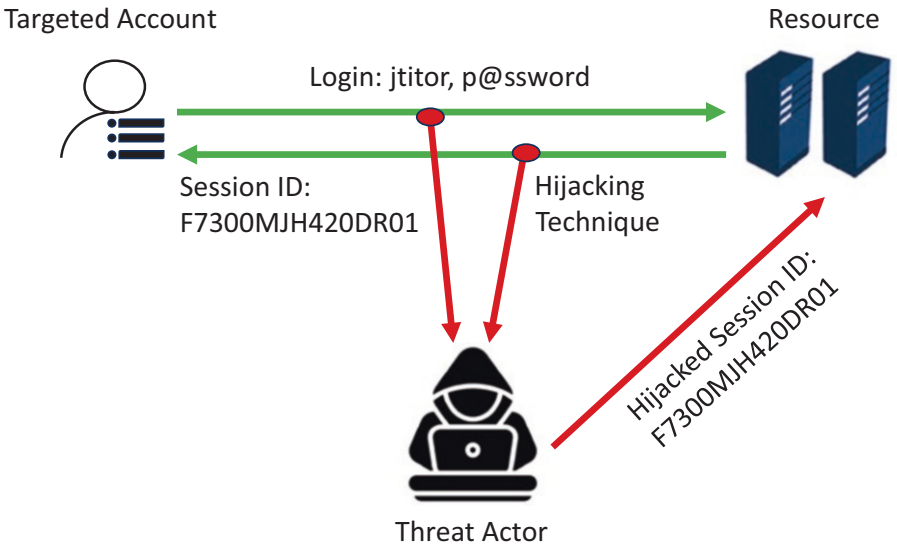


Figure 10-8. *Token hijacking where the technique for session ID theft has been simplified as a part of the workflow*

Identity-Based IoCs

Based on all these attack vectors and potential targeted roles, IoCs for identity theft can vary widely depending on the specific techniques and tactics employed by threat actors from nation-states to script kiddies. However, to simplify the lesson, the most common IoCs that organizations should be vigilant for when trying to detect an identity compromise are:

1. Unauthorized Access
    - Suspicious login attempts, especially from unfamiliar IP addresses or locations
    - Unusual or excessive failed login attempts on an account regardless of whether MFA is used or not
2. Phishing Emails
    - Emails with suspicious links or attachments, often designed to steal login credentials through techniques like watering hole attacks[2]
    - Emails requesting sensitive personal information, such as Social Security numbers, business banking account information for wire transfers, or even passwords
3. Social Engineering
    - Unusual requests for sensitive information over the phone, text messages, or other electronic communications media
    - Inconsistent or suspicious behavior from individuals seeking personal information regardless of whether their identity has been verified or not

[2] www.techtarget.com/searchsecurity/definition/watering-hole-attack

4. Account Takeovers
    - Notifications of password changes, address, or email address changes that the identity did not initiate
    - Unexpected email notifications or confirmations for new account logins or changes to account settings like a password reset that you did not request
5. Malware Infections
    - Suspicious software or files detected on your device, such as keyloggers or data-stealing Trojans that have been reported by your enterprise endpoint protection solutions
6. Unauthorized Transactions
    - Unrecognized charges to corporate business accounts or fraudulent payment requests for illegitimate services or products
    - Notifications of purchases, money transfers, gift cards, or withdrawals you didn't authorize for your organization
7. Identity Verification Requests
    - Unprompted requests for identity verification or confirmation of personal information from unknown or questionable sources. This includes services that claim to represent you or your business on behalf of a specific function.
8. Anomalies in Data Access
    - Sudden, unexplained changes in accounts, data, and other privileged information data requests
    - New account creation or system interaction that was not sanctioned by your organization
9. Social Media Abuses
    - Unfamiliar accounts or profiles impersonating your business on social media platforms

10. Data Breach Notifications
    - Alerts from companies, websites, services, or your business partners indicating that your personal information may have been compromised in a third party data breach
11. Fake Access Requests or Authorizations
    - Access request or access authorization approval that is submitted into the identity management system in order to gain nefarious access

It is essential that every employee stays vigilant and takes immediate action if they detect any of these indicators of compromise. Regularly monitoring your log files, enabling multifactor authentication, and practicing good cybersecurity hygiene can help reduce the risk of an identity security incident. As you can surmise, translating these into automated technology can be tricky, and in the future chapters, we will explore deeper what these IoCs look like from an identity security perspective and how these can be implemented within your organization. And as with every modern attack scenario, the complete list of hacking techniques far exceeds the confines of this book. Intentionally, we have listed the most common and easiest to understand based on the audience of this book. So, please do not think of this as an exhaustive list, but rather the best starting point everyone should engage with in understanding the threats.

CHAPTER 11

Identity Attack Vectors

An identity can be attacked by the person owning the identity or any of the related accounts and applications they are responsible for. As an attack surface, we need to think beyond traditional ports, protocols, and services found in information technology and consider the breadth and depth of identity, accounts, entitlements, and access. Identity attack vectors have a risk surface that is not only digital but also physical, and it can be compromised using old-school paper communications, such as letter from the postal service, or using social engineering via the telephone in a vishing (voice phishing) scam that leverages deepfake AI technology.

However, the point of an identity attack is fairly straightforward: a threat actor wants to find a method to compromise an identity and impersonate it for their own malicious intent. All they need to do is get access to one of your accounts to get started. If it is a privileged account, it is nearly game over from the start, especially if it is associated with SSO. The goal of the threat actor is to own you at the highest level of privileges and impersonate you. To be clear, they want to be an electronic imposter. The threat actor's primary goal is to disrupt the one-to-many relationship of a person to their identity and then compromise the integrity of the identity-to-account relationship. The risk surface encompasses all the methods to disrupt these relationships. This threat model applies to both physical and digital identities.

Once a threat actor can successfully impersonate you, they can authenticate with your accounts. This assumes your authorization has not been restricted, and they can move laterally to own your identity and its accounts in other systems. The attacker then assumes the ability to perform the tasks you are privileged to perform. By leveraging other attack vectors, the threat actor can potentially elevate their privileges to administrator or root. This is why mapping of accounts to identities is so important for understanding how they can be used for IoCs.

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_11

Methods

What methods does a threat actor leverage to steal your identity? In a digital world, they target your accounts. They steal the associated credentials via vulnerabilities and exploit combinations by attacking assets, or they leverage privileged attack vectors against available accounts via techniques like social engineering.

In the physical world, criminals attempt social engineering, mail fraud, stealing identification, or even tricking you to commit to verbal or written commands. While physical identity theft can result in fraudulent loans, credit cards opened under your name, and purchases made on your behalf, physical identity theft typically translates into an electronic form at some juncture in the attack chain. Only physical impersonations of claiming to be someone, such as by wearing their uniform and name badge, manifest purely in the physical world (physical persona). The damage a threat actor can perform can be severe if they also have stolen your electronic identity in the form of a smart card or door badge. Edward Snowden demonstrated the ramifications of this electronic identity attack, which he perpetrated while he was a trusted insider. He did not need to perform any physical impersonations to steal information as he owned the credentials. He did, however, also steal the credential from a colleague to enable privileged escalation and identity theft.

The case of Edward Snowden[1] demonstrates the inappropriate use of correctly assigned entitlements and highlights a significant area for further analysis and consideration based on an insider threat. The subject area of appropriate use, supported by usage and behavioral baselining and analysis, is an area that we will come back to many times in this book.

To that end, consider the following methods a threat actor will use to exploit an identity risk surface:

- Electronic
  - *Vulnerabilities and Exploits*: Software flaws that can lead to exploitation and ownership of an account
  - *Misconfigurations*: Poor configuration hygiene that can allow an attacker to hijack or create accounts

[1] www.whistleblowers.org/news/the-case-of-edward-snowden/

    - *Secrets-Based Attacks*: Credential and secrets attacks based on poor account hygiene that give a threat actor unauthorized access
    - *Privileged Attacks*: Credential and secrets attacks based on poor privileged account management that give a threat actor unrestricted administrator or root-based privileged access
    - *Social Engineering*: Broad electronic misuse to target a person to obtain sensitive information
- Physical
    - *Imposter*: A physical representation of another person for the sake of inappropriate access using role-based clothing and fake identification badges.
    - *Documentation*: False physical paperwork designed to invoke a state of compromise and mislead the target to provide information or access.
    - *Audible*: Verbal command or responsive social engineering, typically done over the phone or via an always-listening microphone, to capture sensitive information or grant unintended privileges. Artificial intelligence and deepfake technology have increased the risk for this type of attack.
    - *Biometric*: The theft and malicious implementation of biometric data to gain access or compromise additional datasets.

These are all basic classifications, but nonetheless form the basis of every breach we experience today. All of these can be linked back to an identity and used as attack vectors. Managing the digital aspects is the primary focus of every identity security solution today.

Tactics

Today, the favored tactic used by threat actors for bulk compromise of accounts is the Dark Web. This is where illicitly obtained information can be sold to other criminals in the form of raw data or even as a service to target accounts. This information may have details regarding a user's identity (like address and phone number). Fortunately,

the Dark Web information available today still rarely links an identity to the multiple accounts required to build a complete access profile. However, when that is done, threat actors can create fully synthetic identities. This is covered in a later chapter.

While individual attacks may be opportunistic or targeted, large-scale tactics are typically based on some form of financial gain and will target accounts that have already been compromised. The goal could be to steal additional data or start the process of a ransomware infection. Tactics for identity attack vectors can be performed using:

- The art of hacking one person at a time by any means available, including targeted acts with detailed information about the user.
- Bulk premediated attacks using techniques like credential stuffing or password spray attacks to compromise any vulnerable accounts, including faults in some MFA technologies.
- Targeting vendors, contractors, APIs, etc., that are available and accessible outside of the organization and vulnerable to an attack due to insecure credential practices, dormant accounts, or software vulnerabilities. Supply chain attacks against the wider supply chain can jeopardize the entire interconnected service delivery chain of almost everything we consume today.

All of these apply to both insider threats and external attacks. With these in mind, the most common methods threat actors use to compromise an account (covered in detail for the most common ones in the previous chapter) and compromise an identity are:

- *Interception*: Passwords are captured as they are transmitted electronically through email, the network, and even SMS texts. This includes SMS cloning, SIM jacking, or hijacking attacks.
- *Brute Force*: Automated guessing of passwords using dictionaries or other related password libraries that may target password reuse.
- *Searching*: The manual or electronic searching of passwords stored in insecure files, scripts, or another inappropriate electronic medium. This also falls under the category of sensitive unstructured data.
- *Manual Guessing*: Based on social engineering and knowledge of the identity, a threat actor will try to manually guess the password for an account.

- *Social Engineering*: Threat actors use human trust and social tricks to reveal identities and credentials or other sensitive information used for access.
- *Stealing Passwords*: The theft of passwords insecurely stored on paper or other non-electronic media. This could be as simple as posting a secure WiFi password on the whiteboard of a conference room.
- *Shoulder Surfing*: Physically observing someone typing in their password by looking over their shoulder.
- *MFA Fatigue*: MFA fatigue refers to the frustration and inconvenience experienced by users when constantly required to use multifactor authentication for online security, potentially leading to reduced compliance and security risks when the requests are triggered by a threat actor.
- *Key Logging*: Malware used to capture sensitive keystrokes, including passwords, as they are being entered and transmitted.

Once passwords have been stolen, they are used by the threat actor directly or placed on the Dark Web for a future sale. In reality, the Dark Web is nothing more than a collection of criminal sites that use service models to transact the purchase of credentials, secrets, tools, and data to leverage the stolen information. Regardless, the tactics are the same once the data is exposed; leverage it to steal an account and escalate to own an identity.

Threat actors from nation-states and other criminal entities may not rely on the Dark Web for their missions. They can operate as the sources of Dark Web data and can be actively engaged in hacking organizations to obtain illegal information. Criminal entities may build a long-term, persistent presence to achieve their goals while building a profile of identities for a future attack. These criminal enterprises are typically well funded. Their motives for identity theft are beyond a quick monetization of the stolen information. The discovery and reconciliation process within an IGA system is a very effective method for determining deviations in established business rules that could be used as IoCs for these types of persistent threats.

Implications

The implications of identity theft can be profound, disheartening, and even gruesome. The elderly (often targets of consumer identity theft) could have their entire financial savings depleted. For a business, it could mean the gross theft of intellectual property or, more profoundly, an event that causes major financial disruption to normal operations. These breaches have been in the news for years and are not expected to subside anytime soon.

Even the deceased can have their identities or accounts compromised in ways that make it difficult for their heirs to reconcile their estates. How do businesses even cope with an employee leaver situation when the trigger is unplanned, like the death of an active employee or a catastrophic event like 9/11? Organizations can find it difficult to manage these situations without a good governance solution, and if accounts and users are not managed after an unexpected event, the ramifications can be long-lasting for the health of the business. Looking for these abnormalities using ITDR is a good start when events trigger disruptions to normal business operations.

To illustrate how depraved a threat actor may act to achieve their goals, consider the following. A recent, far-reaching data breach in South Africa[2] contained all the typical traits of a government or credit reporting service compromise. Within the datasets themselves, however, was an interesting field not commonly seen in other datasets: "deceased_status." This personally identifiable information included whether the identity, and their associated account, was alive or dead. While it did not reveal the date of their death, it does open a very morbid question: "Can you hack the dead or compromise their identity?" The simple and grave answer is "yes." And the ramifications can be appalling if this type of trigger is not managed within a business. Was the identity of a deceased account still authenticated into the business? And if so, for what purpose? Was it their heir using the system or a threat actor?

Next, take this to a lower level of extreme identity security use cases. Consider a recently deceased individual and what might occur within their former employer or personal services:

[2] https://portswigger.net/daily-swig/pretty-much-the-entire-population-south-africa-hit-by-major-data-leak#:~:text=According%20to%20Hunt%2C%20the%20breach,to%20nine%20million%20deceased%20individuals.

1. Their bank accounts have not been closed or frozen nor has their employer's ability for direct deposit.
2. Social media sites may allow active postings and messages, including ones used for company business.
3. They probably still receive email at work and home, but no one can process them.
4. Their cell phones and landlines may still work, including voice mail.
5. They may not have loved ones immediately available to manage their estates.

All of these make their estate a prime target for identity theft. The cybercriminals could siphon off or even liquidate the deceased's assets since potentially, no one is monitoring their assets, services, and resources.

Considering the interconnected financial world we live in, hacking the dead may seem like a morbid topic, but facts suggest that, if we are not protecting an identity well for planned and unplanned triggers and organizational changes, the assets they have access to are at risk. The implications are true for any trigger, including illness, parental leave, sabbaticals, etc. All must be a part of your identity security model, or they will eventually result in unmonitored identity attack vectors. Therefore, we must not only consider daily operations for IAM and IGA but also all the edge cases that could lead to an incident. Hacking deceased employees is just one potential illustration of how morbid these edge use cases can be.

Privileges

Identity attack vectors have two real-world implications, regardless of methods and tactics. These will affect you regardless of whether you are a consumer or operating in a business entity. Both implications require identity theft and account penetration to conclude with "some type of access." After all, if someone compromises an account associated with your identity, you may never even know they own you or the business until it is too late.

To get started, assume your identity has been compromised. This means a threat actor(s) has access to your account(s). Whether these accounts are privileged, standard user, or guest implies how much damage they can inflict on your identity without additional attack vectors. In addition, how many accounts are compromised, and their financial or legal importance, also implies how much trouble it could be to undo the damage. This is true for a business or commercial account.

The privileges and entitlements your identity has are extremely relevant for a threat actor. For example, if you are a doctor, a threat actor could compromise even more sensitive data and personally identifiable information if your identity is stolen. A cybercriminal could gain access to your medical accounts and patient records. The privileges in this case only matter as much as the entitlements granted to you and the resources available to that account. As the doctor, you are not the administrator for the application itself, but typically would have privileges to retrieve sensitive information for all your patients, present and former. This makes your identity, accounts, and entitlements a valuable target and a high risk. In this scenario, the only higher risk would be presented by the system administrator for the application itself or the database owner for the solution. If either of their identities were stolen, not only would it be possible for the application to be at risk but all the data and all the users of that resource as well. Knowing who these account owners are, and monitoring access, is another key point in managing identity attack vectors.

Thus, this scenario again exemplifies why users should always be given the least amount of privilege needed for any identity to access a resource or application. Just managing with IGA does not mean everyone can be root. Enforcing least privilege is accomplished using privileged access management solutions and should be applied using roles to all identities and their associated accounts.

Regardless of its electronic demark for remote access, identities should always have at least three different types of account privileges. Remember, an identity can have multiple accounts, and each one should have the lowest form of privileges. While each of these can have granularity within them, a privileged account is typically the highest level of rights, while “none” contains no rights whatsoever and is actually lower than guest access. This is the first indicator of compromise. A threat actor directly gaining privileged access to an account assigned administrative privileges and owned by an important identity is the worst-case scenario for any organization. The second implication is the converse. A threat actor gaining access to a lower privileged account where they are required to use other attack vectors for lateral movement or to escalate privileges. The

latter can be managed by Endpoint Detection and Response (EDR) solutions, while the former cannot since it is typically modeled as authorized privileged activity. User behavioral analytics typically have a more difficult time interpreting malicious activity when valid credentials are being used and the account itself is already considered privileged. This was a key point in the Foreword of this book.

In the world of IGA, all accounts and their associated credentials can be placed under management, regardless of the privileges. Doing so helps mitigate the threats from both scenarios. In the world of Privileged Access Management (PAM), only accounts with administrative, root, or superuser privileges are typically placed under management. The latter, as we have concluded, is what threat actors seek. However, if they can gain access to even a lower-level privileged account, privileged attack vectors or asset attack vectors can be leveraged to elevate the threat actor's rights. This is how an incident can turn into a full-fledged breach. In Figure 11-1, this is illustrated as the privileged attack chain and shows how privileged escalation can occur as an identity crisis.

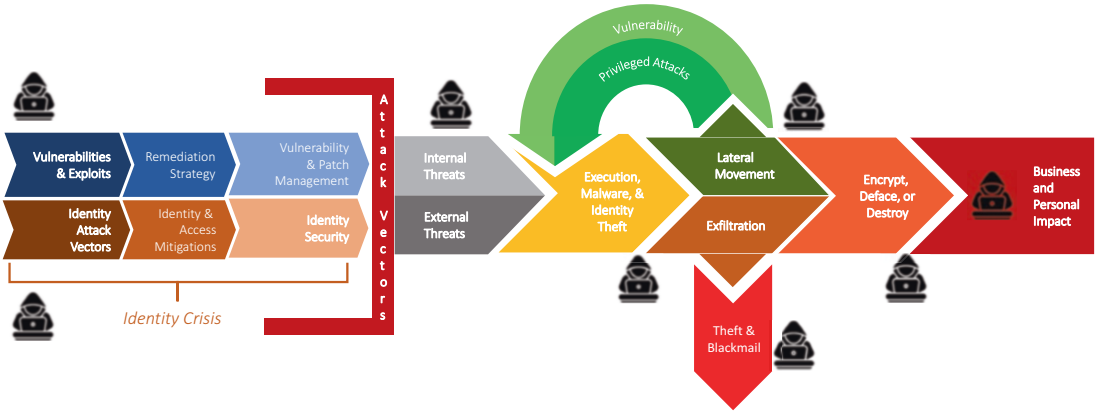


Figure 11-1. *Privileged attack chain*

On the top left of the diagram, the attack vectors can be attributed vulnerabilities, exploits, and unpatched systems before entering the attack chain. On the bottom left, identity attack vectors prevail that can compromise your identity security strategy and any ITDR you may have in place as part of your identity security philosophy. The rest of the diagram illustrates how any threat actor, internal or external, can leverage your environment with lateral movement and privileged escalation to exfiltrate data, deploy malware like ransomware, or damage your organization in numerous other ways.

With this in mind, here are the definitions for each user account type that can be assigned under IGA management to mitigate unwarranted privileged escalation and lateral movement:

- *Privileged User*: A privileged user is typically the administrator or root for a resource.
- *Super User*: A super user (or superuser) has elevated privileges in various graduations above a standard user but does not have full administrative capabilities.
- *Standard User*: A standard user is void of all elevated privileges except for normal runtime of a resource.
- *Guest*: A guest is typically the lowest form of access, which is usually less than a standard user. Interaction with a guest account only provides basic services.
- *Anonymous*: Controls access to specific resources only using an account with a null password or keys, which are not exposed to the end user.
- *Disabled*: A disabled account may have any level of privileges but is explicitly denied access and interaction with assigned resources.
- *None*: No privileges at all and may not even be defined as an identity or account.

In following with the principles of least privilege previously discussed, assigning standard user (or below) to all users is a mitigation strategy to prevent any lateral movement through the attack chain, and in many cases, deny privileges from malware from even executing. This is a crucial concept in preventing any identity-based attack vectors. If you have minimal privileges, rights, permissions, and entitlements, authentication and authorization for inappropriate resources can be mitigated.

CHAPTER 12

The Identity Cyber Kill Chain

It's obvious there are some problems in identity security to be solved, but the question remains: How do we better protect ourselves? Managing identities with IGA, IAM, and PAM solutions and following best practices really does make a difference to the security posture of an organization. To best understand how, we must first learn from our collective past mistakes and embrace the knowledge of others who have tamed the path. The best example of this is the NIST Cybersecurity Framework.[1] In fairness, this framework is a must read for all aspiring cybersecurity professionals and does provide a foundation for many of the concepts we will be discussing and impacts many regulatory compliance initiatives like GDPR and HIPAA.

Looking back over recent data breach reports and examining the forensic and post-incident analysis that is sometimes available, we notice two things relative to the discussion on identity management controls. The first is that threat sophistication is increasing at a rapid rate. The adversary is persistent, often well funded, and knows where best to attack and pursue persistence. The second is that these forensic reports clearly illuminate common faults that occur across many breaches. These identity management mistakes and process weaknesses are spread over the Cyber Kill Chain and encompass poor account controls, weak passwords, orphan accounts, dormant accounts, rogue accounts, weak inventory, poor cataloging of entitlements, and the over-assignment of user privileges. As we explore this material, we will reserve recommendations for preventive measures until the next chapter. The concepts presented in the following text have been empirically battle tested and need to be understood end to end before any mitigation strategies can be implemented.

[1] www.nist.gov/cyberframework

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_12

Part I: The Cyber Kill Chain

The Cyber Kill Chain[2] was introduced by Lockheed Martin in the late 1990s and documents the anatomy of a typical cyber breach by plotting the path of attack from start to finish. In many ways, the Cyber Kill Chain has stood as a cyber defense thinking reference model for 20+ years.

The privileged attack chain previously discussed is a modern subset derived from this model focusing solely on identity attack vectors. To better understand the steps a threat actor takes to breach an organization based on identity attacks, consider this expanded version of the attack chain, step by step.

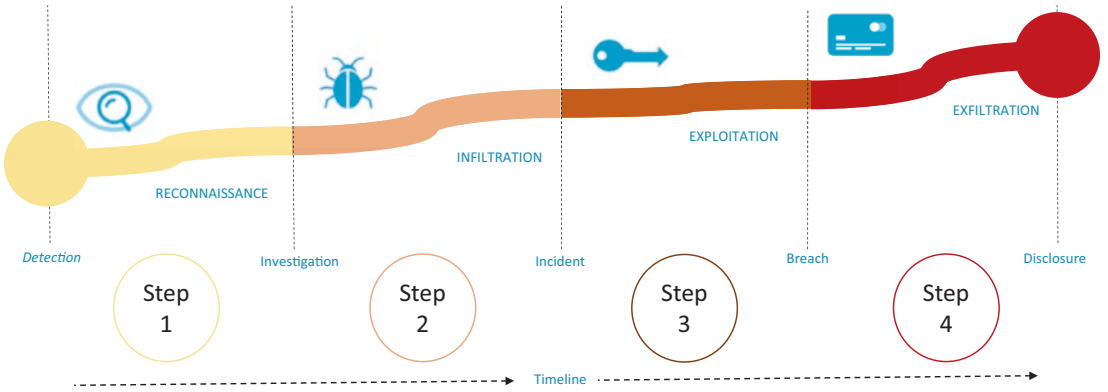


Figure 12-1. *Summary view of the phases of a typical cyberattack*

Figure 12-1 shows an adaptation of the Cyber Kill Chain to model and map phases of a fictional, but highly relative, identity-based cyberattack cycle. For our discussion, let us use this modified definition to highlight where weakness in identity security (and its controls) are often at fault. This goes beyond what was discussed as identity attack vectors, since it's focused on flaws in the processes themselves, vs. the potential ramifications already outlined in this book. Again, this is a fictional attack timeline, but its details are derived from several recent real-world breach reports.

[2] www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html#:~:text=Developed%20by%20Lockheed%20Martin%2C%20the,order%20to%20achieve%20their%20objective

Reconnaissance

For this breach, things start as they often do—with active reconnaissance. The elements of this attack phase are summarized in Figure 12-2. The first phase of the Cyber Kill Chain is all about the adversary gaining knowledge about the target and how best to attack it. As a starting tactic, the attacker will often begin scanning all the externally available resources the target enterprise uses.

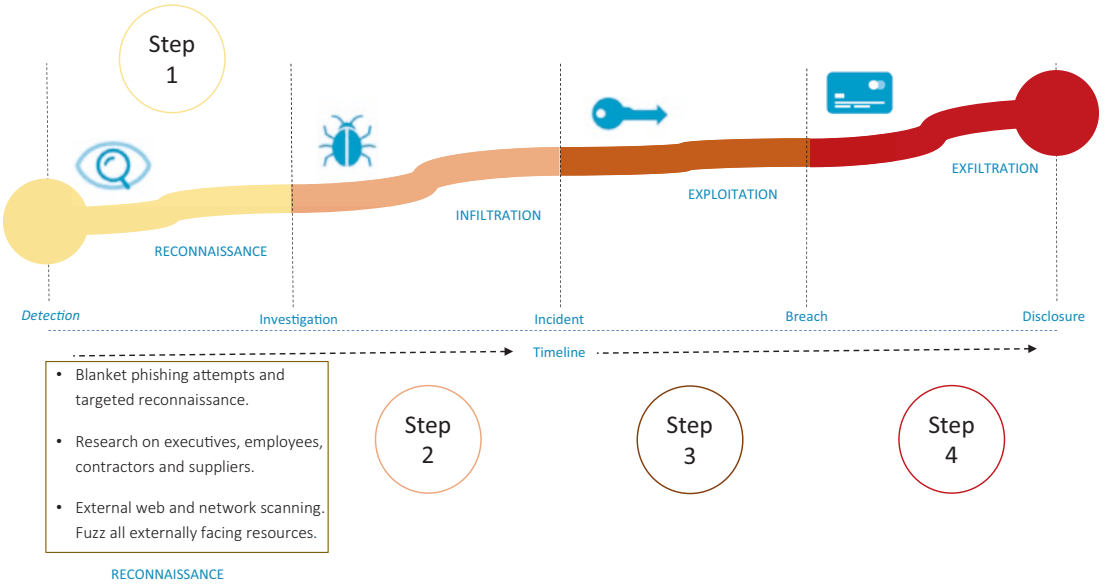


Figure 12-2. *Reconnaissance phase IAM weaknesses*

This phase is also when social engineering begins. Anyone and everyone connected with the company—employees, customers, vendors, partners, etc.—will be researched, and those with potential access into the enterprise will be sent blanket phishing emails. Executives and other high-value targets will be subject to spear-phishing or whaling campaigns.

Infiltration

Infiltration is the next phase of our Kill Chain. This is summarized in Figure 12-3. With ever-increasing numbers of people accessing our systems and data, eventually someone always makes a mistake. Often, that mistake is simply clicking the wrong link in the wrong email. Spear-phishing has become sophisticated and so widespread that in our

scenario, an executive clicked a link that downloaded some "drive-by" malware onto their computer. With that, the local admin account could be compromised, granting the bad guys access to a range of enterprise network resources. With local admin access, the attacker could move laterally with virtually no limits to the organization's servers, install exploitative software on the network, and begin scanning systems for any and all information that may be worth something.

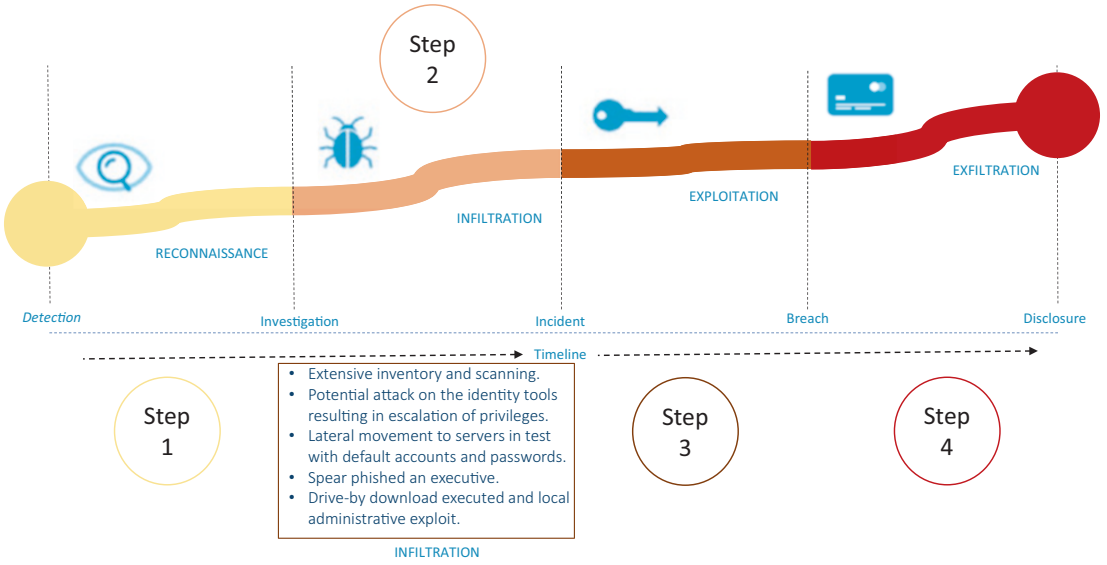


Figure 12-3. *Infiltration phase IAM weaknesses*

Exploitation

During this phase of the attack, the attackers have found their way into several resources and are looking for the best pathways to gain higher levels of privilege and access to the most valuable data resources. This phase of the attack is highlighted in Figure 12-4. Typically, threat actors will launch brute force password attacks on administrative accounts or other accounts to crack into the account and perform lateral movement. Brute force attacks can be effective at cracking the weakest passwords. In addition, attacks on logical business processes, such as access requests, will help the threat actor attempt to impersonate a user in the workflow. When a threat actor has gained access and can compromise communications like email, messages are easily spoofed and incorrect access is often granted. All of these are identity attack vectors that can be waged against the IGA system itself, without any visibility by security teams.

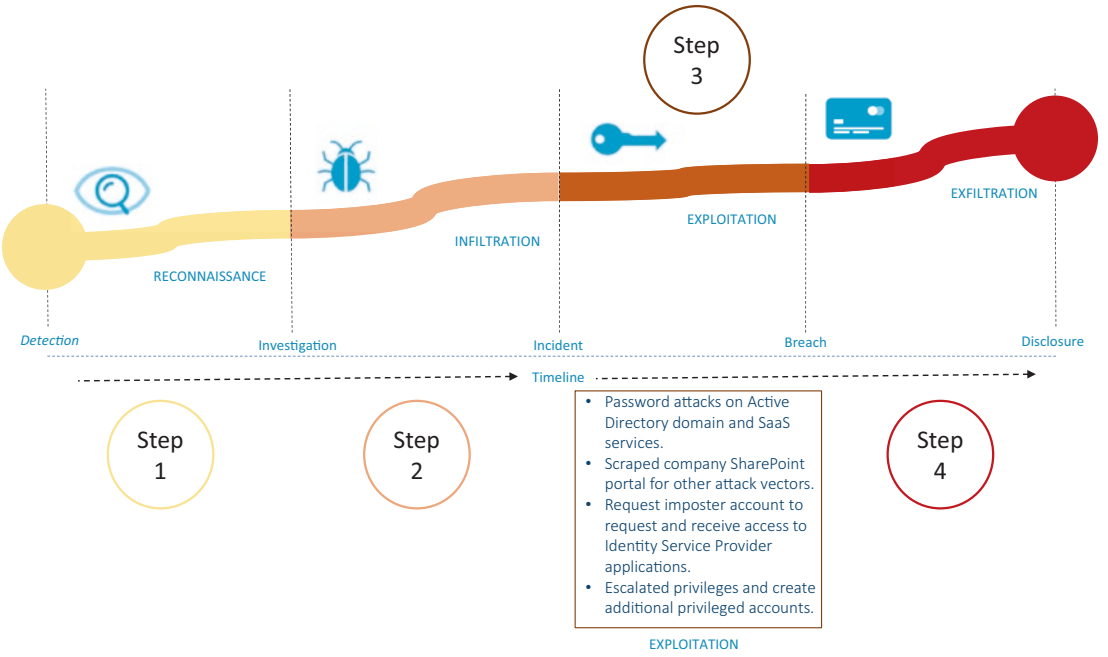


Figure 12-4. *Exploitation phase IAM weaknesses*

Exfiltration

Once a good set of target systems are owned, the last phase of the Kill Chain is the exfiltration of data. This is shown in Figure 12-5. This usually involves downloading password databases for internal systems, the collection of customer data, the theft of intellectual property, or the deployment of additional malware. It is not uncommon to see the archiving and removal a large number of files from internal and cloud file storage systems too. And lastly, in modern attacks, the threats of ransomware to hold the organization hostage have shown to be extremely profitable for organized cybercriminals and a huge liability for businesses and cyber insurance carriers (more on this later).

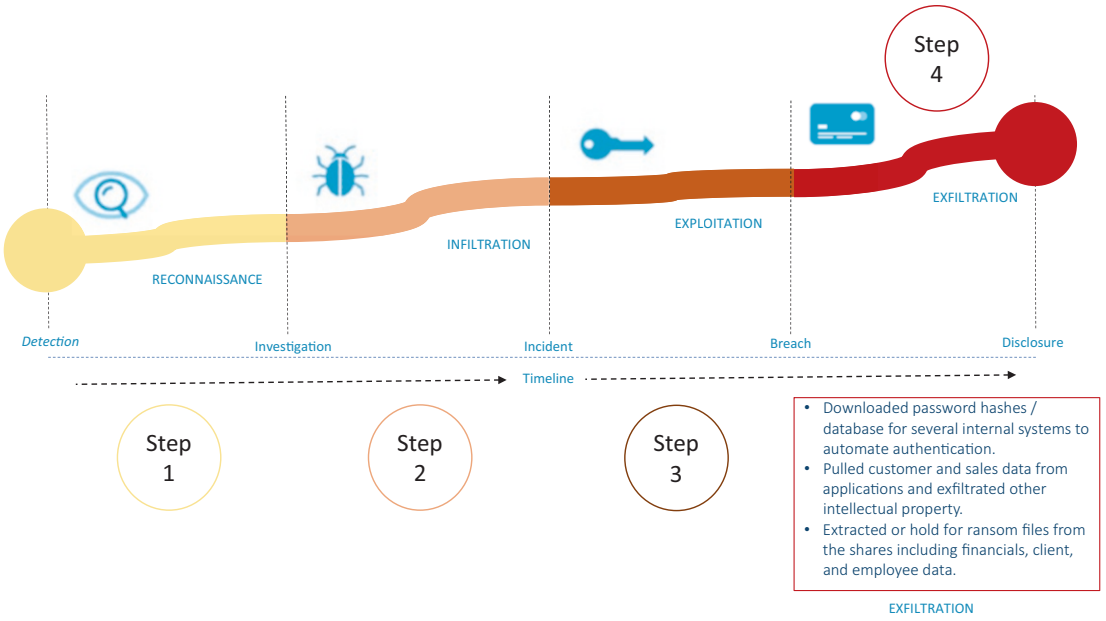


Figure 12-5. *Exploitation phase IAM weaknesses*

This leads to a startling conclusion based on the Cyber Kill Chain: let your IGA implementation and the technology solution manage itself and all other IAM and PAM tools. To the business, they are just another application. Where applicable, control it just like you would manage any other asset, application, and resource. This includes everything from daily usage for RBAC to functional accounts. IGA should not be the source of weakness in your organization, despite its management power. It can govern itself, too, and be your first indication with ITDR that an identity has been attacked or compromised.

Part II: Real-World Identity Attack

While Part I focused on the common attack vectors most organizations face, Part II highlights a real-world case study that could have had devastating results for an organization. While the innocent (names, companies, and vendors) has been sanitized from this analysis, modern identity attacks share one thing in common in almost every scenario: humans are the weakest link, and their mistakes ultimately lead to a breach. So while Part I is theoretical, based on common attacks, this section highlights how bad it can really get.

During a company's daily monitoring, an individual's account activity was observed performing novel Single Sign-On (SSO) behavior for their role:

- From an IP address in a foreign geolocation not serviced by the company, the account performed successful API (application programming interface) calls without an authentication event.
- The end-user Identity and Access Management agent reported: "(Mozilla/5 (Macintosh; Mac OS X 10) AppleWebKit (KHTML, like Gecko) Chrome/99 Safari/537)" as the browser and operating system.
- The account "jtitor@timetraveler.io" is observed attempting to log in to the SSO Administrative Console. The authentication failed due to the lack of a valid MFA request.
- Following the authentication failure, the threat actor continued to leverage the API to create an account called "service_network@timetraveler.io" since the API session did not enforce policy for MFA. This is actually a design flaw in the vendor's SSO solution.
- Using the API, the threat actor escalated the privileges of this account by assigning the malicious account the Service Desk Administrator role.
- The threat actor is then observed giving the "service_network@timetraveler.io" membership to the SSO Administrator Console application.
- This account is also pushed to the external application SSO Help Center.
- Identification of a new administrative account and changes in membership were detected by the Security Operations Center, and containment and remediation occurred within a few hours. Dwell time was less than one day.
- This account was deactivated and removed by the security team from the SSO application.
- At this point, forensics was started to determine the identity attack vector and confirm no additional persistence.

- The identity's asset, "Laptop-JTitor," was quarantined using the installed EDR (Endpoint Detection and Response) solution. The end user's daily driver is based on macOS mimicking the original end-user agent request to obfuscate whether the asset was real or not. Next, the end user was contacted and interviewed regarding the incident to help identify any anomalous behavior or phishing attempts.
- After extensive investigations, a HAR (HTTP Archive) file with SSO Cookies was found on the end-user's laptop. This file was created just before the malicious event began.
- The HAR (HTTP Archive) file was associated with a support request for the SSO vendor themselves and was generated and uploaded to the support case earlier in the day.
- The daily driver (standard user) account for "John Titor" has administrative rights within the SSO solution. This is needed to manage support tickets.
- The HAR file did contain a session hash that could authenticate via the API, but not the user interface, until it expired. This is how API access allowed the creation of a new account but failed via the user interface.
- The threat actor obtained the HAR file via only three possible attack vectors:

    a. An insider threat at the SSO vendor accessed the HAR file. This was either a trusted insider or the vendor compromised themselves (probable, due to the timeline).

    b. The end user's laptop was compromised and allowed exfiltration of the HAR file (unlikely, due to there being no other indicators of compromise).

    c. A vulnerability allowed external access to the HAR file via poor configuration or exploit via the upload URL (a stretch since the ticketing system was a popular SaaS vendor).

While some read this timeline and question, "how?", the point is that this real-world attack was based on identity attack vectors. A trusted user was compromised, and their behavior to create an HAR was not unusual in support of a service ticket. This is why the introduction to this book was so important. The user was trusted, honest, truthful, and behaving in a way that caused no question about their integrity. However, the compromise of the HAR file locally, at the third-party supplier or via an exploitable vulnerability, allowed a valid session token to impersonate the user as a masquerade attack (discussed later), with administrative privileges, and to execute API commands. The creation of the account, changing of roles, and the attempted login to an administrative session without MFA set off every alarm possible within the organization. And for those that really care, the actual root cause was that the third-party vendor compromised themselves,[3] and the root cause was single-factor authentication stored in a synchronized personal password management browser account.[4] Your security is only as good as the weakest link, or in this case, the weakest link of your supplier.

This helps prove even further the need to embrace identity threat detection and response (ITDR) to mitigate this type of attack vector. Therefore, what type of detections could have helped identify this type of identity attack and trip "every alarm," regardless of the actual root cause? Ask your business if you have a security solution that can detect

- A new SSO or account management synchronization agent registered out of band or without proper change control
- An account that has changed critical identity infrastructure or a sensitive configuration setting immediately after an MFA request
- An MFA factor that was added to and then removed from any account within 6 months
- A new identity provider that has been enrolled within your identity services or SSO environment
- An administrator that has multiple phone numbers or physical devices as active MFA factors for administrative accounts
- An account that has had all MFA factors reset from an anomalous IP or an inappropriate geolocation

[3] www.beyondtrust.com/blog/entry/okta-support-unit-breach

[4] https://sec.okta.com/harfiles

- An administrative account that is performing administrative actions using a proxy
- An administrative account that is being used without MFA

With a real-world identity attack vector, timely detection and the minimization of dwell time are critical. This attack had multiple mitigations to ensure it did not happen again, but the most important lesson is to always protect administrative accounts from any type of misuse or tampering.

Part III: Identities Under Attack

Once again, attacks against privileged users are hitting the headlines with some high-profile breaches and a warning from IAM vendors that a threat actor is using social engineering to attain access to highly privileged "super admin" roles in licensed customer tenants. While attacks against privileged users are a tale as old as tech, there are important lessons to learn about how modern attacks against identities and identity infrastructure work, and more importantly, how they can be mitigated.

Let us start by breaking down some of the common Tactics, Techniques, and Procedures (TTPs) that have been observed in recent attacks by threat actors, such as ALPHV,[5] Scattered Spider,[6] LAPSUS$,[7] and others. Recently, there have been multiple reports of coordinated attacks against SSO customers. Figure 12-6 describes how that attack flow works.

[5] https://securityscorecard.com/research/deep-dive-into-alphv-blackcat-ransomware/
[6] www.darkreading.com/attacks-breaches/-scattered-spider-mgm-cyberattack-casinos
[7] www.cisa.gov/sites/default/files/2023-08/CSRB_Lapsus%24_508c.pdf

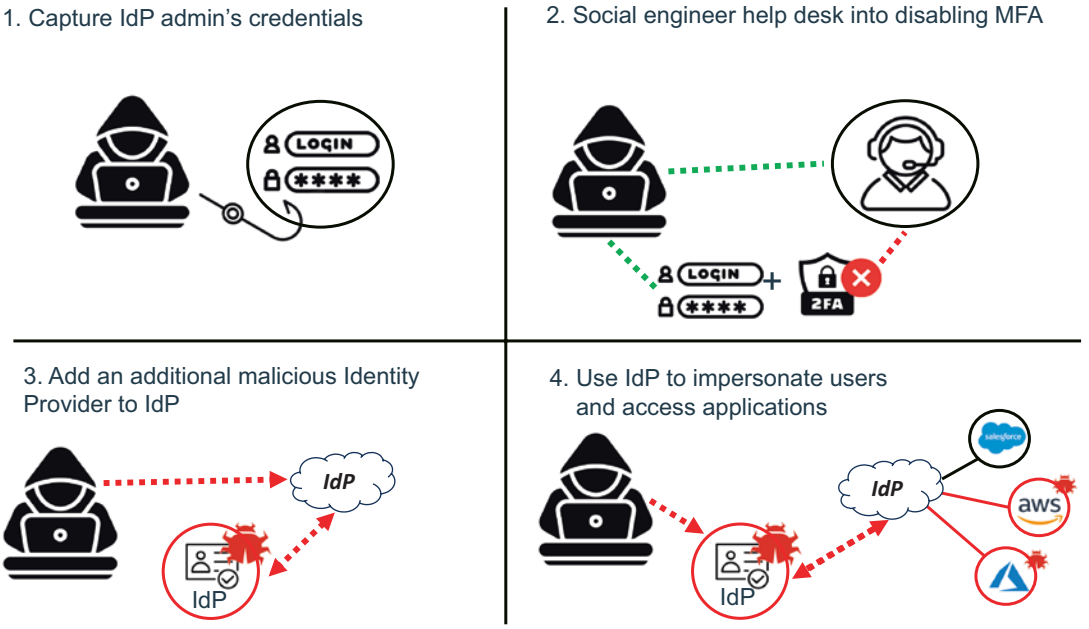


Figure 12-6. *IAM and IdP (identity providers) attack vector flow*

Attackers leveraging identity provider (IdP) generally begin by targeting user accounts with high levels of privilege—specifically, the super administrator permissions. The attackers are able to gain the passwords of privileged user accounts, likely via phishing, device compromise, or insider threat. However, when MFA is implemented, passwords alone are generally not enough to authenticate into the accounts.

And while organizations have become much better at protecting accounts using MFA, this alone does not deter a determined attacker. Weaker forms of MFA can be phished, subjected to MFA fatigue/MFA bombing, or even to SIM swapping attacks, leaving organizations vulnerable.

Even when phishing-resistant MFA, such as FIDO2, is used, attackers are now turning to help desk technicians, who can be socially engineered into resetting an account. This exploitation of the account recovery process means an attacker armed with some basic information about a user can request the account MFA be reset. Once the help desk is socially engineered to reset the account, the attacker can remove MFA entirely or enroll a device that is under the attacker's control as the second factor. In extreme cases, an attacker might even be able to get both the password and the MFA reset by literally talking their way around your security defenses. Threat actors, such as LAPSUS$, have successfully used this technique against a range of organizations and accounts.

Now that the attacker has gained access to a super administrator, organization administrator, or a custom administrative role with sufficient privilege, they want to be able to move laterally, elevate privileges, and maintain persistence by establishing their own backdoor into the environment. To achieve this, they abuse identity federation by adding in their own malicious identity provider and potentially using SSO to gain unfederated access to all the applications in the environment.

The intent of inbound federation provided by an IdP and SSO vendor is to quickly allow users to sign into applications using the IdP that part of the business already uses, reducing the friction of having to create new accounts. While this feature makes it easier for organizations to bring together multiple subsidiary companies that might all use different IdPs (identity providers) but need to access common business applications, it also opens up a wealth of access to an attacker who is able to add their own IdP. This is why the ability to add a new IdP is restricted to the highly privileged super administrator accounts, which are the ones targeted in these attacks.

With an IdP under the attacker's control, the attackers can impersonate users and gain access to applications downstream. This is done by configuring their IdP to act as an "impersonation app," allowing an attacker-controlled account to impersonate any legitimate user account. This allows the attacker to SSO into applications by posing as any legitimate account.

Because the attacker has compromised the identity infrastructure using inbound federation trust relationships (also known as "Org2Org"), many common breach responses will be ineffective. For example, if you suspect a legitimate user has been compromised and reset their credentials, this will not prevent the attacker from continuing to impersonate them. This is why having the ability to identify these types of risks and attacks as early as possible is critical.

With this in mind, there are a few additional identity attack techniques threat actors have recently leveraged:

- *Anonymizing Proxy Services*: These allow an attacker to mask their true location and make it appear like they are using an ISP that is close to the user whose account they are targeting. This tactic allows the attacker to blend in, avoiding triggering impossible travel (no way to physically get from one location to another in a short period of time) or other IP geolocation controls. In the recent attacks, accounts and devices not previously associated with an account were observed to mimic the valid account's identity to make identification more difficult.

- *Cloud Connectors*: The abuse of on-premises connector agents and the cloud is another key target in the identity infrastructure. Some alleged threat actors involved in recent high-profile attacks have specifically referenced targeting agents used for password and credential synchronization. These agents are used to sync account information, including passwords, between Active Directory on-premises and cloud applications. This minimizes user friction, but it also makes the agents a tempting target for attackers seeking to compromise them and capture credentials in bulk.

These kinds of attacks are not new, but they are increasing. If we look back to the SolarWinds attack of 2020, we can see how the attackers used a similar approach. In that case, the attackers modified trust domains in Azure AD (now known as Microsoft Entra ID), so they could add a new federated identity provider (IdP) that was under their control. This enabled them to create a backdoor into Azure that allowed for widespread access.

In other cases, we observed the SolarWinds attackers compromising the credentials of high-privilege on-premise accounts that were synced to Microsoft 365 (formerly Microsoft Office 365). From there, they were able to exploit cloud apps by adding their own rogue credentials that could be used to exploit legitimate permissions assigned to the cloud application.

Overall, these techniques are very powerful, not only because they can be hard to detect but also because they can allow the attacker to effectively bypass the multifactor authentication controls in place. The classic organizational playbooks of resetting passwords and enforcing MFA can't remediate against compromised identity infrastructure, so it is vitally important to be aware of these techniques and gain visibility around where you may be vulnerable to them. These are identity attack vectors that can bypass almost all traditional security controls.

Let's explore the best defense and mitigation strategies you can set up against this type of modern identity attacker:

- *Factor in Phishing Resistance*: Using phishing-resistant authentication, such as FIDO2, is highly recommended, especially on any type of privileged account. Remember, not all forms of MFA (multifactor authentication) are equal. We have seen phone-based MFA present little-to-no defense against threat actors in recent attacks. Even time-based one-time passwords (OTPs) can be targeted

by modern phishing kits or token hijacking attacks. Sign-in policies and conditional access policies should be in place to ensure that users must reauthenticate from the right device, location, or network to conduct privileged activities.

- *Implement and Enforce Least Privilege:* Understand where privileged roles exist and try to limit privileges to just those who absolutely need them. As probably the biggest example of this, does your help desk absolutely need the ability to reset the passwords of a limited number of super administrators in your identity infrastructure? Often, accounts are granted privileges for a one-time task, such as "adding a new HR application," but after the task is completed, the privileges are never reviewed or removed. Privilege creep represents a real risk to the business while providing no real value. So review, reduce, and remove privileged access wherever possible.
- *Perform Privilege Audits*: Review your internal processes and consider "who, why, and how" for processes involving account password or MFA resets that an attacker could exploit. Pay extra attention to any accounts with privileges. These technically could be privileged accounts, like a super administrator, or privileged accounts in the context of the business, such as members of the management team. Often, privileges can be quite subtle, and it is assumed that certain roles just need a level of privilege or to be members of a group to grant privileges and access to systems. It is important to minimize assumptions and ensure that you understand what you are granting and why it is absolutely needed. And to add misery to the scope of this problem, this needs to be done for all humans, machines, and APIs throughout the entire environment to help prevent an incident. As mentioned in Part II of this section, APIs are not all created equally, and many do not honor policies like MFA.

With a rising number of identity security attacks that exploit the gaps in visibility between IAM tools and traditional security tooling, it is vital to start thinking about how you can protect your organization. Identities, especially those with significant privileges, are at the heart of modern attacks as attackers turn to identities and social engineering rather than exploits and malware.

Shifting to deal with a world of identity security not only requires tools that can help you gain visibility and control of identities and privileges, reduce risk, and detect threats; it also requires an organizational shift to bridge the worlds of identity, access, and security. Super administrators may be hitting the headlines today, but they are only one part of a broader ecosystem we need to protect.

Part IV: Old School

There is a trend among all generations to go vintage. That is the love, desire, and utilization of something many would consider antiques but still fully functional and practical in today's modern digital world. While the term vintage can be applied to anything from clothing to old gaming consoles, threat actors can leverage vintage attack vectors to circumvent even the best identity-based security controls. This section covers an old-school identity vector that can still be devastating to any business, employee, or individual via email.

To begin, let's set up the scenario. A business user, shared email account, or a personal account is compromised using single-factor authentication or another more advanced attack vector like SIM hijacking. What is important is that the user's email has been compromised and a threat actor has access. Once in, the threat actor can:

- Access any email sent, deleted, or stored in the email box.
- Delete or send any email of their own potentially to any recipient.
- Access any contacts stored in the system for additional social engineering, including personally identifiable information such as phone numbers, addresses, birthdates, etc.
- Depending on the email service, have access to your calendar and any appointments past or present.
- Change the configuration of your email, including resetting your password.
- Leverage your email to perform password resets on other applications like banking or social media.
- Request a one-time password PIN for any websites that use multifactor authentication via email.

Once an organization or person detects an email compromise, their first reaction is to reset their password. In many cases, this may work if you are only using single-factor authentication. However, for many unsuspecting individuals, this may fail to work, and their email continues to be abused.

The next obvious step is to enable multifactor authentication or even application-specific passwords. In some cases, this may resolve the issue, but even creating unique passwords for your email application using the provider's configuration settings (application-specific passwords[8]) may also continue to not remediate the threat. The organization has followed best practices to block the threat actor, but even password resets and MFA are not blocking their intrusion. The question is how and what can be done next?

This is where we all need to stay vigilant about vintage attack vectors. Just because something is old does not mean it does not work. Vulnerabilities and exploits from years past with CVEs[9] (Common Vulnerabilities and Exposures) from decades ago are still relevant, and if an old system is exposed, then it can still be exploited. Email is no different if inappropriate configuration settings have been set.

In this scenario, the first setting the threat actor performed was to modify the configuration of the email box to forward all email to another account they owned, typically outside of the organization. This is not a privileged setting in most email systems and was only applied to the single account after it was compromised. The forwarded email allowed a persistent presence into the account even when the password changed, and MFA was enabled. This allowed the threat actor to monitor all activity, continue to have access to sensitive systems by deleting PIN or password reset emails after they were received, and exfiltrate sensitive information as desired.

A simple feature added for convenience many years ago has become a vintage attack vector that can provide persistence even when all modern best practices have been enabled. Therefore, the lesson learned from this section is simple: do not forget about the attack vectors of the past that are simple to execute and have huge rewards. Threat actors almost always look for the path of least resistance, and something as simple as an email forwarding setting can lead to a monumental identity security threat using simple old-school tactics, and no malware was needed. Vintage is cool and vintage attack vectors, especially for identity security, should never be overlooked in your planning your organization's mitigation strategies.

[8] https://support.google.com/accounts/answer/185833?hl=en

[9] https://cve.mitre.org

CHAPTER 13

Six Steps to Identity Security

Identity security is the risk management for *everything* included in Identity Access and Management. This chapter was written to help identify the core areas of identity security, presenting each key capability that you, as a security professional, should implement across your entire identity fabric and ensure your IAM vendors are aligned to providing information to identify identity threats. The illustration in Figure 13-1 is designed to help you visualize these recommendations as a part of your organization's identity security strategy based on all the concepts we have already discussed.

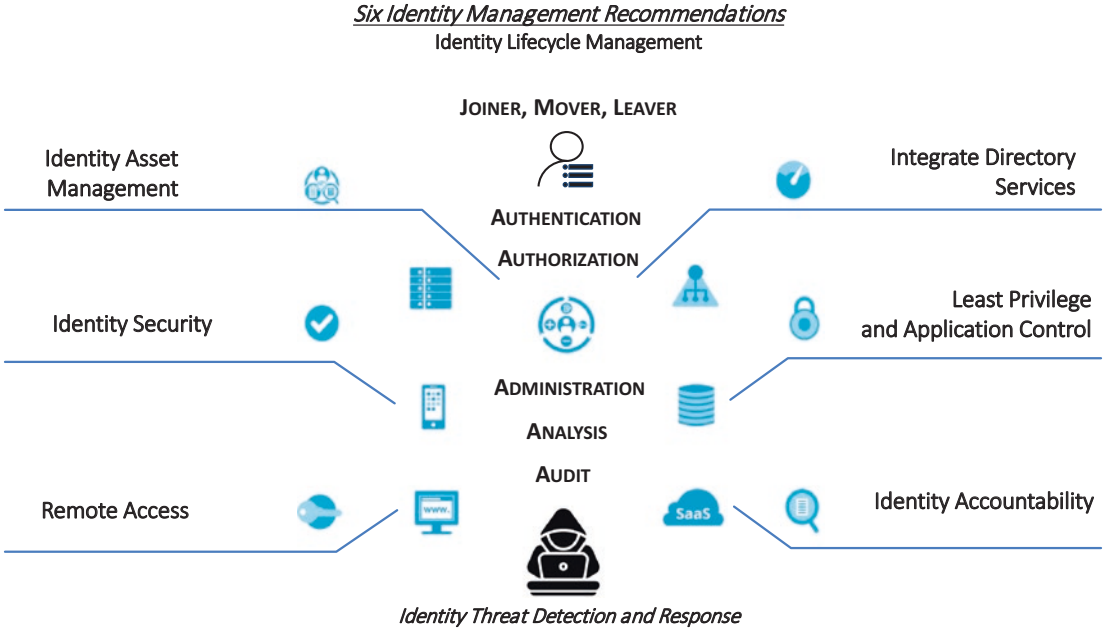


Figure 13-1. *Recommendations for your organization's overall identity security strategy*

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_13

When implemented, each core area will give you greater control and accountability over the identities, accounts, assets, users, systems, and activity that comprise your privilege and non-privileged environment while eliminating and mitigating many attack vectors. You can address these areas all at once or more commonly, phase in controls for one or several areas of IAM at a time. The more of these areas you implement, the more synergies you will see, and the more impactful the reduction in enterprise risk and operational improvements.

Throughout the process of selecting and deploying your identity security solutions, keep in mind these business requirements. They will help you articulate the value of this program higher in the organization:

- *Total Cost of Ownership*: Does it result in time-savings (such as replacing manual processes with automation) and allow you to redeploy resources for other initiatives?
- *Time-to-Value*: How soon does it help you measurably improve security controls and dial down risk? How long will it take to achieve your end-state goals with the solution?
- *Integrations*: How does it integrate with the rest of your security ecosystem (IAM, SIEM, service desk, analytics)? Does it help you make better decisions on risk? If it only works well as a stand-alone/point solution, it's probably only a stopgap vs. a long-term solution. On the other hand, if the solution has synergies with your existing security solutions, it will also help you maximize existing investments.
- *Longevity*: Will the solution vendor grow with you or even pull you toward growth through security enablement? Is the vendor resourced to evolve capabilities and deepen feature-richness to meet the PAM use cases of tomorrow?

1. Identity and Asset Management

There is a traditional business axiom that states, "You cannot monitor, manage, or measure something you do not know anything about." This is gospel when managing identity security. There is no way to manage and monitor the risks of any type of identity

if you do not know if it exists and where it can be used. The latter implies you know all about the assets and resources in your environment, including the cloud. Now, step back and reread the title for this first recommendation. It is "identity and *asset* management," not "…access management." This is key for our discussion and reflective of the Venn diagram earlier in the book.

So, how does one go about getting started with step one? Electronic discovery of all identities, accounts, privileges, entitlements, rights, and permissions is the preferred approach. While the task fits easily in one sentence, empirically, it can be very difficult to perform and maintain an accurate accounting at any given time.

Answer: Solicit vendors and open source solutions that can perform comprehensive inventory functions in as near to real time as possible. Ensure that they are not just performing network-based scanning but that they also have extensive connector technologies into other IAM, IGA, IdP, MFA, SSO, and PAM solutions.

Based on these basic requirements, consider features that the solution must have based on your environment:

- The ability to automatically establish identity and account relationships. This can come from existing directory services, email addresses, or unique designators highlighting the relationship.
- Store any discovered results in an asset database. Preferably, this database is a true CMDB (Configuration Management Database) and is ITIL (Information Technology Infrastructure Library) compliant. This will give you visibility across the entire organization.
- Classify accounts as human or machine, and ensure any applicable traits, like service accounts or interactive login, are enabled.
- Enumerate privileges for the account, including highlighting any privileged accounts. This is crucial for step two.
- Determine the age, longevity, and dormancy of any account to isolate flaws in the IGA process. This will help you identify flaws in the joiner, mover, and leaver process, as well as any accounts that may be left over from former staff and decommissioned applications.
- Identify up front any accounts not linked to MFA or SSO solutions. And if your organization is not already using MFA for all user accounts today, stop, close this book, and make that your highest priority.

- Highlight any accounts that are local to assets and resources and not centrally managed using directory services. These accounts will need special care when removing administrative rights for least privilege or ensuring their passwords are unique.
- Support a wide variety of operating systems, applications, and the cloud to ensure you can cover everything!

2. Identity Accountability

The most logical starting point for gaining greater control over identity accountability is managing the most sensitive accounts within the environment. Those are the accounts with privileged credentials. Privileged credentials include privileged account passwords, secrets for DevOps and CI/CD toolsets, SSH keys, certificates, and any file needed to start and maintain DevOps systems, such as JSON and XML files.

Unfortunately, administrators commonly share passwords, which makes it nearly impossible to get a clean audit trail required for identity accountability. Many systems, applications, and devices (IoT, etc.) have embedded or hardcoded passwords, exposing opportunities for misuse. Passwords are needed for application-to-application and application-to-database access. New privileged credentials are born when new cloud/virtual instances are spun up. The list goes on and the risks become myriad.

Many organizations still embrace manual password management measures (discovery, rotation, enforcement of best security practices) that are notoriously unreliable, complex, time-consuming, and impractical to scale. No wonder many environments still have problems. Even some best practices—like eliminating and centrally managing some types of embedded passwords—are virtually impossible without enterprise tools. And we are not referring to personal password managers that suffer from a wide variety of other flaws outside the scope of this book.

So, how do organizations ensure security and accountability over all the different types of identities without disrupting productivity or other workflows and processes?

Answer: An automated, comprehensive solution that can seamlessly discover the ever-expanding list of identity and account types (both human and nonhuman) in your environment, place those accounts/credentials under management, and satisfy auditor requests that they are being adequately managed. Such a solution will eliminate some identity attack vectors outright while mitigating many others, thus helping to drastically

reduce enterprise security exposures. This requires a purpose-built enterprise password management and privileged credential management solution that can automate each phase of the password lifecycle consistent with your security policies. This will create identity accountability because the accounts that matter the most are consistently being managed. Consider these core characteristics that should be a part of this solution:

- Performs full network scanning, discovery, and profiling with auto-onboarding of privileged identities and accounts of all types—shared admin, user, application, and service accounts, SSH keys, database accounts, cloud identities and accounts (Entra ID, etc.), social media accounts, machine, DevOps secrets, API keys, and robotic process automation credentials—including by vendors
- Illuminates where and how privileged passwords are being used, revealing security blind spots and malpractice (default, shared, and/or, embedded passwords, use of the same admin account across multiple service accounts, reuse of SSH keys across multiple servers, etc.)
- Manages credentials across every platform (Windows, Unix, Linux, cloud, on-prem, etc.), directory, hardware device, application, services/daemons, firewalls, routers, etc.
- Provides one unified, comprehensive solution to manage human (privileged users) and non-person (application, machine, service account, etc.) identities, and that includes session monitoring/management—no requirement for multiple/different interfaces or to be charged separately for each
- Centralizes, secures, and encrypts all privileged credentials in a tamper-proof safe/vault (Ideally, the solution supports industry-standard encryption algorithms, such as AES 256.)
- Builds permission sets dynamically according to data from scans
- Implements API calls to eliminate embedded/hardcoded credentials in files, applications, scripts, and other code

- Automates rotation of passwords, SSH keys, and other secrets according to a defined schedule, including after each use for the most sensitive accounts or for those accounts facing heightened security risks or compromise
- Enforces your privileged password management policies, such as password complexity, uniqueness (different passwords per asset, account, etc.), expiration, rotation, check-in and checkout, elimination of default passwords, and other rules
- Automates workflows across the entire password management lifecycle
- Provides granular access control
- Enables SSO and never reveals the password to the end user
- Performs rigorous session monitoring and management to ensure a clean audit of all privileged activity and to immediately pause or stop suspicious sessions until a determination can be made regarding legitimacy
- Requires no additional third-party tools or Java for session management; utilizes native tools (MSTSC, PuTTY) instead
- Enables true least privilege by enabling a security model of just-enough access and just-in-time access
- Leverages industry standards, like SAML and RADIUS, to integrate with any MFA solution
- Provides break-glass options for password checkout
- Enables privileged task automation to reduce risk by automating multistep, repetitive tasks
- Provides comprehensive reporting and analytics for SOC teams and for executive visibility into the management of privileged credentials
- Delivers enterprise-grade audit and compliance support by providing clear and distinct audit trails for all activities involving credentials under management

In addition, please consider these other traits that may be relevant to your organization:

- How important is scale? Do you have just a few thousand privileged credentials or hundreds of thousands? A handful of PAM solutions may be able to scale to manage tens of thousands or even hundreds of thousands of privileged user credentials. Fewer still can also manage high numbers of SSH keys. And if it is important to you (it should be) to monitor and manage all privileged sessions, understand that just a couple elite vendors can monitor/manage hundreds of thousands of concurrent sessions.
- How adverse are you to security complexity, solution overlap, and security vendor redundancies? Many security vendors sell various components of privileged credential management separately—each with their own management console. Of course, there are also niche security vendors that may offer stand-alone capabilities for SSH key management or application password management or secrets management. Ideally, you want complete solutions that manage, monitor, and audit all types of privileged credentials in a centralized and unified way vs. managing a sprawl of point products across the enterprise.

3. Remote Access

Remote access pathways have long represented weak links for most organizations. Today's era of elevated remote work and distributed workforces provides ample opportunity for attackers. Researchers have consistently found that ransomware attacks gain an initial foothold by exploiting RDP (Remote Desktop Protocol). According to an IBM X-Force Report, 76%[1] of cloud accounts for sale are for RDP access.

[1] www.ibm.com/reports/threat-intelligence?utm_content=SRCWW&p1=Search&p4=43700077724064084&p5=p&gclid=CjOKCQjw1aOpBhCOARIsACXYv-dK8BaKMFiY7rWFleADJDo5-iEhm5HxxJYoTwCvjqi4o8Yw1YmReX4aAjaDEALw_wcB&gclsrc=aw.ds

Third-party vendors and internal employees alike (including cloud ops engineers, developers, and other distributed workers) need the right level of access to effectively do their jobs. IT administrators need the ability to effectively manage, grant, and elevate privileges.

Organizations often lack visibility into what vendors and remote workers are doing when they access their network. VPNs provide far more access than is usually required. Most other remote access solutions also share similar pitfalls as VPN, including

- All-or-nothing access with a lack of granular security settings
- No visibility into, or record of, activity—meaning no audit trail
- Lack of support across diverse operating systems and use cases

With these shortcomings, a single compromised account could take down your entire network. When you consider the scale of your organization and the security risk of dozens or hundreds of third parties on your network, it is plainly apparent how critical this deficiency is. With so many remote access points and with (typically) suboptimal visibility, auditing, and security controls over this access, it is just a matter of time before a weak link across the remote access surface is compromised—either via an employee or a third-party vendor.

How can organizations better monitor remote access for privileged users without inhibiting business agility?

Answer: Eliminate "all-or-nothing" remote access for vendors and other remote workers by implementing granular, role-based access to specific systems with defined session parameters. Allow vendors or internal users access to specific systems for an allotted time and for specific applications or purposes. Administrators can approve or deny access requests from anywhere and any device to anywhere across major platforms. And finally, only allow root or domain administrative access from trusted devices, and never allow a BYOD (Bring Your Own Device) to obtain or access privileged accounts. Therefore, consider these security controls and capabilities for a remote access solution to manage identity access, regardless of privileges:

- Enforces least privilege by giving authorized users just enough access to complete activities just in time for remote sessions.
- Controls and monitors sessions using standard protocols for RDP, VNC (Virtual Network Computing), HTTP/S, and SSH connections.

- Enables granular access to specific systems, improving security and eliminating "all-or-nothing" access.
- Enables the user to inject credentials directly into the access session; the user never needs to know or see the credential (includes accounts with MFA enabled during a Web Jump Access session).
- Creates an audit trail to provide visibility into vendor activity on your network, as well as meet compliance mandates, by controlling the access pathways into IT networks used by vendors.
- Manages privileged access to infrastructure and business assets that leverage web-based management consoles, including IaaS servers, hypervisor environments, and web-based configuration interfaces for core network infrastructure.
- Provides seamless, out-of-the-box integrations with ITSM (Information Technology Service Management), SIEM, SCIM, and password management, as well as other common business software solutions.
- Enables MFA and alternative authentication methods, such as TouchID/FaceID or FIDO2.
- Leverages industry standards like SAML and RADIUS to integrate with any multifactor authentication (MFA) tool.

4. Implement Least Privilege and Application Control

Once identities, accounts, and credentials are being consistently discovered, onboarded, and managed via IAM, IGA, and PAM, the next step is to implement least privilege on end-user machines by eliminating local administrative rights. Similarly, if you have Windows servers, you also want to dial in the proper privileged access for your various administrator accounts (Network, Microsoft Exchange Active Directory, Database, Developers, Help Desk, IT Staff/Power Users, etc.). This will be covered in greater depth in the second half of this section.

With a least privilege approach, users only receive permissions to the systems, applications, and data they need based on their current role. Rather than having privileges enabled and always on (zero standing privileges), thus always ripe for misuse

or abuse, the privileges are only elevated on an as-needed basis. By defaulting most users to standard users and only elevating privileges as needed, you drastically reduce the threat surface, sharply curtail the ability for lateral movement, and minimize the risk of threats, such as phishing and ransomware, to land and expand. By tightly controlling and auditing admin access, you also ensure your most sensitive assets are protected from identity abuse.

Relying on native and ad hoc, in-house toolsets to restrict or enable end-user privileges is onerous and time-consuming. Candidly, it just does not scale. Moreover, although users should not be granted local administrator or power user privileges in the first place, sometimes certain applications require elevated privileges to run. This typically happens with older applications, applications that truly need system access, or ones compiled with unnecessary privileged settings.

So, how do IT organizations reduce the risk of users having excessive privileges without obstructing their productivity or overburdening the help desk with requests for privileges/permissions to perform required tasks?

Answer: Organizations need the ability to efficiently eliminate local admin rights across Windows and macOS systems, tightly control and audit admin access to servers and sensitive systems, and enforce granular control over applications, all without hindering end-user productivity. This requires enterprise endpoint privilege management solutions that remove end-user privileges while automating rules-based technology to elevate application privileges without ever elevating user privileges. To accomplish this identity security goal, consider the following characteristics you will need for a solution:

- Implements true least privilege by removing standing local administrative rights across desktop and server users while enabling dynamic, just-in-time elevation of privileges for specific applications and tasks
- Provides powerful application control that enables management of which applications users can install or run, with the flexibility to set both broad and granular rules through a nonresource-intensive operational process
- Enforces policy-based restrictions on software installation, usage, and OS configuration changes

- Sets policies via the cloud as a SaaS solution or Active Directory Group Policy with support for air-gapped systems and non-domain assets
- Reports on privileged user behavior, including applications installed or run, system and configuration changes, as well as changes to critical policy or data files, eliminating unauthorized software installs, workarounds, or gaps that could lead to exploitation
- Provides a single, unimpeachable audit trail of all user activity that simplifies compliance and streamlines forensic investigations
- Simplifies operations by eliminating the need for end users to require two accounts
- Provides a technique for using real domain or local privileges, when required
- Centralizes management, policy, reporting, and analytics into one streamlined solution
- Integrates with identity security, ITSM, SIEM, and other privilege management tools to embed into and enhance the existing security tech stack, improving workflows and allowing for a more comprehensive understanding of risk

Next, let us consider servers, which encompass Windows, Unix, Linux, and any other flavor of server-based operating systems. First ask, "do you have a Unix/Linux server estate or nontraditional endpoints that touch your network?" If you do not know the answer and are doing any work in the cloud, you probably do. Many vendors that offer Windows privilege management capabilities lack similar capabilities for Unix, Linux, and macOS, let alone nontraditional endpoints that may be present in the cloud. An ideal endpoint and server privilege management solution would enforce least privilege and application control best practices across all assets. In this age of identity security, there truly is no reason not to consolidate vendors and solutions. With that said, consider the following:

- Business-critical, Tier 1 applications running on Unix and Linux servers are prime targets for cyber threat actors. Privileged user credentials for these resources can provide access to ecommerce data, ERP systems with employee data, customer information, and sensitive financial data.
- Having root passwords, superuser status, or other elevated privileges is important for IT admins to do their jobs. Unfortunately, this practice presents significant security risks stemming from intentional, accidental, or indirect misuse of privileges.

Native, open source, and ad hoc tools are often used to "get by." But in server environments with even modest complexity, you end up paying a high price for these "free" tools in several ways. Some shortcomings of Sudo and other basic tools include

- Deficiencies in oversight, forensics, and auditing: lack of file integrity monitoring, log securing, or the ability to record sessions and keystrokes for audits.
- Existing gaps in security for these tools that do not account for activity inside scripts and third-party applications, leaving a shortcut to unapproved applications.
- Native OS tools lack the ability to delegate authorization without disclosing passwords.
- Basic and free tools also tend to contribute to administrative complexity and lack of scalability. Policies typically need to be managed on each individual server when using Sudo or other basic tools and do not scale to large, complex enterprise environments or clouds.

With Sudo and other tools, it is virtually impossible to maintain best-practice security and compliance in all but the most primitive of IT environments. And simply put, the stakes of inadequate privileged access controls in your Unix/Linux environments are far too high to depend on manual processes.

So, how does an organization address these server privilege management shortcomings of native tools?

Answer: An enterprise server privilege management solution that provides visibility and control over all privileged activities across Unix and Linux. This includes consistent enforcement of least privilege, efficient delegation of Unix and Linux privileges, and authorization, all without disclosing passwords for root or other accounts. The ability to either do away with Sudo outright or make the most of Sudo by layering on enterprise capabilities that resolve security and auditing deficiencies makes administration simpler and less prone to error. Consider the following requirements for identity security and least privilege within your non-Windows environments:

- Enforces least privilege and eliminates the use of root—without hindering user productivity
- Enables just-in-time administration (JIT), with the ability to assign dynamic privileges to accounts and assets to ensure identities only have the appropriate privileges when necessary and for a limited amount of time
- Exercises granular control and audit over applications, commands, files, and scripts to help protect against malicious threats, as well as innocent errors
- Records and indexes all sessions for quick discovery during audits
- Adaptively enforces full keystroke logging for the most sensitive sessions to prove appropriate behavior
- Provides a clear view and clean audit trail into who is doing what and when
- Consolidates audit logs and centralizes reporting across all your server domains
- Supports Pluggable Authentication Module to enable utilization of industry-standard authentication systems and provide a high confidence in identity authentication and authorization
- Offers a powerful and flexible policy language to provide a migration path from Sudo that supports change control and scripting best practices
- Provisions/deprovisions privileges transparently, helping to ensure compliance and honoring the concept of least privilege

- Includes file integrity monitoring to protect critical files and binaries from tampering or inappropriate access, including attempts at exfiltration
- Offers REST API for easier integration with third-party products
- Leverages an integrated data warehouse and threat analytics across the privilege landscape
- Integrates with identity security, ITSM, SIEM, and other privilege management products to improve workflows, better understand risk, and implement context-based privilege elevation/delegation decisions

5. Integrate Directory Services

Once you have greater control over privileged access in Unix and Linux environments, the next logical step is to bring those systems under consistent identity management, policy, and Single Sign-On. Unix, Linux, and macOS have traditionally been managed as stand-alone systems with each operating as a silo with its own set of users, groups, access control policies, configuration files, and passwords to remember. Managing a heterogeneous environment that contains these silos, plus a Microsoft Windows environment, leads to inconsistent administration for IT, unnecessary complexity for end users, and added risk to the business due to the wide variety of identity management solutions.

So, how do IT organizations manage identity policy and authentication consistently across diverse platforms and provide a streamlined user experience that reduces administration time and errors?

Answer: Centralized authentication for Windows, Unix, Linux, and macOS environments to reduce the risk and complexity of managing a heterogeneous environment. This creates a drastic improvement by reducing the number of logins (and the resulting help desk calls when they are forgotten) and the number of different systems, configurations, and policies to manage. This requires a directory bridging solution that allows all platforms to authenticate against one directory service, like Microsoft Active Directory (AD) or Entra ID. To achieve this goal, the implementation and solution should have the following traits:

- Single Sign-On for any enterprise application that supports Kerberos or LDAP
- A single familiar toolset to manage both Windows and Unix/Linux systems (e.g., Active Directory users and computers, ADUC (Active Directory User Controls))
- Allows users to use their Active Directory credentials to gain access to Unix, Linux, and macOS, eliminating various password files, NIS (Network Information Services), and LDAP repositories into Active Directory and removing the need to manage user accounts separately
- Support for adding Linux, Unix, or macOS systems to the network without requiring Active Directory schema modifications
- Provides a pluggable framework with an interface similar to Microsoft's Management Console on Linux or macOS (full support for Apple's Workgroup Manager application would allow for seamless management and control of macOS system settings)
- Supports a wide range of Unix, Linux, and macOS platforms (CentOS, Debian, Fedora, FreeBSD, HP-UX, IBM AIX, Oracle Enterprise Linux, SUSE, Red Hat, Solaris, Ubuntu, etc.)
- Supports compliance with SOX, PCI, HIPAA, and other regulations to demonstrate compliance with identity security standards

6. Identity Security

Organizations are struggling to protect themselves in large part due to two trends:

1. Their digital estates and attack surfaces are increasing. In a nutshell, we have more and more to manage, and the locations for technology are sprawling everywhere.
2. Their threat visibility is decreasing due to the lack of tools, best practices, and understanding of how anything electronic can be hacked.

Now, if we add the explosion of human and machine identities and the proliferation of new access paths to critical systems and data, most security teams are left with poor visibility into threats and other security exposures that can come from almost any angle (poor Qui-Gon, he never saw it coming). And simply put, identity security is the overall security posture protecting every aspect and workflow of the entire IAM stack. The days of "I did not see that identity-based attack coming" are nearing the end for most organizations to use as an excuse.

On the other hand (since Luke has lost one of them), IT and security professionals are overloaded with vulnerability and threat data. Advanced persistent threats (APTs) often go undetected because traditional security analytics solutions are unable to correlate diverse identity-related data, such as privileged accounts, users, assets, cloud entitlements, etc., to discern hidden risks. Seemingly isolated events are written off as exceptions, filtered out, or lost in a sea of data. The intruder continues to traverse the network, and the damage continues to multiply. AI-driven attacks are already proving more evasive as an up-and-coming identity attack vector that has left defensive teams fighting unseeingly unwinnable battles.

Today, organizations lack a centralized view of identities, accounts, and privileged access across their IT estate to help identify identity-based security risks. Solutions tend to operate in fractured siloes. This creates identity security challenges for enterprises of all sizes, including:

- Heightened risk that their solutions will not integrate or communicate well with each other, resulting in downtime, security gaps, and frustration
- Persistently higher administrative burden just to determine indicators of compromise or identity risks
- Delayed orchestration in response to threats since information is not well presented
- Inability to satisfy auditors or address forensic requests in a timely manner, if at all
- Security risks to the identity infrastructure itself since targeting IAM solutions is just another attack vector

So, how do security and IT operations teams gain an understanding of where threats are coming from, prioritize them, and quickly mitigate the risks?

Answer: A holistic, intelligent view of all identities and access across your entire multicloud and on-premises estate that can provide a clear picture of risk. This requires a modern identity security solution that can

- Provide a centralized, holistic lens of identities and access across your multicloud and on-premises estate.
- Provide a clear, easy-to-understand picture of the accounts, privileges, and access associated with each identity.
- Promote identity hygiene with actionable recommendations, before they become a threat.
- Identify and right-size problematic privileges and cloud entitlements, and remediate other security misconfigurations.
- Identify overprivileged and high-risk privileged accounts, inactive and orphaned accounts, partially revoked identities, and other security issues.
- Provide the ability to see new attack paths you could not detect in a siloed environment or with non-integrated toolsets.
- Detect identity-based threats and proactively respond. This allows for the acceleration of threat investigations and timely risk mitigation.
- Detect and alert on suspicious activities, including events involving multiple identities and accounts.
- Understand complex attack chains, attack paths, and the blast radius of compromised identities and accounts.
- Marshall the intelligence needed to orchestrate the optimal response to a threat or breach.
- Correlate low-level data from a variety of leading third-party solutions to pinpoint high-risk users and assets and identify critical threats.
- Report on compliance, benchmarks, threat analytics, and what-if scenarios.

CHAPTER 14

Evolving Identity Security Threats

By executing well on the preceding steps, you will address most of your identity security needs, eliminate or mitigate many identity attack vectors, and vastly reduce your threat surface. This is something you can tangibly measure when you see the number of alerts, investigations, and incidents decrease in your Security Operations Center (SOC).

However, most emerging technologies that have the power to transform IT come with security challenges, such as how to manage identity and authentication models and privileges for new technologies, workflows, and regulations. These present the types of gaps that savvy attackers seek to exploit simply because the maturity of security is always behind technology and laws.

While there are many edge use cases solutions can meet that are not covered in this chapter, let us briefly touch on several important areas that have emerged in recent years that can present unique challenges.

DevOps (Development Operations)

Most organizations today have adopted DevOps practices. Yet security is often an afterthought, or even a casualty, of the speed and tools used, which are often open source in DevOps environments.

While DevOps achieves condensed development cycles through automation and frequently leverages the scale of the cloud, the downside is that it can also "automate insecurity," creating massive security, compliance, and operational gaps. Some common DevOps risks include:

- Insecure code, hardcoded passwords, and other privilege exposures that could lead a threat actor to easily "owning" the environment

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_14

- Scripts or vulnerabilities in CI/CD tools—such as Ansible, Chef, or Puppet—that could deploy malware or sabotage code and compromise the supply chain
- Excessive provisioning of privileges across the DevOps landscape between zones
- Sharing of secrets used in the pipeline via insecure storage or retrieval processes
- Vulnerabilities, misconfigurations, and other weaknesses in containers that could lead to an exploit or identity compromise

While additional security controls clearly need to be built into DevOps, how do you do so without hampering speed and agility?

Answer: Implement solutions that reduce DevOps and CI/CD-related risks by improving visibility and control over secrets, admin privileges, and system configurations. By uniting these capabilities across on-prem, virtual, cloud, and DevOps use cases, IT organizations can achieve their agility goals without burdensome processes. Consider the following characteristics for this type of solution:

- Inventories and auto-onboards all DevOps assets and identities to automate workflows for increased visibility and support for audit and compliance
- Finds, secures, and centrally manages the use of all hardcoded passwords, secrets, keys, and certificates, including developer access to source code, DevOps tools/applications, scripts, test servers, and production builds, thereby eliminating a common threat vector frequently exploited by attackers
- Enforces least privilege by granting only required permissions for only the finite moments needed to appropriately build machines and images or deploy, configure, and remediate production issues on machines and images
- Applies application control to ensure the use of only the right tools in the right context, limiting the exposure of lateral movement, should an attacker gain inappropriate access

- Enforces boundaries between development, quality assurance (test), and production systems
- Manages and audits all privileged sessions, delivering much-needed visibility for security teams, as well as audit and compliance support

Finally, while some may consider DevOps a mature Agile discipline with plenty of secrets management solutions on the market today, I have chosen to include it in evolving identity security challenges since even the most recent tools are barely able to keep pace with newer workflows being created in the cloud. DevOps will continue to rapidly evolve, and just like betting on the Millennium Falcon, it will change drastically again in the coming years.

Synthetic Identities

For entrepreneurial threat actors, accessing datasets from past data leaks and breaches has long been a common attack vector. We experience them daily—from phishing attacks and scam phone calls to credit card fraud—however, an unnerving sophistication has recently taken place in these common attack vectors. Organizations and individuals are falling victim to criminals who have developed innovative methods of linking compromised datasets based on common fields and attributes. Previously, simple data leaks for email addresses, names and birthdates, phone numbers, and even something as obfuscated as the last four digits of an identity's Social Security number are now being linked, merged, and correlated with other data breaches to produce partial profiles for millions of people. The end result makes it easier for threat actors to commit identity-based crimes with enough information to spoof a user's identity with a high degree of electronic confidence.

In fairness, this technique is an old attack vector with a new name, but it has been wreaking havoc across unsuspecting entities recently. It has even been cited recently in the consumer banking industry as an attack vector that has left thousands of people with fraudulent bank accounts associated with their identities. Dubbed "synthetic identities," this resurrected identity attack vector contains an individual's nearly complete profile based on previous data breaches. While this is not a new technique for banks to contend with, the ramifications have created a void in the cybersecurity industry, and worse, a gap in security best practices for identity validation and verification because the electronic profiles of the fraudulent accounts are near complete at the time of creation.

To understand this risk better, let us start with a modern definition for a synthetic identity. According to Equifax,[1] synthetic identities are a form of financial fraud in which a real person's information, such as their Social Security number or date of birth, is stolen and combined with other falsified personal information to create a new identity.

The weakness that leads to this type of attack manifests itself in the lack of validation of the falsified information used in the creation of the synthetic identity. The consumer's name and Social Security number may be correct, but subtle nuances—such as their home address or phone number—are often falsified to conduct the attack.

Moreover, it's not unusual for someone's contact information to change from time to time, and that's why it's an unreliable attribute when validating an identity. Contact information can change when someone moves, takes a new job, or even has a change in a relationship. Businesses rely on name, birthdate, and Social Security number (or the last four SSN digits), and that is how it becomes a new liability. Even if the organization bundles this information and sends it to a third-party identity verification service, confidence in the data is only as good as the threat actor's hack and partial semblance of the synthetic identity. After all, the more real data the attacker possesses, and the more careful the manipulation of the synthetic data they inject, the more likely they are to succeed in their attack.

While this has obvious consequences for businesses and consumers, businesses employing staff and leveraging contractors and vendors can easily suffer from similar attack vectors. Ask a very basic question: How much personal data does the organization collect and verify in its human resources system when a new employee, contractor, or vendor gets onboarded? And does the company periodically revalidate the information to ensure personal changes do not nullify the information?

While these may sound like standard operating procedures, gaps in this process can easily lead to a wide range of financial fraud within a business. For those not sure how these attack vectors could materialize, consider the following:

- *How does an employee notify the business of a contact information change, including electronic banking information?* If the company allows these changes solely through email, the human resources or accounting department is open to a phishing attack to misdirect correspondence or an electronic deposit of payroll. Never use email alone to change this information. Consider verification via a phone call or service desk ticket.

[1] www.equifax.com/personal/education/identity-theft/articles/-/learn/synthetic-identity-theft/#:~:text=Highlights%3A,to%20create%20a%20new%20identity

- *When a new team member is onboarded, what is the verification procedure the organization requires to validate the information provided?* Odds are the new employee does not behave in a malicious manner, but how does the company validate their driver's license information? While state laws often require changes within 30 days, it is common to ignore this legislation. An old address in employee records can lead to a variety of issues, including validation of I-9 forms for employment authorization and incorrect correspondence when postal mail is required.
- *What kind of communications protocols for an emergency have been established?* While synthetic identities generally do not involve emergency contact information, recent cyber and social engineering attacks involving threat actors posing as faux kidnappers or individuals that have been incarcerated have left family members in sheer panic, while they reconcile how to respond. Employers should collect, protect, and periodically revalidate emergency contact information, including a "code word" or "passphrase" to manage any emergency scenario that may arise. Voice phishing (vishing) has become a legitimate attack vector. Having the business manage emergency contact information, just like any other personally identifiable information (PII), can mitigate this threat and cause a threat actor posing as a fake family member to be promptly identified by their inability to provide a shared "secret" in the event of a vishing attempt.

I said before that synthetic identities are not new; it's simply new terminology for an old problem. The primary difference today is the amount of data used to falsify an identity and the techniques used to create that profile. Falsifying contact information and posing as someone else's identity is a technique criminals have plied for hundreds of years. While consumers bear the brunt of these attacks, businesses can modify their policies to ensure employees, contractors, and vendors offer complete, up-to-date information to mitigate an attack. And to ensure staff does not become a victim of an attack, deny the changing or dissemination of this information using unsecure and non-verifiable communications, like email. These simple changes can help mitigate a company's risks from synthetic identity attack vectors.

Masquerade Attack

A masquerade attack is an emerging form of cyberattack that involves the use of a manipulated, manufactured, spoofed, or stolen user identifier and is based on the electronic representation of an identity. A device, digital signature, network address, session token, GUID, certificate, etc., are used to bypass access control lists and gain access to systems or authorization to conduct certain privileged actions based on the spoofed electronic representation of an identity. Masquerade attacks can be used to perpetuate financial crime, compromise assets on-premises or in the cloud, or access sensitive data without creating a new or modified human or machine identity. It is their unique electronic representation that is stolen and abused using other attack vectors like brute force attacks, MFA fatigue, or password spraying.

This emerging identity attack vector can be carried out from either inside or outside a network by either insider threats or external threat actors. Essentially, any demark where access is allowed and can be spoofed has a risk surface. Therefore, any place a human or machine can submit credentials or a secret for access, including by IP address, can be included in the risk surface.

Masquerade attacks are categorized as a new identity attack vector simply because the human or machine identity is stolen and used a limited number of times, and then the electronic representation provides persistent presence in order for the threat actor to maintain a beachhead in an environment. Its definition is merely an extension of many of the other identity attack vectors organizations experience today. Unfortunately, if you search the Internet for more information on masquerade attacks, you will find a myriad of definitions that just illustrate a threat actor impersonating another trusted user. Those representations are deficient because that is just an impersonation attack, and it has existed as an attack vector since the first use of credentials for identity verification. The electronic identity component is crucial to a proper definition because the masquerade is the reuse of the electronic component of an identified authenticated representation that makes the definition possible. Therefore, a simple masquerade attack chain could look like this:

1. A threat actor sends a phishing email to various members of an organization. The phishing campaign may be targeted or random.
2. The phishing campaign uses a watering hole technique to attempt to scrape the credentials from the suspecting recipient. (A watering hole attack is a spoofed website impersonating a real

website designed to capture credentials via a successful login and then pass that information to the real site for authentication.)

3. Based on the pass-thru-design of the watering hole website, the threat actor can capture the website's session token and continue access until it is revoked or expires. This can expose data or be leveraged in other forms of privileged attack vectors and was illustrated previously in the attack chain chapter of this book.
4. If the credentials stolen via the watering hole are based on a corporate identity, then leveraging a vulnerability, exploit, or hardening weakness, the threat actor could authenticate into the end user's host using the same credentials and leverage the host directly.
5. Once local access is obtained, the threat actor can scrape the host for additional password hashes and use them to authenticate into other assets within the environment.

This technique is what defines a masquerade attack versus just reusing the same credentials over and over again based on compromised credentials. The stolen website token and locally obtained password hashes can make this technique difficult to detect in real-world environments because authentication has already been performed successfully as a part of normal operations, and the threat actor is just reusing the electronic representation of an authenticated identity to continue their nefarious mission. This is why we have a new definition, new attack vector, and threat actors target the electronic authenticated representation of an identity trying to steal more credentials.

Operational Technology, IoT, and Nontraditional Endpoints

Internet of Things (IoT) has gone mainstream in almost every enterprise. Other nontraditional endpoints are also pervasive across most organizations. These endpoints often lack basic security features, frequently have default and hardcoded or embedded credentials, may have firmware that is difficult to patch or update, and carry with them many other risks—from being susceptible to denial-of-service (DoS) attacks to data leakage.

Many of these devices and systems were never designed or envisioned with the intention of being connected to the corporate network in such broad fashion, especially for mission-critical work, as is the case in some operational technology (OT) environments. As a best practice, industrial control systems (ICS) and supervisory control and data acquisition (SCADA) systems were traditionally "air-gapped" to safeguard their mission-critical functions. Today, we even find those systems connected to traditional IT networks and communicating with the cloud. Additionally, many ICS vendors now use standard IT technologies within their solutions, making them more accessible to attacks by using COTS (Commercial of the Shelf) operating systems, applications, and remote access protocols.

Management tools for these assets typically lack the ability to discover, onboard, and securely manage diverse device types and secure their access at scale. The result is many dangerous security exposures sprawled across OT and IT environments. Mirai and other botnets, which caused widespread disruption and brought some businesses to a standstill, are just the tip of the iceberg in terms of what can occur from lapses in security for IoT devices. Compromises of OT systems could lead to catastrophic damage to infrastructure and could jeopardize human lives, supply chains, and critical infrastructure.

So, how can organizations consistently account for and secure, at scale, the ever-increasing number of nontraditional endpoints, from IoT/Industrial IoT (IIoT) devices, SCADA, ICS, and common network devices (routers, switches, firewalls) used in these environments?

Consider a solution that offers granular command control and audit over privileged user activity on a network that is compatible using agnostic technology for all of these device types. The technology should extend the identity security best practices we have discussed, coupled with zero-trust security principles (covered later in this book), applied to OT systems and nontraditional endpoints. The solution should contain the following characteristics, at a minimum:

- Discovers and onboards all nontraditional network devices for management and asset inventory.
- Enforces credential management best practices, such as eliminating embedded/hardcoded credentials and securing credentials in a centralized, tamper-proof vault to mitigate threat risk and prevent lateral movement.

- Removes admin rights and applies fine-grained least privilege control for all endpoints and access, especially when the device itself has poor embedded RBAC technology. Good identity security can layer on top and provide functionality beyond what the manufacturer offers.
- Secures remote access (for employees, vendors, contractors, to/between systems, etc.) and implements least privileged access. This requires VPN-less technology that also layers on MFA for traditional remote access protocols, as well as any vendor proprietary technologies.
- Enables segmentation and micro-segmentation to isolate networks, assets, and resources from inappropriate access or lateral movement.
- Monitors and records sessions to provide a complete audit trail of user activity for training, compliance, and assurance of appropriate user behavior.
- Analyzes behavior to detect suspicious user activity based on commands, logs, access patterns, etc.
- Supports SSH, RDP, Telnet, HTTP(s), and many other COTS remote access protocols to limit the risk surface for remote access.
- Supports the Purdue Model and zero trust for architectural deployment.

Robotic Process Automation (RPA)

Robotic process automation is a fast-emerging and evolving method of using software robots to eliminate mundane and routine tasks that would otherwise burden other IT resources. However, RPA security controls are often inadequate. For instance, RPA toolsets typically have excessive rights and embed, or hardcode, credentials in order to establish connections for automation between assets, resources, and applications.

A solution for solving RPA challenges should have the following characteristics:

- Scans, identifies, profiles, dynamically categorizes, and auto-onboards all assets that may be included in an RPA workflow and supporting resources
- Enforces best practices for password management, including the elimination of hardcoded or embedded RPA credentials, and secures the organization from automated exploitation via an extensive, RPA-compatible API
- Ensures passwords can be automatically reset after RPA usage to ensure the security of the workflow and just-in-time access
- Enforces least privilege and granular control across RPA processes, toolsets, and workflows, such that any automated process cannot operate outside its designated scope
- Locks down access to authorized applications only, ensuring malicious behavior or malware cannot tamper with an automation workflow
- Integrates with and supports a wide range of RPA tools (Blue Prism, UiPath, Pega, etc.)

Cyber Insurance

Cyber insurance has become an essential component in today's digital landscape. The looming risk threat actors pose to business finances has left some C-suite executives relying on their cyber insurance policies to ease some of the pressure. Understanding the history of cyber insurance, how cyber insurance has evolved, and what key risk areas are being focused on can help your organization when it comes to qualifying or renewing a cyber insurance policy.

Although cyberattacks and data breaches have become more frequent across the world, most businesses do not have adequate protection through existing general insurance policies. Cyber insurance was first developed in the early 1990s to mitigate risks faced by tech and security firms that depended heavily on technology for their

operations. Initially, these policies only covered third-party liabilities. As technology evolved and high-profile breaches occurred, insurance brokers began offering first-party coverage as well.

Over the years, the cyber insurance market has experienced significant changes, and I foresee additional changes in its future related to emerging technology, artificial intelligence, and new attack vectors. One of the biggest disruptions we saw in the market with cyber insurance occurred during COVID. Prior to COVID, you infrequently heard the term "cyber insurance." In fact, cyber insurance was not even commonplace, despite existing for almost 30 years. COVID forced businesses to stretch their technology beyond its means, resulting in poor security decisions that facilitated the emergence of new cyber threats.

Threat actors capitalized on pandemic-created security gaps—they evolved just as the businesses evolved. This truly marked the dawn of an increase in the frequency of breaches—ransomware, phishing campaigns, and social engineering attacks that contributed to cyber insurance claims being submitted far faster than deductibles were being collected. Insurance carriers were losing so much money due to claims they had no choice but to deny renewals, reduce coverages, increase premiums, and realign policies to stop the financial bleeding. This knee-jerk reaction left businesses and organizations with limited options: either pay an absurd premium for minimal coverage or continue business without any coverage.

Today, cyber insurance is in a much better position than it was a few years ago. While it continues to evolve, cyber insurance brokers have a firm understanding of what they expect to see from businesses when it comes to addressing threats and risk mitigation. Both of these focus heavily on "identity."

Almost every breach requires two things: an identity and privilege (rights) associated with the identity. If you look at the top attack vectors, all involve some form of identity—whether it be stolen or weak passwords associated with an identity, phishing or social engineering focused on an identity, cloud access identities, vulnerabilities or exploits leveraged through an identity, or the malicious insider that has an identity. Some form of identity is at the heart of every breach, whether it be a human or nonhuman identity. Ensuring they are protected is just as crucial to businesses as it is to their cyber insurance broker.

Identity plays a pivotal role in cyber insurance, and identity attack vectors have greatly influenced the evolution of cyber insurance. Threat actors often target individuals to gain unauthorized access to compromise a business. With tactics like phishing emails and social engineering becoming increasingly sophisticated, insurers must stay ahead

of these threats by offering coverage that addresses identity-related risks. This includes coverage for identity theft resolution services and legal assistance for victims of identity attacks.

Cyber insurance brokers have modernized their approach with regard to assessing risk. Cyber insurance policies are reviewed by underwriters. Cyber insurance brokers also have started performing non-evasive scanning and probing of businesses to uncover any potential risk that might be exposed to the public. In addition, they have modernized the due diligence questionnaire that businesses seeking a cyber insurance policy or renewal must answer. This questionnaire is often referred to as the "Ransomware Supplemental Addendum." Over the years, it has grown from a couple of pages to an elaborate and dynamic flowing questionnaire, with as many as 50+ questions requiring potential responses. The responses are then used to determine whether a business is insurable or if it possesses too much risk to be insured.

One major area of a cyber insurance policy that has garnered heavy attention is the "exclusions." This is often referred to as the "fine print" of a contract, and in most cases, is overlooked. Exclusions are now the backbone of policy carriers protecting their vested interest. As companies are evaluated for risk (certain areas may possess high risks due to missing controls or risk mitigation factors), insurance brokers will write exclusions in the policy. These exclusions protect the brokers and carriers by calling out security deficiencies within a business. Thus, if a breach or compromise is a result of the exclusion, the policy will not cover it.

When it comes to "What is NOT covered" in terms of exclusions, a sensitive area that has recently undergone revisions is "Acts of War." Acts of War exclusions have been in cyber insurance policies since day one. Until recently, the Acts of War exclusion had not faced a rewording and categorization overhaul. Due to the legal fallout of the "NotPetya" attack in 2017 (and subsequent legal battles that arose from this), Lloyd's of London, a major marketplace for insurance and reinsurance, proceeded to outline, categorize, and define specific terms and verbiage for "Acts of War." As a result, many other firms have now adopted the updated verbiage outlined in Lloyd's Market Bulletin Y5381.

Businesses seeking to obtain a new policy or renew their existing policy need to understand that identity lies at the core of several risk factors and plays a pivotal part in a broker's evaluation. While several cyber insurance brokerage firms have created their own security checklists for evaluating businesses, most offer similar coverage with slight wording variances.

The following are key areas of importance involving identity and cyber insurance broker requirements:

- *Multifactor Authentication*: MFA is now viewed as the de facto standard for all organizations. While most cyber insurance brokers specifically call out the MFA requirement, some do not, instead assuming it is already implemented. If you have not instituted some form of MFA across your organization, you might not even qualify for a policy, or there could be exclusions written around MFA, should a breach or compromise occur.
- *Vulnerability Management*: Whether it's operating systems, applications, databases, etc., vulnerability management is critical for reducing risk and managing patches for exploits and vulnerabilities. A user's identity is the target; once compromised, a user's identity with the proper privileges can leverage an exploit or vulnerability to compromise an organization. Cyber insurance policies outline patching times and durations, and if either of these is exceeded and the vulnerable system is the result of a breach or compromise, your cyber insurance policy may deny your claim.
- *Service Accounts*: Service accounts[2] can include everything from human, nonhuman/machine, and any other privileged identities that exist in an organization. These accounts must be properly discovered, inventoried, managed, and audited. Ensuring each of these controls is in place will reduce identity risk and provide necessary audit trails when needed.
- *Endpoint Security*: With remote workers no longer protected by brick-and-mortar network security boundaries, organizations have focused even more on protecting their endpoints. Some security professionals focus heavily on detection and response. If you are detecting and responding, that means a breach has already occurred, and you are simply reacting. Detection and response are not preventative or even proactive. If you happen to read the annual Microsoft Vulnerability

[2] www.beyondtrust.com/blog/entry/how-to-manage-and-secure-service-accounts-best-practices

Report,[3] "Elevation of Privileges" and "Remote Execution" top the list of Microsoft vulnerabilities. Endpoint privilege management focuses on the preventative steps aimed for endpoint security. Make sure you evaluate how you are protecting your endpoints and the identities that use them.

- *Third-Party Access*: Despite their widespread use, virtual private networks (VPNs) are often a dismal security choice when it comes to enabling access for third parties/vendors. There are many other use cases for which VPNs are inappropriate, such as when mixed with employee BYOD and other sensitive access for internal employees. For adequate security, and to satisfy cyber insurers, organizations need to be able to restrict and contain an identity (enforce least privilege), track where they went, what they did, etc. VPNs are unable to provide that kind of granular access control and auditing. Cyber insurers are keenly interested in how third parties, vendors, and other users are gaining access to your protected resources. Typically, third-party vendors have secure remote access by means of a bastion or jump host that can broker connections, target, and limit specific resource access using PAM technology and the tenets of zero trust.
- *Cyber Awareness Training*: As I mentioned earlier, social engineering and phishing campaigns rank at the top of the list when it comes to how threat actors prey on victims. Users need to be trained on what to look for, how to respond, and what they should do if they suspect a phishing email or social engineering scheme. Being able to own an identity and all of the privileges associated with it, including its password, is the foothold bad actors seek to get started in an environment.

There can be several other risk factors cyber insurance brokers use to evaluate businesses. Make sure you understand where those risks are, if they are covered in or excluded by the policy, and make sure your policy covers both first-party and third-party losses resulting from identity-related incidents. This will provide comprehensive protection for your business and any affected individuals or clients.

[3] www.beyondtrust.com/resources/whitepapers/microsoft-vulnerability-report

Additionally, regularly assess your security measures and update them accordingly. Implementing strong authentication protocols and encryption methods can help safeguard sensitive data and prevent unauthorized access.

In conclusion, cyber insurance has undergone massive reform to keep up with the threat landscape. Cyber insurance will continue to evolve. For instance, the onset of generative artificial intelligence and quantum computing both expose areas of concern for cyber insurance policies.

Artificial Intelligence

The early, roaring 2020s may be remembered as the years generative artificial intelligence went mainstream, perhaps even more so than the COVID pandemic it closely followed. The potential uses and abuses of AI are starting to materialize, but there is one thing for certain: some use cases are raising concerns for specific business roles, like software development.

For those who haven't heard of generative artificial intelligence or tools like ChatGPT, the technology is based on revolutionary language models originally developed by OpenAI. The technology can be used for a wide variety of applications—from answering questions to writing letters and essays—and even for creating software source code. Generative AI allows users to interact with the model in plain English (or localized spoken and written languages), ask questions, and receive responses in natural language, including properly tagged and formed source code. While generative artificial intelligence is a breakthrough in machine identity personification, it is important to be aware of the potential risks associated with using the platform, especially for software development, documentation, and other material your organization may create.

One of the biggest risks of using generative AI for software development is the potential for the platform to generate code that is vulnerable to security threats, weaknesses, exploits, and the best and most honest way to do things. In my opinion, this is why the human introduction to this book was so important. Generative AI leverages vast amounts of public data on the Internet to formulate responses based on user inputs, but it cannot discern whether the generated code is bug or vulnerability free or whether it may be subject to intellectual property rights owned by someone else. This can make the AI-generated code exploitable by threat actors, despite being fully functional

and structurally correct. Any source code developed by generative AI should be fully inspected, vetted, and tested to ensure it is not a liability. In addition, the rights to create material may be called into question if it breaches patents or other ownership principles.

The second risk of generative AI is the relationship to personally identifiable information (PII) and the identities within your organization. While generative AI can process the written word to create a response, any PII in the initial request becomes public domain and a potential part of future inquiries—by anyone. That is how the system learns. Therefore, be mindful of the personal information that may be disclosed to anyone while interacting with generative AI. Asking the platform to write a configuration file or source code with specific server names, IP addresses, and even secrets could represent a potential data leak. The platform may collect and store this data, including interactive conversations that could potentially be accessed by unauthorized individuals. Inadvertently, a user's simple request with specific information to obtain an accurate response could lead to a breach, source code that has hardcoded vulnerabilities, or leaks in an identity that could be used as an attack vector.

To mitigate these risks involving the disclosure of sensitive information, we must exercise certain precautions when using generative AI for software development or other business roles:

- Never assume source code output from generative AI is vulnerability or risk-free.
- Never input a question with personally identifiable information. If needed, obfuscate key variables with easily substitutable information that is also not easily traceable.
- Never use generative AI to create software documentation or other information that contains specific code samples from the organization. This is a natural combination of the risks to developing code and the inclusion of PII.
- Consider that generative AI may use source information that violates the copyrighted works from other sources. Inclusion of derivatives of this work may require disclosure or licensing considerations.

While generative AI can be a valuable tool for software development, be aware of the potential risks associated with the platform and artificial intelligence. This is especially true since this is the first mainstream application to tackle conversational English and create output that uses the vast resources of the Internet to formulate answers with

unnerving confidence. I recommend taking the necessary precautions when trusting the solution for software development, and as a user, the information you provide it for meaningful results.

Secure Remote Access

In the age of client-server applications (circa 1990s and early 2000s), applications were developed with an installed client on your workstation and a back-end application that communicated over a priority port. The application could have multiple tiers of middleware, including gateways, databases, load balancers, and processing engines to offload complex workloads from underpowered clients that processed raw information when transmitted back. Most organizations still have many of these applications. Some of the applications were produced by leading vendors; others are homegrown.

Fast-forwarding to the late 2000s and early 2010s, modern software development saw a shift from client-based installations to applications powered via a web browser for end-user interaction. Early web-based applications lack many necessary characteristics compared to the full client–installed brethren, like context-rich right-click client menus and printing capabilities, due to the maturity of HTTPS and programming languages like JavaScript. Today, that has all changed. Advanced applications, like MS365 (Microsoft Office) in a browser, are nearly as good as their client-installed versions. So, what happened to all the client/server-based applications that we still need today? Where did they go since the onset of COVID and the widespread emergence of work-from-anywhere/hybrid work environments?

At the beginning of the COVID pandemic, the information technology industry saw a ramp-up of VDI (virtual desktop interface) technology and the expansion of VPN (virtual private networks). These two technologies were the primary ways to allow an end user working from home the use of legacy client-server applications. Either the organization hosted the application in a VDI environment (Citrix, Microsoft, VMware, etc.) or it was loaded on the end user's laptop and connectivity was provided via a VPN. This approach works but is incredibly expensive and difficult to maintain compared to having the same laptop connected in an office environment within traditional on-premises perimeter-based network controls, like VLAN segmentation, firewalls, IDS/IPS, etc.

A bit further ahead in the early 2020s, alternative secure remote access technology became the primary vehicle for managing connectivity as the perimeter dissolved, due to COVID and strategic business initiatives like zero trust. VDI and VPN technology

became a cost and maintenance problem. Enterprises no longer wanted to invest in VDI and VPN for the purpose of extending the life of client-server applications. The cost to rewrite the applications, with all the years of business logic, was cost-prohibitive as well. Therefore, the industry sought a better solution. The results were unexpected and signaled the death and slow decline of VDI and VPN.

During the last few years, secure remote access technology has focused on providing access to native protocols like RDP, SSH, Telnet, and HTTP using protocol encapsulation and routing technology. This is commonly found in the form of proxies or gateways. The question became—why can't this be done at the application and data layers, too? The answer was simple: yes, it can—and a new method of enabling client-server applications to operate was born in the form of infrastructure access. In lieu of the client communicating with the server-based application one-to-one (on-premises), the traffic is captured locally on the client, encrypted, and routed to the server-based application using the same proxy technology that was previously used for RDP, SSH, Telnet, etc., remote access. The application is still installed on the end user's device (laptop), and no VPN or VDI is needed to host the client or route the traffic through a VPN concentrator to the organization's private network (or private cloud). Infrastructure access performs all these functions at a fraction of the cost, without the maintenance of VDI, and as a bonus, with a much lower security risk. You are no longer routing network traffic for the device as a whole via VPN, but rather tunneling the trusted application only via infrastructure dedicated to remote session access. The use case has now just been expanded to allow client applications too, including fat clients like MS SQL Studio and Oracle Toad as well, for database maintenance.

This new way of supporting legacy client-server applications serves as the death knell for antiquated VDI and VPN technology. There is almost no need for them when web applications (browser-based) and traditional x86, x64, Java, macOS, and other clients can now operate locally and communicate anywhere in your environment (data center or cloud) without additional tooling. If you are trying to modernize your environment and save money, consider this approach to extending the life of your client-server applications in a work-from-anywhere world. This modern approach provides the benefits of zero trust and much better identity security, as opposed to just having an open RDP port exposed to the Internet.

Biometrics

In 2015, biometric data from the US Office of Personnel Management (OPM) was stolen in a well-published attack. Unlike accounts, usernames, and credentials, biometrics can never be changed and are permanently linked to an identity based on physiology. For this reason alone, the safety of biometric data should always be questioned and never considered as the sole viable means for authentication without some other verifiable attributes to establish confidence in an identity. In reality, biometrics are just a better single-factor (1FA) solution than credentials, but a system still needs to verify the owner of the biometric request for authentication.

While many vendors seem to regard biometrics as a holy grail for authentication, breaches such as those that occurred with OPM demonstrate why biometrics-based security has flaws. For these reasons, biometrics should only be applied for authorization and never for authentication alone. Today, multiple techniques are available for taking digital biometric information and creating hacking tools like fake fingerprints to bypass biometric scanners, plant false fingerprints, or even falsify applications that need fingerprint data using traditional ink techniques.

As we reviewed earlier, authorization, in the simplest terms, is permission to perform a task. It's the ability to proceed without verifying who you are or who you say you are. The most common form of biometric authorization used today is Apple Pay. When placing your finger on the touch identification sensor or scanning your face, you are authorizing payment (depending on the iPad or iPhone model). It is just permission.

Conversely, authentication is the verification of you as a person and who you say you are. It does not authorize you to perform any tasks; it just proves your identity. Authentication is primarily performed today by usernames and passwords, two-factor authentication, smart cards, and other techniques, like one-time-passwords. The technology for authentication generally ties secret knowledge to a second physical media, or to the creation of a unique code, that only you can know or possess, like biometrics. The various components of an authentication system are designed to prove your identity, but they do not authorize you as a person to anything.

So, here is where the problem lies. Newer biometric technologies are blurring the line between authorization and authentication. They are taking a sophisticated approach based on technology to identify an individual (authentication) and are now merging it with permissions to perform a task (authorization). When biometric data is compromised, security will suffer, and it can be used to infiltrate either the user's identity or the ability to perform tasks.

While OPM was the first major breach of this type, it had far-reaching ramifications. The biometric data stolen can be used to craft authorization and authentication attacks against anyone breached, including some of the highest-valued personas and assets in the world. Either type of attack would completely bypass traditional username and password paradigms if biometrics alone were used for both, as some vendors are proposing. This is why there needs to be a separation between authentication and authorization, as previously discussed, and why the definitions between the two need to be clearly understood if biometrics is going to succeed in authorization.

Biometric data is like any other form of data—it gets stored electronically. Granted, biometric data is encrypted, and it uses various forms of key mechanisms to ensure that one data source alone cannot reveal the contents, but that is not the point. Vendors claim biometric data is not hackable, but history has shown us that every type of encryption can be compromised and probably will be. It is just a matter of time, persistence, and a fast enough computer. This includes the promises of quantum computing to break old encryption algorithms and also to create new ones that are quantum-cracking resistant.

As of today, while there are systems that are "impossible to hack," that claim only has a finite lifetime. Biometric data is stuck with you for a lifetime. You cannot change your fingerprints or the blood vessels in your eyes. So, by the time you are old, there is a good chance that the system securing your biometrics might not be as secure as it is today. Considering that we have computers from the 1950s running air traffic flight control, power plants, and other infrastructure, it is reasonable to assume that some of today's systems and databases may also be around in 60 years.

There is one final point worth noting: the debate regarding biometrics is not new. The ability to steal someone's likeness is not limited to science fiction and has been documented repeatedly via the Internet and TV shows like *MythBusters*. What is new is the extent that hardware and software vendors are pushing biometrics as the next generation of security solutions.

In the last ten years, we have seen fingerprint readers on laptops (most of which have been cracked) and facial recognition software that uses local cameras to verify user identities as part of a login prompt. In fact, most Android phones suffer from these flaws, and it has been proven that 3D printed masks (heads) can trick these systems as long as they are not using infrared for confidence. This next-generation push of biometrics applies multiple techniques for facial recognition (visual, infrared, etc.) and ultimately stores the data electronically, just like every other previous technology. These devices are still computers with storage devices—encrypted or not—and biometric information

needs to be referenced somehow to complete the authentication process. Consequently, somehow, sometime, these biometrics will be compromised. A threat actor just needs to find a weak connection in the workflow to implement a successful attack vector.

The OPM breach has shown it is possible to steal biometric data. Consumers and businesses adopting biometrics need to be mindful to implement separation of tasks and to protect accordingly.

The implosion of biometric technology is not far off if we continue to blur the lines of authorization and authentication. As long as we can keep the two separate and distinct, the risk can be managed, and in the future, resultant predictions of identity chaos can be avoided.

Multifactor Authentication

Multifactor authentication (MFA) has been marketed as a security solution to combat credential and identity theft for the past two decades. In fact, when it was introduced in early 1996 using text messages as a second factor for authentication, it was heralded as an unhackable method to provide authentication—even when a threat actor knows your username and password. Like any other security technology from the Enigma[4] machine invented during World War II, through RSA SecurID Keys[5] from the early 2000s, every security technology has been proven to be hackable over time.

MFA represents an emerging identity risk. With this in mind, some implementations of MFA have proven to be fallible to cyberattacks and should be avoided by enterprises and consumers.

Some modern MFA attack vectors include

- *SIM Jacking*: SIM jacking (or SIM swapping) is a malicious technique where a threat actor fraudulently takes control of a victim's mobile phone number by convincing the victim's mobile carrier to transfer the number to a new SIM card or using a cellular sniffer called a StingRay to clone the user's SIM card. The former is often accomplished through social engineering, like tricking customer

[4] www.nms.ac.uk/explore-our-collections/stories/science-and-technology/enigma-encoding-machine/

[5] https://arstechnica.com/information-technology/2012/06/securid-crypto-attack-steals-keys/

support into believing the attacker is the legitimate owner of the device. Once the threat actor succeeds, they can intercept the victim's text messages and phone calls, which may contain sensitive information used for two-factor authentication. This allows the threat actor to gain unauthorized access to the victim's online accounts, potentially leading to identity theft, financial fraud, and other cybercrimes including extortion. To protect against SIM jacking, individuals should use strong and unique passwords, employ two-factor authentication methods not reliant on SMS text messages, and exercise caution when sharing personal information with unknown parties. Some cellular phone carriers offer added security features, such as PIN codes or security questions, to bolster defenses against SIM cards from inappropriately being copied or swapped.

- *MFA Fatigue*: MFA fatigue refers to a condition where individuals experience frustration, inconvenience, or reduced security compliance when prompted to provide multiple forms of verification during the authentication process legitimately or when prompted via nefarious activities. We have covered this multiple times already, but the faults should be noted in depth. While MFA significantly enhances security by requiring at least two forms of authentication (something you know and something you have), it can lead to fatigue when overused or abused as part of an attack, causing a form of denial of service. This fatigue can manifest as users avoid secure computing practices, implement shadow IT for authentication, or become less vigilant when flooded with frequent MFA requests and by ultimately allowing the user to authenticate in lieu of continuously receiving a barrage of requests. Balancing security with user convenience is essential to mitigate MFA fatigue, and from a threat actor, inappropriately flooding the end user with authentication requests (sometimes called MFA bombing) could lead to an identity breach.
- *SMS Text Messaging*: SMS texting for MFA carries several inherent risks that can compromise identity security. First, SMS messages are vulnerable to interception through various methods, such as SIM jacking as described earlier. Once in control of the phone number, a

threat actor can intercept MFA codes sent via SMS, just like any other application. Another attack vector can utilize social engineering to trick end users into revealing sensitive information via SMS messages that appear to be from legitimate sources. In addition, SMS-based MFA does not provide the same level of security as other methods, like time-based one-time passwords (TOTP) or hardware tokens since the results are sent unencrypted. For every organization, and based on current security best practices, it is advisable to use other methods besides SMS for MFA methods. Alternative solutions like one-time passwords, biometrics, passkeys, or hardware tokens are a better choice to enhance identity security.

- *Push Notifications*: Using push notifications for MFA introduces certain security and privacy risks. Push notifications often rely on mobile applications to deliver secondary authentication prompts and can be a source of MFA fatigue. Some potential risks associated with push notifications include:
    - *Application Vulnerabilities*: If the MFA application itself has security vulnerabilities or the vendor is compromised, threat actors may intercept or manipulate push notifications, potentially granting themselves unauthorized access to protected resources.
    - *Phishing Attacks*: Cybercriminals may attempt to mimic legitimate push notifications via pop-ups, messages, or even browser pages, tricking end users into approving access requests on malicious sites or applications.
    - *User Error*: Users might accidentally approve a push notification without verifying its legitimacy, especially if they're distracted, unaware of the potential risks, or as a result of a message storm associated with MFA fatigue.
    - *Lost or Stolen Devices*: If a user's device is lost or stolen, an unauthorized person may have access to push notifications and can potentially approve authentication requests, especially if the device has a weak pin code or no biometrics protecting access.

- *Sole Device MFA*: Relying solely on a single device for MFA can be problematic if the device malfunctions, is unavailable, stolen, lost, or runs out of battery, potentially causing the inability to authenticate or recover access.
- *Privacy Concerns*: Push notifications can collect and transmit user data, including geolocation, raising privacy concerns if the information collected violates the organization privacy policy or regional laws.

To mitigate these risks, it's important to use phishing-resistant MFA technology. Organizations should avoid MFA solutions that are dependent on the technology approaches listed earlier.

Biometrics, hardware passcode tokens, and MFA solutions that are based on FIDO2 for authentication are the best and currently the most resilient approach to mitigating these risks.

MFA has evolved in the last two decades; ask your organization if you have kept up with the changes and have plans to depreciate these obsolete approaches.

Passwordless

At the start of every year, businesses plan next year's budget, and people plan for their New Year's resolutions. Some people will resolve to eat better or exercise more and others to shed bad habits. For many organizations, there is a trend to go passwordless (or eliminate passwords, as much as possible) in the next year. The big question is how can a business actually achieve this and not compromise security? In fact, the end goal should be to improve your security and lower the risk of being hacked while allowing better protection against identity-based attacks.

Let us first consider all the nuances of passwords. What are they, where are they used, and how are they stored?

There are three primary locations where employees use passwords outside of a privileged access management solution:

1. *Operating Systems*: Passwords used for OSs are typically used to log in to the operating system after booting, rebooting, changing sensitive settings, or installing updates.

2. *Websites*: Any secure website, from banking to commerce, may require a password to authenticate an identity. Security-conscious websites have added one form or another, including multifactor authentication or basic two-factor authentication.
3. *Locally Installed Applications*: Some locally installed applications may require a password to access data or perform sensitive operations. These passwords are typically associated with an application, so a local client can authenticate across the network in a client-server architecture.

With these in mind, the technique to remove passwords varies based on your technology stack. Please consider:

- *Microsoft Windows*: Microsoft Windows users can store passwords and secrets within their browser (e.g., MS Edge and Google Chrome) and within Microsoft Hello technology that uses biometrics for identity verification. Microsoft Hello can be used during all aspects of runtime, including when the operating system boots. However, enterprises will still want centralized management and security over credentials to mitigate any threats that could compromise localized secrets storage.
- *Apple macOS*: Apple macOS users can leverage Apple's Touch ID technology to access Apple's iCloud Keychain for password storage and automatic retrieval. Note that this function only works during the normal runtime of macOS. A password is still required during the initial macOS boot process when the disk is encrypted, and this capability has no centralized enterprise management features for any business.
- *Android*: For Android users, the fragmentation of the Google Android market has left passwordless technology up to third-party vendors and Android phone manufacturers. Although the base operating system APIs (application programming interfaces) support passwordless implementations, a universal approach for password managers or biometrics isn't present. This means a third-party password management solution to support passwordless business implementations is required.

- *iPhone/iPad*: All iOS users can leverage numerous built-in technologies from Apple, including pin codes, Touch ID, and biometric facial recognition, depending on the phone or tablet model. Similar to macOS, pin codes are typically required on any reboot, sensitive updates, or application purchase (unless biometrics is explicitly turned on). During normal operations, Touch ID or biometrics can manage passwordless authentication. In addition, Apple has created an application for Windows to share browser-based passwords when a user has an iPhone and a Windows computer, which is good for individuals and small businesses to leverage for passwordless implementations. Unfortunately, there are still no enterprise-ready or business versions of this technology, only integrations into authentication mechanisms like FIDO2.
- *Generic*: Regardless of device and vendor, third-party solutions have filled the gap to create passwordless solutions when manufacturers have fallen short or when you use a mixed variety of vendors. These are password managers, biometric devices using industry standards like FIDO2, and privileged access management solutions. These solutions support installations on a wide variety of devices and operating system versions in order to synchronize passwords or provide password obfuscation via injection to minimize exposure.

With these basics in mind, let's apply your organization's operating systems, websites, and applications. Most users have a mixed ecosystem, such as a Windows computer and an Apple iPhone. Based on each manufacturer's solutions, you can use Microsoft Hello via biometrics for authentication on the device and share passwordless technology using iCloud Keychain across your iPhone and Windows browser (MS Edge or Chrome). You can enforce these settings via group policy. For Android and Windows users, third-party password managers provide the best approach for sharing passwords across their devices while leveraging Microsoft Hello for their Windows devices.

If you have a platform bias (like me, I am a galaxy starship kind of guy) and try to use all of one vendor (like Apple technology), you can eliminate almost all your passwords with native technology built into the devices—without licensing any third-party solutions. For many, iCloud Keychain allows you to sync passwords across all devices and auto-inject them into the operating system, browser, and applications when a password field is detected. If you take advantage of the technology to recommend

a complex, unique password for every entry, you can maximize your protection and lower your risk. Your business, however, may balk at this approach, due to the lack of centralized management, auditing, and monitoring capabilities. In that case, you will need a platform-agnostic tool, like an enterprise password manager, to achieve passwordless implementation goals.

Apple isn't the only vendor that offers these passwordless capabilities. Many third-party solutions, including enterprise-ready privileged access management solutions, contain these capabilities for businesses, too. Apple, however, is the only vendor that offers it across all devices; Microsoft doesn't have a phone-based operating system and relies on Android for its mobile phones. In these cases, a third-party application has been added to manage passwords and achieve passwordless management goals.

For many organizations, achieving a passwordless goal is made easier by limiting the number of vendors (i.e., standardizing as a Windows or Mac-only shop and limiting exceptions) used for end-user computing.

Remember (and the point is not to remember passwords), if you're tired of managing passwords and want to achieve a stronger level of security for your organization's daily runtime, consider going passwordless. The only caveat is that you may need newer devices and operating systems (at a cost) to achieve these goals natively. If you have older systems, third-party applications can help achieve similar goals via software, but without the benefit of having built-in biometrics. This password management approach uses one master password to secure all other passwords. This isn't a bad approach, but with modern technology, there's a better way.

Ransomware

Over the years, we have seen the toll ransomware has taken on businesses and organizations around the world. From notable breaches to common incidents at enterprises that you might not hear about on social media or mainstream news outlets, ransomware is a focal point of every modern cybersecurity strategy. If you have not considered the impact ransomware could have on your business and the financial loss and/or headache it may cause, then you are the perfect candidate a threat actor is looking for!

Here are five key elements to know about ransomware to help you assess the impact, how it works, and what you should consider in protecting against it:

1. The first element is understanding the impact ransomware is having across the world and how the attack surface ransomware leverages is expanding. Increased cloud vulnerabilities, as reported by CrowdStrike at a rate of 48%,[6] have expanded the attack surface and overall exposure to ransomware threats. Additionally, the exploitation of Remote Desktop Protocol (RDP) to gain initial access involving ransomware has become a significant entry point. It is also no surprise that a high percentage of cloud accounts for sale on the Dark Web are for RDP access. As information technologies mature, the attack surface proliferates. There are more connections or pathways for ransomware to gain access, more identities, more privileges, and the list goes on and on. Awareness around growth and activities as businesses continue to digitally transform is key to understanding your vulnerabilities to an attack involving ransomware.

2. The second element is understanding that ransomware is a successful business model for threat actors. When you compare the win rates of ransomware in general, the success builds a consistent monetization model for the threat actors. That model is driven by "naming and shaming" businesses. Threat actors will often publicize an attack to a victim's customers and leverage media to coerce the victim to pay the ransom. Cyber insurance also increases the payout, with 50%[7] of the firms in the United States claiming their insurance covers the ransom. In addition, hacking skills are no longer required. Prepackaged exploits and Ransomware as a Service (RaaS) make it simple for the threat actors to capitalize on a ransomware attack.

[6] https://blog.checkpoint.com/2023/01/17/check-point-research-flags-a-48-growth-in-cloud-based-networks-attacks-in-2022-compared-to-2021/#:~:text=Check%20Point%20Research%20(CPR)%20examines,in%202022%20compared%20to%202021

[7] www.bloomberg.com/news/articles/2023-06-14/cyber-insurance-premiums-surge-by-50-amid-ransomware-attacks

3. The third element is understanding that paying the ransom does not necessarily end the threat. Studies have reported that only 14%[8] of organizations that paid for the ransom received 100% of their data back. There is no guarantee that hackers will return all your data, and they could also sell it or even re-encrypt it again. The other thing to consider is the "Blood in the Water" effect; if the root cause, such as a vulnerability, unsecure identity or privilege, or even unsecure remote access, is not addressed, there is the potential for another savvy ransomware operator to exploit the victim again. Effectively mitigating the risk(s) is paramount to ensuring that the same compromise cannot be leveraged twice.

4. The fourth element is understanding that ransomware is mostly malware. It involves various attack techniques linked together and is usually a multistep process involving multiple pieces of malware. Understanding the classes of ransomware is also important because, if you can stop the malware, you can stop most ransomware.

5. The final aspect of ransomware is based on people. Human-operated ransomware is on the rise, and these "hands-on-keyboard" attacks target an organization and actively steer an attack using human insights vs. automation and bots. Recent examples of these attacks include Ryuk[9] and TrickBot.[10] This is in contrast to automated and tailored attacks like WannaCry.[11] Human-operated ransomware attacks are typically initiated via an identity-based attack (social engineering, brute forcing, etc.) and then use lateral movement techniques (excess privileges, poorly managed credentials, etc.) to continue their infection and lateral movement. Having elevated privileges (an administrative

[8] www.cio.inc/whitepapers/esg-report-i-long-road-ahead-to-ransomware-preparedness-w-11527

[9] www.crowdstrike.com/blog/big-game-hunting-with-ryuk-another-lucrative-targeted-ransomware/

[10] www.cisa.gov/news-events/cybersecurity-advisories/aa21-076a

[11] www.techtarget.com/searchsecurity/definition/WannaCry-ransomware?Offer=abMeterCharCount_var3

account), the threat actors can steal and/or encrypt data by deploying a ransomware payload. This is in contrast to using an exploitable vulnerability, typically in the form of a worm, to automate the infection. Automated ransomware can only go so far, but with human operators, creativity becomes the key to the threat actor's success.

Here are the main classes of ransomware that could impact your organization:

1. **Crypto malware or encryptors** that block access to data and applications by encrypting devices.
2. **Lockers** that completely block access to a computer system creating a denial-of-service attack.
3. **Doxware/leakware** that steals sensitive information from your computer and threatens to release it online.
4. **Scareware** that claims to identify other malware like viruses on your computer and then demands money to remove them. The threat itself may or may not even be real.
5. **Ransomware as a Service (RaaS)** where an attacker pays the service operator a subscription fee to leverage ready-packaged ransomware toolkits/malware to leverage in attacks.

Protecting against ransomware is not exactly rocket science, but understanding the shifting attack focus is key in knowing what needs protection. This shift originates from the days of the perimeter network, endpoints, and now the identity. If you can focus on protecting the identity, then the endpoints, then the perimeter, you can hamper the success ransomware is having.

So, why is ransomware winning? As I mentioned earlier, ransomware has gained huge success on the backs of business growth, digital transformation, and poor security practices. Ransomware finds success gaining footholds via things like unsecured remote access through virtual private networks (VPN) or Remote Desktop Protocol. Unmanaged device and application access and a large amount of remote working and vendor access have led to compromise.

In addition, there is the user risk. Phishing and social engineering attacks continue to advance in their sophistication and how they prey on user behavior. With AI now in the picture, attacker capabilities and scale are rapidly improving. Poorly managed

remote access and application control is a huge contributor to the ransomware success, including unpatched vulnerabilities and even poor identity management.

The role of identity security and least privilege is critical in the fight against ransomware—both in preventing an attack from gaining a foothold, as well as during any steps that follow, including performing reconnaissance, execution of lateral movement, and the deployment of more malware.

For example, PAM remote access technology can provide a VPN-less approach that ensures no inbound connections and eliminates unsecured remote access pathways that are used as a primary tunnel to deliver ransomware. Endpoint privilege management for Windows, macOS, and Unix and Linux provides enforcement of least privilege by removing and managing privileged accounts via just-in-time (JIT) access. This minimizes the attack surface and threat windows commonly used by threat actors to elevate privileges that are required to execute ransomware. And to compromise privileges, the hacking of an identity/account relationship is always required.

When you break down the steps of a successful defense against ransomware, it starts with employee training. Security awareness training is key in developing good user habits and warding off human behaviors that could lead to a compromise. Next is vulnerability management. Proactively prioritizing and patching dangerous vulnerabilities reduces the attack surface that bad actors seek to exploit. Identity security follows closely behind by providing controls in a proactive manner that address many ransomware, malware, and attacker targets. Identity Threat Detection and Response (ITDR) is also key in streamlining the detection, followed by a trusty Antivirus (AV) and Endpoint Detection and Response (EDR) solution. Last is a good disaster recovery (DR) plan. Just because you say you are doing backups does not guarantee recovery when it counts most. Make sure you test and validate your recovery steps to ensure that data can be recovered and is intact.

Protecting against known and unknown (zero day) threats is key in any identity security strategy. Consider the following identity security measures as part of a blended defense against ransomware threats:

- Lock down remote access pathways (RDP, VNC, VPNs) and eliminate default remote access protocols when possible.
- Prevent account hijacking by managing all identities, accounts, and credentials using identity security best practices, from IGA to PAM.

- Prevent ransomware execution by enforcing least privilege and using application control as recommended in security best practices, like zero trust.
- Apply context-aware controls to defend against fileless attacks, such as those that exploit trusted applications and embedded application macros.
- Block malicious code from delivering ransomware payloads at the source by shielding your identity from threat actors.
- Ensure credentials and vulnerabilities cannot be exploited to infect other devices on the same network.

CHAPTER 15

Complexity Inherent in the IAM System

There is an inherent complexity in the entire fabric of Identity and Access Management that cannot be ignored during our discussion. This complexity is latent with flaws from multiple vendor interactions, various integration protocols and synchronization mechanisms, and the complex relationship between identities and assets. To better understand these problems, we will dive into a few that cause the most grief within an organization.

Separate Products and Isolated Infrastructure

A significant threat comes from the infrastructure itself. Identity management is a wide and varied technical field. As discussed in earlier chapters, it is often boiled down to the "three legs of a stool", consisting of access management, privilege management, and governance solutions. Sadly, today these fields of expertise are still separate, isolated solutions and technology buying cycles that are deployed and managed separately. Yes, it is true to say that a three-legged stool never wobbles, but the surface can be tilted so badly that risk can be introduced when one discipline is grossly biased over another. If we consider the cybersecurity industry has become continuously complex and a deeply competitive market, identity tools have progressed toward some form of minimal unification and integration. To further our analogy, each of the legs of the stool has taken on some part of the responsibility of the others. Today, a market-leading access management solution will provide some privilege management capabilities. Similarly, a market-leading privilege management solution will deliver parts of the access management value proposition. That said, the unfortunate ongoing state of the union today still requires the average enterprise buyer to buy, install, maintain, and operate

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_15

three products if their use cases are anything beyond basic functionality. Of course, this means more technology, more deployments, and more complexity in order to mitigate modern risks.

For some considerable time, thinkers in and around the IAM space have been forecasting the emergence of a single identity infrastructure stack. With recent big market investment moves driving consolidation at a company and financial level (as of this writing, Ping, ForgeRock, and SailPoint all have the same financial owner), you might conclude that we were on the path to this elusive convergence state. Sadly, however, this financial engineering and company-level consolidation path is unlikely to generate the single identity management infrastructure stack most customers would benefit from. The reasons why are not always financial, but rather the merging of technical debt, strategies, workflows, and even user interfaces becomes a task overwhelming for most development organizations.

In reality, the inevitable specialization requirements indicative of a typical security and identity technology project continue to drive most medium- and large-scale deployments into separate, isolated, and only marginally integrated swim lanes. Inevitably, this means more cost, more complexity, and probably of greatest concern, a trend toward increasingly long project timelines.

All of this leaves us at a disadvantage pf having no single place to see view, manage or control, separate tears of what should be, and in all practical terms are, a single identity infrastructure. The growing importance of establishing a comprehensive identity-centric control plane cannot be met by the products themselves. All this identity technology, all this complex infrastructure, and all of the accounts and privileges embedded within it are part of the attack surface in focus for the adversary.

Doors and Corners

If you're a sci-fi fan (as we are) and have read *The Expanse* series of novels by James Corey (if you haven't done so, you should), the term "Doors and Corners" and the title of Episode 2 in Season 2 will conjure up images of the crusty old Belter cop, "Miller", as his ghostly figure laments the best approach to entering a dangerous room. That reference neatly codifies the perpetual truth that security weakness is always lurking in the forgotten corners of complex systems design. In today's typical identity control plane

implementations, this is especially true. The systems themselves are large, complex configuration machines that, when woven together, form an intricate fabric of potential vulnerabilities. As the use cases cross IAM systems boundaries, they drive sophisticated integration—and many doors and many corners.

Again, these complex integration issues can only be addressed when you take a comprehensive and unified approach to net IAM systems visibility, detection, and response.

Managed Identity Services Platforms

Considering the issues and potential vulnerabilities highlighted earlier, we are now starting to see the emergence of a new tier of the IAM management layer—cake. Here, the identity management service provider steps up to take on more of the complexity of the end-to-end IAM system. Today, we see an increasing customer preference for IAM outsourcing and the delivery of core identity capabilities as a net managed service. Leading identity service providers are rushing in to provide single point management services that span the three legs of the rickety old IAM stool. The smart ones are seeing the potential here for differentiation in their service offerings and are delivering their own net service control plane technology. The idea here is to create a single management overlay that can view, control, and monitor all aspects of the identity infrastructure stack, regardless of the vendor that supplied it.

This type of stand-alone management overlay is nothing new. Those who are familiar with the IBM Tivoli or Computer Associates Unicenter management platforms from the 1990s and early 2000s will fully understand this concept and see how it could apply to great effect here. The repeating pattern manifests in the creation of an application-centric management environment that abstracts the infrastructure to deliver unified management capabilities.

Imagine if you will, rather than seeing your application at the bottom of an identity product embedded in the IAM tools custom user interface, instead the paradigm is turned around to create views like in Figures 15-1 and 15-2.

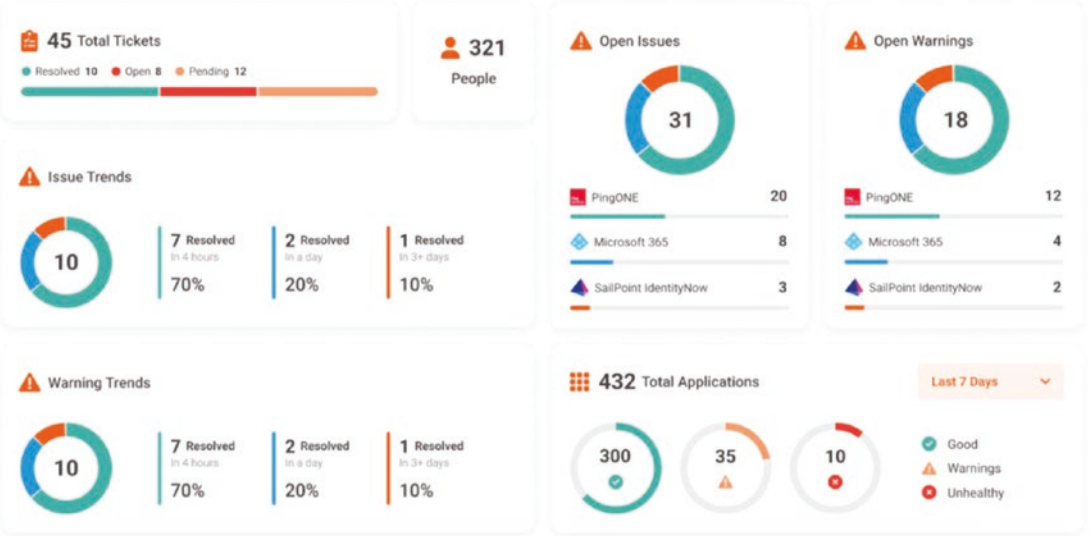


Figure 15-1. *Identity dashboard of anomalies related to identity security with legacy solutions reflected as a summary*

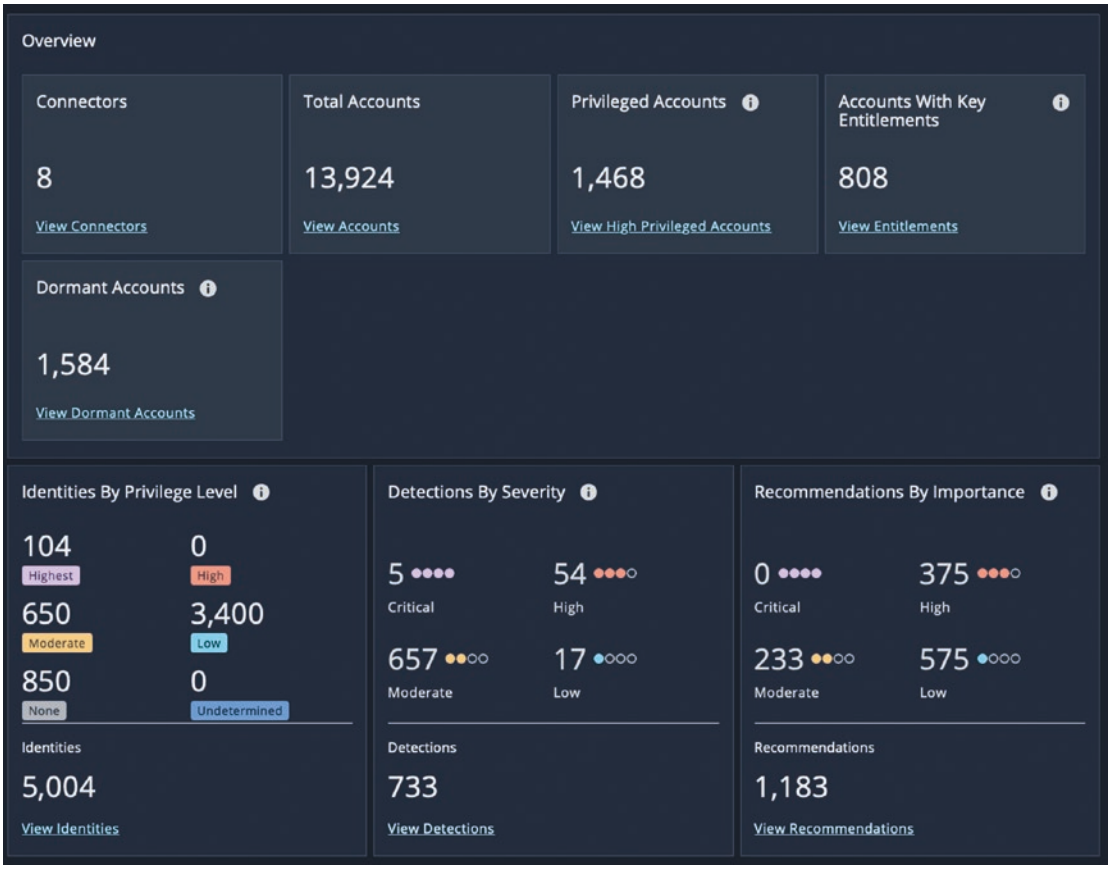


Figure 15-2. *Identity security view of account and privileged risks with an organization as an integrated solution overlay*

Here, the various tiers of the identity infrastructure that support the application are shown underneath it, not on top of it. It's from the application that you derive the status of the identity tooling beneath. So, the new frontier, the new client-focused viewpoint and the truly business-centric way to address the problem here, is to manage multiple IAM tools from a common console, unified dashboard, single control plane, and ultimately a single pane of glass. This still means separate commercial contracts with separate identity specialty vendors, but its delivery and management on top of these solutions provides a unifying value. We believe the delivery of identity security management platforms has the potential to lower the attack surface, identify and mitigate vulnerabilities, and do so while significantly reducing the cost and complexity of a net IAM system deployment by bringing solutions together vs. the overwhelming task of a complete rewrite or integrating decades of technical debt.

CHAPTER 16

Identity Technical Debt

Identity technical debt is kryptonite for the prosperity and longevity of any technology-based business. At some point in time, legacy components, software, and aging assets and resources will no longer meet modern business demands and information security requirements. After the short duration of seven years, many components will be designated as end-of-life and will be queued for replacement or modernization. After all, most endpoint hardware does not even last that long, and we are lucky if back-end systems might last a touch longer. However, some infrastructure resources, like identity governance and access management, may lag behind other initiatives and lead to identity technical debt, simply due to their complexity and scope of deployment. The reasons for this offset are not trivial and are laden with a variety of dependencies—from vendor-update complexities to customized environments that have no path for modernization to the solutions that are simply too cost-prohibitive to maintain.

To begin understanding this risk, let us define identity technical debt:

- "Identity Technical Debt is the aging of identity management tools and role/policy-based access models that can lead to isolation, segmentation, or obsolesce of the identities and accounts they are designed to manage."
- This definition applies to dedicated identity management tooling and role- or policy-based access models that are embedded within solutions. Essentially, the tools or models fail to keep up with modern requirements.

With this in mind, here are some common sources of identity technical debt:

- The legacy remnants of Active Directory when migrating to a new identity service, like Entra ID, due to integrations or the inability to fully decommission the previous solutions for any reason or dependency. The old installation of Active Directory represents identity technical debt since components still remain.

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_16

- The inability to integrate a solution into your identity governance and authentication model due to incompatibility, customization, or lack of SAML, OAUTH, or MFA support. Essentially, the solution operates as an identity island for compatibility reasons, and the supporting components are identity technical debt.
- The effort and cost to lift and shift from an on-premises technology to the cloud for identity management is just not feasible. The proper solution is to start from scratch, and based on budget, politics, technology, etc., this is not achievable. Therefore, the organization is left with a legacy identity governance and access solution to manage modern initiatives, including digital transformation and Web3 projects. The identity technical debt of a Tier 1 project—to manage all identities and accounts everywhere—has become a liability as the organization shifts to the cloud and other newer platforms.

Identity technical debt is essentially managing identities, accounts, and their relationship to the business using outdated technology that creates undue burden on the business to manage and modernize. What makes this situation even more complex is that organizations may be unaware of where identity technical debt exists. When this happens, how do you develop a strategy, even if it is long term, to resolve this dilemma?

To solve this problem, organizations must identify where this technical debt exists. Some common methods include:

- Performing an audit or health check of your IGA (Identity Governance and Administration) solution and determining which technologies cannot migrate to a new identity service provider or be managed when a new project is onboarded, without extensive customization. These commonly include resources that have manual workflows, custom connectors, or even flat file import/export routines for identity and account management that have been integrated manually into your identity governance solution.

- Licensing is an identity detection or identity posture management solution that integrates identity services, authentication, and authorization logs to determine if access is appropriate and centrally managed. Requests that are not authenticated via an identity service may be locally managed using another identity service that represents technical debt or shadow IT.
- A simple review of end-of-life systems that have exceptions granted for business or cost reasons may have an aging role- or policy-based access model. These findings may potentially be incompatible with future changes, policy changes, and modern governance integrations, like SCIM (System for Cross-Domain Identity Management). Unfortunately, end-of-life systems become an island for role-based access or require end-of-life identity governance exceptions that represent identity technical debt.

Once you identify all identities and where their accounts are being managed from, you can determine if you have identity technical debt. These will be the locations that are near end-of-life, past end-of-life, unmanaged, or which have bluntly been implemented with some form of shadow IT. In fairness, some of this discussion is the function of a good IGA tool to provide attestation reporting on identity management. However, as organizations transition to the cloud, embrace new technologies, and solve privileged access challenges with technologies like PAM, having rogue identity services, even for least privilege, becomes a potential problem. Considering the techniques described earlier, ask the following question: "Do we have identity technical debt, and how can we prevent it in the future?" Once you know your answer, then implementing standards like SAML, OAuth, SCIM, FIDO2, etc., for identity management, authentication, and authorization may help you plan and minimize identity technical debt. This is especially true as you try to remove single-factor authentication throughout your organization to mitigate risk. The ultimate goal is not to select new technology or existing standards that could leave you stranded in the future, but rather to integrate and embrace solutions that have a long supportable life span.

CHAPTER 17

Identity Digital Transformations

Have you ever pondered the difference between customization and configuration? It is eerily similar to driving a stock Ford Mustang[1] vs. a Shelby GT500.[2] Both vehicles started in the same manufacturing plant, but in the end, one can be maintained by the dealer, and the other needs a specialized mechanic with expertise in extensive modifications and tuning. While the analogy may sound a little obtuse up front, the real-world implications when deploying a new identity solution or migrating to the cloud could be showstoppers, or an expensive lesson, if not fully understood by your organization. Did you buy a stock solution with out-of-the box functionality or did you license a solution that needs to be tailored in order to fit your business needs? To better unpack this, let's first provide clear definitions:

- *Customization*: The ability to modify or enhance a product to meet desired objectives. This can be achieved by using supported (or unsupported) methods to implement changes and functionality that are not provided by the manufacturer out of the box. This can include custom code or third-party add-ons that perform desired functions.
- *Configuration*: The ability to alter a solution via a user interface, configuration files, command line, etc., to change runtime behavior and attributes to support a business and compliance requirement. Configuration changes are typically supported by the manufacturer,

[1] www.ford.com/cars/mustang/

[2] www.shelby.com/Vehicles/GT500-Signature-Edition

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_17

> or, if not, it is clearly stated that the changes may have undesirable outcomes. Configuration changes generally do not have any custom code to implement or use a software developer's kit for implementation, in contrast to customization.

In years past, organizations typically deployed enterprise identity solutions on-premise and customized the solution to meet business objectives. The configuration of the solution was implemented based on business policies, but the customization allowed it to be tailored to established workflows, use cases, and jobs to be done as best as possible.

Today, consider the cloud. Customization of cloud SaaS solutions is rarely available, and if it is, may inhibit upgrades based on the customizations you may have implemented. While some SaaS vendors allow extensive customization, major changes in their platform often require a complete code review of existing customization to ensure the upgrade will be successful and that intended changes will "hold" in the latest versions. To avoid this problem, most SaaS solutions prohibit extensive customization, and any changes are considered feature enhancements. This does not include any configuration options that may be a part of the SaaS solution, unless the functionality is specifically depreciated or enhanced between versions.

You may be asking yourself, what is the problem here? The answer is that, for most organizations continuing their digital transformation journey for identity security, legacy on-premises implementations that have been customized (not configured) rarely support a migration to the cloud. If they can be adapted, the work is typically extensive, expensive, and complicated rework, unless you cloud wash the entire implementation in a private cloud environment.

Modern SaaS solutions have predefined workflows, preset configurations, and software development kits designed to enhance the solution, but not necessarily customize them extensively to change workflows and add functionality. When upgrading from a legacy system, organizations must grapple with either changing their workflows to meet the SaaS solution's implementation or compromising to some middle ground for the solution to work. The customization that worked on-premises probably does not exist in the cloud and, unfortunately, may not even be possible to recreate. This is a hard truth and far outside the bounds of any configuration that you may implement.

The next time your organization considers another project to move to the cloud, consider customization vs. configuration, especially for identity security. Ask the business:

- What customization has been done to the solution to meet internal business objectives?
- Can customizations be replicated in the cloud without changing existing workflows, required jobs to be done, and use cases?
- Do the configuration settings on-premises have equivalent functionality in the cloud?
- Is the business using a third-party solution to enhance customization?
- What is the effort to migrate customization to the cloud? What are the costs?
- What expertise, internally or through a partner, is needed to recreate any customization?

While this topic may seem straightforward, the real-world implications are profound for any business moving to the cloud for identity management and security. Our experience has demonstrated that even the simplest solutions on-premises can have exorbitant hidden costs when moving to the cloud, especially if these nuances are not addressed up front and need to be created using custom code for your unique business requirements.

CHAPTER 18

Just-in-Time Access Management

The concept of just-in-time (JIT) access management is a strategy that aligns real-time requests for usage of accounts directly with entitlements, workflows, and appropriate access policies. Companies use this strategy to secure accounts from continuous real-time access (known as standing privileges) by restricting them based on appropriate behavior, context, and other ephemeral properties. This decreases the risk of an always-on account that can be leveraged by a threat actor outside of acceptable use policies and procedures. This method requires organizations to establish criteria for just-in-time access and accept that these accounts are not available outside of potentially break-glass scenarios.

While similar concepts for JIT manufacturing are well established, using the model for a security and operations solution, like IGA, does present some technical considerations during implementation. The first is around the just-in-time account delegated for access. An account is granted entitlements, privileges, and permissions only when they are actually needed. Most of the time, this is a privileged account—commonly an administrator account or some special account based on some form of ITSM exception. The goal of a JIT account is to assign the necessary privileges "on the fly" based on an approved task and workflow and subsequently remove them once the task is complete or the window or context for authorized access has expired.

The modeling required to take an account and apply the appropriate privileges can be implemented using the following JIT techniques:

- *JIT Account Creation and Deletion*: The creation and deletion of an appropriate account to meet mission objectives. The account should have traits to link it back to the requesting identity or service performing the operation for logging and forensics. Connectors in the IGA solution can typically manage this requirement.

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_18

- *JIT Group Membership*: The automatic addition and removal of an account into a privileged administrative group for the duration of the mission. The account should only be added to an elevated group when the appropriate criteria are met and subsequently removed when the mission is complete. Connectors based on defined entitlements can typically manage these group membership requests, similarly to how organizational changes are processed.
- *JIT Privileges*: The account has individual privileges, permissions, or entitlements added to perform a mission, but only for a limited duration once all criteria are met. These rights need to be revoked once the mission is complete and should include certification that no other privileges were inappropriately altered. This process can be managed by IGA connectors or integration into a PAM solution.
- *JIT Impersonation*: The account is linked to a preexisting administrative account(s), and when a specific application or task is performed, the function is elevated using those credentials. This is commonly done using automation or scripting with Windows "RunAs" or Unix/Linux Sudo. Typically, the end user is not aware of the impersonation account for this type of operation and may overlap with always-on privileged account delegation. JIT impersonation is normally done only with the integration into a PAM solution.
- *JIT Disabled Administrative Accounts*: Disabled administrator accounts are present in a system with all the permissions, privileges, and entitlements to perform a function. They are enabled to perform a specific mission and then subsequently disabled again once operational criteria have been satisfied. This concept is no different than having always-on administrative accounts accept native enablement functionality leveraged to control JIT access. This functionality can be achieved by IGA connectors or PAM endpoint privilege management solutions.

- *JIT Tokenization*: The application or resource has its privileged token modified before injection into the operating system kernel. This form of least privilege is commonly used on endpoints to elevate the privileges and priority of an application and not the end user themselves. This technology is the cornerstone of endpoint privilege management for PAM solutions.

For any of these account privileges and entitlements to occur just-in-time, the following criteria should be considered as triggers. These triggers should also include variables like time and date for change control windows, and suspension or termination criteria if indicators of compromise are detected:

- *Entitlements*: When privileged access management is integrated with Identity and Access Management solutions, entitlements between solutions can be synchronized for privileged access. To that end, JIT access can be assigned via PAM solutions directly or through IAM entitlements programmatically. While the IAM entitlement workflow is a longer technology process for synchronization and does not occur real-time, it does provide a vehicle for account certifications based on privileges that are void when linking with PAM solutions to control access.
- *Workflow*: The concept of workflow approval is commonly associated with call centers, help desks, and other information technology service management solutions. A request is made for access, and using a defined workflow of approvers, access is either granted or denied. Once the workflow satisfies an approval, a JIT account can be enabled. This typically corresponds to the user, asset, application, time/date, and associated ticket in a change control or help desk solution.
- *Context-Aware*: Context-aware access is based on criteria like source IP address, geolocation, group membership, host operating system, applications installed or operating in memory, documented vulnerabilities, etc. Based on any logical combination of these traits, JIT account access can be granted or revoked to satisfy business requirements and mitigate risk.

- *Two-Factor (2FA) or Multifactor Authentication (MFA)*: A common method for authorizing privileged access to always-on or JIT privileged accounts is 2FA or MFA. While this does not distinguish between the two access techniques, it does provide additional risk mitigation in assuring the identity does have proper access to a privileged account. It can, however, be used as a JIT trigger for an account using any of the techniques listed earlier.

JIT triggers are just that—conditions for an account to be placed in a temporary just-in-time state. They are the same for PAM as they are for IGA organizational changes. They can be used stand-alone or can be logically grouped with other triggers to instantiate privileged account access. The key takeaways for teams to consider are what policies govern a JIT account for proper access and what conditions should be met for its revocation. These could include:

- Time and date windows for access and change control
- Commands or applications that may indicate a compromise
- Detection of access to sensitive information
- Termination of the primary session
- Existence of corresponding collateral in a ticketing solution
- Inappropriate modification of resources, including installing software or modifying files
- Inappropriate attempts at lateral movement
- The manipulation, creation, or deletion of user accounts or datasets

This list isn't exhaustive, but it can, nonetheless, help filter the criteria for a JIT account to be made available based on corresponding triggers. To help illustrate this potentially complex workflow, consider Figure 18-1.

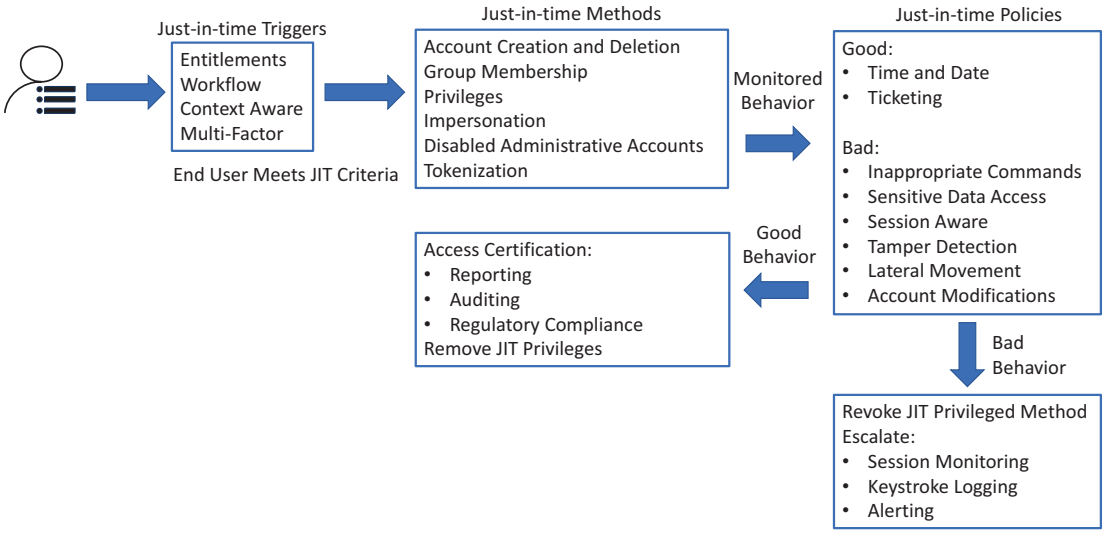


Figure 18-1. *Just-in-time workflows based on triggers, methods, policies, and privileges*

While just-in-time (JIT) access management is not a new concept, the utilization of always-on accounts has been the primary vehicle for administrative access for the last 40 years, but the risks of always-on accounts are expanding. New, highly entitled, and privileged accounts are required for virtual, cloud, IoT, and DevOps environments to administer solutions. The quantity and location create complexity, and this complexity, in turn, often translates into increased risks around security, operational continuity, and compliance. Traditional perimeter-based security technologies can only protect privileged accounts within their boundaries. Privileged accounts are now truly everywhere for your organization. They are even present in the IGA connector technology we need today for a successful IGA deployment. Every one of these accounts is potentially another privilege and identity attack vector, and some of them are accessible directly on the Internet. This is where JIT access management with IGA can make a significant impact in securing your environment from identity attack vectors: by removing standing privileged access and advancing toward a zero standing privilege (ZSP) state.

CHAPTER 19

Zero Trust for Identity Security

Zero-trust principles and architectures are being widely embraced and adopted by the public and private sector as a security best practice. In fact, zero trust may be the best strategy to protect against identity attack vectors.

Legacy security architectures and network defenses are simply ineffective at managing a world more reliant on the cloud and a remote workforce. In this world, zero-trust principles have become one of the most effective approaches to mitigating the heightened risks of highly sensitive identities, assets, and resources.

The NIST SP (Special Publication) 800-207[1] whitepaper has been widely accepted as a framework through which to view zero trust as a technology and security model. NIST refers to this special publication as "a definitive source of ZTA concepts and principles," where ZTA refers to zero-trust architectures. However, as helpful as NIST 800-207 is, it primarily covers zero-trust concepts at a high level. Additional publications, NIST 1800-35B[2] specifically, exist to bridge theory and reality.

To implement zero trust in practical terms, an organization must grasp which technologies and configurations can connect theory to practicality. Zero trust, after all, is not a product or solution, but rather a methodology and philosophy to access and identity security. It should be considered by use case and workflow, and it needs to be architected and measured against each one. That is where this chapter comes in. Figure 19-1 illustrates how an environment should consider end-to-end workflows with zero trust. Every component that interconnects technology uses either human or machine identities and can be a part of a zero-trust workflow.

[1] https://csrc.nist.gov/pubs/sp/800/207/final

[2] https://csrc.nist.gov/pubs/sp/1800/35/2prd-(1)

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_19

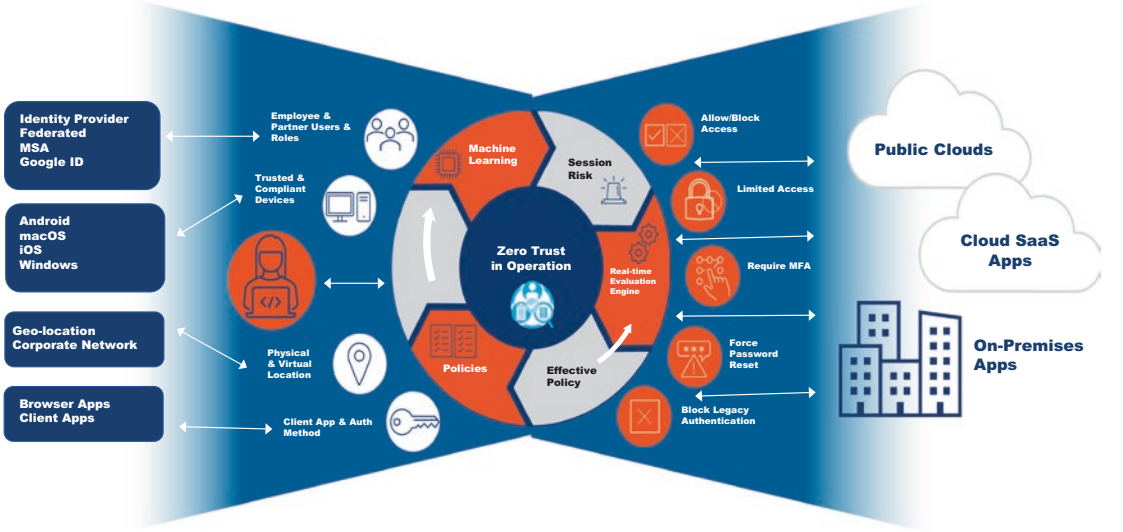


Figure 19-1. *End-to-end zero-trust workflows in an environment*

No identities are more imperative to secure than those with potentially sensitive access to systems, data, applications, and other resources that could be leveraged against an organization. Almost every attack today requires some form of privilege the threat actor can leverage to gain an initial foothold or to laterally move within a network.

Identity security protects credentials, granularly enforces least privilege, and monitors and manages sessions, whether human, machine, employee, contractor, or vendor, for inappropriate user behavior. Identity security is also essential to enable a zero-trust architecture (ZTA) and to address the seven tenets of zero trust, as put forth by NIST. These are illustrated in Figure 19-2.



Figure 19-2. *Seven tenets of zero trust as defined by NIST*

Zero-Trust Details

The National Institute of Standards and Technology (NIST) defines zero trust[3] as "an evolving set of cybersecurity paradigms that move defenses from static, network-based perimeters to focus on users, assets, and resources." NIST further explains that the collection of concepts that comprise the zero-trust principle is designed to "minimize uncertainty in enforcing accurate, least privilege per-request access decisions in information systems and services in the face of a network viewed as contested."

In practical terms, this entails eliminating persistent trust, performing continuous authentication, granularly restricting access to the minimum needed, applying segmentation and micro-segmentation strategies, and continuously auditing access. Identity security is a foundational technology stack for implementing each of these essential zero-trust security controls and to both prevent and mitigate internal and external threats.

[3] https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.20.pdf

The primary goal of zero trust is to protect enterprise resources (particularly, but not solely limited to, data) and allow containment (like a bubble) if something inappropriate occurs. As outlined in NIST SP 1800-35B, this goal has become increasingly important in response to today's network and remote working challenges:

- Networks have become decentralized, perimeter-less environments with resources distributed across both on-premises environments and multiple clouds.
- Many users need access from anywhere, at any time, to any device to support the organization's mission.
- Data is programmatically stored, transmitted, and processed across different boundaries under the control of different organizations to meet ever-evolving business use cases.
- It is no longer feasible to simply enforce access controls at the perimeter of the enterprise environment and assume that all subjects within it can be trusted.

Thus, network location can no longer be treated as the prime component to the security posture of a resource. Instead, the zero-trust model operates under the assumption that no asset or user account can be implicitly trusted based solely on their physical or network location or asset ownership. Identity and device authentication and authorization must be required before establishing a session for an enterprise resource.

Defining Zero-Trust Architectures

NIST Special Publication (SP) 1800-207 defines a zero-trust architecture (ZTA) as "an enterprise's cybersecurity plan that utilizes zero trust concepts and encompasses component relationships, workflow planning, and access policies. Therefore, a zero-trust enterprise is the network infrastructure (physical and virtual) and operational policies in place for an enterprise as a product of a zero-trust architecture plan."

NIST further articulates that the primary focus of a ZTA is "protecting data and resources. It enables secure authorized access to enterprise resources that are distributed across on-premises and multiple cloud environments, while enabling a hybrid workforce and partners to access resources from anywhere, at any time, from any device in support of the organization's mission."

The concepts and principles in NIST SP 800-207 Zero Trust Architecture are applied to "enterprise networks that are composed of pre-established devices and components and that store critical corporate resources both on-premises and in the cloud."

According to SP 1800-35B, ZTA performs real-time, continuous behavioral analysis and risk-based assessment of the access transaction or session. "For each access request, ZTA verifies the requester's identity and role, the requesting device's health and credentials, and possibly other information based on session, device, and activity attributes. If defined policy is met, ZTA dynamically creates a secure connection to protect all information transferred to and from the accessed resource." Each access request is evaluated by verifying:

- The context available at access time, including the requester's identity and role
- The requesting device's health and credentials
- The sensitivity of the resource

Figure 19-3 provides an example of a ZTA, as outlined in NIST SP 1800-35B.

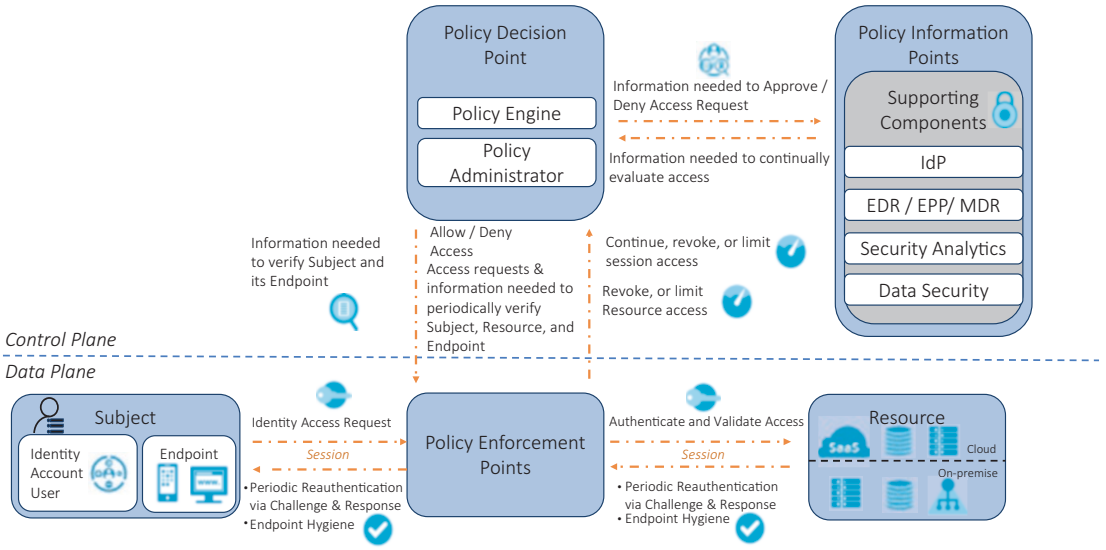


Figure 19-3. *Sample zero-trust architecture from NIST 1800-35B*

The core ZTA components depicted in the architecture diagram include:

1. *Policy Engine (PE)*: The PE handles the ultimate decision to grant, deny, or revoke access to a resource for a given subject. The PE calculates the trust scores/confidence levels and ultimate access decisions based on enterprise policy and information from supporting components. The PE executes its trust algorithm to evaluate each resource request it receives. In simple terms, it is the logic engine that processes policy to make enforcement decisions.
2. *Policy Administrator (PA)*: The PA executes the PE's policy decision by sending commands to the Policy Enforcement Point (PEP) to establish and terminate the communications path between the subject and the resource. It generates any session-specific authentication and authorization token or credential used by the subject to access the enterprise resource.
3. *Policy Enforcement Point (PEP)*: The PEP guards the trust zone that hosts one or more enterprise resources. It handles enabling, monitoring, and eventually terminating connections between subjects and enterprise resources. It operates based on commands received from the PA. In simple terms, it provides enforcement of policy at a layer where transactions actually occur.

Other important components include

- *Policy Decision Point (PDP)*: The PE and PA combine to comprise the PDP, which executes the decision of whether a subject is permitted access to a resource.
- *Policy Information Point (PIP)*: The PIP provides telemetry and other data to enable the PDP to make informed access decisions.
- *Subject*: An end user, application, and other nonhuman entities that request information from resources.
- *The Gateway*: Responsible for enabling, monitoring, and terminating the connection between the subject (user or application) and the resource via the agent, so all activity can be assessed and documented.

Zero-Trust Architectural Models

As a security best practice, network and access controls should be configured, tested, and monitored for traditional on-premises resources to prevent inappropriate access. Unfortunately, in many modern environments, these security controls have been loosened or bypassed to support myriad use cases—from hybrid cloud environments to remote workers to solutions that span multiple geolocations and untrusted networks. This does not mean that the security controls have been removed, but rather that enough exceptions have been implemented to enable additional business use cases, and the added risks have been accepted in order to provide business functionality. This "unhardening" of the environment runs contrary to the zero-trust model and, in many cases, uses static credentials or secrets for solutions to interoperate. Thus, a compromise in one location jeopardizes the entire environment that previously had strong segmentation.

To achieve zero trust, while supporting these novel use cases, a new perimeter must be established. This entails a zone-based approach to creating a secure enclave. Access into the enclave itself, as well as between resources that reside within it, should be tightly restricted.

An enclave is like a network perimeter within an unsecure network, and the containment, with all the security controls, is built within. As a simple analogy, everything is a bubble (enclave), and bubbles range in size from very small, for a single asset or application, to very large, encompassing a complete system. Figure 19-4 illustrates this containment compared to a traditional, network-based security model.

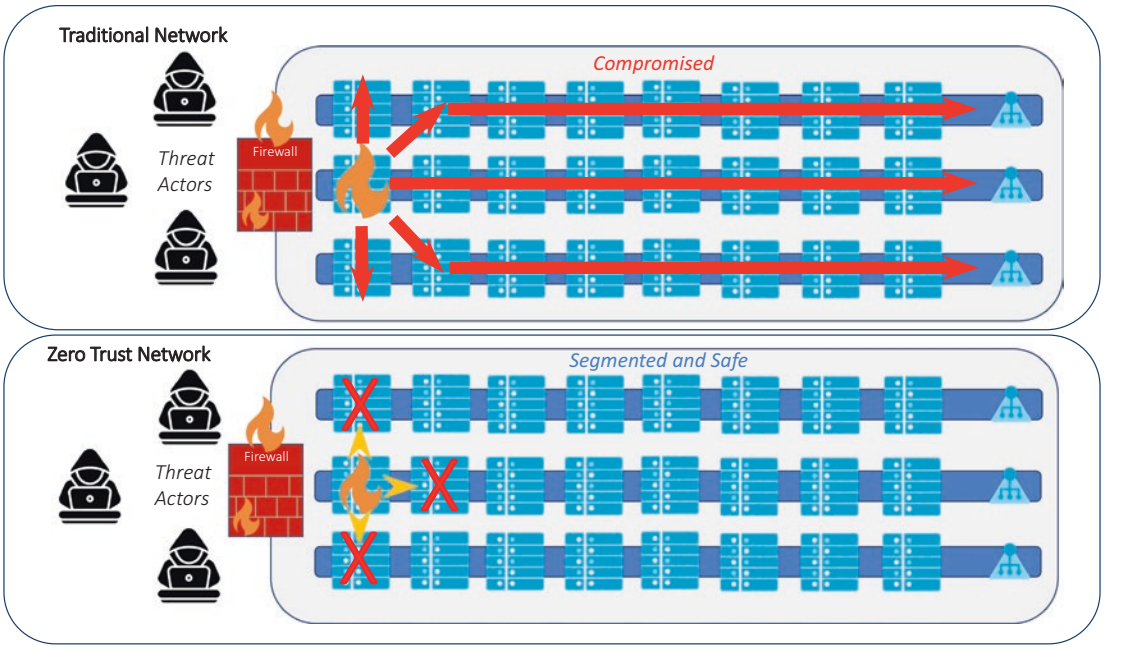


Figure 19-4. *Zero-trust exposure vs. network-based security controls*

A zero-trust enclave model establishes granular segmentation. While resources within the enclave can have loosened security controls to meet the operational business requirements, they are still monitored for inappropriate behavior when activity originates from outside the enclave. Think of the enclave as a mini trusted network (bubble) within another network.

Gaining access to the enclave requires security controls beyond what is enabled via a typical bastion host or restricted network tunneling and routing controls. This is key for establishing segmentation. The necessary requirements to enable zero-trust functionality within an enclave include:

- An external resource holds the policy and administration for all access.
- Access is brokered by a logical engine that processes the policy and attributes to determine or limit access.
- Access is fully documented, from authentication to complete session management and session recording.

- Sessions are monitored for behavioral activity, from inception through any required lateral movement, to complete a task. Inappropriate activity can be responded to in real time and via automation.
- Access is not absolute and should follow workflows like just-in-time, including ephemeral connectivity and integration into service desks and other change management solutions.
- No sessions or traffic can access the enclave unless processed through the control plane. In other words, all lateral movement into the enclave should be denied using traditional network and access security controls. The enclave should operate as a secure segment of the network that is fully gated and lacks back doors.

Success with this model demands a balance between all these requirements. Authorized users and applications must initiate a session that can be verified and connected into the enclave. Session monitoring, credential injection, and least privilege must be applied to overcome the security and compliance concerns governing an organization consistent with zero-trust objectives.

In addition, these capabilities must be enforceable, whether the user or application is operating from the corporate office or from a remote location. This paradigm can help a zero-trust architecture play a pivotal role in overcoming legacy network challenges when a broad network perimeter is no longer used for security enforcement.

As we consider the concept of an enclave, NIST guidance does provide variations in models to meet different business and technology objectives. For example, the resource enclave is a variation of a network zone. It is a collection of resources (applications, operating systems, network devices, databases, etc.) with a hardened perimeter around all the assets. In lieu of a perimeter being a broad network zone, it is isolated to only critical assets within the resource enclave for a given purpose. Having a PCI (Payment Control Industry) zone is a simple example of this resource model. Essentially, a resource enclave is a secure network zone with limited external access and is fully segmented.

In any organization, there will potentially be dozens, or hundreds, of resource enclaves secured by traditional access control lists (ACLs) and network access hardening. This represents an expansion of the micro-segmentation concept to unique implementations to meet business or technology requirements. A resource enclave is adaptable to legacy systems, and the architecture is compatible with zero trust.

Visualizing all these concepts helps solidify the validity of this approach. Resources within the enclave rely on traditional security control models and do not necessarily mitigate threats (i.e., inappropriate lateral movement) within the enclave itself unless connectivity is fully monitored. For example, threats from lateral movement are only mitigated between subjects and from resource enclave to resource enclave. Figure 19-5 illustrates these concepts based on communication pathways.

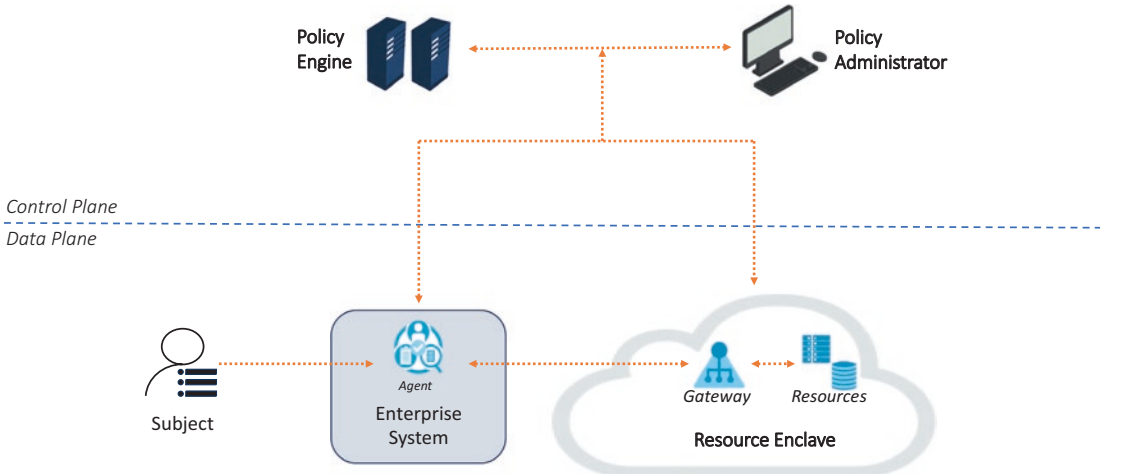


Figure 19-5. *Basic zero-trust reference architecture*

Least Privilege and Zero Trust

Privileged Access Management (covered in depth in Chapter 13) consists of cybersecurity strategies and technologies for exerting control over the elevated ("privileged") access and permissions for users, accounts, processes, and systems across an IT environment. Use cases range from enforcing least privilege to managing privileged accounts/credentials to securing all privileged remote access. PAM is a fundamental security control and discipline that falls within the broader identity security umbrella. PAM controls are also necessary to enable and achieve zero trust.

Today, enterprises and public agencies are challenged with securing significantly more privileged accounts and supporting vastly larger remote, hybrid, and decentralized networks than in years past, and many of them are not even considered traditional privileged accounts operated by administrators or system owners. We have more tools, users, and machines requiring privileged access than ever before. Resources

and privileged accounts can be provisioned at great scale across cloud environments. However, even ephemeral resources can create a risk surface and must be appropriately accounted for with security controls.

Increasingly, resources that require authentication, privileges, and access may reside outside of corporate governance in our always-connected world. This can include other untrusted resources or identities, accounts, and processes. These realities have given rise to the concept of the Data Plane, which is important to manage via ZT (zero trust).

Simply defined, a zero-trust architecture (ZTA) addresses modern privileged access security challenges by enforcing granular, secure, authorized access near the resources, whether located on-premises or in the cloud, for a remote workforce and partners based on an organization's defined access policy.

When applying the granularity of privileged access management (PAM), zero trust can ensure all access is managed and documented for appropriate behavior. Today, this is a particularly important challenge to solve in order for organizations to meet regulatory compliance requirements as well as cyber insurance eligibility requirements.

As envisioned by NIST (SP 800-207), a zero-trust security model eliminates persistent trust and enforces continuous authentication, least privilege, and adaptive access control. This strategy also applies segmentation and micro-segmentation for secure access. A zero-trust approach is about constant visibility and control over who is doing what on your network.

A modern PAM solution supports the smart, practical implementation of NIST's zero-trust security model—all without disrupting business processes. At a high level, PAM can provide zero trust with the following capabilities across on-premises and cloud environments:

- Discovers, onboards, and catalogs all privileged accounts and assets. Inventory is crucial for zero trust.
- Enforces adaptive access and continuous authentication to ensure all devices, users, and identities who have access to a network are who they say they are with confidence.
- Right-sizes privileged access and entitlements by applying least privilege, including just-in-time access, to all sessions, applications, and assets from vetted endpoints.

- Enables secure least privilege remote access for vendors, employees, and service desks to the solutions that are needed to complete a mission.
- Implements segmentation and micro-segmentation to isolate assets and users and prevent lateral movement, which is especially critical should a security incident occur.
- Monitors and manages every privileged session, providing continuous visibility and control over who is doing what, and why, so any suspicious behavior can have permissions and access revoked immediately.
- Extends Microsoft Active Directory authentication, Single Sign-On capabilities, and Group Policy configuration management to Unix and Linux systems, simplifying the secure management of identities and the path to implementing zero trust enterprise-wide.

Together, these capabilities provide high levels of control, flexibility, and adaptability while vastly minimizing the threat surface, reducing lateral movement, and providing continuous oversight over identities and access. Unlike other security technologies, PAM provides robust protection whether the threat originates on the outside or from within.

Table 19-1 illustrates how NIST zero-trust concepts and architectures map to actual PAM disciplines, like password and secrets management, remote access, and least privilege management.

Table 19-1. *How PAM fits within the NIST (National Institute of Standards and Technology) zero-trust architecture*

Identity Security Solutions	NIST ZTA Core Concepts Policy Engine (PE)	Policy Administrator (PA)	Policy Enforcement Point (PEP)	Gateway
Privileged password and secrets management (passwords, keys, and certificates including session and behavioral monitoring with just-in-time access)	Found in the management capabilities governing secrets management and the role and attribute-based access models defined by the Policy Administrator. A PAM policy engine decides on access availability based on a wide variety of criteria and tracks privileged access—even if it is ephemeral	Creates, updates, and manages the policy for end users and machine identities to grant access and automate a privileged session. Access to the resource enclave is granted to the Policy Administrator and can be managed through a PAM solution's management console		Responds to a privileged access request for a session or application from a trusted policy engine. This is fundamental to zero trust since a user is never granted direct access to a resource when using a native operating system protocol. All connectivity is managed and monitored by the agent, and only session activity (screen, terminal, or web page) is rendered to the requesting subject

(*continued*)

Table 19-1. (*continued*)

Identity Security Solutions	NIST ZTA Core Concepts Policy Engine (PE)	Policy Administrator (PA)	Policy Enforcement Point (PEP)	Gateway
PAM remote access solutions (on-premises zone-to-zone, hybrid cloud, and untrusted network access)	Found in the management capabilities governing remote access and the role and attribute-based access models defined by the Policy Administrator. A secure remote access solution can manage assets and users in any network zone, regardless of perimeter, as long as Internet access is available	Creates, updates, and manages the policy for end users, grants access, and automates application remote access; this is the basis for zero trust. Access to the resource or application is granted to the Policy Administrator and can be managed through the remote access solution's management console	Implemented using a remote access bastion host. It responds to a remote access request for a session or application from a trusted policy engine or intermediate proxy and enables a jump technology to route the communication. A user is never granted direct access to a resource as when using a native operating system protocol. All connectivity is managed and monitored, and only session activity (screen, terminal, or web page) is rendered to the requesting user	

Least privilege management/ endpoint privilege management (Windows, Mac, Unix, Linux, etc. Removal of administrative rights and just-in-time elevation)	The policy engine can be found in management capabilities of the rules, policies, and log engine governing least privilege endpoint access and in the role and attribute-based access models defined by the Policy Administrator	The Web Policy Administrator creates, updates, and manages the policy for end users, grants access, and automates application access. This is the basis for zero trust. Access to the resource or application is granted to the Policy Administrator and can be managed through the Policy Administrator	Policy Enforcement Point is the least privilege client installed on Windows, macOS, Unix, and Linux endpoints. For Windows and macOS, it initiates privileged applications and performs application control on the endpoints on behalf of the user or application. The Policy Enforcement Point capability compares the application execution request to the defined policy and launches (or denies) the application with the appropriate privileges, without actually using privileged credentials. This capability is fundamental to zero trust since the application or user is never granted administrative credentials but can execute an application with privileges. This is why PAM is essential for least privilege and an integral part of zero trust. For Unix/Linux, it initiates commands on the host on behalf of the user or application—without the end user actually logging in—and renders the results transparent to the end user

Privileged Account Control

Organizations have experienced an explosion of privileged accounts (human, application, machine, etc.) that must be managed. These privileged accounts are supporting on-premises technology, the cloud, hybrid environments, and many of the SaaS, IaaS, and PaaS solutions that help power modern businesses.

Privileged credentials and other secrets must be managed using a model that can support your modern environment. Access to passwords and secrets must address human and machine identities, whether on-premises or in the cloud or being licensed from a vendor, supplier, or solutions provider.

Privileged password management (also known as privileged account and session management) provides an optimal way to architect access to sensitive resources.

A PAM solution providing privileged password management can ensure your resources are managed and protected from potential inappropriate connection abuse, and can ensure all resources contained within an enclave are executed within a zero-trust model. This means no end users or machine identities are ever trusted for a direct privileged session unless their access can be brokered through a gateway. All session activity is fully monitored. This holds true for any location in which a resource enclave may reside, irrespective of the perimeter. Consider the following use cases that zero trust and privileged account management must observe:

- Securely storing and allowing for retrieval of passwords, credentials, and other secrets to provide access to human and machine identities, without exposing them to the wild, end user, or allowing them to be leaked or copied.
- Automatically rotating passwords on resources based on workflows, scheduling, and on-demand and by providing a feature-rich API (application programming interface) to enable other automation based on least privilege.
- Providing a historical record of passwords and secrets for access to backups, virtual machine snapshots, and disaster recovery environments using a highly secure workflow to prevent tampering.

- Enabling complete session management, including monitoring, recording, playback, and analysis of sessions in both real time and from a historical perspective for compliance, auditing, forensics, and training purposes.
- Allowing access only through a secure gateway technology to prevent or monitor lateral movement and broker access via discrete components to enumerate and enforce an appropriate access policy.
- Allowing for the privileged password management architecture to be applied to existing environments, with minimal distribution. When properly implemented, it should actually streamline and enhance end-user access.

Based on the design put forth in this chapter, these use cases can follow the model of least privilege and just-in-time access, integrate with ITSM solutions, and help achieve the benefits of a zero-trust enclave deployment for administration and access.

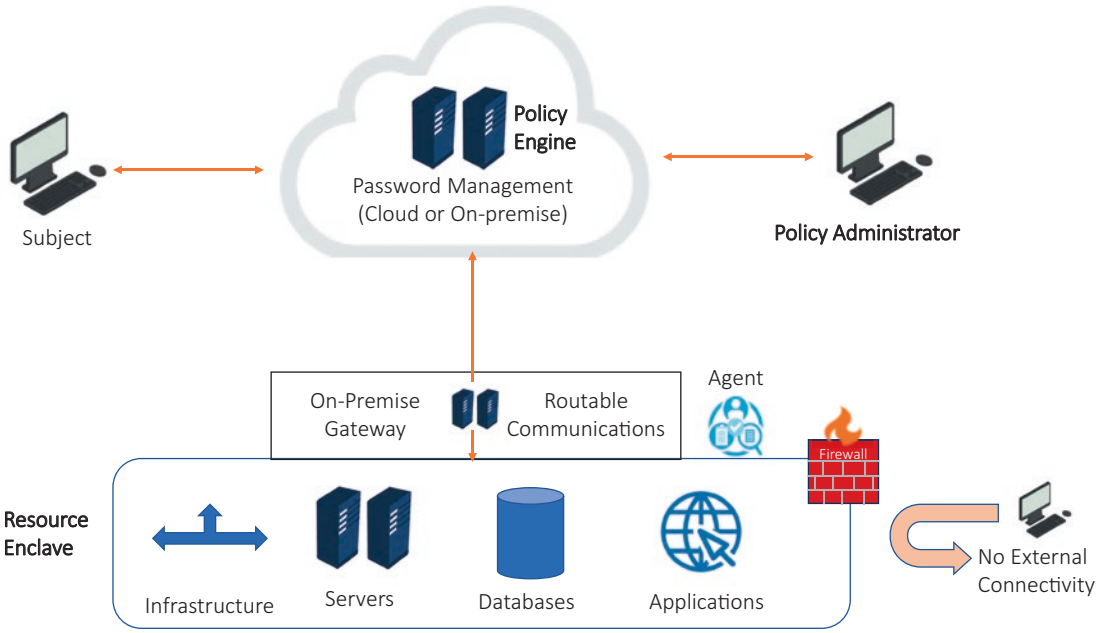


Figure 19-6. *Privileged password and secrets management applied to zero trust*

In Figure 19-6, the ZTA consists of a password and secrets management console and REST API to assist with automation and nonhuman accounts. The agent's role is tightly coupled to the policy engine to provide localized enforcement. Its purpose is to release the password or secrets from encrypted storage to the subject and to perform checks and rotation on existing managed resources. This includes accessing historical passwords and secrets and managing decisions from the policy engine, including the duration of access.

Any partial implementation of this model can be an improvement for privileged password and secrets management when considering a software-defined perimeter. This architecture is much more secure than allowing unrestricted access to any resource from any place using static credentials—especially if the password or secrets are single-factor and reused.

Remote Access

As a security best practice, native remote access protocols should be disabled for corporate-issued computing device(s). Unfortunately, in many environments (especially for users working from home), this security control has either been bypassed or not implemented, and remote devices may be accessing corporate resources using inadequately secured remote access pathways directly or via VPN.

The rationale behind enabling protocols like RDP, SSH, and VNC has been a source of contention between IT and infosec teams. One argument is the need for low-cost remote access technology natively supported by the operating system. This is generally advocated by IT. Security and compliance teams, on the other hand, are wary of the inherent vulnerabilities, wormable exploits, and lack of auditing and secure network routing of native protocols. Such protocols are a high risk when used from untrusted networks and devices, and they can potentially be accessed by anyone on the Internet.

There must be a balance between these approaches. Authorized users need to initiate a secure remote access session to any device, any place, regardless of protocol. In addition, session monitoring, credential injection, and least privilege must be applied to overcome the security and compliance concerns governing an organization. These capabilities must be in place whether the employee is working from the corporate office or from a remote location.

A zero-trust architecture for remote access can play a pivotal role in overcoming native remote access protocol challenges. A zero-trust remote access implementation can accommodate almost any environment and allow a just-in-time (JIT) workflow with zero standing privileges[4] (ZSP).

Figure 19-7 illustrates risks based on remote working within a decentralized, perimeter-less environment and reflects the combination of assets, connectivity, remote access technology, and prime locations for a threat actor to infiltrate an environment.

In addition, based on normal connectivity, applications could operate remotely or using a remote session. To secure all activity, connectivity itself must be secured and all access provided only via a secure remote access technology.

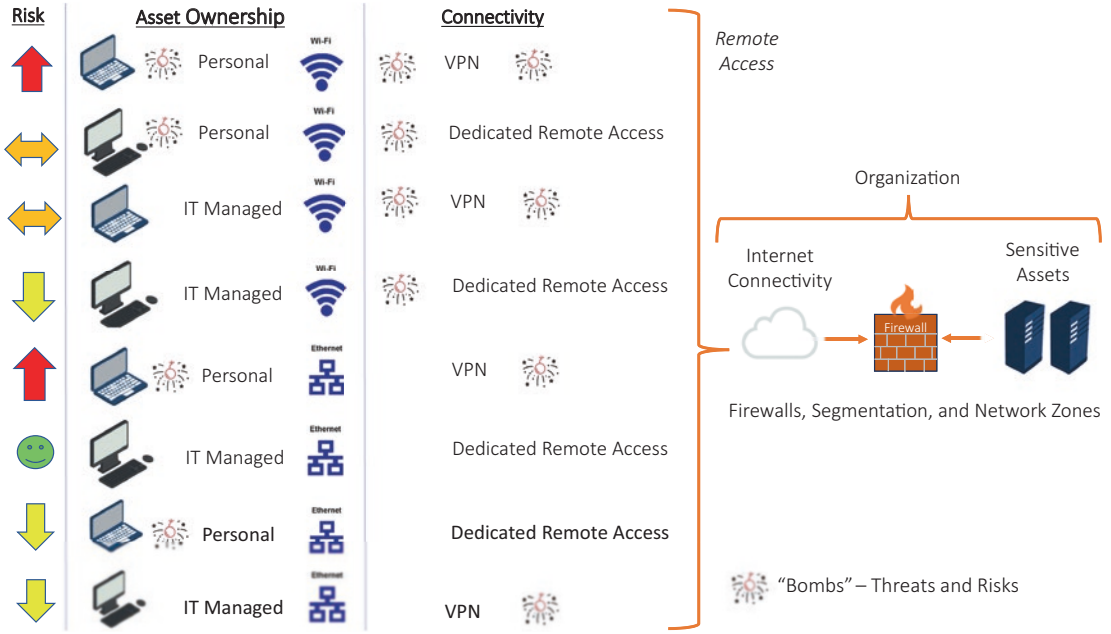


Figure 19-7. *Risks with remote worker connectivity*

In Figure 19-7, each "bomb" represents a risk:

- *Three Bombs*: Unacceptable, critical risk
- *Two Bombs*: Medium level of acceptable risk

[4] www.beyondtrust.com/resources/glossary/zero-standing-privileges#:~:text=Zero%20Standing%20Privileges%20(ZSP)%20refers,elimination%20of%20all%20standing%20privileges.

- *One Bomb*: Low risk for remote access
- *Zero/No Bombs*: Best case for acceptable remote connectivity

Note that using a personal device with a business-issued VPN client is always a critical risk, regardless of whether the connection is wired or wireless. This is because the device is unmanaged and the organization has no control over how it is used, updated, or operated.

In this decentralized environment, threats exist when accessing sensitive internal resources from:

1) Personal or Bring Your Own Device (BYOD) hardware that is unmanaged, unpatched, multi-user, end-of-life, or may otherwise be susceptible to phishing or malware. In addition, BYOD users are typically their own local administrators, amplifying the risk.
2) Unsecure home networks based on Wi-Fi connectivity where the connection is potentially insecure, has a weak password, is wide open, or may allow a man-in-the-middle attack due to a common SSID or poor encryption. Also, other devices could compromise the wireless network or monitor communications. This includes privileged accounts outside of corporate governance used by home networks accessing consumer SaaS solutions.
3) VPN technology, which typically uses split tunneling and should never be installed on personal devices that could compromise communications and provide a conduit for lateral movement via the flaws in the home network. Because VPN technology only operates at the network layer, it is unable to monitor remote sessions. And typically, remote users only need application access (application layer).

To mitigate these threats, a combination of zero trust, IT-managed devices, identity security, and privileged access management can succeed where traditional technology alone may pose unacceptable risk:

- *IT Managed*: Managed security controls for risk assessment, including core disciplines for vulnerability and patch management.
- *Connectivity*: Minimizing network risk with a wired connection in lieu of unknown wireless connectivity.

- *Privileged Access Management*: Remote access sessions are initiated at the application layer based on role, including credential obfuscation, eliminating the need for network layer traffic per application, per user. Privilege elevation is strictly controlled locally and across the network, thereby eliminating local administrative credentials and admin rights for end users.
- *Identity Security*: Documenting all activity for compliance, including remote access user behavior by asset and identity, regardless of human or machine.
- *Zero Trust*: Implementing a cloud-based management architecture for all remote access sessions honors zero trust, and strict application control is enforced. This applies the concept of zero trust with least privilege to all sessions, regardless of their origin or destination. It ensures the risks can be fully mitigated by never exposing root or administrative privileges outside of the extended perimeter nor to the end user.

This combination of technologies and strategies works because you are not only securing the source device, but also minimizing network risk with a wired connection, strictly controlling remote access, not performing any protocol routing like SSH, and recording all session activity for compliance and behavioral analysis.

Finally, applying the concept of zero trust with least privilege ensures the risks can be fully mitigated by never exposing root or administrative privileges outside of the perimeter. VPNs themselves cannot achieve this without the use of privilege management solutions.

To that end, VPNs cannot effectively manage risks outside of a defined perimeter. For zero trust to succeed, the network and environment need to be secured before a ZTA can be implemented.

A zero-trust remote access implementation should be designed to manage sessions and, with a few considerations, can be implemented using a zero-trust model. The solution should extend password management best practices beyond the perimeter. This model should enable organizations to apply the least privileged and robust audit controls to all remote access required by employees, vendors, and service desks. Users should be able to securely access any remote system, running any platform, located anywhere, and should be able to leverage integrated password management capabilities to discover, onboard, and manage privileged credentials.

Realistically, not all native protocols can be removed, especially when IoT, ICS, and OT (Operational Technology) are involved. A zero-trust remote access solution using the same gateway technology should be able to perform protocol translation from one remote access protocol to another (even proprietary) to minimize the risk of exposing native protocols on the Internet or via VPN. In addition, this capability should combine native workflows, zero-trust principles (like JIT and ZSP), and a comprehensive audit trail, regardless of remote access paradigm.

For technology workers, like developers and cloud ops engineers, that believe they need to use VPNs (and sometimes multiple are needed in disparate environments), the resistance to zero trust and remote access technology can be fierce. While VPNs are encrypted and are generally secure, they do offer broader access than is typically needed. Additionally, from a workflow perspective, the user needs to remember what VPN should be used for each specific device (based on network access), and they must know credentials for that system. The potential broad access alone violates the segmentation portion of zero-trust tenets and could allow lateral movement during an attack.

With a zero-trust remote access architecture, the user can use a single console (full client or browser), leverage their native tools (e.g., SQL, SSH, and RDP), inject credentials from a secure password manager, and generate a full (and consolidated) audit trail for every session. This methodology gives users their desired workflow for managing across infrastructure, and it also gives the organization the airtight security needed during an audit. Consider the following traits for a zero-trust remote access implementation to protect identities:

- Applies least privilege and robust audit controls to all remote access for employees, vendors, contractors, and service desk personnel.
- Manages and automatically injects credentials into remote sessions, so the end user never sees or has knowledge of them for appropriate usage.
- Secures infrastructure access and implements a secured bastion host with multifactor authentication, adaptive authorization, and session monitoring for administrator consoles. This also applies to access that crosses trusted network zones.
- Provides access to web pages, such as the Azure or Microsoft 365 portal, through a locked-down and embedded browser.

- Enforces boundaries between development, test, and production systems for DevOps security best practices.
- Provides application-level micro-segmentation that prevents users from executing applications and other resources they are unauthorized to access.

A zero-trust privileged remote access implementation can provide the advantages over VPN alone, as documented in Table 19-2.

Table 19-2. *How a zero-trust remote access architecture compares to a typical VPN*

Use Case	Virtual Private Network (VPN)	Identity Security Solution
Remote Access	Y	Y
Secure Connectivity	Y	Y
Network Layer Access (Protocol Tunneling)	Y	Y
Encrypted Traffic	Y	Y
Application Layer Virtualization		Y
Remote Desktop Access		Y
Proxied RDP Access		Y
Proxied SSH Access		Y
Proxied VNC Access		Y
Proxied HTTP(s) Access		Y
Application Session Monitoring		Y
Application Session Recording		Y
Just-in-Time Access	Y	Y
Zero-Trust Architecture	Y	Y
Privileged Access Management Integration		Y
Secure BYOT (Bring Your Own Terminal)		Y
ITSM Integration		Y

(*continued*)

Table 19-2. *(continued)*

Use Case	Virtual Private Network (VPN)	Identity Security Solution
Password Management and Credential Storage		Y
Cloud Deployment	Y	Y
On-Premise Deployment	Y	Y
Agentless Access		Y
Operating System Agnostic		Y
Prevention of Lateral Movement		Y
Application and Session Auditing		Y

To make this become a reality, consider applying this model to remote access sessions, regardless of where they are deployed and operate from or to. Consider Figure 19-8 of a simplified NIST-based zero-trust architecture for remote access.

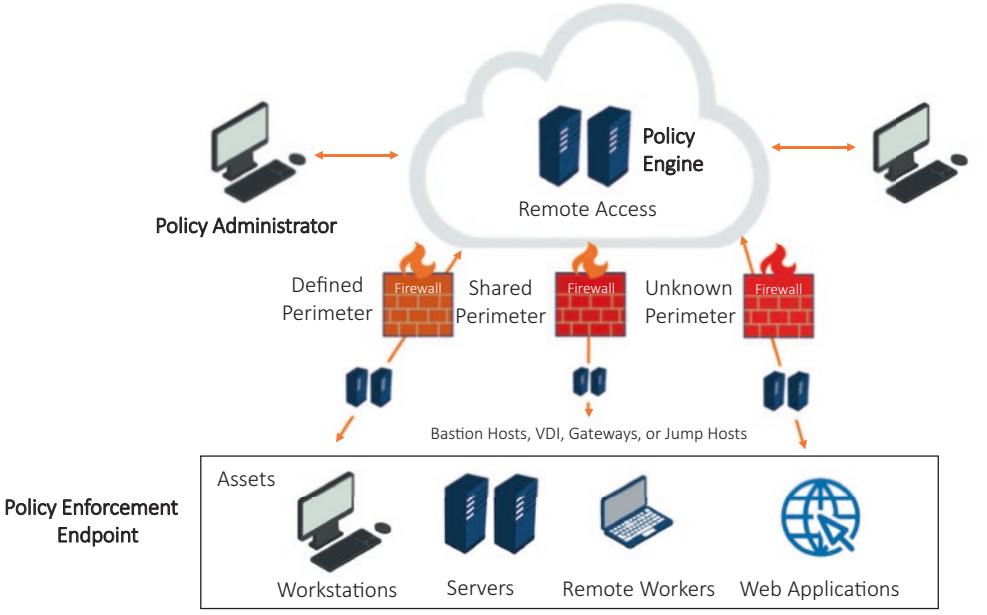


Figure 19-8. *A zero-trust architecture for remote access technology*

Based on this design, all remote sessions honor the model of least privilege and just-in-time access, integrate with ITSM solutions, and follow a zero-trust architecture for policy and administration. Any partial implementation of this model can be an improvement in secure computing for a software-defined perimeter. This architecture is much more secure than allowing a home computer with VPN access into your environment to perform administrative functions.

Zero-Trust Least Privilege

For nearly all organizations, the simplest way to take control of their Windows and macOS endpoint estates is to implement least privilege by removing standing local administrative rights from their end users. Unfortunately, many organizations have not implemented this key security control to its full effect, leaving them vulnerable to malware, ransomware, and lateral movement–based attacks.

To successfully implement least privilege, the organization must find a solution that allows the following two goals to operate in harmony:

- *Productivity*: Many tasks that end users need to complete in their day-to-day work require admin rights, including installing applications or changing system settings. If their administrative rights are removed without the right solution in place, their productivity can be completely diminished.
- *Security*: The organization must remove all standing local administrative rights, exercise control over which applications are allowed to be installed or run by different end-user roles, and limit which tasks end users can execute on their endpoints to reduce their threat surface.

These goals underscore the importance of a ZTA for endpoint privilege management, especially given the flexible modern work environment. Adopting the right endpoint privilege management solution as a part of a zero-trust architecture can accommodate corporate office-based and work-from-home environments and can close the gap often found between managing privileges and applications, all while achieving strict governance over authentication.

An endpoint privilege management solution elegantly achieves both goals by managing applications and identities based on privileges and performing these functions securely, without the need for the end user to reenter administrative credentials when tasks require elevation. The strategy removes excessive administrative rights and applies modern application control while dynamically elevating rights to end users, so they have just the access they need at just the right time. This functionality is crucial and must be achieved if the session is air-gapped, offline, remote, or present on the corporate network, since zero-trust principles are designed to be perimeter-less.

For a zero-trust least privilege solution to be effective on the endpoint, consider the following characteristics:

- Enables zero-trust security by removing excessive admin rights and eliminating persistent privileges.
- Applies granular application control and enforces least privilege across applications, web browsers, systems, and other resources by giving users just enough access at just the right time.
- Stops attacks that take advantage of trusted applications, like Office, Adobe, and web browsers, by applying built-in, context-based security controls.
- Drastically reduces the cyberattack surface and protects against malware, ransomware, and phishing attacks that need administrative privileges to execute.
- Provides a single audit trail of all user activity with graphical dashboards and reports.
- Provides application control by enforcing child process execution and privileges assigned to child processes.

This zero-trust approach combines endpoint privilege management and application control to minimize the endpoint attack surface and eliminate unwanted lateral movement across your entire heterogeneous environment. Consider Figure 19-9 for this architecture.

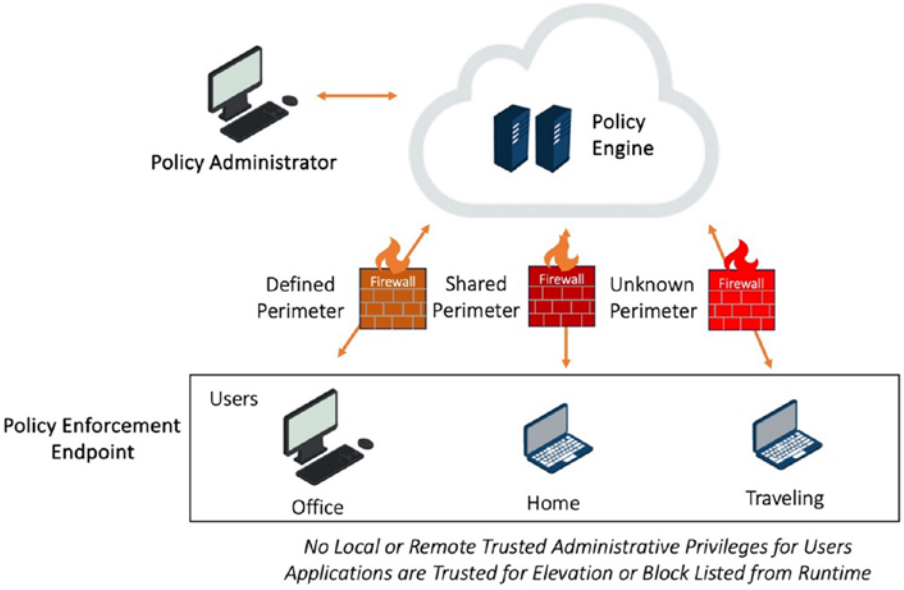


Figure 19-9. *Zero-trust least privilege and endpoint privilege management architecture*

If you are considering rearchitecting, redeploying, or modernizing your endpoint security model, you can achieve zero trust for privilege elevation, least privilege, and application control using this paradigm. This model satisfies all the NIST zero-trust requirements and allows endpoint privilege management to extend the implementation to additional security models, such as just-in-time access. This is especially true if the policy engine and management components are in the cloud to truly allow for a perimeter-less implementation and operations anywhere.

Any partial implementation of this model can be an improvement in secure computing for a software-defined perimeter. This architecture is much more secure than allowing a home computer with VPN access into your environment to perform administrative functions.

Next, let us discuss non-Microsoft and Apple assets. Unix and Linux administrators rarely, if ever, physically operate the keyboard directly behind their asset, if one is even connected at all. This is nearly always true when operating within the cloud.

Administration, and even root access, is granted to the individual based on an identity/account relationship and is often shared for root access. That individual uses remote access technology and protocols like SSH to perform a task. If administration for Unix and Linux is primarily performed remotely, it begs the question, where does

that remote administration originate? Is it on-premises and within a trusted network, or is the user operating remotely, such as from a home office or maybe their local coffee shop? And if you consider every remote access session is some form of privileged remote access, since access is actually granted from someone operating remotely, then the keyboard and mouse are nowhere near the physical computing device.

Today, unsecure networks of work-from-anywhere users are an extension of our IT perimeters. Consequently, we have introduced new attack vectors and potential regulatory compliance issues that need to be addressed. For Unix and Linux administration, this represents an unacceptable risk since a typical business's most sensitive data and applications tend to reside on these mission-critical platforms. All connectivity is dependent on secure credentials that follow the model of least privilege, just-in-time access, and single-use authentication.

A zero-trust solution for Unix and Linux enables organizations to granularly control privileged access, achieve compliance, and vastly dial down cyber risk. The implementation should apply parameters such as time, day, location, and application or asset vulnerability status to enable better privilege elevation decision-making. An enterprise solution provides capabilities far beyond Sudo, including centralized administration, session monitoring and management, file integrity monitoring, and more. A zero-trust implementation for Unix and Linux access that incorporates least privilege should consider the following traits:

- Removes admin rights and enforces least privilege across all Unix and Linux endpoints, regardless of the server class operating system or Linux desktop version.
- Enables users to securely run specific commands and sessions remotely, without logging in as an administrator or root via the command-line or graphical interface.
- Advances toward a zero standing privilege (ZSP) state by dynamically elevating privileges just-in-time for processes, applications, etc., but not for end users.
- Enforces separation of duties and privilege separation to limit the privileges associated with any account or process.

Consider Figure 19-10 for zero trust applied to Unix and Linux for least privilege.

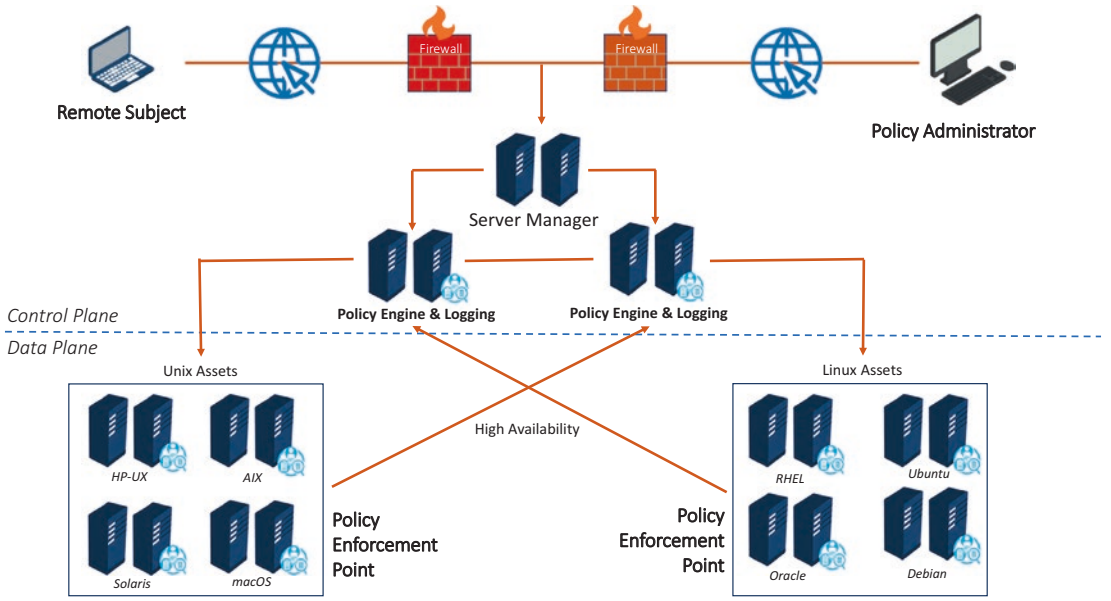


Figure 19-10. *Unix and Linux zero-trust least privilege implementation including remote access support*

This zero-trust architecture for Unix and Linux uses a model for user and application access, regardless of the network topology. And if access is secured using a dedicated zero-trust remote access technology, you can achieve these goals without the need for protocol tunneling or a VPN.

While any partial implementation of this architecture can be a substantial step toward more secure computing, it represents only a hybrid approach. An example of this would be to use VPN technology in lieu of a dedicated zero-trust remote access solution. In fairness, this scenario is much more secure than allowing a home computer with VPN access into your environment to perform administrative tasks on your sensitive Unix and Linux resources, especially as root.

Directory Bridging

Directory bridging technology is paramount in supporting and simplifying a zero-trust strategy for Unix/Linux machines while also helping organizations comply with other regulatory mandates. Directory bridging centralizes authentication for Unix and Linux environments by extending Microsoft AD's Kerberos authentication and Single Sign-On. Users leverage their AD credentials (or Entra ID) to access Unix and Linux systems for a seamless experience.

Organizations can attain consistent policies and controls across the enterprise by extending native group policy management tools to include settings for Unix and Linux, and to transition users from desktops to remote machines or between systems without requiring credential reentry. Leveraging Microsoft Group Policy across non-Windows platforms also enables centralized configuration management, reducing the risk and complexity of managing a heterogeneous environment. This concept is foundational for all the identity security discussions to date since it eliminates identity and account sprawl based on operating systems and helps keep a consistent approach for authentication by having one authority for directory services.

Security administrators have struggled to convert, apply, enforce, and audit policies across the enterprise due to dissimilar operating systems. Directory bridging makes it easy to map settings automatically and apply them to the machines, whether they are on-premises or in the cloud. Specific zero-trust security controls can be enforced by Active Directory GPOs (Group Policy Objects), so if there is an attempt to alter those settings, even by root, the settings will be immediately reapplied, and an alert will be generated indicating a policy violation that can be forwarded to your SIEM. Therefore, when considering a directory bridging technology as a part of your zero-trust initiatives and identity security model, look for these attributes:

- Removes admin rights and enforces least privilege across all Unix/Linux endpoints
- Securely allows authentication into non-Windows systems using directory services provided by another platform, like Microsoft Active Directory
- Allows for zero-trust policies to be applied via group policy options (or similar) to non-Windows assets for policy enforcement
- Supports a variety of zero-trust architectures from enclave to gateway models

Measuring Zero Trust

With all these zero-trust considerations in mind, how does an organization measure success? Luckily, the US Department of Defense has created a Zero-Trust Framework[5] with measurable controls to quantify the success of each of these implementations against your workflows.

Figure 19-11 is a simplified version of security controls created by the Department of Defense to help measure your own control frameworks against zero trust and their seven tenets at a granular level.

Execution Enablers	Zero Trust Maturity		
• Doctrine • Organization • Training • Material • Leadership • Education • Personal • Facilities • Policy	Zero Trust Category	Zero Trust Target Level (Number of Controls)	Advanced Zero Trust (Number of Controls)
	1. User	13	15
	2. Device	14	10
	3. Application and Workload	12	6
	4. Data	17	14
	5. Network and Environment	10	3
	6. Automation & Orchestration	13	7
	7. Visibility & Analytics	12	6
	Total	91	61
Appendix B enumerates all the controls		Total Activities	152

Figure 19-11. *Simplified US Department of Defense Zero Trust Framework measurement, based on seven tenets and security controls*

While these controls can map to other regulatory frameworks from NIST, like SP 800-53, the goal is to demonstrate compliance per workflow using identity and asset-security best practices. To simplify this approach for all organizations, this fan chart has been recreated into a table (available in Appendix B) to help your organization get started.

[5] https://dodcio.defense.gov/Portals/0/Documents/Library/DoD-ZTStrategy.pdf

Zero-Trust Design Considerations

Legacy applications, infrastructure, and operating systems are most certainly not zero-trust aware. They have no concept of least privilege, balk at application control, and lack privilege escalation models or authentication models that dynamically allow for modifications based on contextual usage. They have no concept of a remote worker. They rely on direct network connectivity to operate them. They lack session monitoring or lateral movement monitoring capabilities. In fact, their security is probably highly network dependent, and redeploying internal applications can be costly and potentially disruptive. Only a serious business need can justify undertaking these types of initiatives. Adding security controls to existing applications to make them zero-trust aware is not always feasible. It is likely that your existing applications have no facilities to accommodate the connection models in the specification and are not coded to operate in a resource enclave model as specified by NIST.

Depending on the architecture of your custom application, consider using:

- Zero trust with password and secrets management as the mechanism for any managed session when connectivity should be monitored and managed
- Zero trust with endpoint privilege management as the mechanism for remote worker privilege elevation, least privilege, and application control
- Zero trust with remote access management as the mechanism for remote worker authentication, least privilege, and session monitoring
- Zero trust with directory bridging for consolidated directory service authentication

This approach will allow you to implement zero-trust controls and enhance your security posture, but without having to reengineer established systems.

Any zero-trust implementation requires a layered or wrapper approach to enable legacy systems. However, a pure zero-trust approach entails enveloping all resources, regardless of their location, with these concepts. You can, however:

- Log remote session activity, record interactive screen sessions, and monitor events to look for potentially malicious behavior. This is a partial implementation of zero trust with remote access and may be sufficient for some environments to mitigate risks. This is an important consideration when a single remote session may interact with multiple systems behind the scenes that are not being managed, yet represent a high value to a threat actor.
- Log privileged activity, capture process launches, and monitor events to look for potentially malicious behavior. This is a partial implementation of zero trust with identity security solutions and may be sufficient for some environments to mitigate remote application access risks.
- Log screen activity, capture process launches, tally keystrokes, and monitor logs to look for potentially malicious behavior. This is a partial implementation of zero trust with identity security solutions. In conjunction with well-defined access control lists, the implementation is good enough to manage, monitor, and mitigate the risks of remote access to your Unix and Linux assets from virtually any source location.

If you think your organization does not use peer-to-peer (P2P) networking technology, you may be unaware of the default settings in Windows 10+. Starting in 2015, Windows 10 enabled peer-to-peer technology to share Windows updates among peer systems to save Internet bandwidth. While some organizations turn this feature off, others may not know it exists. In addition, tools like Tanium use agents that cross-communicate in a linear or mesh network architecture to perform queries at scale that do not allow zero-trust endpoint segmentation on shared networks.

This represents a risk of privileged lateral movement between systems. While no vulnerabilities and exploits have materialized for this type of peer-to-peer connectivity yet, it does present communications that violate the zero-trust model in much the same way as a mail server needs to communicate with every end user's mail client. There should be no unauthorized lateral movement—even within a specified micro-perimeter.

In addition, if remote workers have protocols like ZigBee or other mesh network technologies for IoT, you will find that they operate counter to zero trust. They require

peer-to-peer communications to operate, and the trust model is based strictly on keys or passwords, with no dynamic models for authentication modifications.

With all these in mind, consider the following:

1. It is impractical to build a resource enclave for the entire installation. Resource enclaves should be well-defined and as small as possible. Resource enclaves that require peer-to-peer communications to operate, and whose trust models are based strictly on keys or passwords, with no dynamic models for authentication to support access via a gateway should be minimized in any given implementation. These environments can be managed using a password and secrets manager, but are not representative of a zero-trust architecture. Therefore, if you decide to embrace zero trust and password management, consider hardening your resources to allow an enclave model and disallow any inappropriate network communications to anything within that defined enclave. While there will be exceptions for communications within the enclave itself, conceptually, the perimeter stops at the collection of resources. Password management using a gateway should be the vehicle for allowing any external access.

2. If you decide to embrace zero trust and identity security, consider hardening your endpoint security model to disallow any inappropriate network communications on the same subnet as the source or destination. While there may be exceptions for devices like local printers, conceptually, the perimeter stops at the device itself and is defined through applications and privileges, and remote access only controls the connection point to point.

3. Investigate whether your Unix and Linux environment is using similar technologies, has P2P, or has mesh network technologies enabled—even for policy or network management. These present a huge barrier to embracing any trusted remote access paradigm because lateral movement will always inherently be present in some form.

CHAPTER 20

Identity Obfuscation

In the age of identity attack vectors, protecting one's identity cannot be accomplished solely with a single strategy or product. Organizations need to step up and consider other mitigation strategies outside of identity security that can help, too. There are plenty of opportunities for a threat actor to steal your identity using information technology that we embrace every day. However, there are other mitigating approaches, called identity obfuscation, that can limit a threat actor's ability to create the critical linkage between account, identity, and data—especially when thinking about synthetic identities.

Identity obfuscation features are typically application settings, dedicated software, or even physical additions to devices that shield your data and protect your identity. They are required by law in some cases (GDPR, for example) to obfuscate a user's identity when collecting performance and analytics data. Their use can have far-reaching ramifications in the form of financial penalties if they breach regulation compliance requirements for data and identity collection.

Identity obfuscation features can shield your identity from many physical and electronic tactics a threat actor may apply to gain an advantage. Consider the following identity obfuscation technologies, both electronic and physical:

- *Incognito Browsing Mode (Private Browsing)*: The ability of a web-based browser to block cookies, browser version information, historic data, and other sensitive information that could be used to determine your persona, identity, or even to launch a targeted attack based on runtime data submitted by your computer during a browsing session.
- *Performance Obfuscation*: The ability of software to collect performance, analytics, event and support information, and automatically scrub it for personally identifiable information about the user, applications, or even environment before submitting to a

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_20

vendor or installed solution. This type of technology is typically used to protect data from being sent over a regional or country boundary when data privacy laws prohibit a company or organization from storing or sending it with granular identity information.

- *Screen Privacy Filters*: These are physical, polarized filters added to computer or mobile device screens that prevent threat actors from viewing a screen from obtuse angles. A user can clearly see the screen when operating directly in front of it or from acute angles off a perpendicular axis. This is designed to stop shoulder surfing and the inadvertent linkage of information that could be present on a user's machine and visible to inappropriate users.
- *Guest Shopping Carts*: While not directly considered a privacy filter, an online retailer that allows the purchasing of items anonymously (or using a guest account) is a form of privacy filter. The user's identity is restricted to the transaction and an account, with detailed identity information not stored for future use. This reduces risk by not having an account created in a potentially untrustworthy retailer. Using a guest account for shopping is highly recommended for merchants that you infrequently (or one time) visit.
- *IP Address Relays and Virtual Private Networks*: The ability to shield your assets' IP address from the destination in order to obfuscate the geolocation or source of request across the Internet. Most modern operating systems have this capability; however, certain applications will fail to operate correctly when geolocation is legally required to perform the transaction. Using this feature with a browser in incognito mode will shield your identity from most threats on the Internet.

Depending on your organizational requirements, your businesses may consider implementing identity obfuscation settings to minimize risk and meet regulatory compliance requirements. For example, information technology owners may install computer-based privacy filters on all laptops to prevent data leakage as employees travel, or on desktops in a financial institution, to limit visibility of privileged information. Security teams may request data obfuscation in the Windows Event Logs for

a solution to prevent identity information from showing up in a SIEM. And businesses may use guest accounts on operating systems to allow many-to-one usage of computing devices (vs. creating accounts for every identity) in order to save costs. While this last use case may sound extreme, many applications are licensed by user, and if the data can be represented in a generic, nonsensitive fashion, then protecting someone's identity (by not making an account) can have profound effects on lowering cost, too.

CHAPTER 21

Regulatory Compliance

Overview

Regulatory compliance is an organization's adherence to laws, regulations, guidelines, frameworks, and specifications relevant to its business processes that are mandated by law or required for a business to operate in a given jurisdiction. Violations of regulatory compliance often result in legal punishment, including government enforced fines, and in extreme cases, the dissolution of the business.

Organizations must approach regulatory compliance requirements with sustainability in mind if they are to effectively manage their identity risk. This is a security-driven compliance approach, and if we sustain and maintain compliance, we are secure. If you do nothing more than what is necessary to pass a SOX or HIPAA audit, you are not likely to address your logical access risks or security requirements. Effectively managing identity risk requires meaningful diligence above and beyond "checking a box" for compliance. Achieving a sustainable level of transparency and risk management to protect against the palpable security threats that exist inside the organization should be the target goal. While the goal post is continually moving, your organization should still try to accomplish getting there and, depending on the region or local government, may have severe penalties even if the basics are in noncompliance.

Compliance Example

For example, consider "The New York State Department of Financial Services (DFS)."[1] In 2017, DFS created state rules for financial organizations regarding data and identity privacy, public disclosure, oversight, and cybersecurity best practices. While the

[1] www.wsj.com/articles/new-york-adds-stiffer-requirements-to-cybersecurity-rules-68d49fd1?mod=djemCybersecruityPro&tpl=cy

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_21

initial policy mimics many existing initiatives, recent changes in 2023, "23 CRR-NY 500 NY-CRR,"[2] have new provisions that explicitly focus on identity security and privileged access management. Consider this section directly from the updated law:

> *500.7 Access privileges and management.*
>
> *(a) As part of its cybersecurity program, based on the covered entity's risk assessment each covered entity shall:*
>
> *(1) limit user access privileges to information systems that provide access to nonpublic information to only those necessary to perform the user's job;*
>
> *(2) limit the number of privileged accounts and limit the access functions of privileged accounts to only those necessary to perform the user's job;*
>
> *(3) limit the use of privileged accounts to only when performing functions requiring the use of such access;*
>
> *(4) periodically, but at a minimum annually, review all user access privileges and remove or disable accounts and access that are no longer necessary;*
>
> *(5) disable or securely configure all protocols that permit remote control of devices; and*
>
> *(6) promptly terminate access following departures.*
>
> *(b) To the extent passwords are employed as a method of authentication, the covered entity shall implement a written password policy that meets industry standards*
>
> *(c) Each class A company shall monitor privileged access activity and shall implement:*
>
> *(1) a privileged access management solution; and*
>
> *(2) an automated method of blocking commonly used passwords for all accounts on information systems owned or controlled by the class A company and wherever feasible for all other accounts. To the extent the class A company determines that blocking commonly used passwords is infeasible, the covered entity's CISO may instead approve in writing at least annually the infeasibility and the use of reasonably equivalent or more secure compensating controls.*

[2] www.dfs.ny.gov/system/files/documents/2023/11/rf_fs_part500_amend2_20231101_alt.pdf

The State of New York has explicitly stated the importance of identity security and has even gone as far as mandating PAM and secure remote access solutions for all financial organizations. The state recognizes the importance of this technology in mitigating the risks of identity attack vectors, and it is expected that other states and the federal government will soon follow these recommendations. If you read this far in the book and are still not convinced that identity security requires special attention, hopefully seeing state governments embracing the approach in this book will convince you of the importance and guidance for success.

US Regulatory Compliance

With this in mind, consider Table 21-1. The table enumerates common compliance requirements for organizations based in the United States that provide high-level identity security mandates per vertical or financial model.

Table 21-1. *Common compliance regulations applied to identity security*

Regulation	Vertical Industry	Focus	Information Security Requirements
Sarbanes-Oxley Act (SOX)	All public companies traded on US stock exchanges, including entities owned by foreign entities, but traded within the United States	Information integrity	Ensure the accuracy of financial information and the reliability of systems that generated associated content. Section 404 requires management to assess internal controls and obtain attestation from external auditors annually
Security Management Act (FISMA)	Federal agencies and associated affiliates	Information integrity	Develop, document, and implement programs to secure data and information systems supporting agency operations and assets

(*continued*)

Table 21-1. (*continued*)

Regulation	Vertical Industry	Focus	Information Security Requirements
General Data Protection Regulation (GDPR)	All organizations, regardless of ownership, that conduct business in the European Union (EU)	Privacy	Protect consumer data from theft and fraud. Notify all involved parties when a breach occurs within 72 hours (about 3 days), and "remove/forget" consumer data when requested
Payment Card Industry (PCI) Data Security Standard (DSS)	All members, service providers, and merchants that store, process, or transmit cardholder (credit card) data	Fraud prevention and privacy	Meet 14 information security requirements in areas such as data protection, access control, encryption, and monitoring and intrusion prevention.
Health Insurance Portability and Accountability Act (HIPAA)	US healthcare providers, payers, clearing houses, and their business associates	Privacy	Protect the security and privacy of personally identifiable health information from unauthorized access, alteration, deletion, or inappropriate transmission
Gramm-Leach-Bliley Act (GLBA)	US-based financial organizations	Privacy	Establish administrative, physical, and technical safeguards to protect the security, confidentiality, and integrity of consumer financial information
North American Electric Reliability Council (NERC)	All entities responsible for the planning, operation, and use of the bulk electric system in North America	Critical infrastructure protection	Protect information technology assets essential to the reliability of the bulk electric system, including monitoring, access control, and change/configuration management
Children's Online Privacy Protection Act (COPPA)	Any organization producing products or services that target minors	Protection of minors	Data protection and marketing restrictions, when engaging minors

Global Regulatory Compliance

In addition, organizations with a global presence should familiarize themselves with data privacy and identity laws around the world. Table 21-2 highlights some of the most prominent laws by country or region that will have similar requirements.

Table 21-2. *Sample of data privacy and identity protection regulations around the world*

Country/Region	Legislation
Canada	Personal Information Protection and Electronic Documents Act (PIPEDA)
	Personal Information Protection Act (PIPA Alberta and PIPA BC)
	The Act Respecting the Protection of Personal Information in the Private Sector (Quebec Privacy Act)
Mexico	Federal Law on the Protection of Personal Data Held by Private Parties
Argentina	Personal Data Protection Act 25.326
Brazil	General Data Protection Law 13.709/2018 (LGPD)
Columbia	Law 1266/80, Law 1581/2012, Decree 1377/2013
Costa Rica	Undisclosed Information, Law 7975
	Law on the Protection of Persons Regarding the Processing of their Personal Data, Law 8968
Peru	Protection of Personal Data 29.733
Uruguay	Data Protection Act 18.331
United Kingdom	General Data Protection Regulation (GDPR)
	Privacy and Electronic Communications Regulations 2003 (PECR)
	Financial Conduct Authority's Principles for Business (PRIN 2.1)
European Union	General Data Protection Regulation (GDPR)
Israel	Protection of Privacy Law 5741-1981 (PPL)
Turkey	Law on the Protection of Personal Data (LPPD)

(*continued*)

Table 21-2. (*continued*)

Country/Region	Legislation
United Arab Emirates	Digital Payment Regulation
China	PRC Cybersecurity Law
	The Decision on Strengthening Online Information Protection
	National Standard of Information Security Technology
South Korea	Personal Information Protection Act (PIPA)
	Act on Promotion of Information and Communication Network Utilization and Information Protection (IT Network Act)
	Use and Protection of Credit Information Act (UPCIA)
	Act on Real Name Financial Transactions and Guarantee of Secrecy
Japan	Act on the Protection of Personal Information (APPI)
Australia	Privacy Act 1988
	Australian Privacy Principles

Regulatory Compliance Best Practices

Taking the correct approach to compliance can enable an organization to manage user access as a sustainable ongoing process, rather than a one-time audit that does very little to support a sustainable, secure computing environment.

To proactively address compliance requirements, many organizations look to identity security to define and manage the overall process. Identity security is a cross-organizational enterprise discipline that provides the intelligence and business insights needed to strengthen controls and protect information assets. With identity security, organizations gain a 360-degree control plane that answers the question: "Who has access to what?" This control plane provides the process and tracking transparency needed to reduce potential security and compliance exposures.

Identity security also helps organizations improve efficiency by replacing paper-based, manual processes with automated tools. Not only can an organization significantly reduce the cost of compliance, but it also establishes a repeatable process that is more consistent, auditable, and reliable over time. Taking an automated approach

helps to build predictability, repeatability, and sustainability into the compliance workflows while improving the end-user experience and overall satisfaction.

The steps in Table 21-3 illustrate a base methodology and timeline for implementing identity security. The key to success is defining measurable steps (timeline based) to build that repeatable and sustainable regulatory compliance process across all identity tasks and activities.

Table 21-3. *Best-practice steps toward sustainable compliance and identity security*

Step 1	Step 2	Step 3	Step 4	Step 5
Access your current state	Build identity security model	Automate monitoring controls	Automate preventive and response controls	Complete audit of process, end to end
Aggregate and correlate identity data	Define policy model	Access identity certifications	Access deviations	Audit data
Conduct baseline assessment	Define role models	Policy detection	Access identity security risk	Identity exceptions and service-level violations
	Define risk models	Manual remediation workflows	Implement automation routines	Provide attestation for compliance
				Identify exceptions
				Identify corrective actions

The Future of Identity-Based Compliance

In the late 1990s, exploitable vulnerabilities began to traverse the young Internet and prove that poorly coded and tested software could be exploited for fun, financial gain, and exfiltrate sensitive information. In order to detect these software flaws, vendors like Eye, ISS, and Nessus began creating vulnerability scanners to identify operating systems and software with these weaknesses. As you could imagine, the detection methods, risk

scoring, descriptions, and the accuracy of each vulnerability assessment scanner led to a myriad of false positives and false negatives between the tools. While one vendor may identify a vulnerability as a high (there were no critical ratings back then), another vendor may score the risk as a medium or even low depending on their research and personal opinions.

In order to solve this discrepancy, standards organizations like MITRE and NIST began building compliance standards, like CVE, CVSS, and even CWE, so everyone could talk the same language and assess vulnerabilities the same way. This brief history lesson has taught us one important thing: without standardization and a common language to describe threats, then organizations, and every vendor, could represent the same problem in a different way, ignoring the knowledge and risk obtained by other perspectives. This includes governments that learned very early on that vulnerabilities and exploits could be used for cyber warfare against military and commercial installations.

Today, if we consider identities as the new perimeter, and we know the flaws from SIM (Subscriber Identification Module) jacking to MFA fatigue, the world is in need of a new compliance standard to identify, classify, and score identity-based security risks, just like we did for vulnerabilities. The difference in our history lesson from vulnerability assessments to identity security lies in the technology improvements from network based vulnerabilitiy scanners to APIs and agents. Just like early vulnerability scoring systems, the cybersecurity industry needs to develop a methodology for detection, recommendation, and remediation that will ultimately become ubiquitous for identity based threats. This will form a foundation for a common identity-based risk language and, regardless of vendor, level the playing field for organizations trying to mitigate this threat. In simple terms, the cybersecurity industry needs a new regulatory standard called something like Common Identity Exposures (CIE) to bring worldwide discussions on identity risks under one umbrella, much like CVEs did in the early 2000s. The current plethora of regulatory compliance standards and global and regional laws, as we have just discussed, helps prove this point.

This new standard needs to be developed from the ground up since scoring like CVSS is not applicable to identity-based risks and the current terminology is nonstandard across vendors. Regardless, here are a few attributes that should be considered as we explore the *possibility* of developing a new compliance standard:

- *Name*: A simple title for the detection of an identity security–based risk
- *Description*: A verbose representation of the identity security–based risk, including the possibility of compromise and identity attack vectors that could lead to exploitation

- *Occurrence*: The number of times the event has occurred with appropriate time and date references
- *Risk*: The risk of the identity-based threat based on a simplified scoring mechanism, like critical, high, medium, low, and informational used in similar standards.
- *Identity*: The associated identity for the detection, including username, email address, and other relevant designators
- *Identity Provider*: The identity provider and directory store affected by the detection and the owner of the identity
- *Detection*: The actual API, log, event, or detection method including XML or JSON that triggered the finding
- *Remediation*: The manual or automated method to resolve the identity-based risk and what potential negative impact it could have on the environment

With these attributes in mind, all vendors assessing identity security–based risks can talk a common language regardless of their technological implementation. As many of my peers would state, you cannot fix something if you do not know how broken it is, and you cannot prioritize what to fix first if you do not know the severity of all the measured risks. A new compliance standard can help ensure that everyone works on the most critical problems first and communicates the problems in the same way. Then organizations can truly understand the risks of poor identity hygiene within their environment, and we can ultimately solve the problems associated with identity being the new perimeter.

CHAPTER 22

Key Takeaways

For the conclusion of this book, I would like to offer the following five critical tenets of identity attack vectors and how identity security should be embraced to make your implementation successful and resilient:

1. **Think Identity, Not Account**

   Even before the advent of cloud computing, we learned that, most often, an end user has multiple accounts and multiple entitlements per account across the organization's infrastructure. If an enterprise only focuses its IAM program on managing at the account level, it will never get the full visibility needed to properly know: "Who has access to what?" Understanding the **overlapping** relationship between (a) an identity and its accounts, (b) the account and assets, and (c) the entitlement and the data/information that it protects, is crucial. By centralizing data around an identity, enterprises have a single place to model roles, policies, privileges, and risk; a single place to build out compliance and audit policies; and a unified approach to provisioning access management across the organization. This is the foundation of identity security.

2. **Visibility and Transparency**

   With technologies like cloud, mobile, DevOps, and IoT being mixed together with established enterprise mainstays like SAP, Oracle, and RACF (yes, RACF is still out there, folks), everyone needs a central point of visibility. All enterprise applications that contain valuable or sensitive data, or any process that performs mission-critical IT operations, must be managed with a focus on identity security, compliance, and automation. A single point of

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_22

visibility allows the organization to leverage common detective and preventative controls and ensures an enterprise-wide view of its identities and their associated accounts. This enables the business and IT to effectively analyze risk, make informed decisions, and implement appropriate controls in an automated and sustainable fashion to ensure the entire IT estate is protected. It should be a goal of every organization to strive for centralized visibility for all identity, log, and security data.

3. **Identity Governance**

   It's critical to always manage the lifecycle of an identity and its access by tying it to the business policies and owners responsible for it. We must allow best-practice identity governance policies to span the entire lifecycle of an identity—from joiners, movers, and leavers through to the processes of access request, provisioning, and access review. By embedding policies and controls throughout the full identity lifecycle process, organizations can achieve higher levels of automation, sustainable compliance, and reduced security risk. It's more important than ever to apply centralized policies and automated controls to all key identity business processes. Adding consistency and repeatability will allow organizations to strengthen controls, work more efficiently, and promote good governance policies over the long term. Most importantly, regardless of whether a resource is on-premises or in the cloud, identity governance should always be handled using the same control and governance policies.

4. **Entitlements**

   As we have covered in detail, entitlements are the privileges, rights, and permissions associated with an account, role, or group. Right-sizing entitlements across the entire organization, and honoring concepts like least privilege and privileged access management, is key to minimizing your organization's risk surface. The aforementioned is only possible with comprehensive inventory and discovery technology that can ensure shadow

IT and rogue identities, accounts, and assets do not become a liability. Measuring entitlements and limiting privileged use of them are foundational for identity security.

5. **User Experience**

   Identity security must help provide a better overall user experience. Having efficient user experiences for these security processes is a critical part of ongoing participation from all users—both inside and outside the organization—throughout the identity and access lifecycle. The experience must feel like part of a natural business flow and must encompass everything from daily authentication to the privileges we expect to utilize in our everyday job functions. With this in mind, if any solution is implemented without a good end-user experience, workers will find ways to circumvent the technology and stunt implementations, making any risk mitigations potentially a moot point. Users must be champions for identity solutions—ensure everyone is involved in its implementation and usage.

CHAPTER 23

A Final Thought on Vendors

As the budget holder for a leading cybersecurity company, my team and I are frequently asked by outside vendors what it will take to replace an incumbent solution. Keep in mind, my peers and I typically field at least five cold calls a day and dozens of solicitation emails, as well as social media messages to review a new technology, consider another product as a replacement, or please watch our demo and we will send you a gift. The thought that this level of cold solicitation actually works is mind-boggling. It's rare for vendors to get a C-level executive to respond to a cold call or spam message.

So, with all that aside, what will it take to replace an existing solution with a competitor? The honest answer from senior leaders in the security space (regardless of vertical) is that a replacement of a cybersecurity solution (including identity security) is based solely on technical or business reasons that are intolerable to the organization. The impetus to make such a change and embark on such a journey is rarely, if ever, emotionally driven. That should not be a surprise, but rather something that young sales executives need to learn.

Therefore, why *would* an executive or budget owner even consider replacing an existing solution? Here are key reasons:

- *Security Breach*: When the incumbent vendor incurs a cybersecurity breach that affects your organization, the immediate response is, "Should we continue to license their solution?" This is an obvious knee-jerk reaction, but, often, the gut check by the security team is correct. The incumbent solution and vendor should be considered at a high risk of replacement, even in the middle of an existing contract, in order to mitigate future risks. Depending on the severity of the breach, the replacement may be required to maintain business continuity or to satisfy cyber insurance coverage.

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_23

- *Excessive Cost*: As the world has shifted from perpetual licensing to subscription and term structures, annual costs can easily increase in whole number multiples, if your contract does not have any restrictions. Vendors believe that their solutions are so "sticky" that a 2x or 3x cost structure will be absorbed, since the cost to replace their solution could be even higher. That is an incorrect assumption. Burning your clients by excessively increasing annual costs will never end well, and expect that, at some point in the near future, they will replace you. You may be able to get away with a cost increase like this once, but odds are, if the stunt is pulled again, you will lose that client.
- *Strategic Direction*: Some cybersecurity vendors will be very open about their strategic direction and road maps. Their strategic direction may be open to feedback and minor adjustments, but major changes in the cloud, or support of on-premises technology, may be incompatible with your own business model. If the deviations are drastic enough, your organization may be forced to look for new technology to stay relevant. While this consideration is rare, when it does happen, it can be a significant event that uproots foundation technologies used to support the business.
- *Incompatibilities*: The lifecycle of any cybersecurity solution includes operating support, data retention, integrations, etc., that are compatible with the ecosystem for your organization. In addition, the end-of-life of some technology may be negatively impacted when a vendor drops support for a legacy technology stack that you are dependent upon. This could include older desktop operating systems and legacy server platforms that function perfectly and have a longer lifecycle than originally planned. This is especially true if the implementation is based on OEM[1] (Original Equipment Management) solutions. When these changes occur, cybersecurity coverage for a technology could fail and require a change to a vendor that still provides support. And remember, for the right amount of money, any vendor will potentially still provide support.

[1] www.investopedia.com/terms/o/oem.asp

- *Technical Support*: Every cybersecurity solution needs technical support. After all, it is all based on software, and humans write software, and humans make mistakes. If the technical support provided by any vendor fails to correct a problem in a timely manner, or if a solution requires excessive technical support just to operate, it is ripe for replacement. Make no mistake about it; high maintenance solutions with lots of external support requirements tend to fail over time and become strong candidates for replacement or upgrades.
- *Vendor Consolidation*: Saving money, minimizing the number of negotiated contracts, and having a single "neck to choke" are a common theme for cybersecurity solutions, especially identity security solutions. While having best-of-breed technology is always desirable for the best protection, many technology stacks can be "good enough," just to meet vendor consolidation initiatives. Therefore, if you are a vendor with a point solution to solve a single use case, you are ripe for vendor consolidation when someone else offers similar features and reduces costs.

While there are obviously other business reasons to replace a solution, the preceding reasons are the most common. Random inbound correspondence that "sticks" is opportunistic and strictly based on luck and timing. Vendors that have an inside scoop or target an event that has been published, tend to have a better chance of succeeding, as long as they are not perceived as "ambulance chasers." So, for all my peers marketing cybersecurity and identity security solutions, remember, change typically occurs when change is needed, and positive brand awareness is key. Flooding a prospect's inbox with solicitations is not brand awareness, but rather an annoyance, especially at an executive level. Focusing on who you are vs. what you're selling is always a better approach to help an executive decide when they should replace a solution and who they should consider.

CHAPTER 24

Conclusion

Have you heard of the *Darwin Awards*?[1] They are satirical awards that recognize the very poor choices of the "award recipient" that led to their own death or sterility. Darwin Award recipients essentially self-select themselves out of the human gene pool by embarking on stupid activities.

While the Darwin Awards are a morbid attempt at humor—and some may wonder why the reference is part of the conclusion for an identity security book—the concept has relevance for cybersecurity because many people do stupid things that could result in a "game over" security event for an organization. This simple conclusion underscores personal lessons learned, anecdotes overheard, and activities observed that I personally would like to nominate for the Darwin Cybersecurity Awards:

- "The server crashed. Do you have a backup?" … "Yes, but it was on the server, too."
- "Please turn off and remove any antivirus software in order for our solution to run correctly."
- "We have purchased a technology that is future-proof."
- "Just email the human resources department; they can change your bank account information for electronic payroll deposits."
- "We do not care about government regulations. We can demonstrate we're doing the best we can, and that should be good enough for anyone."
- "The same 'secret' needs to be installed on every system for the solution to communicate."

[1] https://darwinawards.com/

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1_24

- "We do not change the default passwords, so someone always knows how to access the 'asset' in case of an emergency."
- "We are not a target for a cybersecurity breach; we only make edible products."
- "I trust my employees will not click on a phishing email."
- "My antivirus solution will stop ransomware."
- "That is my vendor's security problem, not mine."
- "We do not need to remove administrative rights from our end users. If there is an issue, we can reimage their workstation within a few hours or overnight."

While this book was written with some mild science fiction humor in mind, and only one political commentary, the lessons learned are real. A cyberattack can affect any organization or any individual. No one is immune.

Humbly understanding and striving to learn identity security best practices is your best bet for avoiding a cybersecurity incident. The simple Darwin Cybersecurity Award Nominees list earlier is a reminder to favor positive intent and avoid philosophies that deny history and common attack vectors. And finally, only a human can admit failure or lie with intent. We all should learn from our mistakes and learn to succeed and fail as a team. No identity should ever be left to fail alone.

APPENDIX A

Identity Security Sample RFP Questions

Business Justification

- What metrics are we trying to improve/change with this project quantify success?
- Why are we pursuing this outcome now and not before?
- What broader business strategy is this initiative tied to (security, compliance, cyber insurance, zero trust, operational excellence, etc.)?
- What management KPI (Key Performance Indicators) does this project support?
- Which business unit is driving this program/project?
- How is this security risk being graded in terms of level of operational risk (negligible, low, medium, high, severe, very severe)?
- How is this security risk being graded in terms of inherent probability (very unlikely, unlikely, likely, very likely, almost certain, certain)?
- What is the specific/measurable business result expected by making the change/selection?

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1

- Cost of inaction: What are you losing by not acting on this problem (measuring risk and impact)?
- How is this initiative being funded? Has the budget been approved?
- What solutions are being considered?
- What decision-making governance process will this project follow?
- Description of "compelling pressures"/timeline of action?
- What are the risks for this project (internal and external)?

Universal Identity Security Capabilities

- Provides a centralized, holistic lens of identities and access across your multicloud and on-premises estate.
- Provides a clear, easy-to-understand picture of the accounts, privileges, and access associated with each identity.
- Identifies problematic privileges and multicloud entitlements and helps you right-size them.
- Identifies potential security misconfigurations and assists in proactively mitigating them.
- Identifies overprivileged and high-risk privileged accounts, inactive and orphaned accounts, partially revoked identities, and other security issues.
- Detects and alerts on suspicious activities, including events involving multiple identities and accounts.
- Documents identity-based risk—including the potential blast radius of each account and identity—enabling you to take decisive action.
- Correlates low-level data from a variety of third-party solutions to pinpoint high-risk users and assets and identifies critical threats.

- Helps unlock ITDR capabilities, enabling a rapid orchestration of security response to stop or mitigate threats.
- Reports on compliance, benchmarks, threat analytics, and other what-if scenarios.
- Supports phishing-resistant multifactor authentication technology like FIDO2.
- Integrates with Privileged Access Management solutions.
- Integrates with Identity Governance and Administration solutions.
- Can identify identity-based attack vectors:
    - Partially revoked accounts
    - Incomplete disabled accounts, depending on the asset or application
    - Dormant account activity
    - Lack of multifactor authentication
    - Privilege activity outside of PAM or SSO
    - Unmanaged privileged accounts
    - Privileged activity from to end-of-life operating systems or applications
    - Shadow identity detection
    - Bring your own cloud detection
    - Unmanaged accounts in cloud or production
    - Bulk forwarding of email
    - MFA has multiple associated phone numbers for access
    - Nondomain email addresses registered to accounts or MFA

Privileged Access Management Capabilities

- Performs full network and cloud discovery and profiling with auto-onboarding of privileged identities and accounts of all types—including shared admin, user, application, and service accounts; SSH keys, database accounts; cloud identities and accounts (Azure AD, etc.); social media accounts; machine accounts; DevOps secrets; API keys; and robotic process automation (RPA) credentials. This also includes vendor identities and accounts.
- Identifies where and how privileged passwords are being used, revealing security blind spots and malpractice—including default, shared and/or embedded passwords, use of the same admin account across multiple service accounts, reuse of SSH keys across multiple servers, etc.).
- Provides one unified, comprehensive solution to manage human (privileged users) and nonhuman (application, machine, service account, etc.) identities and includes session monitoring/management.
- Manages credentials across every platform (Windows, Unix, Linux, cloud, on-premises, etc.), directory, hardware device, application, service/daemon, firewall, router, and other organizational assets.
- Centralizes, secures, and encrypts all privileged credentials in a tamper-proof safe or vault. Ideally, the solution supports industry-standard encryption algorithms, such as AES 256.
- Builds permission sets dynamically according to data retrieved from scans.
- Implements secure API calls to eliminate embedded or hardcoded credentials in files, applications, scripts, and other code.
- Automates the rotation of passwords, SSH keys, and other secrets according to a defined schedule, including after each use for the most sensitive accounts or for accounts facing heightened security risk or compromise.

- Enforces your privileged password management policy—including password complexity, uniqueness (different passwords per asset, account, etc.), expiration, rotation, check-in and checkout, elimination of default passwords, and other rules.
- Automates workflows across the entire password management lifecycle.
- Provides granular access control for all end users, assets, and system integrations.
- Enables better security for SSO and never reveals the password to the end user by using modern passwordless technology and password injection techniques.
- Performs rigorous session monitoring and management to ensure a clean audit of all privileged activity and to immediately pause or stop suspicious sessions until a determination can be made regarding legitimacy.
- Requires no additional third-party tools or Java for session management—utilizes native tools (MSTSC, PuTTY) for a better end-user experience.
- Enables true least privilege by enabling a security model of just-enough access and just-in-time (JIT) access.
- Leverages industry standards, like SAML and RADIUS, to integrate with any MFA solution.
- Provides break-glass options for password checkout in the event of an emergency.
- Leverages an integrated data warehouse and threat analytics across the privilege landscape.
- Enables privileged task automation to reduce risk by automating multistep, repetitive tasks.

- Provides comprehensive reporting and analytics for SOC teams and executive visibility into the management of privileged credentials.
- Delivers enterprise-grade audit and compliance support by providing clear and distinct audit trails for all activities involving credentials under management.

Zero-Trust Remote Access Technology

- Enforces least privilege by giving authorized users just-enough access to complete activities just-in-time for remote sessions.
- Controls and monitors sessions using standard protocols for RDP, VNC, HTTP/S, and SSH connections.
- Enables granular access to specific systems, improving security and eliminating "all-or-nothing" access.
- Enables the user to inject credentials directly into the access session; the user never needs to know or see the credential.
- Creates an audit trail to provide visibility into vendor activity on your network and meet compliance mandates by controlling the access pathways into IT networks used by vendors.
- Manages privileged access to infrastructure and business assets that leverage web-based management consoles, including IaaS servers, hypervisor environments, and web-based configuration interfaces for core network infrastructure.
- Provides seamless out-of-the-box integrations with ITSM, SIEM, SCIM, and password management, as well as other common business software solutions.
- Enables MFA and alternative authentication methods, such as TouchID, FaceID, and Hello.
- Leverages industry standards, like SAML and RADIUS, to integrate with any MFA tool for privileged escalation, based on policy.

Endpoint Least Privilege

- Implements true least privilege by removing standing local administrative rights across desktop and server (Windows, macOS, Unix, and Linux) users while enabling dynamic, just-in-time elevation of privileges for specific applications and tasks.
- Provides smart application control, enabling management of which applications users can install or run, with the flexibility to set both broad and granular rules through a non resource-intensive operational process.
- Enforces policy-based restrictions on software installation, usage, and OS configuration changes.
- Sets policies via Active Directory Group Policy and Web Services with support for air-gapped systems and nondomain assets.
- Eliminates unauthorized software installations, workarounds, or gaps that could lead to privileged exploitation.
- Reports on privileged user behavior, including applications installed or run, system and configuration changes, as well as changes to critical policy or data files.
- Provides a single, unimpeachable audit trail of all user activity that simplifies compliance and streamlines forensic investigations.
- Simplifies operations by eliminating the need for end users to require two accounts.
- Allows for granular control over the access and permissions of APIs, extending the principle of least privilege to API accounts.
- Provides a technique for using real domain or local privileges when required.
- Centralizes management, policy, reporting, and analytics into one streamlined solution.

- Integrates with identity security, ITSM, SIEM, and other privilege management tools to enhance and embed into the existing security tech stack, improving workflows and allowing for a more comprehensive understanding of risk.
- Exercises granular control and audit over applications, commands, files, and scripts—protects against malicious threats as well as innocent errors.
- Records and indexes all remote privileged sessions for quick discovery during audits.
- Adaptively enforces full keystroke logging for the most sensitive sessions.
- Provides a clear view and clean audit trail into who is doing what and where.
- Consolidates audit logs and centralizes reporting across all server domains.
- Provisions and deprovisions privileges transparently, ensuring compliance satisfaction.
- Offers REST API for easier integration with third-party products.

Identity and Directory Services Bridging

- Features a Single Sign-On for any enterprise application that supports Kerberos or LDAP.
- Provides a single familiar toolset to manage both Windows and Unix/Linux systems (e.g., Active Directory users and computers, ADUC).
- Allows users to use their Active Directory or Entra ID credentials to gain access to Unix and Linux, consolidating various password files, NIS, and LDAP repositories into one identity directory service.
- Provides integration with Linux and Unix services via PAM and Kerberos (Samba, NFS, Apache, etc.).

- Adds Linux or Unix systems to the network without requiring Active Directory or Entra ID schema modifications.
- Provides a pluggable framework with an interface similar to Microsoft's Management Console on Linux.
- Supports a wide range of Unix and Linux platforms, including CentOS, Debian, Fedora, FreeBSD, HP-UX, IBM AIX, Oracle Enterprise Linux, SUSE, Red Hat, Solaris, Ubuntu, and architectures such as x86_64, SPARC, PPC, PPCLE, and s390.
- Supports compliance with SOX, PCI, HIPAA, and other regulations.
- Leverages industry standards, like SAML and RADIUS, to integrate with any MFA tool for privileged or step-up authentication.

APPENDIX B

Department of Defense (DoD) Zero-Trust Framework

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1

ID #		Capability/Activity	Pillar	Level	Security Controls	Compliant (Y/N)
1.1		**User Inventory**				
	1.1.1	User Inventory	1 - User	Zero Trust Target		
1.2		**Conditional User Access**				
	1.2.1	App-Based Permission	1 - User	Zero Trust Target		
	1.2.2	Rule-Based Dynamic Access Pt. 1	1 - User	Zero Trust Target		
	1.2.3	Rule-Based Dynamic Access Pt. 2	1 - User	Advanced Zero Trust		
	1.2.4	Enterprise Gov't roles and Permissions Pt. 1	1 - User	Advanced Zero Trust		
	1.2.5	Enterprise Gov't roles and Permissions Pt. 2	1 - User	Advanced Zero Trust		
1.3		**Multi-Factor Authentication (MFA)**				
	1.3.1	Organizational MFA/IDP	1 - User	Zero Trust Target		
	1.3.2	Alternative Flexible MFA Pt1	1 - User	Advanced Zero Trust		
	1.3.3	Alternative Flexible MFA Pt2	1 - User	Advanced Zero Trust		
1.4		**Privileged Access Management (PAM)**				
	1.4.1	Implement System and Migrate Privileged Users Pt. 1	1 - User	Zero Trust Target		
	1.4.2	Implement System and Migrate Privileged Users Pt. 2	1 - User	Zero Trust Target		
	1.4.3	Real-time Approvals & JIT/JEA[1] Analytics Pt. 1	1 - User	Advanced Zero Trust		
	1.4.4	Real-time Approvals & JIT/JEA Analytics Pt. 2	1 - User	Advanced Zero Trust		

1.5		**Identity Federation & User Credentialing**		
	1.5.1	Organization Identity Lifecycle Management	1 - User	Zero Trust Target
	1.5.2	Enterprise Identity Life-Cycle Management Pt. 1	1 - User	Zero Trust Target
	1.5.3	Enterprise Identity Life-Cycle Management Pt. 2	1 - User	Advanced Zero Trust
	1.5.4	Enterprise Identity Life-Cycle Management Pt. 3	1 - User	Advanced Zero Trust
1.6		**Behavioral, Contextual ID, and Biometrics**		
	1.6.1	Implement User & Entity Behavior Activity (UEBA) and User Activity Monitoring (UAM) Tooling	1 - User	Zero Trust Target
	1.6.2	User Activity Monitoring Pt. 1	1 - User	Advanced Zero Trust
	1.6.3	User Activity Monitoring Pt. 2	1 - User	Advanced Zero Trust
1.7		**Least Privileged Access**		
	1.7.1	Deny User by Default Policy	1 - User	Zero Trust Target
1.8		**Continuous Authentication**		
	1.8.1	Single Authentication	1 - User	Zero Trust Target
	1.8.2	Periodic Authentication	1 - User	Zero Trust Target
	1.8.3	Continuous Authentication Pt. 1	1 - User	Advanced Zero Trust
	1.8.4	Continuous Authentication Pt. 2	1 - User	Advanced Zero Trust

(*continued*)

[1] https://learn.microsoft.com/en-us/powershell/scripting/learn/remoting/jea/overview?view=powershell-7.3

ID #		Capability/Activity	Pillar	Level	Security Controls	Compliant (Y/N)
1.9		**Integrated ICAM (Identity Credentials and Access Management) Platform**				
	1.9.1	Enterprise PKI/IDP Pt. 1	1 - User	Zero Trust Target		
	1.9.2	Enterprise PKI/IDP Pt. 2	1 - User	Advanced Zero Trust		
	1.9.3	Enterprise PKI/IDP Pt. 3	1 - User	Advanced Zero Trust		
2.1		**Device Inventory**				
	2.1.1	Device Health Tool Gap Analysis	2 - Device	Zero Trust Target		
	2.1.2	NPE/PKI, Device under Management	2 - Device	Zero Trust Target		
	2.1.3	Enterprise IDP (Identity Provider) Pt. 1	2 - Device	Zero Trust Target		
	2.1.4	Enterprise IDP Pt. 2	2 - Device	Advanced Zero Trust		
2.2		**Device Detection and Compliance**				
	2.2.1	Implement C2C/Compliance-Based Network Authorization Pt. 1	2 - Device	Zero Trust Target		
	2.2.2	Implement C2C/Compliance-Based Network Authorization Pt. 2	2 - Device	Advanced Zero Trust		

2.3		**Device Authorization w/ Real Time Inspection**		
	2.3.1	Entity Activity Monitoring Pt. 1	2 - Device	Advanced Zero Trust
	2.3.2	Entity Activity Monitoring Pt. 2	2 - Device	Advanced Zero Trust
	2.3.3	Implement Application Control & File Integrity Monitoring (FIM) Tools	2 - Device	Zero Trust Target
	2.3.4	Integrate Next-Gen AV Tools with C2C	2 - Device	Zero Trust Target
	2.3.5	Fully Integrate Device Security stack with C2C as appropriate	2 - Device	Advanced Zero Trust
	2.3.6	Enterprise PKI Pt. 1	2 - Device	Advanced Zero Trust
	2.3.7	Enterprise PKI Pt. 2	2 - Device	Advanced Zero Trust
2.4		**Remote Access**		
	2.4.1	Deny Device by Default Policy	2 - Device	Zero Trust Target
	2.4.2	Managed and Limited BYOD & IoT Support	2 - Device	Zero Trust Target
	2.4.3	Managed and Full BYOD & IoT Support Pt. 1	2 - Device	Advanced Zero Trust
	2.4.4	Managed and Full BYOD & IoT Support Pt. 2	2 - Device	Advanced Zero Trust
2.5		**Partially & Fully Automated Asset, Vulnerability, and Patch Management**		
	2.5.1	Implement Asset, Vulnerability and Patch Management Tools	2 - Device	Zero Trust Target

(*continued*)

ID #		Capability/Activity	Pillar	Level	Security Controls	Compliant (Y/N)
2.6		**Unified Endpoint Management (UEM) & Mobile Device Management (MDM)**				
	2.6.1	Implement UEDM or Equivalent Tools	2 - Device	Zero Trust Target		
	2.6.2	Enterprise Device Management Pt. 1	2 - Device	Zero Trust Target		
	2.6.3	Enterprise Device Management Pt. 2	2 - Device	Zero Trust Target		
2.7		**Endpoint & Extended Detection & Response (EDR & XDR)**				
	2.7.1	Implement Endpoint Detection & Response (EDR) Tools and Integrate with C2C	2 - Device	Zero Trust Target		
	2.7.2	Implement Extended Detection & Response (XDR) Tools and Integrate with C2C Pt. 1	2 - Device	Zero Trust Target		
	2.7.3	Implement Extended Detection & Response (XDR) Tools and Integrate with C2C Pt. 2	2 - Device	Advanced Zero Trust		
3.1		**Application Inventory**				
	3.1.1	Application/Code Identification	3 - Applications and Workloads	Zero Trust Target		
	3.1.2	Resource Authorization Pt. 1	3 - Applications and Workloads	Zero Trust Target		
	3.1.3	Resource Authorization Pt. 2	3 - Applications and Workloads	Zero Trust Target		

3.2		**Secure Software Development & Integration**		
	3.2.1	Build DevSecOps Software Factory Pt. 1	3 - Applications and Workloads	Zero Trust Target
	3.2.2	Build DevSecOps Software Factory Pt. 2	3 - Applications and Workloads	Zero Trust Target
	3.2.3	Automate Application Security & Code Remediation Pt. 1	3 - Applications and Workloads	Zero Trust Target
	3.2.4	Automate Application Security & Code Remediation Pt. 2	3 - Applications and Workloads	Advanced Zero Trust
3.3		**Software Risk Management**		
	3.3.1	Approved Binaries/Code	3 - Applications and Workloads	Zero Trust Target
	3.3.2	Vulnerability Management Program Pt. 1	3 - Applications and Workloads	Zero Trust Target
	3.3.3	Vulnerability Management Program Pt. 2	3 - Applications and Workloads	Zero Trust Target
	3.3.4	Continual Validation	3 - Applications and Workloads	Zero Trust Target

(*continued*)

ID #	Capability/Activity	Pillar	Level	Security Controls	Compliant (Y/N)
3.4	**Resource Authorization & Integration**				
3.4.1	SDC Resource Authorization Pt. 1	3 - Applications and Workloads	Zero Trust Target		
3.4.2	SDC Resource Authorization Pt. 2	3 - Applications and Workloads	Zero Trust Target		
3.4.3	Enrich Attributes for Resource Authorization Pt. 1	3 - Applications and Workloads	Advanced Zero Trust		
3.4.4	Enrich Attributes for Resource Authorization Pt. 2	3 - Applications and Workloads	Advanced Zero Trust		
3.4.5	REST API Micro-Segments	3 - Applications and Workloads	Advanced Zero Trust		
3.5	**Continuous Monitoring and Ongoing Authorizations**				
3.5.1	Continuous Authorization to Operate (cATO) Pt.1	3 - Applications and Workloads	Advanced Zero Trust		
3.5.2	Continuous Authorization to Operate (cATO) Pt.2	3 - Applications and Workloads	Advanced Zero Trust		
4.1	**Data Catalog Risk Alignment**				
4.1.1	Data Analysis	4 - Data	Zero Trust Target		

4.2		**DoD (DEPARTMENT OF DEFENSE) Enterprise Data Governance**		
	4.2.1	Define Data Tagging Standards	4 - Data	Zero Trust Target
	4.2.2	Interoperability Standards	4 - Data	Zero Trust Target
	4.2.3	Develop Software Defined Storage (SDS) Policy	4 - Data	Zero Trust Target
4.3		**Data Labeling and Tagging**		
	4.3.1	Implement Data Tagging & Classification Tools	4 - Data	Zero Trust Target
	4.3.2	Manual Data Tagging Pt. 1	4 - Data	Zero Trust Target
	4.3.3	Automated Data Tagging & Support Pt. 1	4 - Data	Advanced Zero Trust
	4.3.4	Automated Data Tagging & Support Pt. 2	4 - Data	Advanced Zero Trust
4.4		**Data Monitoring and Sensing**		
	4.4.1	DLP Enforcement Point Logging and Analysis	4 - Data	Zero Trust Target
	4.4.2	DRM Enforcement Point Logging and Analysis	4 - Data	Zero Trust Target
	4.4.3	File Activity Monitoring Pt. 1	4 - Data	Zero Trust Target
	4.4.4	File Activity Monitoring Pt. 2	4 - Data	Zero Trust Target
	4.4.5	Database Activity Monitoring	4 - Data	Advanced Zero Trust
	4.4.6	Comprehensive Data Activity Monitoring	4 - Data	Advanced Zero Trust

(*continued*)

ID #		Capability/Activity	Pillar	Level	Security Controls	Compliant (Y/N)
4.5		**Data Encryption & Rights Management**				
	4.5.1	Implement DRM and Protection Tools Pt. 1	4 - Data	Zero Trust Target		
	4.5.2	Implement DRM and Protection Tools Pt. 2	4 - Data	Zero Trust Target		
	4.5.3	DRM Enforcement via Data Tags and Analytics Pt. 1	4 - Data	Zero Trust Target		
	4.5.4	DRM Enforcement via Data Tags and Analytics Pt. 2	4 - Data	Advanced Zero Trust		
	4.5.5	DRM Enforcement via Data Tags and Analytics Pt. 3	4 - Data	Advanced Zero Trust		
4.6		**Data Loss Prevention (DLP)**				
	4.6.1	Implement Enforcement Points	4 - Data	Zero Trust Target		
	4.6.2	DLP Enforcement via Data Tags and Analytics Pt1	4 - Data	Zero Trust Target		
	4.6.3	DLP Enforcement via Data Tags and Analytics Pt. 2	4 - Data	Advanced Zero Trust		
	4.6.4	DLP Enforcement via Data Tags and Analytics Pt. 3	4 - Data	Advanced Zero Trust		
4.7		**Data Access Control**				
	4.7.1	Integrate DAAS Access w/ SDS Policy Pt. 1	4 - Data	Zero Trust Target		
	4.7.2	Integrate DAAS Access w/ SDS Policy Pt. 2	4 - Data	Advanced Zero Trust		
	4.7.3	Integrate DAAS Access w/ SDS Policy Pt. 3	4 - Data	Advanced Zero Trust		
	4.7.4	Integrate Solution(s) and Policy with Enterprise IDP Pt. 1	4 - Data	Zero Trust Target		

	4.7.5	Integrate Solution(s) and Policy with Enterprise IDP Pt. 2	4 - Data	Advanced Zero Trust
	4.7.6	Integrate Solution(s) and Policy with Enterprise IDP Pt. 3	4 - Data	Advanced Zero Trust
	4.7.7	Integrate Solution(s) and Policy with Enterprise IDP Pt. 4	4 - Data	Advanced Zero Trust
5.1		**Data Flow Mapping**		
	5.1.1	Define Granular Control Access Rules & Policies Pt. 1	5 - Network and Environment	Zero Trust Target
	5.1.2	Define Granular Control Access Rules & Policies Pt. 2	5 - Network and Environment	Zero Trust Target
5.2		**Software Defined Networking (SDN)**		
	5.2.1	Define SDN APIs	5 - Network and Environment	Zero Trust Target
	5.2.2	Implement SDN Programmable Infrastructure	5 - Network and Environment	Zero Trust Target
	5.2.3	Segment Flows into Control, Management, and Data Planes	5 - Network and Environment	Zero Trust Target
	5.2.4	Network Asset Discovery & Optimization	5 - Network and Environment	Advanced Zero Trust
	5.2.5	Real-Time Access Decisions	5 - Network and Environment	Advanced Zero Trust

(*continued*)

ID #		Capability/Activity	Pillar	Level	Security Controls	Compliant (Y/N)
5.3		**Macro Segmentation**				
	5.3.1	Datacenter Macro segmentation	5 - Network and Environment	Zero Trust Target		
	5.3.2	B/C/P/S Macro segmentation	5 - Network and Environment	Zero Trust Target		
5.4		**Micro Segmentation**				
	5.4.1	Implement Micro segmentation	5 - Network and Environment	Zero Trust Target		
	5.4.2	Application & Device Micro segmentation	5 - Network and Environment	Zero Trust Target		
	5.4.3	Process Micro segmentation	5 - Network and Environment	Advanced Zero Trust		
	5.4.4	Protect Data In Transit	5 - Network and Environment	Zero Trust Target		
6.1		**Policy Decision Point (PDP) & Policy Orchestration**				
	6.1.1	Policy Inventory & Development	6 - Automation and Orchestration	Zero Trust Target		
	6.1.2	Organization Access Profile	6 - Automation and Orchestration	Zero Trust Target		

	6.1.3	Enterprise Security Profile Pt. 1	6 - Automation and Orchestration	Zero Trust Target
	6.1.4	Enterprise Security Profile Pt. 2	6 - Automation and Orchestration	Advanced Zero Trust
6.2		**Critical Process Automation**		
	6.2.1	Task Automation Analysis	6 - Automation and Orchestration	Zero Trust Target
	6.2.2	Enterprise Integration & Workflow Provisioning Pt. 1	6 - Automation and Orchestration	Zero Trust Target
	6.2.3	Enterprise Integration & Workflow Provisioning Pt. 2	6 - Automation and Orchestration	Advanced Zero Trust
6.3		**Machine Learning**		
	6.3.1	Implement Data Tagging & Classification ML (Machine Learning) Tools	6 - Automation and Orchestration	Zero Trust Target
6.4		**Artificial Intelligence**		
	6.4.1	Implement AI automation tools	6 - Automation and Orchestration	Advanced Zero Trust
	6.4.2	AI Driven by Analytics decides A&O (Authorizing Official) modifications	6 - Automation and Orchestration	Advanced Zero Trust

(*continued*)

ID #	Capability/Activity	Pillar	Level	Security Controls	Compliant (Y/N)
6.5	**Security Orchestration, Automation & Response (SOAR)**				
6.5.1	Response Automation Analysis	6 - Automation and Orchestration	Zero Trust Target		
6.5.2	Implement SOAR Tools	6 - Automation and Orchestration	Zero Trust Target		
6.5.3	Implement Playbooks	6 - Automation and Orchestration	Advanced Zero Trust		
6.6	**API Standardization**				
6.6.1	Tool Compliance Analysis	6 - Automation and Orchestration	Zero Trust Target		
6.6.2	Standardized API Calls & Schemas Pt. 1	6 - Automation and Orchestration	Zero Trust Target		
6.6.3	Standardized API Calls & Schemas Pt. 2	6 - Automation and Orchestration	Zero Trust Target		
6.7	**Security Operations Center (SOC) & Incident Response (IR)**				
6.7.1	Workflow Enrichment Pt. 1	6 - Automation and Orchestration	Zero Trust Target		
6.7.2	Workflow Enrichment Pt. 2	6 - Automation and Orchestration	Zero Trust Target		

	6.7.3	Workflow Enrichment Pt. 3	6 - Automation and Orchestration	Advanced Zero Trust
	6.7.4	Automated Workflow	6 - Automation and Orchestration	Advanced Zero Trust
7.1		**Log All Traffic (Network, Data, Apps, Users)**		
	7.1.1	Scale Considerations	7 - Visibility and Analytics	Zero Trust Target
	7.1.2	Log Parsing	7 - Visibility and Analytics	Zero Trust Target
	7.1.3	Log Analysis	7 - Visibility and Analytics	Zero Trust Target
7.2		**Security Information and Event Management (SIEM)**		
	7.2.1	Threat Alerting Pt. 1	7 - Visibility and Analytics	Zero Trust Target
	7.2.2	Threat Alerting Pt. 2	7 - Visibility and Analytics	Zero Trust Target
	7.2.3	Threat Alerting Pt. 3	7 - Visibility and Analytics	Advanced Zero Trust
	7.2.4	Asset ID & Alert Correlation	7 - Visibility and Analytics	Zero Trust Target
	7.2.5	User/Device Baselines	7 - Visibility and Analytics	Zero Trust Target
7.3		**Common Security and Risk Analytics**		
	7.3.1	Implement Analytics Tools	7 - Visibility and Analytics	Zero Trust Target
	7.3.2	Establish User Baseline Behavior	7 - Visibility and Analytics	Zero Trust Target
7.4		**User and Entity Behavior Analytics**		
	7.4.1	Baseline & Profiling Pt. 1	7 - Visibility and Analytics	Zero Trust Target

(*continued*)

ID #	Capability/Activity	Pillar	Level	Security Controls	Compliant (Y/N)
7.4.2	Baseline & Profiling Pt. 2	7 - Visibility and Analytics	Advanced Zero Trust		
7.4.3	UEBA Baseline Support Pt. 1	7 - Visibility and Analytics	Advanced Zero Trust		
7.4.4	UEBA Baseline Support Pt. 2	7 - Visibility and Analytics	Advanced Zero Trust		
7.5	**Threat Intelligence Integration**				
7.5.1	Cyber Threat Intelligence Program Pt. 1	7 - Visibility and Analytics	Zero Trust Target		
7.5.2	Cyber Threat Intelligence Program Pt. 2	7 - Visibility and Analytics	Advanced Zero Trust		
7.6	**Automated Dynamic Policies**				
7.6.1	AI-enabled Network Access	7 - Visibility and Analytics	Advanced Zero Trust		
7.6.2	AI-enabled Dynamic Access Control	7 - Visibility and Analytics	Advanced Zero Trust		

Index

A

Access control lists (ACLs), 49, 158, 213, 237
Access management (AM), 60, 84, 185, 236
Access rights, 51, 87
Accountability, 65, 136, 138, 139
Account Password Management (APM), 77
Accounts, 32, 136, 137, 142, 147, 214, 215, 220, 222, 224
  administrator accounts, 90
  application accounts, 48, 49, 91
  centralized accounts, 45, 46
  cloud accounts, 49, 50
  daily drivers accounts, 92
  emergency access accounts, 92
  functional accounts, 46, 47
  and identity/credentials relationships, 43
  local accounts, 44, 45
  managed accounts, 47
  privileged user accounts, 91
  root, 9, 23, 33, 43, 45, 49, 65, 90
  service accounts, 47, 48, 91
  uses, 43
Active Directory (AD), 45, 85, 88, 216
Active Directory users and computers (ADUC), 270
Act on the Protection of Personal Information (APPI), 248
Administration, 24–26, 147, 148, 212, 221, 229, 231, 232
Administrative rights, 31, 46, 75, 138, 143, 144, 219, 229, 230, 269
Administrator, 8, 23, 26, 28, 31, 37, 43, 49, 56, 66, 67, 75, 127
Advanced persistent threats (APTs), 150
AI-driven attacks, 150
Analytics, 25, 26, 136, 140, 148, 150
Anonymizing proxy services, 130
Antivirus (AV), 13, 17, 18, 183, 261, 262
API standardization, 286
Application control, 19, 71, 144, 145, 154, 183, 184, 225, 230, 231, 236, 269, 277
Application inventory, 278
Application password management (APM), 78, 141
Application privileges, 14, 70, 144, 145
Application programming interfaces (API), 48, 125, 177, 220
Applications, 37–39, 48, 49
Application-to-application accounts (A2A), 48, 78
Application-to-Application Password Management (AAPM), 78
Architectures, 46, 205, 210, 222, 230, 271
Artificial intelligence (AI), 7, 26, 28, 37, 57, 167, 285
Asset management, 19, 137
Attackers, 62, 82, 83, 122, 129–132, 154
Audible crimes, 31
Audit, 24, 25, 138, 141, 144, 154, 155, 160, 225, 226, 230, 234
Authentication (AuthN), 22–25, 60–62, 153, 159, 167, 171–178, 275

© Morey J. Haber, Darran Rolls 2024
M. J. Haber and D. Rolls, *Identity Attack Vectors*, https://doi.org/10.1007/979-8-8688-0233-1

Authorization (AuthZ), 22–25, 157, 158, 171–173
Automated dynamic policies, 288
Automated systems, 1
Automation, 28, 42–43, 46, 47, 97, 100, 140, 162, 181, 285
Azure AD, 23, 45, 131, 266

B

Bastion hosts, 16, 212, 218, 226
Behavioral analysis, 1, 209, 225
Biometrics, 22, 34, 60, 171, 172, 275
Bring Your Own Device (BYOD), 142, 166, 224
Business justification, 263–264
Business requirements, 58, 136, 197, 201, 212
Business roles, 52–55, 88, 167, 168

C

Certificate Authority (CA), 34
Certificates, 34, 65, 69, 138
Certification, 25, 50, 75, 78, 200, 201
ChatGPT, 167
Chief Financial Officer (CFO), 29
Child exploitation, 32
Children's Online Privacy Protection Act (COPPA), 246
Client-server applications, 169, 170
Cloud, 13, 16, 49–51, 68, 71, 73, 81, 82, 85, 86, 192, 193, 265, 266
Cloud connectors, 131
Cloud Infrastructure Entitlement Management (CIEM)
  best practices, 73
  cloud providers, 72
  definition, 71
  integration, 72
  least privilege, 72
Cloud service providers (CSP), 71, 72
Commercial of the Shelf (COTS), 160, 161
Common Identity Exposures (CIE), 250
Compliance, 26, 58, 62, 63, 72, 78, 153, 155, 174, 213, 221, 222, 225, 232, 235, 243, 248–250, 265, 268, 271
Compliance violations, 61, 62
Conditional user access, 274
Configuration, 19, 25, 26, 54, 82, 84, 85, 195, 269
Context-aware access, 201
Control Objectives for Information and Related Technology (COBIT), 21
COVID, 163, 167, 169
Credentials, 28, 34–37, 45, 48, 158, 159, 161, 171, 177, 181, 266
Credential theft, 62, 91
Critical process automation, 285
Customization, 16, 95, 98, 192, 195
Cyberattacks, 10, 82, 120, 158, 173, 262
Cyber awareness training, 166
Cyber insurance, 162–167, 180, 215, 257
Cyber insurance market, 163
Cyber Kill Chain, 79
  exfiltration, 123, 124
  exploitation, 122, 123
  infiltration, 121, 122
  reconnaissance, 121
Cybersecurity, 4, 13, 24, 214
  circles, 18
  correlation, 15
  EPM, 14
  layers, 15, 17
  oversight and control, 15
  security vendors, 17

SIEM, 14
stabilization, 18

D

Darwin, 261, 262
Data access control, 282
Database administrator (DBA), 30, 31, 89, 91
Data breaches, 61, 100, 155, 162
Data encryption & rights management, 282
Data flow mapping, 283
Data Loss Prevention (DLP), 281, 282
Data monitoring and sensing, 281
Data Security Standard (DSS), 246
Denial-of-service (DoS), 8, 159, 182
Department of Defense (DoD) framework
AI, 285
API standardization, 286
application inventory, 278
asset/vulnerability/patch management, 277
authentication, 275
automated dynamic policies, 288
behavioral/contextual ID/biometrics, 275
common security and risk analytics, 287
conditional user access, 274
continuous monitoring, 280
critical process automation, 285
data access control, 282
data catalog risk alignment, 280
data encryption & rights management, 282
data flow mapping, 283
data labeling and tagging, 281
data monitoring and sensing, 281
device detection and compliance, 276
device detection authorization, 277
device inventory, 276
DLP, 282
EDR, 278
enterprise data governance, 281
ICAM, 276
identity federation & user credentialing, 275
IR, 286, 287
least privileged access, 275
macro segmentation, 284
MDM, 278
MFA, 274
micro segmentation, 284
ML, 285
PAM, 274
PDP, 284
policy orchestration, 284
remote access, 277
resource authorization & integration, 280
SDN, 283
secure software development & integration, 279
SIEM, 287
SOAR, 286
SOC, 286, 287
software risk management, 279
threat intelligence integration, 288
UEBA, 287, 288
UEM, 278
user inventory, 274
XDR, 278
Department of Financial Services (DFS), 243
Designators, 28, 35, 36, 39, 56, 137, 251
Device detection and compliance, 276
Device inventory, 276

DevOps, 48, 65, 69, 71, 253, 266
  characteristics, 154
  CI/CD-related risks, 154
  risks, 153
Digital certificate, 22, 34
Digital transformation, 86, 182, 192, 196
Directory bridging technology, 60, 234
Directory services, 32, 36, 45, 60, 148
Directory services bridging, 71, 270, 271
Disaster recovery (DR), 75, 81, 183, 220
Discovery, 73, 76, 113, 138, 139, 147, 266

E

Electronic discovery, 137
Endpoint Detection and Response (EDR), 83, 84, 117, 126, 183, 278
Endpoint privilege management (EPM), 14, 55, 78, 230
Endpoint security, 16, 83, 165, 166, 231, 238
End-user machines, 143
End-user roles, 229
Entitlements, 50, 137, 150, 151, 199–201, 215, 254
  complex entitlements, 51
  simple entitlements, 51
Entra ID, 23, 45, 59, 131, 139, 148, 191, 234, 270
Exclusions, 164, 165
Executive team member, 4, 58
Extended Detection and Response (XDR), 83, 84, 278
Extortion, 31, 174

F

Fabric, 10, 58, 59, 61, 135, 185, 187
Failure, 3–5, 35, 56, 61, 96, 97, 125
File access monitoring, 148
File Integrity Monitoring (FIM), 146, 148, 232, 277
Financial fraud, 156, 174
Financial organizations, 10, 243, 245, 246
Forensics, 29, 39, 76, 79, 125, 146, 199
Fraud, 31, 110, 155, 246

G

Gateway, 16, 169, 170, 210, 217, 218, 220, 226, 238
General Data Protection Regulation (GDPR), 62, 119, 246, 247
Generative AI, 28, 167, 168
Gramm-Leach-Bliley Act (GLBA), 246
Group Policy Objects (GPO), 234
Guest shopping carts, 240

H

Hacking, 31, 35, 92, 98, 112, 115, 180, 183
Health Insurance Portability and Accountability Act (HIPAA), 62, 119, 149, 243, 246
Help desk and ticketing, 74
Honest mistakes, 5
HTTP Archive file (HAR), 104, 126
Human identity, 3, 5, 27, 33, 39, 42, 69, 163

I

Identity, 27, 28, 55, 82, 83, 109, 251, 253, 270, 271
Identity accountability
  characteristics, 139–141
  privileged credentials, 138
  types, 138

Identity and access management (IAM), 7, 25, 27, 42, 57, 63, 65
  architecture, 59–61
  best practices, 62, 63
  discipline model, 57–59
  features, 137, 138
  ITDR, 84, 85
  risks, 61, 62
  tools, 82
  vendors, 129
Identity attack, 109, 110, 130
Identity attack vectors, 109, 120, 122, 127, 131, 138, 153, 158, 163, 203, 205, 239
  implications, 114, 115
  methods, 110, 111
  privileges
    accounts, 116
    account types, 116, 118
    concept, 118
    entitlements, 116
    IGA, 117
    least privilege, 116, 118
    PAM, 117
    privileged attack chain, 117
  tactics, 111–113
Identity-based compliance, 249–251
Identity Credentials and Access Management (ICAM), 276
Identity crisis, 7
  detections/recommendations, 7–9
  events, 10
  IAM tools, 10
  identity-based attack, 9
  ITDR, 9
  organization, 7
Identity Defined Security Alliance (IDSA), 82
Identity digital transformations, 195–197
Identity federation & user credentialing, 275
Identity governance, 25, 63, 71, 191–193, 254
Identity Governance and Administration (IGA), 7, 23–25, 42, 44, 59, 65, 84, 113, 192
Identity lifecycle management, 62, 84, 275
Identity management, 21, 22, 42–44, 119, 185, 191
Identity obfuscation, 239–241
Identity Provider (IdP), 59, 71, 127, 129–131, 251
Identity risk, 8, 110, 150, 173, 243, 250
Identity security, 244, 262
  best practices, 249
  challenges, 150
  compliance regulations, 245, 246
  cost of ownership, 136
  definition, 248
  importance, 245
  integrations, 136
  longevity, 136
  modern, 151
  overall strategy, 135
  time-to-value, 136
  trends, 149
  uses, 248
Identity security attacks, 132
Identity security capabilities, 264, 265
Identity technical debt
  definition, 191
  methods, 192
  sources, 191, 192
Identity Theft, 32, 62, 105, 114, 115, 173
Identity threat detection and response (ITDR), 1–3, 9, 127, 183
  definition, 81
  EDR, 83, 84
  IAM, 84, 85

Identity threat detection and response (ITDR) (*cont.*)
IGA, 84
organizations, 83
PAM, 85, 86
XDR, 83, 84
Incident response (IR), 21, 75, 286, 287
Incident response plan, 63
Incognito browsing mode, 239
Indicators of compromise (IoCs), 29, 126, 150, 201
attack vectors
brute force, 97
credential stuffing, 100
dictionary attacks, 95, 96
password guessing, 92–94
password resets, 102, 103
password spraying, 100, 101
PtH, 98
security questions, 98, 99
shoulder surfing, 94, 95
token hijacking, 104
characteristics, 87
end users, 88
forms, 87
goal, 87
identity-based, 105–107
role-based
account types, 90–92
AD, 88
attack vectors, 90–92
best practices, 89, 90
DBA, 89
identities, 90
IGA tools, 88
tracking, 88
Industrial control systems (ICS), 65, 160, 226
Industrial IoT (IIoT) devices, 160
Infiltration, 121, 122
Information technology (IT), 33, 37, 51, 53, 54, 240
Infrastructure as a Service (IaaS), 49, 104, 143, 220, 268
Insider threats, 61, 87, 91, 110, 129, 158
Insurance, 162, 164, 257, 263
International Standards Organization (ISO), 21
Internet of Things (IoT), 18, 138, 159, 160, 226, 237
IP address relays, 240

J

Just-in-time (JIT), 183, 267
administration, 147
context-aware, 201
creation and deletion, 199
disabled administrator accounts, 200
entitlements, 201
group membership, 200
impersonation, 200
privileges, 200
tokenization, 201
triggers, 202
2FA, 202
workflow, 201, 203

K

Key Performance Indicators (KPIs), 76, 263
Key Risk Indictors (KRIs), 76
Kill Chain, 121, 123

L

Land and expand techniques, 82
Least privilege, 43, 54, 55, 63, 269, 270
  characteristics, 230
  endpoint privilege management, 231
  productivity, 229
  security, 229
  Unix/Linux, 232, 233
  users, 143
Least privileged access, 61, 73, 161, 275
Lies, 4, 5, 164, 171, 250
Lifecycle management, 25, 54, 84, 275
Lightweight Directory Access Protocol (LDAP), 45, 59, 149, 270
Local login rights, 38
Log aggregator, 7
Log analysis, 287

M

Machine identity, 3–5, 33, 158, 167
Machine learning (ML), 26, 285
Machines, 3, 5, 40–42, 132, 154, 214, 234
Macro segmentation, 284
Masquerade attack, 127, 158, 159
Micro segmentation, 161, 207, 213, 227, 284
Misinformation, 5
Mobile Device Management (MDM), 278
Modern identity attacker
  limit privileges, 132
  phishing-resistant, 131
  privileged accounts, 132
Monumental identity security threat, 134
Multifactor authentication (MFA), 7–9, 60, 74, 134, 143, 165, 174, 202, 274
  fatigue, 174
  push notifications
    application vulnerabilities, 175
    bombing/fatigue, 174
    lost/stolen device, 175
    phishing attacks, 175
    privacy, 176
    single device, 176
    user error, 175
  SIM jacking, 173
  SMS texting, 174

N

National Institute of Standards and Technology (NIST), 21, 119, 205–209, 216–218, 250
Network analysis, 2
Network security holes, 19
New Technology LAN Manager (NTLM), 98
Non-Windows environments, 147
North American Electric Reliability Council (NERC), 246

O

Office of Personnel Management (OPM), 171–173
One-time passwords (OTPs), 131
Operational technology (OT), 160, 226
Org2Org, 130
Organizations, 29, 31, 36, 50, 65, 68, 71, 76, 82, 83, 239, 243, 250
Original Equipment Management (OEM), 258
Ownership, 40–42, 76, 168, 208, 246

P, Q

Passcode, 33, 35, 176
Passkey, 34, 35, 44, 60, 175
Pass-the-Hash (PtH), 98
Password, 33–35, 45, 47
  Android, 177
  Apple macOS, 177
  iPhone/iPad, 178
  locally installed applications, 177
  Microsoft Windows, 177
  operating system, 176
  third-party solutions, 178
  website, 177
Passwordless, 176–179, 267
Password management, 78, 92, 98, 138–141, 143, 162, 179, 225, 238, 267, 268
Payment Card Industry (PCI), 246
Payment Control Industry (PCI), 213
Peer-to-peer (P2P) networking technology, 237, 238
Performance obfuscation, 239
Permissions, 7, 24, 32, 45, 47, 53, 71, 72, 137, 143, 144, 199, 200, 254, 269
Personal Information Protection Act (PIPA), 247, 248
Personally identifiable information (PII), 36, 56, 157, 168
Personas, 29, 32, 172
  electronic persona, 30–32
  physical persona, 30–32
Phishing, 29, 31, 34, 163, 224
Physical security measures, 2
Platform as a Service (PaaS), 49
Policy Administrator (PA), 210, 217, 218
Policy Decision Point (PDP), 210, 284
Policy Enforcement Point (PEP), 210, 217–219
Policy Engine (PE), 210, 217–219, 222, 233
Policy Information Point (PIP), 210
Policy orchestration, 284
Privileged access management (PAM), 7, 44, 50, 61, 244, 245, 274
  analytics/reporting
    disciplines, 78
    forensics, 76
    identity security requirements, 77, 78
    identity security visibility, 78, 79
    KPIs, 76
    KRIs, 76
    privileged audit, 76
  capabilities, 266, 267
  CIEM, 71, 73
  components, 66, 67
  definition, 65
  discovery, 76
  governance, 75, 76
  and IAM, 79
  identities and accounts, 66
  ITDR, 85, 86
  password storage, 67, 68
  privilege management, 70, 71
  resources, 65
  roles/policies, 75
  secret management, 68, 69
  session management, 69, 70
  solutions, 66
  system integration, 74, 75
  zero trust, 215
Privileged account, 220
Privileged Account and Session Management (PASM), 77, 78
Privileged account management (PAM)
  secrets management, 221
  use cases, 220, 221

Privileged audit, 67, 76
Privileged credentials, 138–141, 220
Privileged password management, 77, 140, 220, 221
Privileged rights, 78
Privileged Session Management (PSM), 77
Privilege Elevation and Delegation Management (PEDM), 78
Privileged management, 57, 70, 71, 145, 147, 225, 270
Privileges, 215, 229, 238
Public key infrastructure (PKI), 34
Push notifications, 175, 176

R

Ransomware, 31, 112, 117, 262
  classes, 182
  identity security measures, 183
  working, 180–182
Ransomware as a Service (RaaS), 180, 182
Ransomware Supplemental Addendum, 164
Real-world identity attack, 124, 125, 127, 128
Regulatory compliance, 119, 215, 232
  best practices, 248, 249
  definition, 243
  DFS, 243
  financial organizations, 243
  global, 247, 248
  organizations, 243
  United States (US), 245, 246
  violations, 243
Remote access, 277
  authorized users, 222
  connectivity, 224
  identity security, 225
  low-cost, 222
  managed security, 224
  privileged access management, 225
  privileged users, 142
  RDP, 141
  risks, 223
  security control, 222
  security controls and capabilities, 142, 143
  traits, 226
  VPN, 142, 227
  zero trust, 225, 228
Remote Desktop Protocol (RDP), 141, 180, 182
Request for proposal (RFP), 79
Rights, 126, 137, 161, 163, 200, 230
Robotic process automation (RPA), 48, 161, 162, 266
Robust security program, 15
Role-based access control (RBAC), 37, 60, 62, 67
Roles
  business roles, 53
  definition, 53
  IT roles, 53, 54
  and least privilege, 54, 55
  two-tier role model, 52
  uses, 53
Root, 142, 147

S

Sarbanes-Oxley Act (SOX), 243, 245
Screen privacy filters, 240
Secret management, 68, 69
Secure remote access technology, 169, 170, 223
Secure software development & integration, 279

Security frameworks, 21, 119
Security Identifier (SID), 36
Security industry, 81
Security Information and Event Management (SIEM), 7, 74, 79, 83, 241, 287
Security information and event manager (SIEM), 14
Security Operations Center (SOC), 125, 153, 286, 287
Security Orchestration, Automation & Response (SOAR), 83, 286
Security tools, 81, 83
Security vendors, 17, 81, 141
Separation of duty (SoD), 25, 56
Server privilege management, 145–147
Service accounts, 47–48, 91, 139, 165, 266
Session management, 69, 70, 77, 220
Session Recording and Monitoring (SRM), 77
Shadow IT, 62, 174, 193
SIM hijacking, 133
SIM jacking, 112, 173, 174
SIM swapping, 129, 173
Single Sign-On (SSO), 7, 8, 24, 45, 46, 60, 74, 88, 101, 125, 270
SMS texting, 174
Social media monitoring, 1
Social Security number (SSN), 36, 56, 156
Software as a Service (SaaS), 48, 49, 90, 104, 126, 196
Software Defined Networking (SDN), 283
Software risk management, 279
SSO vendor, 126, 130
Stakeholders, 58
Standard operating procedures (SOP), 69, 75, 156
Super administrator, 129, 130, 132
Super admin roles, 128
Supervisory control and data acquisition (SCADA) systems, 160
Synthetic identities, 56, 112, 155–157, 239
System for Cross-Domain Identity Management (SCIM), 66, 143, 193, 268

T

Tactics, Techniques, and Procedures (TTPs), 128
Technology partnerships, 58, 61
Third-party access, 91, 166
Third-party vendors, 142
Threat actors, 90–92, 95, 97, 101, 102, 105, 109–114, 116, 117
Threat intelligence, 2, 81, 83
3D printed masks, 172
Time-based one-time passwords (TOTP), 175
Titor, J., 3, 36, 88
Transparency, 243, 248, 253
Two-factor authentication (2FA), 23, 46, 98, 99, 103, 171, 202

U

Unauthorized access, 22, 59, 62, 105, 167, 175
Unified Endpoint Management (UEM), 278
Use and Protection of Credit Information Act (UPCIA), 248
User Activity Monitoring (UAM), 275
User & Entity Behavior Activity (UEBA), 275

User and Entity Behavior Analytics (UEBA), 287, 288
User behavior analytics (UBA), 69, 78
User behavior monitoring, 67, 70
User experience, 53, 60, 148, 255
User inventory, 274
Users, 10, 35–37, 269, 270, 287

V

Vendor, 257, 259
  access, 91
  consolidation, 259
  cost, 258
  incompatibilities, 258
  security breach, 257
  strategic direction, 258
  technical support, 259
Venn diagram, 13, 14, 18, 137
Virtual desktop interface (VDI), 169, 170
Virtual private networks (VPNs), 166, 169, 170, 182, 240
Visibility, 253–254
Vulnerability management, 19, 165

W, X, Y

Watering hole, 105, 158, 159
Web-Based Fraud, 31
Windows privilege management, 145

Z

Zero standing privilege (ZSP), 46, 203, 223
Zero Trust, 81, 274–276
Zero-trust architecture (ZTA)
  access request, 209
  cloud environments, 215
  communication pathways, 214
  components, 210
  definition, 208
  design considerations, 236
  goal, 208
  implement, 205
  log privileged activity, 237
  log remote session activity, 237
  log screen activity, 237
  measurable controls, 235
  modern privileged access, 215
  network-based security controls, 212
  NIST, 206, 207
  NIST 1800-35B, 209
  NIST SP 800-207, 209
  P2P, 238
  PAM fits NIST, 217
  requirements, 212
  SP 1800-35B, 209
  workflows, 206
Zero-Trust remote access technology, 233, 268
Zero-trust security principles, 160