Privileged Attack Vectors

Building Effective Cyber-Defense Strategies to Protect Organizations

Morey J. Haber
Brad Hibbert

Apress®

www.allitebooks.com

Privileged Attack Vectors

Building Effective Cyber-Defense Strategies to Protect Organizations

Morey J. Haber
Brad Hibbert

Apress®

www.allitebooks.com

Privileged Attack Vectors: Building Effective Cyber-Defense Strategies to Protect Organizations

Morey J. Haber
Heathrow, Florida, USA

Brad Hibbert
Carp, Ontario, Canada

ISBN-13 (pbk): 978-1-4842-3047-3
ISBN-13 (electronic): 978-1-4842-3048-0
https://doi.org/10.1007/978-1-4842-3048-0

Library of Congress Control Number: 2017962003

Copyright © 2018 by Morey J. Haber and Brad Hibbert

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.

Cover image by Freepik (www.freepik.com)

Managing Director: Welmoed Spahr
Editorial Director: Todd Green
Acquisitions Editor: Susan McDermott
Development Editor: Laura Berendson
Technical Reviewer: Derek A. Smith
Coordinating Editor: Rita Fernando
Copy Editor: Karen Jameson

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. 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 rights@apress.com, or visit http://www.apress.com/rights-permissions.

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 GitHub via the book's product page, located at www.apress.com/9781484230473. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

www.allitebooks.com

Jakob, Daniel, Gabrielle, and Arielle;
you are my world.

—MJH

www.allitebooks.com

Table of Contents

www.allitebooks.com

About the Authors

With 20+ years' of IT industry experience, **Morey J. Haber** joined BeyondTrust in 2012 as a part of the eEye Digital Security acquisition and oversees strategy for both vulnerability and privileged access management. In 2004, Morey joined eEye as the Director of Security Engineering and was responsible for strategic business discussions and vulnerability management architectures in Fortune 500 clients. Prior to eEye, he was a Development Manager for Computer Associates, Inc. (CA), responsible for new product beta cycles and key customer accounts. Morey began his career as a Reliability and Maintainability Engineer for a government contractor building flight and training simulators.

With over 20+ years' experience in product strategy and management, **Brad Hibbert** leads BeyondTrust's solution strategy and development. He joined BeyondTrust via the company's acquisition of eEye Digital Security, where Brad led strategy and products. Under Brad's leadership, eEye launched several market firsts, including vulnerability management solutions for cloud, mobile, and virtualization technologies. Prior to eEye, Brad served as Vice President of Strategy and Products at NetPro before its acquisition in 2008 by Quest Software. Over the years Brad has attained many industry certifications to support his management, consulting, and development activities. Brad has his Bachelor of Commerce, Specialization in Management Information Systems, and MBA from the University of Ottawa.

About the Technical Reviewer

Derek A. Smith is an expert at cybersecurity, cyber forensics, health care IT, SCADA security, physical security, investigations, organizational leadership, and training. He is currently an IT program manager with the federal government; a cybersecurity Associate Professor at the University of Maryland, University College, and the Virginia University of Science and Technology; and runs a small cybersecurity training company. Derek has completed three cybersecurity books and contributed a chapter for a fourth. He currently speaks at cybersecurity events throughout America and performs webinars for several companies as one of their cyber experts. Formerly, Derek worked for a number of IT companies, Computer Sciences Corporation and Booz Allen Hamilton among them. Derek spent 18 years as a special agent for various government agencies and the military. He has also taught business and IT courses at several universities for over 25 years. Derek has served in the U.S. Navy, Air Force, and Army for a total of 24 years. He completed an MBA, MS in IT Information Assurance, Masters in IT Project Management, MS in Digital Forensics, a BS in Education, and several associate degrees. He completed all but the dissertation for a doctorate.

Foreword

Most people who work in marketing are looking for impact with what they do, say, or show to prospects. If you use a word like risk in information security, it can mean 20 different things to 15 different people. Each peddler of information security technology is always looking to go bigger, more dramatic with each blog, webinar, or conference talk. I think the world as we know it has been predicted to end four or five times by now (I just wish IDS would stay dead!).

After spending the first part of my career with exploit writers, penetration testers, etc., you figure out pretty quickly that these folks have zero tolerance for marketing people. Actually, it's not the marketing people they have zero tolerance for; it's for people who overpromise and underdeliver when it comes to technology. They feel that they have been lied to on more than one occasion - and this is a pervasive problem in our space today. And while these technologists get a bad rap for being direct and overly inquisitive, they are some of the smartest people I have ever worked with. And through simple osmosis (because I'm in marketing and clearly not that smart), I have learned a few things about information security … and please excuse the overly pragmatic approach.

- You need to be self-effacing when it comes to security. I always ask potential customers, "what kind of shop are you?" By this I mean do you really - for reals as my kids would say - want the answer to the question, "is all the security stuff I bought actually working and protecting us?" What if you still have serious gaps? Are you comfortable going to management or the board with that? Or are you the totally locked down, 'we've

got it handled' kind of person? Side note, the few folks who actually say that they do have it all handled are the biggest nozzles out there today. You know the guy who has the answers to EVERYTHING? Ugh! I think I just threw up in my mouth.

- Ten years ago, security teams were four people and 10 tools; now there are five people and 20 tools. There are too many threats for teams to handle. Every day there is a new threat or a new problem. Even the most well-funded and staffed security teams simply cannot stop everything. But here is the dirty security secret. Come close, closer … if a hacker with unlimited time and skill wants to get into your network, he or she will most likely be able to do it 100 percent of the time, and there isn't anything you can do to stop them. So why do any of this? In reality, you're building your security to be better than your competitors. Let me explain with an analogy: if there is a group of people being chased by a bear in the woods, you don't have to outrun the bear. You just have to be faster than one of the other people running from the bear. The same applies to security. Hackers aren't looking for an eloquent vulnerability, they just want to get in. They are looking for the path of least resistance, and if you are better than the 10 other banks out there, they'll most likely go somewhere else.
- We have to stop talking about whatever is hot or new or sexy, like today's trends of IoT, SaaS, AI, or MDM, as we are doing a tremendous disservice to the industry as a whole. Because we were early on a particular topic, we get bored with it right around the time the majority of folks start thinking about implementing it. This leads

> me to my last point: the basics. Security isn't about stopping the unknown or doing the cool new things; it's about doing the basic things really well, deciding what you can and can't do, and focusing. There are always going to be new threats and you can't stop them all. People say you need to prepare for the next zero-day attack, which means preparing for something that you don't know about … um, what? Yes, prepare for the unexpected - never mind the fact that you haven't updated your firewalls in five months or patched your servers. You have to pick and choose and make sure you are actually maximizing your investments and helping your organization take positive, incremental steps forward.

So where is this all going? I've worked in information security for just about 20 years in five different technology companies, and at some point in time privilege always comes into play, either intentionally or by user stupidity. If a hacker gets in, they leverage the user's privilege to move around the organization (lateral movement), or if you have disgruntled employees, they just access stuff that they shouldn't. It's the one common link across most threats. In order to do something bad in an organization, you have to be able to move from machine to machine, and for that you always need privilege. I want to be clear: there is no magic security pill. The only way to be secure is to work at it diligently. Privilege is by no means a cure-all, but it is one of the "risks" that you can solve pretty easily today, and maybe that and the steps outlined in this book helps you run just a bit faster than the next guy and not get clubbed by the bear.

Michael Yaffe
Vice President of Marketing, BeyondTrust

Acknowledgments

Contributions by:
Michael Yaffe, Vice President of Marketing, BeyondTrust
Scott Lang, Director of Marketing, BeyondTrust
Martin Cannard, Director Product Management, BeyondTrust
Matt Miller, Content Manager, BeyondTrust
Sandi Green, Product Marketing Manager, BeyondTrust

Illustrations by: Chris Burd and Erin Ferguson, Marketing, BeyondTrust

Introduction

As highlighted in many articles, breach reports, and studies, most cyber-attacks originate from outside the organization. While the specific tactics may vary, the stages of an external attack are similar (see Figure 1).

1. **First, they hack the perimeter.**

   Attackers could penetrate the perimeter directly, but more than likely they execute a successful drive-by download, or launch a phishing attack to compromise a user's system, and establish a foothold inside the network; they do this all the while flying "under the radar" of many traditional security defenses.

2. **Next, they establish a connection.**

   Unless it's ransomware or self-contained malware, the attacker quickly establishes a connection to a command and control (C&C) server to download toolkits, additional payloads, and to receive additional instructions.

Note: Social attacks were utilized in 43% of all breaches in the 2017 Verizon Data Investigations Report dataset. Almost all phishing attacks that led to a breach were followed with some form of malware, and 28% of phishing breaches were targeted. Phishing is the most common social tactic in the Verizon DBIR dataset (93% of social incidents).

---

3. **Now inside the network, the attacker goes to work.**

   Attackers begin to learn about the network, the layout, and the assets. They begin to move laterally to other systems and look for opportunities to collect additional credentials, upgrade privileges, or just use the privileges that they have already compromised to access systems, applications, and data. Note that an insider can either become a hacker, or if they have the necessary privileges, they can jump right to step number 4.

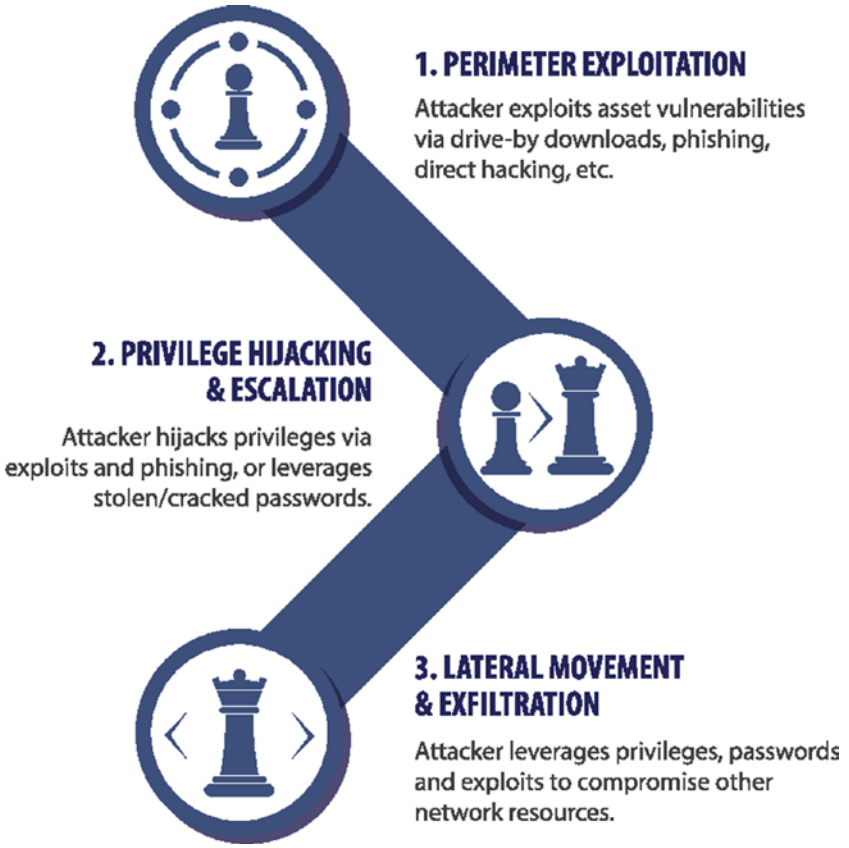

Figure 1. *Stages of an external attack*

4. **Mission Complete.**

   Lastly, the attacker collects, packages, and eventually exfiltrates the data, or in the worst case destroys your resources.

One product will certainly not provide the protection you need against all stages of an attack. And while some new and innovative solutions will help protect against, or detect, the initial infection, they are not guaranteed to stop 100% of malicious activity. In fact, it's not a matter of if, but a matter of when you will be successfully breached. You still need to do the basics – patching, firewalls, endpoint AV, and threat detection and so on. But you also need to protect, control, and audit the privileges in the environment. Properly managing privileges can help at all stages of the attack. From reducing the attack surface, to protecting against lateral movement, to detecting a breach progress, to actively responding and mitigating the impact of that breach, this book will examine where these privilege vulnerabilities exist; how attackers can leverage them; and more importantly, what you can do about it.

Threat Personas

Before we get into the gory details and privileges, let's spend a few minutes on who we are protecting ourselves from. Sources of an attack can come from outside or inside and organization. They may be opportunistic or well planned and targeted. They may be perpetrated by individuals or groups of individuals. To categorize their motives and tactics we may refer to them as hacktivists, terrorists, industrial spies, nation–states, or simply hackers. There is little difference between a hacker, an attacker, a threat actor, and malicious activity that warrants correction during their usage. Many times, security professionals will use the terms interchangeably and with little distinction between the definitions. As security professionals,

we study recent breaches, the forensic investigations, and arrests that follow. It is rare that that largest of breaches go unsolved but they can take years to prosecute based on extradition laws and whether a nation–state was involved. During the course of these events, we learn about incidents, breaches, and whether it was a threat actor, hacker, or even attacker that caused the malicious activity.

The question is: What is the difference, and don't they all mean the same thing? The truth of the matter is that they do not, and many times they are used incorrectly in reporting a breach or cybersecurity incident. The common definitions for each of our threat personas are the following:

- **Threat Actor** – According to Tech Target, "A threat actor, also called a malicious actor, is an entity that is partially or wholly responsible for an incident that impacts – or has the potential to impact – an organization's security."
- **Hacker** – According to Merriam-Webster, "a person who illegally gains access to and sometimes tampers with information in a computer system."
- **Attacker** – In cybersecurity, an attacker is an individual, organization, or managed malware that attempts to destroy, expose, alter, disable, deny services, steal, or obtain unauthorized access to resources, assets, or data. As crazy as it sounds, there is no universally accepted definition for an attacker and that is why it is represented many times out of context.

Based on these definitions, a breach or incident can be conducted by any of the three. A distinction is needed when talking about privileges as a threat vector since the resolution is different for each one.

A threat actor – compared to a hacker or attacker – does not necessarily have any technical skill sets (see Table 1). They are a person

or organization with malintent and a mission to compromise an organization's security or data. This could be anything from physical destruction to simply copying sensitive information. It is a broad term and is intentionally used because it can apply to external and insider threats, including their missions like hacktivism.

Table 1. *Threat Actor Examples*

Threat Actor	Example
Outsiders	Nation–State
	Political Activist
	Organized Crime
	Terrorist Organization
Insiders	Administrators
	Developers
	Systems Users
	Data Owners
	Contractors
	Trusted Third Parties

Hackers and attackers are technical personas or organizations intentionally targeting technology to create incident and hopefully (for them, not you) a breach. They can be solo individuals, groups, or even nation–states with goals and missions anywhere in the world. Their objectives may to destabilize a business, government, to disseminate information, or for financial gains.

The difference between an attacker and hacker is subtle, however. Hackers traditionally use vulnerabilities and exploits to conduct their activities. The results may be damaging or just curiosity. Attackers can use any means necessary to cause havoc. For example, an attacker may be a disgruntled insider that deletes sensitive files or disrupts the business

by any means to achieve their goals. Remember, as these insiders have access to the target systems and data, they can simply use their granted access to accomplish their goal. A hacker might do the same thing but they use vulnerabilities, misconfigurations, stolen credentials, and exploits to compromise a resource outside of their acceptable roles and privileges in order to gain access and accomplish their mission.

The difference between the three is so important. Security solutions are designed to protect against all three types of malicious users and the results will vary per organization:

- In order to defend against a **Threat Actor**, Privileged Access Management (PAM) solutions can manage privileged access, log all activity in the form of session recordings or keystroke logging, and monitor applications to ensure that a threat actor does not gain inappropriate access, and document all sessions just in case they do (insider threats).
- In order to defend against a **Hacker**, Vulnerability Management (VM) solutions are designed to identify vulnerabilities such as missing patches, weak passwords, or insecure configuration across operating systems, applications, and infrastructure to ensure that they can be remediated in a timely manner. This closes the gaps that a hacker can use to compromise your environment, including patch management to streamline the workflow for timely remediation. Most vulnerability solutions help organizations measure the risk associated with these vulnerabilities such that they can prioritize remediation activities to reduce the attack surface as quickly and efficiently as possible.

- In order to defend against an **Attacker**, least privilege solutions and network and host instrusion prevention solutions can be used to reduce the attack surface by removing the level of access threat actors have to resources. This includes removal of unnecessary administrator (or root) rights on applications and operating systems. These solutions can also perform detailed access and behavior auditing to detect compromised accounts and privilege misuse.

A combination of these solutions not only prevents outsider attacks, but limits privileges to assets and users, thereby inhibiting lateral movement. This is the basis for protecting against the privileged attack vector and will be discussed in detail in later chapters.

However, let's not get ahead of ourselves. Let's start with a review of the basic elements of privilege before formulating our defensive and reviewing security best practices.

Regardless of their motives from financial, hacktivism, to nation–state, they will always take the path of least resistance to commit their malicious activity. While this path my sometimes leave obvious trials for forensics, the art of the hack is to be subversive without detection (if possible), and perpetuate the activity under the radar of the implemented security defenses. Attackers, like most people, will choose the path of least resistance. Fortunately, the methods for gaining user and application privileges are well known due to various password attacks and exploits. This book will explore these capabilities and potential defenses so that privileges do not become a successful attack vector for a threat actor within your organization. This discipline is commonly referred to as Privileged Access Management (PAM).

CHAPTER 1

Privileges

Today, privileges based on credentials are one of the lowest-hanging fruits in the attack chain. Threats include the following:

1. Insiders having excessive and unmonitored access to accounts, opening the potential for misuse and abuse.
2. Insiders that have had their accounts compromised through successful phishing, social engineering, or other tactics.
3. Accounts that have been compromised as the result of poor credentials, passwords, devices, and application models allowing attackers to compromise systems and obtain privileges for malicious activity.

Note The 2017 Verizon Data Breech report highlighted that 81% of external related breaches leveraged stolen or weak passwords.

To understand how privileges can be used as a successful attack vector, a clear definition of privileges needs to be established. In a basic definition, a privilege is a special right or an advantage. It is an elevation above the normal and not a setting or permission given to the masses.

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_1

An example is the relationship to education. "Education is a right, not a privilege."[1] Everyone has the right to education and thus a Standard User has the same rights as everyone else. Information technology users have rights that are global to all authenticated users. As these user accounts are created and provisioned, they are granted these standard rights. This could be basic access to a keyboard and mouse, Internet browser, or even office applications such as email. A privileged user has rights above that. That may include the ability to install other software or just change office features and settings, or perform other routine maintenance tasks such as managing backups. This does not mean they are an administrator. It means they have been granted privileges, at a granular level, above the baseline of Standard User. This granularity can have as many levels and features as an organization deems fit. The most basic interpretation is two levels:

1. Standard User – shared rights granted to all users for trusted tasks.
2. Administrator – a broad set of privileged rights granted for managing all aspects of a system and its resources. This includes installing software, managing configuration settings, applying patches, managing users, etc.

However, some organizations will define privileges across four fundamental levels:

1. No access – that is you do not have a user account or your account has been disabled or deleted. This is the denial of any form of privileged access, even anonymously.

[1] http://www.globalpartnership.org/blog/education-right-not-privilege

2. Guest – restricted access and rights below a standard user. Many times this is associated with anonymous access.
3. Standard User – shared rights granted to all users for trusted tasks.
4. Administrator –authorization to effect on the assets runtime, configuration, settings, managed users, and installed software and patches. This can also be further classified into local administrator rights and domain administrative rights affecting more than one resource.

While this perspective of privileges is at a macro user level, it is very important to understand the micro level of permissions down to the token and file to formulate a proper defense. It is myopic to consider privileges are only a part of the application you are executing. Privileges must be built into the operating system, file system, application, database, hyper-visor, cloud management platform, and even network via segmentation to be effective for a user and application-to-application communications. This is true if the authentication is granted by any mechanism from a username and password or a certificate key or pair. The resource interpretation of the privileges cannot be just at any one layer to be truly effective. So let's have a deeper look.

Guest Users

As a Guest User your privileges are strictly limited to specific functions and tasks you can perform. In many organizations guests are restricted to isolated network segments with basic access – perhaps access to the Internet for visiting vendors. If these unmanaged computers are, or become compromised, the risk is mitigated with limited access to organizations' resources. For example, a network scan from a

compromised guest machine will not (or at least should not) provide the attacker direct access to corporate systems and data.

Standard Users

As a Standard User, you have basic privileges above a Guest to perform additional tasks and to fulfill the missions that a specific job function requires. While organizations may forego even Guest Users, it is typical to have granular levels between a Standard User and a Full Administrator. Typical organizations may have 100s or 1000s of different standard user roles designed to balance access and efficiency with risk. Each role has been granted specific access to systems, applications, and data required for their specific job. In many cases a user may be a member of multiple roles depending on their specific job requirements. For example, low-access roles (also called basic roles, basic entitlements, birth rights) are typically provided to each organizational user (employee, contractor) to provide basic access. Perhaps this provides access to an email account and general Intranet for information seeking. Next would be specific roles that would add additional access based on the job itself. See Figure 1-1 for a very basic example of a role hierarchy in a manufacturing environment.

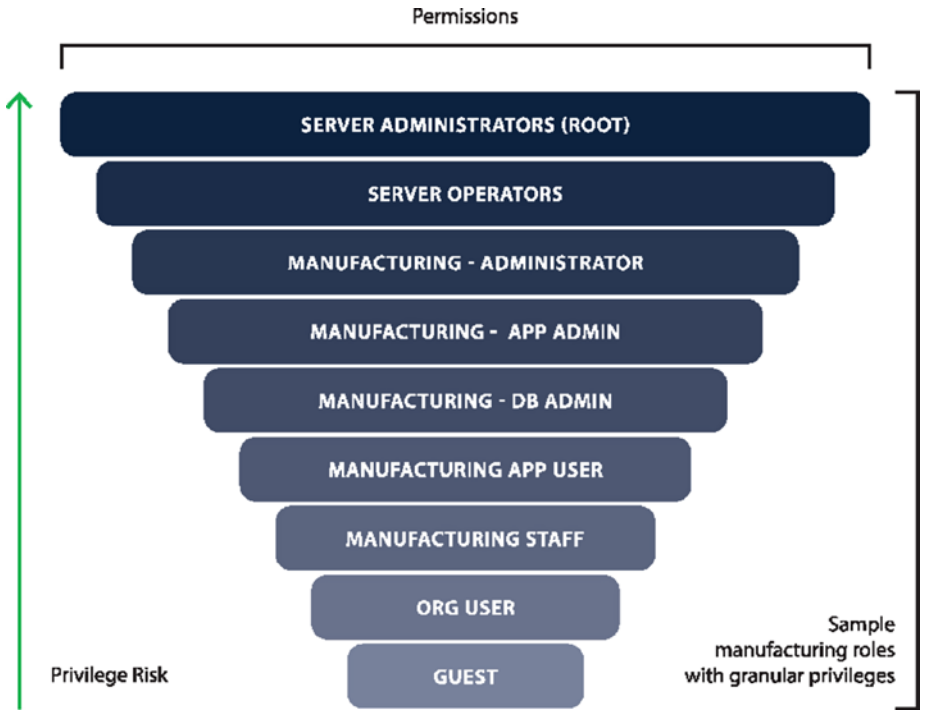

Figure 1-1. *Example of a Role Hierachy in a manufacturing environment*

In this example, the banding and nesting of granular permissions within business roles may allow certain users access to a web server but not access to a database or vice versa. From the perspective of a threat actor, compromising accounts with elevated rights is typically the target as these credentials are the ones that have access to sought-after systems and data.

Malicious activity does not require full domain administrative or root rights (even though that reduces technical barriers and makes it easier for them to conduct nefarious activity). For example, if the user is a manufacturing floor worker, their potential privileges are limited by their job role (barring a vulnerability and successful exploit). If the target user is an information technology administrator such as a server administrator, desktop administrator, database administrator, application administrator,

or executive, the associated privilege risk will be higher as these employees have been granted additional access as defined by their role. This makes them desirable targets for a threat actor. Take, for example, an attacker who wants to gain access to a corporate database or file system with sensitive data (see Figure 1-2).

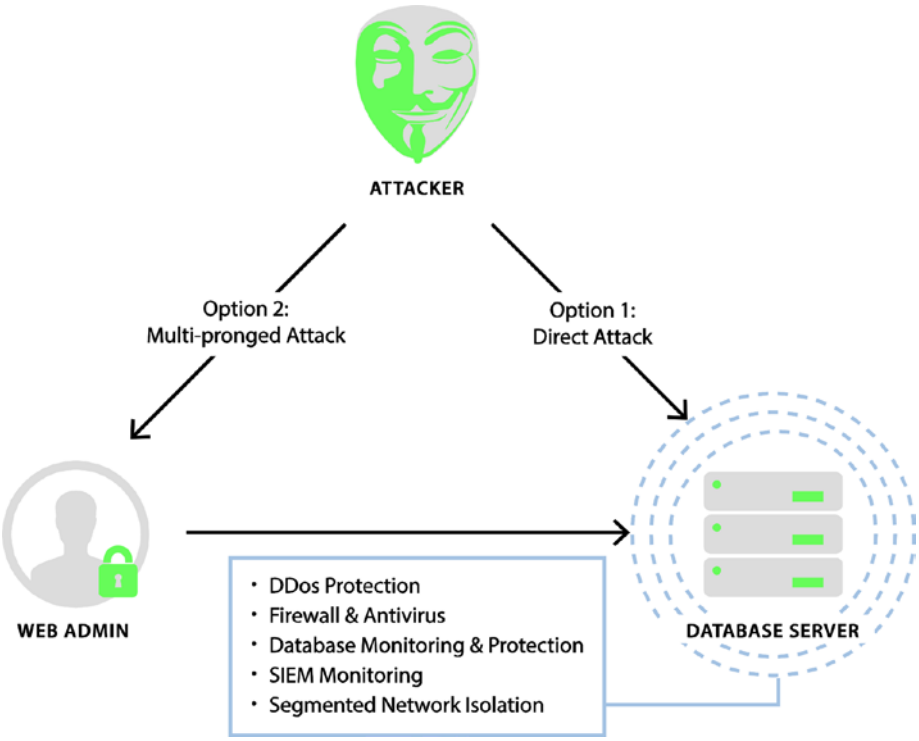

Figure 1-2. *Example of an attacker who wants to gain access to a corporate database or file system with sensitive data*

Do they

1. Directly attack the hardened database or system housing the sensitive data. A system that is likely patched, monitored, and incorporates advanced threat detection and attack shielding technologies.

2. Use a phishing attack to compromise the system\ database administrator and use those credentials to log directly into the target system.

Having privileged access in an applications communication, database, or file system is all that is needed to extract information once an internal beach head has been established, to execute commands, perform lateral movement, and exfiltrate the data.

Additionally, many organizations grant more privileges than are required for a specific job, which leads to increased risk by both hackers and insiders. For example, many organizations still allow users to have administrative control over their desktops.

It is also important to note that recent attacks are beginning to focus on nontraditional assets that may lack the flexibility and control required in today's sophisticated threat environment. With some systems, the access options are very Boolean. You have access or you do not. When you do, you are an administrator and have complete control. This is primarily true for consumer devices that do not have any concept of role-based access but also true for the Internet of Things (IoT), many legacy systems, and even the networking devices that protect the data flow of sensitive information flowing within and "out of" your network.

Administrators

As an Administrator or Root User, you own the system. All functions, tasks, and capabilities are potentially at your control and even if technology is deployed to block an administrator, being an administrator means there is always a way, or back door, around the restrictions. This leads to the premise that once you are an administrator, the security game is over. An administrator can circumvent any protection designed to protect against an administrator, even if the results are destructive to the processes themselves. Obtaining administrator or root access is a privilege and is

the crown jewel to an attacker. Once an attacker has root access and can operate undetected, then any system, application, or data is potentially within their reach. This is despite all the modern malware and attack vectors we defend against. Gaining privileges is the ultimate attack vector for breaching an organization, government, or even end-user–based computing device. Again, in this case, organizations tend to grant too many unmanaged administrator privileges, which leads to significant risk posed by threat actors and insiders.

Identity Management

The process of defining, managing, and assigning these roles to ensure that the "right" people have the "right" access at the "right" time is also known as "Identity and Access Management" or IAM. Privilege Access Management typically complements traditional IAM processes and solutions with additional layers of control and auditing for "privileged" accounts. These are the accounts that pose the greatest risk to the organization.

As you can clearly see in Figure 1-3, a lack of visibility and control over privileged accounts, users, and assets could leave you exposed to a damaging data breach. That visibility often begins with a simple discovery exercise. Ergo, let's first take a look at where these privilege accounts exist. Then, once we get a complete picture of the scope of the challenge, we can discuss some strategies to address it.

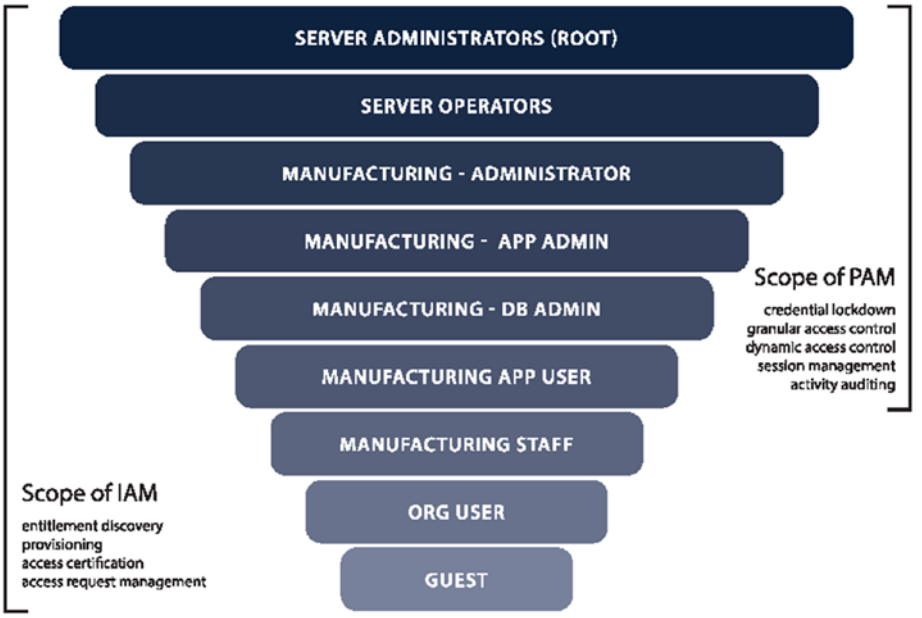

Figure 1-3. *Lack of visibility and control could lead to a data breach*

While this perspective of privileges is at a macro user level (identity management), it is very important to understand the micro level of permissions down to the token and file to formulate a proper defense. It is again a mistake to consider privileges are only a part of the application you are executing. Privileges must be built into the operating system, file system, application, and even network via segmentation to be effective for a user and application-to-application communications. The resource interpretation of the privileges cannot be just at any one layer to be truly effective. Thus, Identity Management only provides access to the resource by scope or role, while Privileged Access Management (PAM) provides the granular permissions needed when the operating system or application is

incapable of providing these privileges itself. It is fair to state that PAM is a subset of Identity Access Management (IAM) and an extension to protect privileges at every level.

Identities

For the sake of definitions, and commonly misused within the industry, an identity is simply a carbon-based life form. It is any human being, or user, that interacts with resources from applications to operating systems. This includes physical and electronic access and is a convenient way of saying I am a person. "I think, therefore I am," and I have an identity. It is important to note that any user should only have one identity.

Unfortunately, this security best practice gets blurred when people assume different names, including having maiden names, and may have duplicate identities referenced electronically in an organization. They still have only one identity but may have electronic instantiations to multiple identities, which should not be confused with having multiple accounts. Organizations should only have one identity for a person, like their social security number (which is a bad practice due to personally identifiable information), or preferably an employee number. One person, one identity, and one electronic reference linking them.

Accounts

An account is an electronic representation of an identity or reference for a set of permissions and privileges needed for an application or resource to connect or operate within the confines of system. 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 one-to-many relationship with identities, be defined locally, grouped together, or managed via directory services.

Accounts can have role-based access applied at the account level, group level, 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.

Therefore, accounts linked to identities are how we gain access to information technology resources. For technology itself, accounts are a vehicle to authorize their usage and operational parameters. Too much privilege given to any type of account can introduce risk, and accounts can literally be named and referenced to almost anything, also dependent of system limitations. It literally is a reference to provide authentication, and an account may or may not have a password or key. When a password is assigned, regardless of its strength, type, or security, it becomes a credential.

Credentials

A credential is an account with an associated password, passcode, certificate, or other type of key. Credentials can have more than one security mechanism assigned for dual or multi-factor authentication, or be basic Guest credentials for anyone to access. 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.

When someone indicates that they have "hacked" an account, what they mean is that they have hacked the credentials associated with the account. Literally hacking an account would only yield a username. Both are needed to successfully compromise a system and potentially its data. Thus, for simplicity in the remainder of this book, hacking an account means the same thing as hacking credentials. It is difficult enough

managing privileges in an environment rather than worrying about the semantics used every day in describing the threat. Security professionals and the press will probably never change in saying one million accounts were compromised when in fact one million credentials where compromised. See the difference?

Default Credentials

Whenever you purchase or license a new resource, whether it is a device, application, or even a cloud resource, it comes with a default credential scheme used for initial access and configuration. The resource is typically in a pristine state, not fully hardened, and vulnerable to a variety of password attacks, especially to the default root or administrator account that could own the entire system. If this account is compromised, a wide variety of persistent privileged attacks could occur by a threat actor and go undetected for years since the defaults governing the solution have not been managed and, more importantly, maintained or monitored. These default credentials are required so that an organization can perform the initial configuration in a consistent manner. Logically, using security best practices, the default credentials should be changed, but many times they are not. This exposes these default accounts as a privileged attack vector. Today, manufacturers have five choices for passwords when they ship a device, application, or other resource:

1. Anonymous Access – full unrestricted default access with no credentials
2. Blank Password – default username but no password
3. Default Password – default credentials with predictable username and password

4. Default Randomized Password – default username with fully randomized password
5. Default Generated Password – default username with predictable password

These are covered in detail below, and if they are not properly configured, it literally is a matter of time before they will be owned. These are the basics for privileged management: changing the predictable or easily obtained to something that requires knowledge to access.

Anonymous Access

Anonymous Access is simple and absolute. No authentication is needed to begin the setup of the resource including advanced settings that may be used to secure the asset from future threat actors. While this method seems completely ludacris in today's security climate, it is often the only way to configure a resource for the first time. Consider the purchase of a new cell phone or mainstream tablet with either iOS or Android. Its initial configuration, allows for anonymous access to set up WiFi. This is typically not required to complete the configuration but if misconfigured, initially or not, could lead to a man-in-the-middle attack. In addition, the primary administrator user account on the device can be set with a null password basically allowing full unrestricted access to the device at any time. These devices do not enforce a password by default even though it is recommended.

What makes Anonymous Access a horrible security threat is when it is not disabled or changed after the initial configuration. Surprisingly, there are plenty of information technology resources that only support anonymous access. These include but are not limited to SCADA sensors like thermocouples, children's IoT (Internet of Things) toys, and digital home assistants (after their configuration) that rely on voice commands. In the end, these are devices that have no programmatic concept of accounts,

role-based access, and every user that interacts with the device has the same level of privileges (see Figure 1-4).

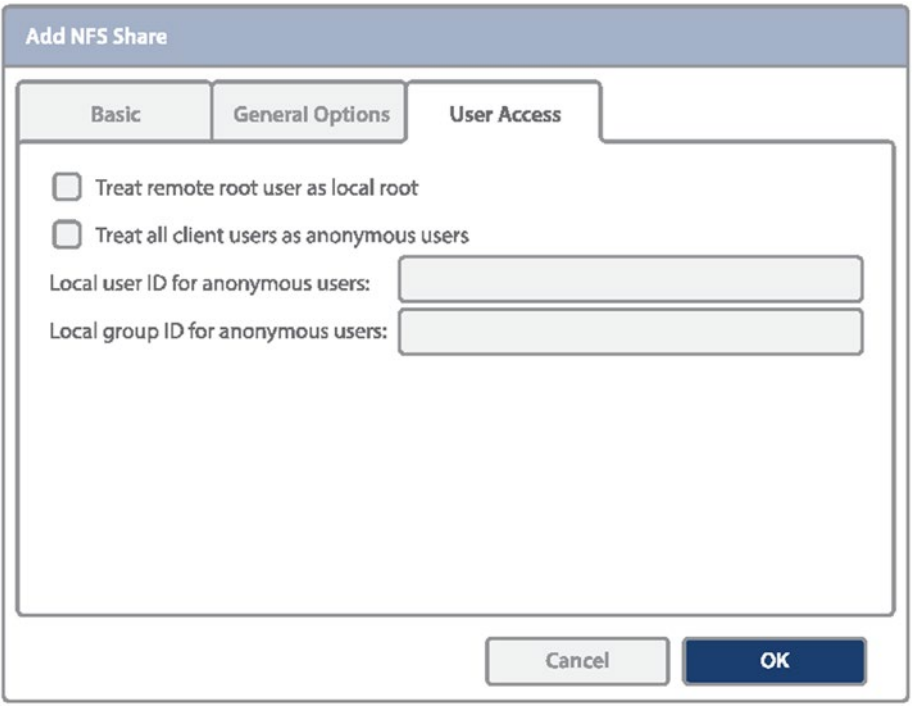

Figure 1-4. *Anonymous Option for Adding an NFS Share*

Blank Password

Blank passwords are commonly used in resources that have multiple accounts but have a null password by default. The security and initial configuration of the resource may require that a password be assigned; however, many technologies, including older databases, do not even prompt for a password assignment after the solution is installed and operating. The risks are obvious. Accounts are present, not properly configured, and depending on the privileges are easy targets for a threat actor (see Figure 1-5).

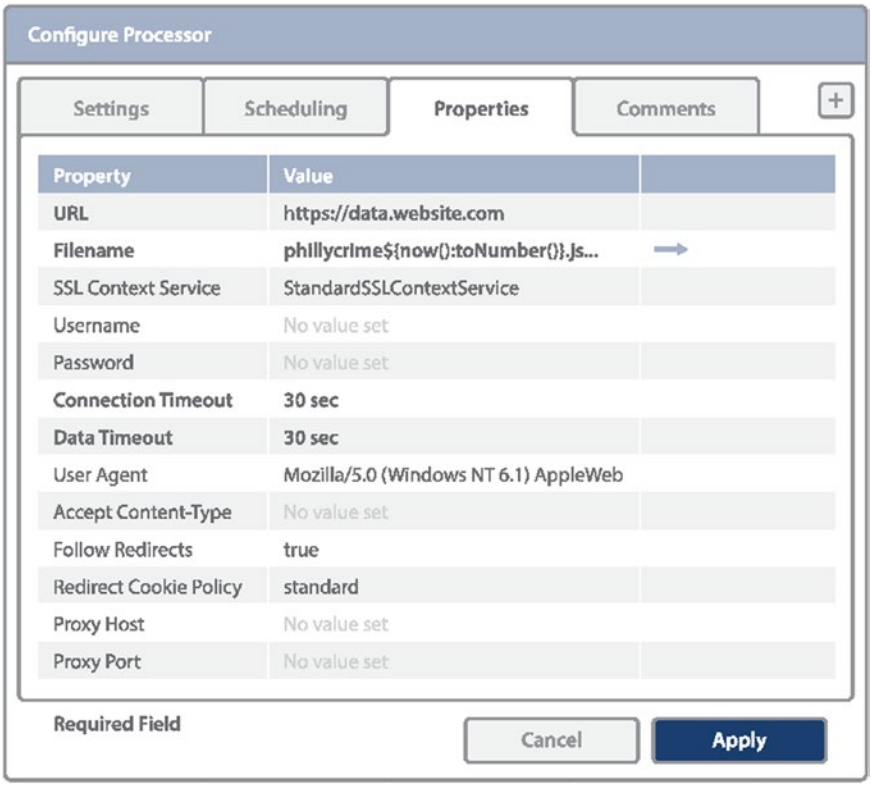

Figure 1-5. *No Account or Password Settings*

Some people may confuse accounts with blank passwords with anonymous access. However, there are two significant differences that should be well understood. With anonymous access the identity of the user is not considered and such access is typically reserved for low-risk activities. With an account with a blank password, the identity of the user is considered, but the security of the authentication process is diminished, usually an oversight that creates undue risk. The most common, and widely used, blank password solutions are systems that support Guest accounts. Anonymous access is independent of whether the guest account is enabled and is typically reserved for all to access. My point is that unauthenticated access is typically purposeful and required for operations, while a blank password is usually an indicator of a privilege vulnerability.

Default Password

For many years manufacturers released solutions with default passwords. Every model series of the device had a unique password and for some manufacturers, the default password is the same for every new resource they produce. While this is a common security practice, it is a glaring security issue. There are volumes of lists on the Internet of these default passwords for every vendor, and all a threat actor needs to do is try them to see if they still are using defaults in order to gain access. In addition, regulatory mandates, discussed later, prohibit the existence of default passwords (of any type) to be used in production due to the risk. Thus, these devices are susceptible to an attack as soon as they are connected to a network or Internet, they may not be properly configured and still have the default credential after the resource is placed in production, and worst off, they may not allow for the default password to be changed. The latter represents a privileged attack vector that is extremely vulnerable just like anonymous or blank password access to the root account.

It is important to note that blank passwords (as defaults) are not just a threat for endpoints and networking devices. In many cases, application vendors will place the onus of implementing security controls on the application users and developers. For example, MongoDB is a popular NoSQL database used by organizations to perform big data and heavy analytics workloads. The default installation of MongoDB on older releases does not actually require authentication to access the database. This resulted in a widespread attack in early 2017 in which application and database administrators were not enabling authentication to the database. To make matters worse, many of these databases were directly accessible to the Internet. For these reasons, the importance of communicating security best practices at all levels of the organizations, including secured coding by the development and application teams, are critical. Figures 1-6 and 1-7 illustrate real-world technologies that are commercially available that have poor default password implementations.

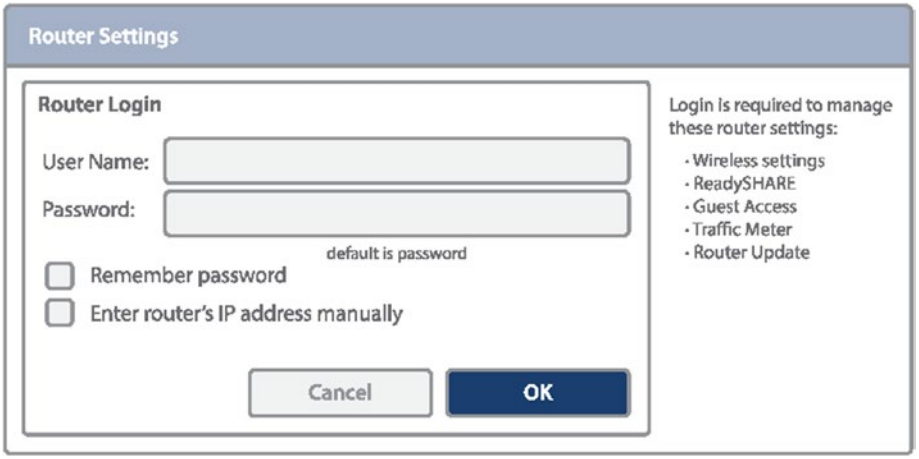

Figure 1-6. *Home-based router with actual text in the user interface stating the default password*

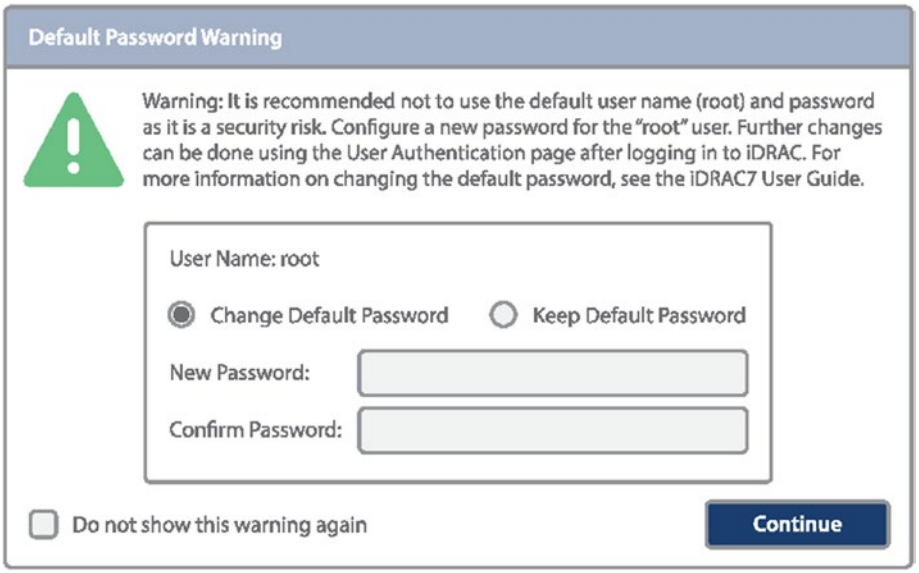

Figure 1-7. *Commercial-based Server solution with an option to keep default credentials*

Default Randomized Password

In modern times, the most secure default password is one that is unique and randomized for every single resource that is produced, licensed, or sold. This password needs to be securely conveyed to the administrator or organization for the initial setup and should be changed upon the initial configuration. Unfortunately, some manufacturers have taken this concept to a level that makes the devices unsecure if physical access to the device is available. Along with the serial number, these vendors have printed the default passwords on the device for anyone to retrieve (see Figure 1-8). A simple press and hold of the reset button restores the password to the default and depending on the device, the configuration too. Once reset, a threat actor now has access to comprise the asset. Mitigation for this type of threat is relatively simple. Copy (photo, scan, or type) the default password documented on the resource, securely store it, and then destroy, mask, or remove the label. In addition, physical access to any device that allows for a reset, or password reset locally, should be secured to prevent this threat. Most compliance regulations mandate this as well. Randomized passwords are currently one of the most secure methods to distribute default passwords but also may have visible risks depending on how that password is initially distributed.

Figure 1-8. *Factory Serial Number*

Default Generated Passwords

Many organizations have processes for onboarding new users and providing access that they require to perform their jobs. When not managed properly, these accounts can create a significant security risk. Have you ever worked for a company where an automated system creates the default login account and password based on something that everyone knows – like your name? Many times, this is how an IT\Help Desk sets up the default access for new employees as it's easy to document, automate, and communicate to new users how to again access.

For Example: if I have a new user named "John Titor" I may have an algorithm that generated the login account and credentials by extracting components of his name. Here my provisioning process is to create the login account using the "first initial of his first name + last name" with a default password of "New" + "first initial first name" + "first initial last name" + "!!!2036$". The result of this calculation results in the following account.

Login Account: JTitor

Password: NewJT!!!2036$

Once created I could then communicate to John Titor how to log into his account via a phone call, email, text, or other means. While this seems secure, to effectively compromise this account all I need to know is the new user's name, and the algorithm to define the default password. And if I am an insider who went through this process, I would have a pretty good idea of what it is. Now you may indicate that this is not really a risk, as these accounts would typically be set to require a password change upon first login, and that is true. However, there are two things to consider:

1. This account would certainly be exposed from the time it is created and the hacker changed the password upon login, to the time the new employee realized that they could not access their pre-created account and has their password reset by the IT team.
2. In some cases, the organization may not enforce resetting of these default passwords, and worse yet, the employee may continue to use it! Believe me – it happens.

Of course, to overcome these issues, more secure best practices can be implemented to reduce these risks, including the enforcement of "change password on next login" and multi-factor authentication. Figure 1-9 shows how to enforce a password reset during the next logon.

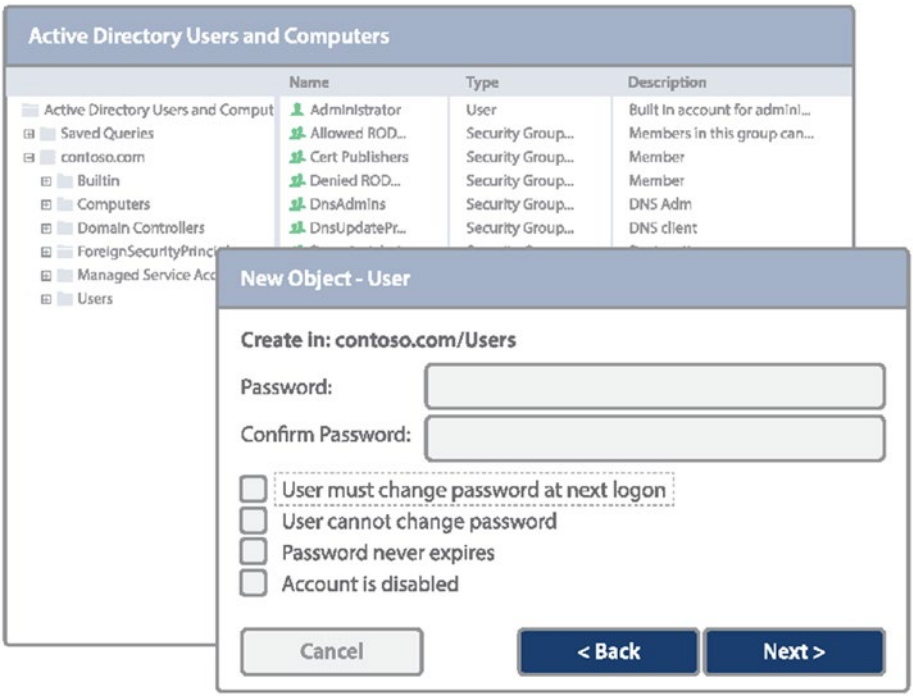

Figure 1-9. *Force a password reset during the next logon*

Third-Party Vendors

Contractors, HVAC companies, building maintenance, managed service providers for routers, and firewalls — the list of third parties that may have access to your network at any given time is endless. Many of these vendors/workers connect to these systems remotely to go about their daily business in supporting your organization. The problem is that many of the systems they interact with are also connected to your corporate network. It has been shown by numerous high-profile breaches that vendor networks can be leveraged to gain access into customer environments.

Hackers can steal credentials to gain access to vendor-controlled systems, and then exploit vulnerabilities and/or poorly managed privileges to move throughout the organization, sometimes machine by

machine. You are only as secure as your weakest link – the security of your environment may rest on the security practices, and controls of a third party.

The big issue with adhering to policy and maintaining security across two companies is that often the credentials used by the remote vendor are not under the direct control of the customer. Two different networks with two different user directories, and perhaps two different security policies make the job of security compliance a challenge. Even if you had a way to ensure security best practices were being followed, you still have no visibility into what activity is being performed on equipment that is connected to your network.

Let's break down the problems:

1. Vendor Credentials – We need a method for making sure that (a) passwords are regularly rotated, and (b) they have not been compromised. Certainly, a privileged password management system would assist here.
2. Network Access – There needs to be controlled inbound access: a VPN, gateway proxy, or preferably both. If we can limit access according to incoming network address, that's gravy.
3. Monitoring – What are users doing when they are connected. There needs to be a tool to be alerted when sessions start, and then 'look over their shoulder'.
4. Control – So what happens if you see something happening that shouldn't? A mechanism to sever the connection is crucial.

To alleviate these challenges, Privilege Account Management solutions that include session management and recording can provide a secure connection gateway, with the ability to proxy access to RDP, SSH, and Windows applications. Passwords can be regularly changed using strong and complex policies to ensure that any credential breach, whether directly by the user or indirectly via malware, has a limited window of exploitation. In addition to these security benefits, all vendor activity accessing the corporate environment through the PAM solution may be recorded for later playback to support both forensic and compliance activities.

CHAPTER 2

Shared User Credentials

One of the most cardinal rules in cybersecurity is never to share your password with anyone and at any time. Realistically, that is not possible. Whether it is a colleague, or contractor, there are no sound use cases when it should be done; ever! That said, many employees continue to share passwords in times of emergency, to delegate tasks, or to overcome issues in planning with sick leave and vacation - being two common culprits.

The problem with shared credentials is that once they are out of your control, how fast and far could they propagate before they are in the hands of a threat actor? This could be anything from a real hacker with malicious intent, to a suspicious spouse or significant other. If multiple users are using the same credential, for example a local administrator account or shared SSH Key, how can an organization reliably associate access and change events to an individual? Unfortunately, even though these risks and challenges exist, there are real-world use cases where shared credentials are absolutely required for an application to work in a multitier architecture, for devices to connect to a network, and multiple users to administer the same resources. Shared credentials, or the act of sharing credentials, is a real privilege problem because once the information is shared, limiting its exposure and measuring the risk of that exposure becomes a difficult threat to quantify. Minimizing privilege risk, or privilege as an attack vector, requires knowledge of all the different

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_2

places shared credentials can exist. And, what can be done to mitigate inadvertent propagation of them being shared. It includes documenting anytime that shared credentials are used, which individual requested them, what actions they performed, and changing the password on a periodic basis to ensure it does not become stale. Shared credentials should also be rotated when organizational events occur such as employee changes and contractor access.

Account Credentials

Users expose their account information in a variety of ways: some intentionally and some inadvertently. The most common methods include verbally, through email, and through text messages. Outside of a listening and recording microphone, the latter leaves a permanent documented paper trail in backups, log files, and text message history, some of which are completely outside of your organization's control. People forget that just because you delete an email or text from your device does not mean the message is truly eradicated. It is just removed from your local view. If a password was sent via one of these methods, it still exists out there, somewhere. Where it exists, and the subsequent exposure to risk, is dependent on how the password was stored. Password storage and retrieval can take many forms including the following:

1. Mentally: Only memorized in the human brain.
2. Documentation: Whether on paper or in an electronic file. These can be secured in a physical safe or have file encryption to protect the contents from a threat actor.
3. Password Manager: A technology solution for the storage and retrieval of credentials and their associated passwords. Advanced versions of this technology can also randomize the passwords and perform session monitoring.

While storing the information solely within your head is presumably the most secure, it has risks that degrade this as a best practice. This is where the expression, "If you got hit by a bus" becomes painfully relevant. Documenting and creating specific accounts for emergency privileged access is a good method for Break Glass and for use case-based sharing but represents risks if the files are shared, copied, or placed in an unsecure location. In this case a threat actor could have unhindered access to your password and to resources you have access to as well. To reduce this risk, many end users utilize Password Managers for storage and retrieval of passwords. This represents one of the best solutions for managing privileged access to mitigate this attack vector.

Shared Administrator Credentials

Many servers, workstations, networking devices, and applications ship with, and rely on, local accounts which provide access to administrative staff in order to perform management activities. In many environments, multiple system administrators will use these accounts to perform specific tasks. The sharing of these accounts and their related passwords, versus creating a unique login for each administrator, may be due to the limitation of the device and\or application as previously discussed. However, in many cases the accounts, and even the account passwords, may be shared across administrators due to the management overhead, complexity, and cost of implementing unique credentials across the environment.

Take, for example, an environment that has 10 administrators managing 1000 systems, as shown in Figure 2-1:

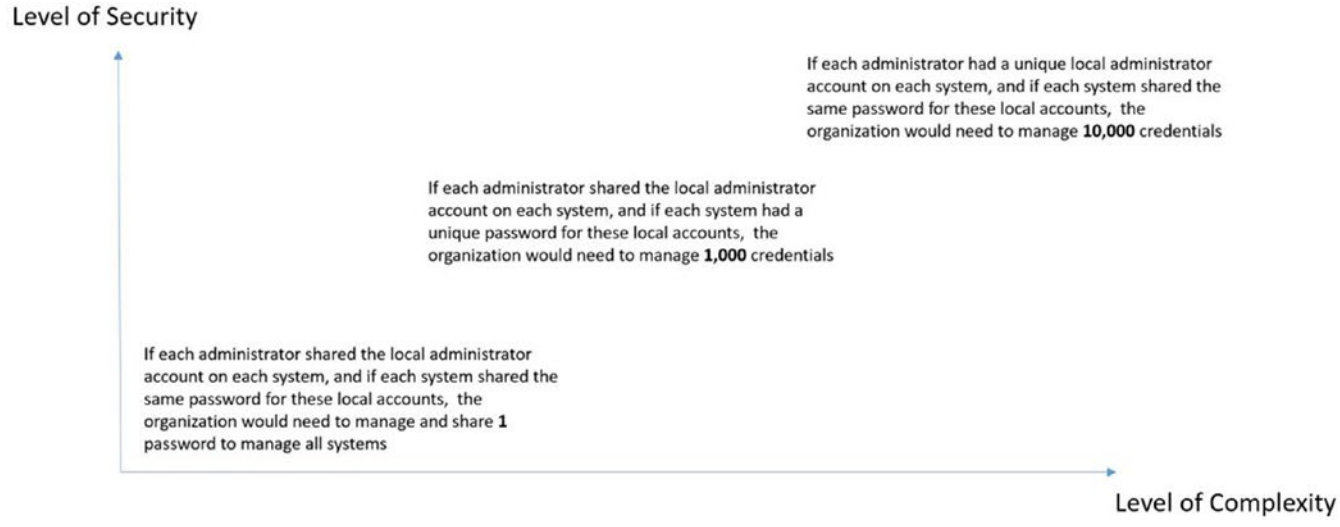

Figure 2-1. *Administrator Environment*

As such, for efficiency sake, many organizations will choose the less secure, less complex, but manageable alternative. Let's examine risks associated with each option in our model:

1. Using the same account on each system with each system account using the same password is the easiest solution from an operational perspective as administrators only need to share and coordinate a single password. However, this option is clearly the most insecure approach. If an administrator's password is compromised, the hacker can easily gain access to all 1000 systems via lateral movement.

2. If the managed systems each had a unique password for the shared account, it reduces the risk and impact of a potential breach. In this case if an administrator's password was compromised, it would only grant the hacker access to that one system. All other systems would have their own unique password. The only challenge with this approach, as with all shared accounts, is that you cannot isolate specific account activity to an individual. In this example, all activities across all administrators would be tracked as "administrator" and not tied to the specific person who performed the action. Also, note that when using a shared account, the password can only be changed if such updates are efficiently coordinated and communicated with everyone that uses the account. The more accounts and passwords, the more complex this coordination exercise can become. In this example, we need to update 1000 passwords across 1000 systems and appropriately notify the 10

administrators when these passwords update occur. The result is that many times, in addition to sharing these passwords, they are infrequently updated, which further increases the risk of compromise. Of course, for such an activity to be efficient and effective, an automated Password Management solution to frequently update these 1000 local accounts with unique and complex passwords can be used.

3. The third option is the most complex option. In this option, users do not use a shared local account. Instead, each user is granted access through their own account. This enables all activities to be logged and tied to a specific user for accountability. However, in our example that would either require that 10 accounts (one for each admin) be created on each local machine, or that each local machine rely on a directory service or centralized identity solutions to perform the authentication process. We will discuss identity solutions and directory services later in this book.

Temporary Accounts

Temporary accounts are commonly associated with interns, contractors, temporary employees, or other resources that will require transient access. These accounts may be shared by users in the same job function - perhaps temporary workers that leverage a shared kiosk, contractors working on plant machinery, professional services contractors, auditors, or other

temporary workers that need to have an account readily available when they start. The risks for temporary accounts include the following:

- Lack of accountability over who performed which task with the accounts.
- Workers may end up having access for longer than they should have.
- Uncontrolled access in environments where these passwords are not frequently changed.
- Accounts not managed or disabled, allowing for unsanctioned access after the temporary period has ended. These are also called gaps in the deprovisioning process.

Personal and Work Passwords

We all have dozens of passwords to remember and forgetting them seems to be commonplace. To reduce the risks and frustrations of forgotten passwords, many users have turned to Password Management solutions, which inventory and secure all their passwords, requiring that they only remember the master password to gain entry. This is a good strategy. What is not a good strategy is to reuse the same password for multiple applications, services, and other resources. Recent breaches in which millions of consumer passwords were disclosed to hackers are bad enough because many were reused in other attacks. Those passwords could also unlock access to your other email accounts, banking applications, Facebook, Twitter and more. It should go without saying, but don't share and use common passwords across both personal and corporate accounts, as a compromise of personal accounts could put your employer and business partners at risk.

Applications

A second cardinal rule of cybersecurity is that users should have a unique password for every application and no two distinct applications should share the same credentials unless required to communicate. This is commonly referred to as password reuse and is one of the largest privilege problems in information technology security today. People use the same password among multiple applications, systems, resources, infrastructure, etc., and if any one of them is compromised, the same reused password can be leveraged against any other device. When this is coupled with noncompliance of security best practices to change your passwords frequently, the risk of shared, reused, and old passwords escalates exponentially as a threat vector. Once one is compromised, all the other resources are now exposed.

Unfortunately, and contrary to this, there are valid use cases where shared passwords between applications are required and represent a unique threat vector. Each application must have the same credentials, and if they are out of sync, the resources fail to function as a desired solution. If one of the resources is compromised, the same problem as password reuse can occur via lateral movement allowing authentication with the same shared credentials. The most common places these shared passwords are used are service accounts, scripts, and application-to-application authentication. There is no simple method to mitigate this problem, but there are methods to ensure the risk is appropriately managed.

- Do not hard-code passwords in scripts, applications, or driver connections.
- Map all services, applications, and accounts that use shared credentials.
- Never place passwords in clear text files or files that can be easily decrypted.

To mitigate this threat, password management solutions provide a vehicle to remediate these risks via an Application Programming Interface (API). In lieu of hard-coding the password, an API call is made to a password safe, or password manager, as a part of a Privileged Access Management (PAM) solution to retrieve the correct password. The PAM solution understands the linkage and mapping of solutions that need the same passwords and either distributes them correctly upon an API call or automatically changes them based on the same relationships. This entire process is secured from a threat actor using its own authentication mechanisms covered later in this book.

A password storage solution (password safe, lockbox, or vault) is the only best practice recommendation for application-to-application password storage versus coding passwords in the solution. Figure 2-2 illustrates an application that uses credentials to secure communications for future application interaction. If they are coded, stored in a separate file, or required to be keyed in during solution runtime, the risk of password theft by a threat actor increases based on the password being documented in a file or manually entered and captured by a keystroke logger.

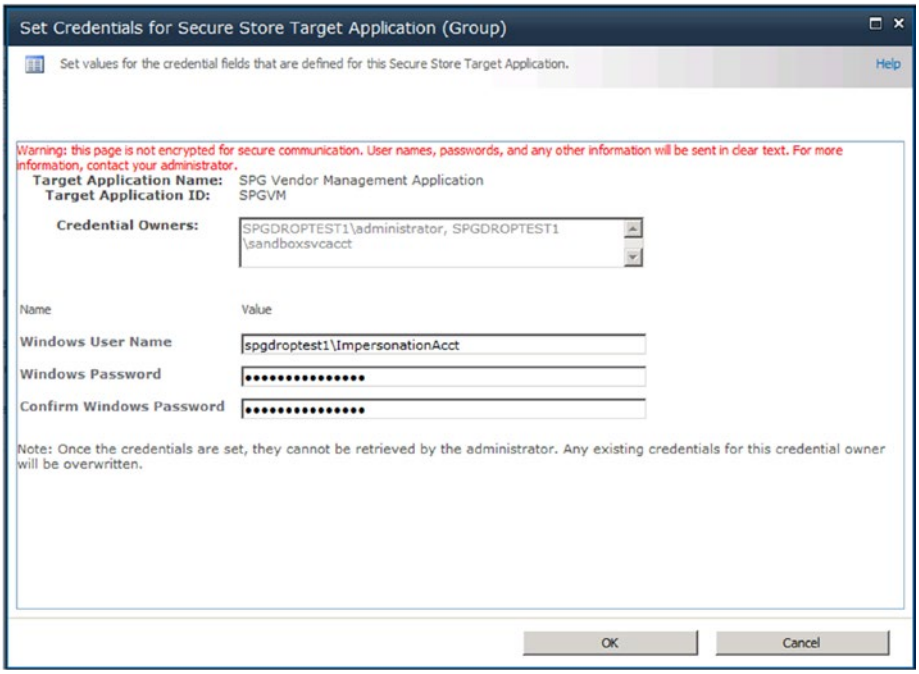

Figure 2-2. *Static Credentials for secure storage authentication between two applications*

Devices

Devices that share passwords are very similar to applications that share credentials but the credentials and password are stored on the device (oftentimes not securely) for continuous usage. These are not the passwords you use for mail or social media accounts, but rather passwords that every device may have in order to connect. These include but are certainly not limited to the following:

- If WEP (hopefully not) or WPA2 is used for WiFi, the shared key or passphrase may be the same for all devices to connect.
- If no PAM solutions are being used, the help desk or local administrative back door on all mobile devices may be the same.

- Tools like appliance-based vulnerability assessment scanners, network management solutions, and security solutions may share the same credentials and passwords across all deployments in order to connect.
- Management of infrastructure devices such as routers and switches using the same root password for either management or network management solutions.

Device passwords represent another vector for privileged attacks. The passwords, or certificates, are rarely changed, and once obtained by a threat actor represent an easy and persistent method to penetrate an environment until they are detected, the services stopped, and all the device passwords changed. For insecure wireless networks using WPA2 or WEP, the likelihood of a passphrase leak increases over time. The more devices out there using it, the more people knowing it, the more likely someone with a rogue device can connect. The best recommendation for shared device passwords is to try and keep them all unique and use advanced authentication technologies to avoid device password or passphrase sharing. A spreadsheet with laptop serial numbers and help desk back-door passwords encrypted on a private share is much more secure than every laptop having the same password, especially if they are never changed. Keeping personally identifiable information out of the spreadsheet is also helpful, since the list would need to be cross-referenced to an owner to be eventually usable by a threat actor. Having all that information in a password manager is the safest approach of all and best practice in lieu of any flat file technique. Table 2-1 illustrates this approach but keep in mind, it is not recommended since all the passwords are exposed. In addition, outside of file security, the data would have to be cross-referenced in order to be usable by any threat actor.

Table 2-1. *Sample Spreadsheet*

Device Serial Number	Help Desk Password	Asset Tag
XDM7GT	1503VaBm@!	2001
PL00HG3	9802PbWd^%	2010
LKJ678	PbUI7650!!	2049
LM7WQ4	RnSs1209)*	1069

Aliases

As a reader of this book, you are a human being. You are unique, even if you have a twin. You are you, and a biometric technique can not necessarily distinguish you from someone else (i.e. iPhone X - think twins and FaceID). When we translate the human aspect of our identities into the digital world, we can have more than aliases, avatars, profiles, and therefore privileges. Information technology users can have multiple aliases, just like having more than one email address. One may be for home, and one may be for work, and we may have multiples of each. They are unique identifiers but ultimately linked back to you. These aliases should never share the same passwords for as we have discussed, this represents a cardinal violation of the password reuse rule.

These accounts, as they are assigned to us, typically have various privileges assigned to them. We may have an everyday account based on our name (Standard User.) We may also have an administrative (or elevated privileges account) based on the same name with a prefix of suffix to indicate it is a privileged account. For example, my Standard User account could be 'jtitor' and my administrative account is 'jtitor-admin.' These are both aliases for my identity and again should never share the same password.

When we work with multiple operating systems, and directory services, we can encounter that these accounts do not inherently sync and traverse platforms or applications. This can leave us with multiple aliases for Unix,

Linux, Windows, Mac, iOS, Android, Social Media, Applications, etc. I think you get the perspective.

From a threat actor's perspective, aliases are a hindrance to their malicious goals, especially if all the alias schemas are different and the passwords are different too. Laterally moving from one resource to another is complicated due to loss of privileges and authentication as they try to navigate an environment. This is good but the problem lies with development and operations. Having potentially hundreds or thousands of local non-synchronized privileged accounts across multiple users is a nightmare and could leave gaping holes in security from rogue and dormant accounts. It is the same reason security best practices prefer domain administrator accounts over local accounts to manage systems. They are easier to control, manage, log, and maintain. Having every administrator have a local non-linked account on every system and on all platforms that they need is logistically an alias nightmare.

From a privileged attack vector standpoint, the fewer the accounts with better visibility can help mitigate this risk. This is where directory services bridging comes into play. This allows one directory store, like Active Directory, to be the authentication store of authority and all supported platforms and applications leverage it for authentication and privileges using the same alias name and (hold your breath) the same password (or two factors). This means that one administrative alias works everywhere, authenticates against one directory store (the password is not stored locally in this model), and attestation reporting on a human user can occur anywhere and at any time, because all you need to do is query for the same alias name across all resources and not potentially multiple derivations. As for the same password, with multiple aliases everywhere, each resource needs to store that password for authentication. That presents yet another attack vector for a threat actor to crack passwords. With a directory bridge, that risk is mitigated.

Minimizing the number of aliases per "human user" is strategically a good best practice for any organization. Removing administrative accounts and only keeping standard users is even better and will be covered later in

the least privileges chapters. Figure 2-3 illustrates how a batch process can be assigned any alias name such that it is not obviously associated with an administrative account.

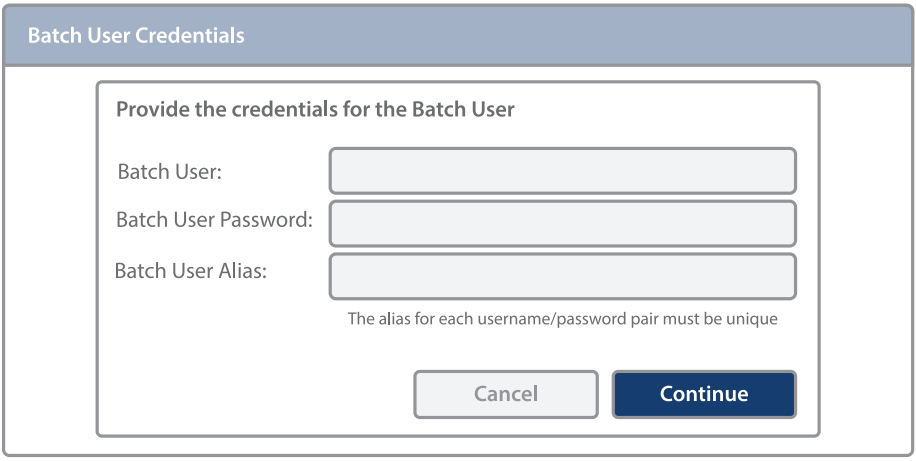

Figure 2-3. *Assigning an Administrator Alias Name*

SSH Keys

Secure Shell (SSH) keys are common tools used by Unix systems administrators to access Unix servers. The keys, when used with passphrases, provide a secure way for admins to access systems and data.

SSH keys are standard and more prevalent in Unix and Linux environments, but are also used across Windows. Admins leverage SSH keys to manage operating systems, networks, file transfers, data tunneling, and more. As with other privileged credentials, SSH keys are not necessarily tied to a single user — multiple people may share the private key and passphrase to a server that holds the public key. As with other types of privileged credentials, when organizations rely on manual processes there is a pronounced tendency to reuse a passphrase across many SSH keys, or to reuse the same public SSH key. This means that one compromised key can be harnessed to infiltrate multiple servers. Of course, there are solutions to this that will be covered in later chapters.

CHAPTER 3

Password Hacking

Hacking of a password by a threat actor can be done using several techniques. Once successful, this can obviously lead to administrator privileges if the account has been granted these rights in the first place. It's yet another reason to limit the number of administrator accounts in an environment. If the account is an administrator, it lessen the barriers to lateral movement and further attempts to crack other passwords on other privileged accounts on the same or reachable systems. As a point of reference, password hacking should not be confused with the former discussions on password exposure such as shared passwords and the insecure documentation of passwords. Password Hacking is a threat that attackers attempt to crack or determine a password using a variety of programmatic techniques and automation.

Guessing

One of the most successful 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 (covered a little later). Flat out guessing is somewhat of an art, but knowing information about the target identity enhances the process and liklihood of success by a threat actor. This information can be gathered via social media,

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_3

direct interaction, or even data gleaned and merged (or aggregated) from prior breaches. The most common variants for passwords that are susceptible to guessing include common patterns. For example:

- The word "password" or basic deviations like "passw0rd"
- Deviations of the account owner's username including initials
- Reformatted or explicit birthdays of users or their relatives
- Memorable places
- Relatives' names and derivations with numbers or special characters
- Pets, colors, foods, or other important items to the individual
- Name of the user

For a threat actor to succeed, they normally do not use automation when using targeted password guessing. While this method may be more labor intensive and have higher success rates, it can have faults and leave evidence like auto locking the 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 latteral movement increase dramatically. Imagine a person that uses only one or two base passwords everywhere for all of their digital presence.

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.

It is important to note, this is one of oldest privileged attack vectors and one of the easiest for anyone to leverage.

Dictionary Attacks

Dictionary Attacks are an automated technique 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 crack 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. A good dictionary attack system lets a threat actor do the following:

- Set complexity requirements for length, character requirements, and character set.
- Allows 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.

Unfortunately, a weakness of dictionary attacks is that it relies 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.

Finally, there are a variety of additional 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.

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 character length and complexity, but can become infeasible even for the fastest modern systems, with a password of 7 characters or more. Therefore, if a password only has alphabetical characters, all in capitals or all in lowercase (not mixed), it would take 26^7(8,031,810,176) guesses (you have a better chance of winning the lottery!). This also assumes that the threat attacker 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.

Pass the Hash

Pass the Hash (PtH) is a hacking technique that allows an attacker to authenticate to a resource by using the underlying NT 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 user name, and hash for the password using a variety of techniques like scrapping a systems' 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 passwords hash remains static for every session until the password itself is actually changed. Pass the Hash can be performed against almost any server or service accepting LM or NTLM authentication, regardless if 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 potentially a target. Once the hash is obtained, command and control or other automation allow for additional lateral movement or the exfiltration of data.

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 about personal information. They are required by many organizations when you set up a new account as a form of two-factor authentication using the challenge response of personal questions only you should know (or limited set of people should know). The end user is then prompted to answer them when logging on 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?

The risk of these security questions is far reaching in obtaining a user's passwords. Think about these scenarios:

- How many people would know the answer to any of these questions?
- Are the answers to these publicly 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 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, already owns your email or text message platform, and your security phrase is the same on multiple sites, they can continue to own you through lateral movement between resources. 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 too.

Password Resets

How often do you change your passwords? Every 30 or 90 days 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.

Keeping all your passwords unique, complex, and rotated frequently is a daunting task even for the most seasoned security professional without a password manager or a mental schema for changing the password. For example, using the month, year, initials, and a few special characters each

time so the pattern can be memorized is a common practice. 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.

Unfortunately, there is a common risk in resetting (not to be confused with changing) passwords that makes them targets for threat actors. These include the following:

- Pattern-based passwords (as described above) when reset
- Passwords that are reset via email 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

Anytime a password is reset, there is a silent acknowledgment that the current password is bad for anyone for a few reasons. It was forgotten, expired, or locked out due to numerous failed attempts. The reset, transmission, and storage of the new password are a risk until it is changed by the end user or worse, not changed by the end user at all. The password itself sits in the "ether" and the security of which is unknown. A threat actor can leverage this by requesting a password reset once an identity has been compromised and change the account to their credentials and traits for future malicious activity. 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 multi-factor authentication to validate.

- Password reset requests should always come from a secure location. Public web sites 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, preferably verbally on the telephone.
- SMS text messages are not secure for sending password reset information.

While changing passwords frequently is a security best practice, resetting passwords and transmitting them through unsecure ether is not. The risks of doing them frequently, and for large numbers of users, represent a risk in themselves since the initial reset password has been communicated using 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 two use cases.

Other Techniques

Consider that almost every word in this book, 7 letters and longer, can be potentially used in a password hacking attack if security best practices are not enforced. The user has chosen a plain English word as a password. In fact, every word even shorter than 7 letters could be a password on a system that does not meet very basic password complexity requirements in length. Once we add simple derivations of these words to include upper- and lowercase, and substitution of specific letters for numbers, like 0 for o, we have a finite list of words that people would statistically choose for a password and can be automated by a program to systematically check

against an account, including default passwords, and see if the user has made a cardinal mistake by selecting a guessable password. While these are basic assumptions for a password hacking, they are relevant for securing passwords and privileges using truly randomized and highly complex passwords found in Privileged Access Management (PAM) solutions. This makes the only choice to guess a password Brute Force or memory stealing hash technology viable like Pass the Hash. Fortunately, these are only minor players as threat actors attempt to steal passwords. Password reuse, default passwords, and poorly secured passwords make up the bulk of all password-related breaches in modern businesses and government. It should be pointed out that there are a wide variety of other techniques to steal passwords that may leverage multiple techniques from watering holes to golden ticket attacks. The list is larger than this book can accommodate. The main point in referencing them is that they are not the initial attack vector for stealing a password. Techniques like watering holes rely first on compromising a web site to subsequently steal a user's login credentials. Social engineering may, or may not, play a factor. Golden ticket attacks are only experienced after the administrative rights of a domain controller are compromised. A threat actor had to compromise the domain administrator account first in order to create additional Kerberos certificates.

The key takeaway is that threat actors will always find another method to steal passwords. We will brand them with clever names, recommend best practices; but in the end, whatever the technique, they are after our privileged accounts.

CHAPTER 4

Password Less Authentication

While there is a movement to remove passwords and traditional credentials from the authentication process, and many emerging solutions are claiming to do so, the unfortunate fact for any of these technologies is still tied to the binary nature of all computing systems. You either have been authenticated, or you have not; the outcome is always Boolean. While you can apply context-aware scenarios to limit access based on other criteria to minimize risk, the user has still been authenticated with yes or no criteria. Their location may limit access, the device may be restricted to specific resources, but in the end, they still have been authenticated in a binary manner. The emerging technologies that layer upon existing solutions such as biometrics, keyboard response time, and even multi-factor authentication still need to translate to that same yes or no answer. For many of these technologies, new security concerns have been raised, and others just may have inherent flaws in their approach (Figure 4-1):

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_4

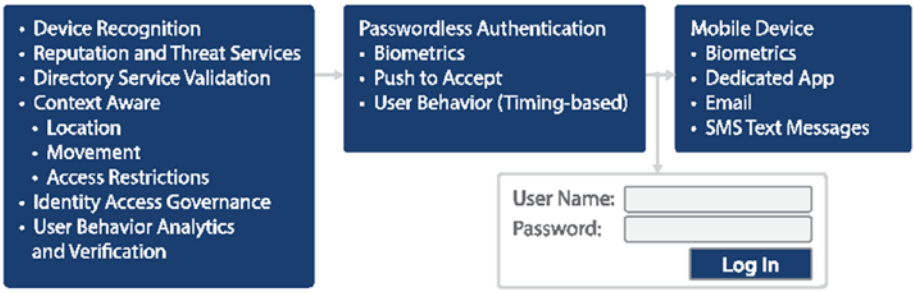

Figure 4-1. *Sample Passwordless Authentication Mechanisms*

- Biometrics – this technology has been deemed by many technologists as the holy grail to replace credentials. While it is true that biometrics should be unique per identity, it has been proven that fingerprints can be replicated, facial recognition bypassed (the twin and child factor for FaceID), and the databases storing biometric information stolen for future malicious activity. Biometrics as an authentication mechanism alone is never a good idea. It should always be used in multi-factor authenticated processes to validated an account and thus not a viable credential replacement technology today.
- Keystroke Timing – an emerging patented technology provides authentication based on the rate keys are typed on a keyboard. Surprisingly the results of this method are very good, but it has been shown to have "known" false positive authentication rates when the user is under duress. For example, if the user breaks their hand, or is only typing with one hand due to something they are carrying, these models falter in authenticating a user since the pattern and rates have

never been documented for them before. The machine-learning portion needs to be retrained for this situation and using traditional credentials are unfortunately the only viable fallback mechanism (with multi-factor authentication just to be secure).

- Federated Services – one of the more promising approaches uses a blended technology approach from single sign-on and multi-factor authentication. The approach requires you to authenticate once to a federated service using a trusted mechanism. This service may be based on traditional credentials and include other multi-factor technology normally hosted in the cloud. Once authenticated, your presence (geolocation, device, asset risk, time and date, etc.) is used to authenticate against other services and applications. This can be seamless or rely on a two-factor code sent to a dedicated mobile application, SMS text, or another vehicle to authorize a new session. Social media accounts like Facebook and Google have been at the forefront of this technology, but outside of Microsoft Active Directory Federated Services and Microsoft Live, the adoption model has been slow to trust this approach unless a dedicated commercial multi-factor vendor has been installed.

At this time, passwordless solutions are a goal that still relies on traditional credentials under the hood within the operating system, application, and authentication standards. They are just a new layer for authentication and currently cannot replace credentials completely.

Please consider the following technology problems that must be resolved to go fully passwordless:

- As a backup when passwordless layers fail, credentials are the only viable backup.
- Legacy technology (and every piece of technology created at the time this book was published) still requires credentials under the hood, whether this is an administrator account or service credentials. Passwordless solutions are just a new security layer on top.
- Credentials may be used (along with multi-factor) as the first step for passwordless authentication in conjunction with single sign-on or other federated authentication services. The initial authentication still uses a password.
- A physical injury to the hand, eye, or face can cause biometrics to fail. Microsoft Hello, Samsung Galaxy Note, and Apple iPhones FaceID are the first generation to take these to consumers, and it is a matter of time whether businesses will accept them as enterprise-ready authentication mechanisms. Their reliability, false positives, and potentially false negatives will also drive whether this is an acceptable passwordless solution.

For a threat actor, passwordless solutions represent a real challenge to gain privileged access compared to traditional credentials. However, just like recent tribulations in election hacking, sometimes it is better to go after the supplier of the technology than trying to hack the organization that has deployed it. If you can break the passwordless solution by stealing a biometric database, finding faults or vulnerabilities with the tool itself, or installing malware on a mobile device, the end results of a breach are virtually the same, and a passwordless solution really would not provide many benefits outside of the identity's ability to remember a password.

CHAPTER 5

Privilege Escalation

Once we have established an authenticated session of any type, whether the session is legitimate or hacked via any of the attacks previously discussed, a threat actor's goal is to elevate privileges and extract data (outside of ransomware and causing business disruptions). See Figure 5-1. A standard user typically does not have rights to a database, sensitive files, or anything of value in mass. So how does a threat actor navigate an environment and gain administrator or root privileges to exploit them as an attack vector? There are six primary methods: passwords, vulnerabilities, configuration, exploits, malware, and social engineering. In addition, there are privileged authentication disciplines that could minimize or lead to additional risk: multi-factor authentication and local or centralized privileges.

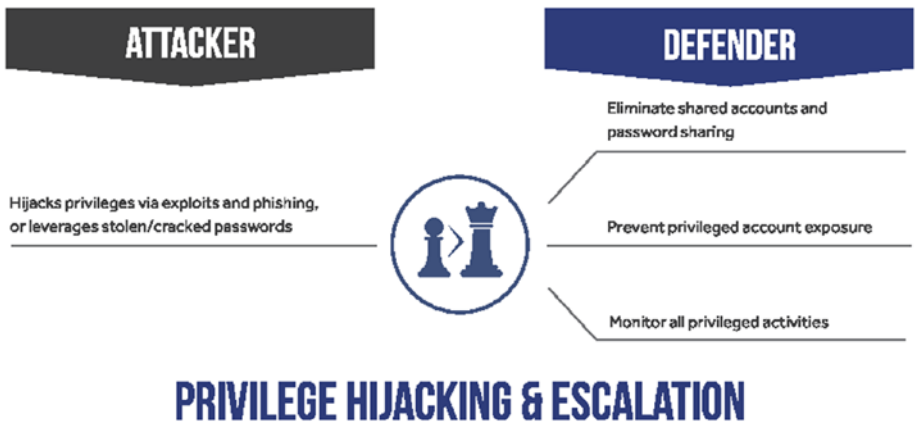

Figure 5-1. *Privilege hijacking and escalation*

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_5

Passwords

We have already established that valid credentials will allow you to authenticate against a resource. Once a credential is known, typically the same name as an email address prefix, obtaining the account's password becomes a hacking exercise. The compromised account could be a standard user (regular employee), an administrator, or some permission and privilege level in between. Often a threat actor will target an administrator or executive directly since their credentials often have privileges to directly access sensitive data and systems, and to move laterally with little inspection. Unfortunately, or fortunately, that is not always possible for a threat actor. They need to start infiltration by gaining a foothold within the environment. Gaining this beachhead could be the result of leveraging missing security patches all the way through social engineering. Once the initial infiltration has been successful, threat actors will perform surveillance and be patient, waiting for the right opportunity to pursue their mission. Typically threat actors will pursue the path of least resistance and will perform steps to clean up their tracks to remain undetected. Whether this involves masking their source IP address or deleting logs, any evidence about their presence can either stop their movement or allow the organization to ramp up forensics to monitor the breach.

There are multiple philosophies on what to do once a breach is detected that are outside of the scope of this book. Regardless, when dealing with compromised privileges and stolen passwords, everything permissioned to that account is now fair game for the attacker. The risk of what is available to them versus allowing them to continue executing for evidence must be clearly defined. Resetting passwords is typically the best strategy and reimaging infected systems a standard practice (especially if they are servers). Simply requesting the end user to change a password does not mitigate the risk if an exploit or malware obtained the password in the first place. Compromised passwords are the easiest vector for a

privileged attack, and the accounts associated with them control almost every aspect of a modern information technology environment. Therefore, passwords compromised on the most sensitive accounts can be a "game over" event for some companies and those should always be treated with care and properly identified for password management and risk assessments.

Vulnerabilities

A vulnerability itself does not allow for a privileged attack vector to succeed. In fact, a vulnerability in and of itself just means that a risk exists. Vulnerabilities are nothing more than mistakes. They are mistakes in the code, design, implementation, or configuration that allows malicious activity potentially to occur via an exploit. Thus, without an exploit, a vulnerability is just a potential problem and used in a risk assessment to gauge what **could** happen. Depending on the vulnerability, available exploit, and resources assessed with the flaw, the actual risk could be limited or a pending disaster. While this is a simplification of a real risk assessment, it provides the foundation for privileges as an attack vector. Not all vulnerabilities and exploits are equal and depending on the privileges of the user or application executing in conjunction with the vulnerability, the escalation and effectiveness of the attack vector can change. For example, a word processor vulnerability executed by a standard user versus an administrator can have two completely different set of risks once exploited. One could be limited to just the user's privileges as a standard user, and the other has full administrative access to the host. And, if the user is using a domain administrator account or other elevated privileges, the exploit could have permissions to the entire environment. This is something a threat actor targets as a low-hanging fruit. Who is operating outside of security best practices and how can I leverage them to infiltrate the environment?

With this in mind, vulnerabilities come in all "shapes and sizes." They can target the operating system, applications, web applications, infrastructure, and so on. They can also target the protocols, transports, and communications in between resources from wired networks, WiFi, to tone-based radio frequencies. However, not all vulnerabilities have exploits. Some are proof of concepts, some are unreliable, and some are easily weaponized and even included in commercial penetration testing tools or free open source. Some are sold on the dark web for cybercrimes and others used exclusively by nation-states until they are patched or made public (intentionally or not). The point is that vulnerabilities can be in anything at any time. It is how they are leveraged that makes them important, and if the vulnerability itself leads to an exploit that can change privileges (privileged escalation from one user's permissions to another), the risk is a very real privileged attack vector. To date, less than 10% of all Microsoft vulnerabilities patched allow for privilege escalation. This is a real threat considering hundreds of patches are released every year for their solutions alone dating back over the last 15 years.

To convey the risks and identification of vulnerabilities, the security industry has multiple security standards to discuss the risk, threat, and relevance of a vulnerability. The most common standards are the following:

- Common Vulnerabilities and Exposure (CVE) – a standard for information security vulnerability names and descriptions.
- Common Vulnerability Scoring System (CVSS) – a mathematical system for scoring the risk of information technology vulnerabilities.
- The Extensible Configuration Checklist Description Format (XCCDF) – a specification language for writing security checklists, benchmarks, and related kinds of documents.

- Open Vulnerability Assessment Language (OVAL) – an information security community effort to standardize how to assess and report upon the machine state of computer systems.
- Common Configuration Enumeration (CCE) – provides unique identifiers to system configuration issues to facilitate fast and accurate correlation of configuration data across multiple information sources and tools.
- Common Weakness Enumeration Specification (CWE) – provides a common language of discourse for discussing, finding, and dealing with the causes of software security vulnerabilities as they are found in code.
- Common Platform Enumeration (CPE) – a structured naming scheme for information technology systems, software, and packages.
- Common Configuration Scoring System (CCSS) – a set of measures of the severity of software security configuration issues. CCSS is a derivation of CVSS.

The results from all this information allow security professionals and management teams to discuss and prioritize the risks from vulnerabilities. In the end, they must prevent exploitation and any of the impactful attack vectors that could come from their abuse. Without a common language and structure among vendors, companies, and government, assessments would be nearly meaningless between organizations based on their implementation of security best practices. A critical risk for one company may not exist for another simply based on their environment. Standards like CVSS allow for that to be communicated correctly to all stakeholders. See Figure 5-2 which illustrates perimeter exploitation.

Figure 5-2. *Perimeter exploitation and considerations*

Configurations

Configuration flaws are just another form of vulnerabilities. They are, nonetheless, flaws that do not require remediation - just mitigation. The difference between remediation and mitigation is key. Remediation implies the deployment of a software or firmware patch to correct the vulnerability. This is commonly referred to as Patch Management. Mitigation is simply a change at some level in the existing deployment that deflects (mitigates) the risk from being exploited. It can be simple change within a file, group policy, or updating certificates. In the end, they are vulnerabilities based on poor configurations and can be exploited as a privileged attack vector just as easily by a threat actor.

The most common configuration problems exploited for privileges involve accounts that have poor default security best practices. This could be blank or default passwords upon initial configuration for administrator or root accounts, or insecure communication paths that are not locked down after an initial install due to a lack of expertise or undocumented back door.

Regardless, configuration flaws just require a change to fix. And, if the flaw is severe enough, a threat actor can have root privileges with little to no effort.

Exploits

Exploits require a vulnerability. Without a documentable flaw, an exploit cannot exist. We may just not understand the vulnerability when a new exploit appears in the wild. It can take some time for security professionals to reverse engineer an exploit to figure out what vulnerability was leveraged. This is typically a very technical forensics exercise. As mentioned in the Vulnerabilities section, Exploits can also take on many different "shapes and sizes" too. They can be used to leak information, install malware, provide surveillance; but ultimately, the goal is to create a sustainable and undetected beachhead within a resource. Exploits themselves can be very destructive in their execution methodology, and the most successful ones do exactly the opposite. An exploit that can gain privileges, execute code, and go undetected is very dependent on the vulnerability but also depends on the privileges the exploit has when it executes. This is why vulnerability management, risk assessments, patch management, and privileges are so important. Exploits can only execute in the confines of the resource they compromise. If no vulnerability exists due to remediation, the exploit cannot execute. If the privileges of the user or application with the vulnerability are low (standard user), and no privileged escalation exploitation is possible, then the attack is limited in its capabilities. However, don't be fooled: exploitation, even at standard user privileges, can cause devastation in the form of ransomware or other vicious attacks. Fortunately, the vast majority can be mitigated (contained) just by lowering privileges and minimizing the surface area for a privileged attack. Exploits succeed the best with the highest privileges. Therefore, minimizing them is a security best practice to thwart many exploits and deflect a threat actor's attempt to compromise an organization.

Malware

Malware, commonly referred to as viruses, spyware, adware, ransomware, etc., are any class of undesirable or unauthorized software designed to have malicious intent on a resource. The intent can range from surveillance, data leakage, disruption, command and control, to extortion. If you pick your favorite crime that can be translated to an information technology resource, malware can provide a vehicle to instrument cyber-criminal activity for a threat actor. Malware, like any other program, can execute at any permission from standard user to administrator (root). Depending on its creation, intent, and privileges, the damage it can do can be anything from an annoyance to a game-over event. Malware can be installed on a resource via a vulnerability and exploit combination or through legitimate installers, weaknesses in the supply chain, or even social engineering such as phishing. Regardless of the delivery mechanism, the motive is to get unauthorized code executing on a resource. Once running, it becomes a battle of detection by anti-malware vendors and threat actors to keep executing, avoid detection, and remove the threat. This includes malware adapting itself to avoid detection as well as disabling defenses to continue proliferation. Malware itself, based on intent, can perform functions like pass-the-hash and keystroke logging. This allows for the stealing of passwords to perform attacks based on privileges by the malware itself or other attack vectors deployed by the threat actor. Malware is just a transport vehicle to continue the propagation of a sustained attack and ultimately needs permissions to obtain the target information sought after by the attacker. It is such a broad category of malicious software that when discussing privileges, the subset that scrapes memory or provides surveillance is the most relevant.

Social Engineering

If you grew up with siblings, you might have had the fortune of being the brunt of a practical joke. Everything from smell my finger, open this box, or taste this. While the examples are rather crude, they are no different from the hacking capabilities we all experience via social engineering and the desire of a threat actor to gain privileges. The main motive from our relatives was to leverage our trust into doing something malicious for the amusement (normally laughter) of our siblings. As harmless as it sounds, we hopefully learned for the next time.

Social engineering is no different. We have a blind trust in the email we receive, the phone call we answer, or even the letter we receive to believe someone is contacting us. If the message is crafted well enough and even potentially spoofing someone we already trust, then the threat actor already gained the first step in deceiving us and potentially carrying out a ruse. If in fact we act on the fake correspondence from a work colleague, friend, or even a sweepstakes, we may just become a victim of social engineering.

Considering the modern threats in the cyber world from ransomware to recording our voices on a phone call, the outcome can become much more severe than eating a dead worm. At the risk of becoming paranoid about every email we receive and phone call we answer, we need to understand how social engineering works and how to identify it in the first place without losing our sanity. This learned behavior is no different than figuring out whether your sibling has lied about a message from your parents or not. Sometimes you just need to verify the message before taking action and understand the risks from the outcome.

From a social engineering perspective, threat actors attempt to capitalize on a few key human traits to meet their goals:

- Trusting – the belief that the correspondence, of any type, is from a trustworthy source.
- Gullible – the belief that the contents, as crazy or simple as they may be, are in fact real.
- Sincere – the intent of the content is in your best interest to respond or open.
- Suspicious – the contents of the correspondence do not raise any concern by having misspellings and poor grammar, or by sounding like a robot corresponding on the phone.
- Curious – the attack technique has not been identified (as part of previous training), or the person remembers the attack vector but does not react accordingly.

If we consider each of these characteristics, we can appropriately train team members not to fall for social engineering. The difficulty is overcoming human traits and not deviating from the education. To that end, please consider the following training parameters and potential self-awareness techniques to stop social engineering:

- Team members should only trust requests for sensitive information from known and trusted team members. An email address alone in the "From" line is not sufficient to verify the request, nor is an email reply. Their account could be compromised. The best option is to learn from two-factor authentication techniques and pick up the phone. Call the party requesting the sensitive information and verify the request. If the request seems absurdly insane like requesting W-2 information or

a wire transfer, verify this is acceptable according to internal policies or other stakeholders such as finance or human resources (it could be an insider attack). Simple verification of the request from an alleged trusted individual, like a superior, can go a long way to stopping social engineering. In addition, all of this should occur before opening any attachments or clicking on any links. If the email is malicious, the payload and exploit may have executed before you have any verification.

- If the request is coming from an unknown source but is moderately trusted—such as a bank or business you interact with—simple techniques can stop you from being gullible. First, check all the links in the email and make sure they actually point back to the proper domain. Just hovering over the link on most computers and email programs will reveal the contents. If the request is over the phone, never give out personal information. Remember, they called you. For example, the IRS will never contact you by phone; they only use USPS for official correspondence. Don't let yourself fall for the "sky is falling" metaphor.
- Teaching how to identify genuine correspondence or not is rather difficult. Social engineering can take on many forms from accounts payable, love letters, resumes, to human resources interventions. Just stating "if it seems too good to be true" or "nothing is ever free" only handles a very small subset of social engineering attempts. In addition, if peers receive the same correspondence, it only eliminates spear phishing attempts as the probable attack vector. The best option is to consider if you should be receiving the request in

the first place. Is this something you normally do or is it out of the ordinary to receive it? If it is, default back to trust. Verify the intent before proceeding.

- Suspicious correspondence is the easiest way to detect and deflect social engineering attempts. This requires a little detective-style investigation into the correspondence by looking for spelling mistakes, poor grammar, bad formatting, or robotic voices on the phone, and if the request is from a source with whom you have no interaction. This could be an offer of a free cruise, or from a bank at which you have no accounts. If there is any reason to be suspicious, it is best to err on the side of caution: do not open any attachments or files, click on any links, or verbally reply, and delete the correspondence. If it is real, the responsible party will call back in due course.
- Curiosity is the worst offender from a social engineering perspective. What could happen, what will happen, and nothing should happen to me since I *believe* I am fully protected by my computer and company's information technology security resources. That's a false assumption. Modern attacks can circumvent the best systems and application control solutions—even leveraging native OS commands to conduct their attacks. The best defense for a person's curiosity is purely self-restraint. Do not reply to "Can you hear me?" from a strange phone call; do not open attachments if any of the above criteria have been realized; and do not believe nothing can happen to me (even for people using Mac OS). The fact is it can, and your curiosity should not be the cause. Being naïve will make you a victim.

Social engineering is a real problem, and there is no technology that is 100% effective. Spam filters can strip out malicious emails, and anti-virus (anti-malware) solutions can find known or behavior-based malware, but nothing can stop the human problem of social engineering and potential insider threats. The best defense for social engineering is education and an understanding of how these attacks leverage our own traits to be successful. If we can understand our own flaws and react accordingly, we can minimize the threat actor's ability to compromise resources and gain privileges within the environment.

Multi-Factor Authentication

While we have been focusing on passwords as the primary form of authentication with credentials, other authentication techniques can strengthen the authentication model. Each have their own merits and flaws, but in the end, an account is authenticated, and privileges applied. As a security best practice, and required by many regulatory authorities, these additional authentication techniques are required to secure access instead of a traditional username and password credentials only. They provide an additional layer that makes it more difficult (but not impossible) to hack and thus are always recommended when securing sensitive information. This model is called multi-factor authentication.

The premise for multi-factor authentication (two-factor is a subset category for authentication) is simple. In addition to a traditional username and password credential, an additional "passcode" or evidence is needed to validate the user. The delivery, passcode, and randomization of passcode varies from technology to technology and from vendor to vendor. These typically take on the form of knowledge (something they know unique to them), possession (something they physical have that's unique to them), and inherence (something they are in a given state).

The use of multiple authentication factors to prove one's identity is based on the converse of a positive identification for authentication. The premise is that an unauthorized threat actor is most likely unable to supply all the factors required for correct access due to an additional authentication variable. During a session, if at least one of the components is in error, the user's identity is not verified with sufficient certainty (2 of 3 criteria match), and access to the resource being protected by multi-factor authentication remains restricted. The authentication factors of a multi-factor authentication model typically include the following:

- A physical device or software (like a phone app that produces a secret passcode re-randomized on a regular frequency.
- A secret code known only to the end user like a PIN that is typically mentally stored.
- A physical characteristic that can be digitally analyzed for uniqueness like a fingerprint, typing speed, or voice. These are called biometric authentication technology.

It is important to note that multi-factor is an identity specific layer for authentication. Once validated, the privileges assigned as a potential attack vector are not significantly altered. For example, if credentials are compromised in a traditional username and password model, a threat actor could authenticate against any target that will accept them locally or remotely. For multi-factor, even though there is an additional variable required, including physical presence, once you are validated, lateral navigation is still possible from your initial location (barring any segmentation technology or policy). The difference is solely your starting point for authentication. Multi-factor has to have all the security conditions met from an entry point while traditional credentials do not. A hacker can leverage credentials within a network to jump from host to host while changing credentials as needed. Unless the multi-factor system itself

is compromised, they cannot target a multi-factor host for authentication unless they have all the security material available to authenticate. Hence, there always needs to be an initial entry point for starting a multi-factor session and once in, using credentials or other means if lateral movement is still possible.

Local versus Centralized Privileges

In subsequent chapters, we will discuss the various approaches to strong and efficient Identity and Privilege Management options that are available to organizations. As we discuss the privileged attack vector in depth, it will become apparent that this goal may be best served by a strong identity and access program that may leverage a directory service foundation. However, as organizations consolidate and simplify identity infrastructures, they must be cautious. If not implemented or secured correctly, they can become a privilege's greatest weakness. If one privileged account is comprised, the risk of lateral movement (Figure 5-3) to other resources relying on, and trusting this service for authentication may be possible.

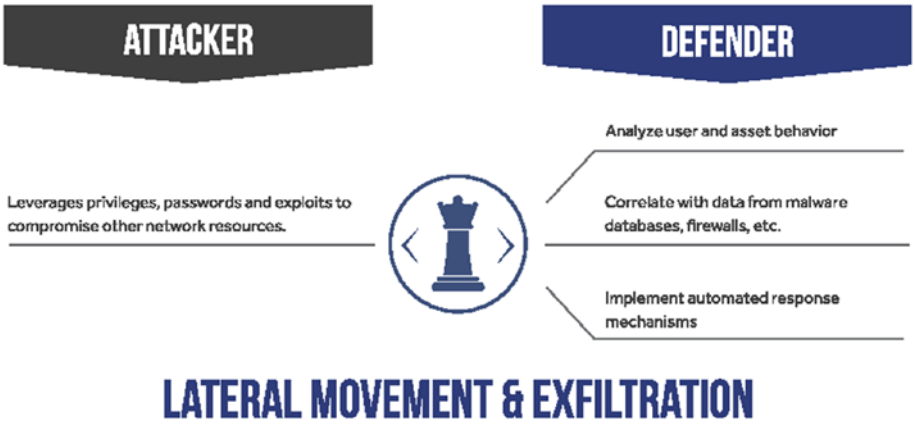

Figure 5-3. *Lateral Movement*

However, without a strong centralized identity and access program, authentication cannot exist between layers from file systems, operating systems, users, applications, data, and even business partners. It is an age-old information technology dilemma to provide the best security but allow for smooth and seamless business functions. Too much security and nothing works. Too little security and it can be an instrument for continued execution by a threat actor to operate anywhere within the environment.

In the end, the best considerations for privileges are granularity and centralization. This allows finite controls for rights and a single place for management. For today's modern infrastructure, this is the best security practices we can implement today.

CHAPTER 6

Insider Threats

For most security professionals, they are tired of hearing about Insider Threats. They are not new; it is an old-school attack that has been made public due to the nature, quantity, and sensitivity of the data being stolen electronically. Years ago, these attacks occurred on a regular basis but did not have the same labels or stigma they have today. I am not saying they were acceptable back then either. We just need to be realistic about what an Insider Threat is and acknowledge that it has been going on in various forms for hundreds of years.

By definition, an Insider Threat is an internal persona behaving as a threat actor. Regardless of the techniques they are using, they are not behaving in the best interest of the company, potentially breaking the law, and exfiltrating information they do not have permission to possess. An old-school example of this type of threat is client lists. It's an Insider Threat that's still relevant today, by the way. A salesperson, executive, etc., that is planning to leave an organization may have photocopied or printed client lists and orders before leaving the organization to have a competitive edge when they start with a new employer. The volume of paper potentially would have to be substantial to make an impact, but leaving with confidential information on printed paper is still an Insider Threat. Obviously, they were not leaving with file cabinets of material, but today with electronic media, and the Internet, that volume of data could easily be egressed without anyone noticing. And, as a reminder, that file cabinet of sensitive information can easily fit on a USB thumb drive in a person's pocket. Therefore, we now have a label for this type of threat

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_6

and Insider Threats are becoming more relevant. It still makes security professionals sick to their stomachs because the crime is old, but the methods and volume are now something to consider and require a new strategy to protect against.

Insider Threats occur for a variety of reasons. This includes aspects of a human persona looking to hurt or gain an advantage against an organization. Regardless of their intent, it's the digital aspect of an Insider Threat that warrants the most attention. Human beings will do the most unusual things in the most dire (or direst) situations, but if they are not permitted to, many of the risks of Insider Threats can be mitigated. Consider the following for your business:

- How many people have access to sensitive information in mass? This is not using a program to retrieve one record at a time, but rather who has direct access to the database or can run a report to dump large quantities of information from a query?
- Are all accounts valid people that are still employed or relevant?
- How often do you change the passwords for sensitive accounts?
- Do you monitor privileged access to sensitive systems and data?

So, in fairness, answering those questions honestly could be opening Pandora's box. Nonetheless, you should answer them if you care about Insider Threats. Here is why:

- Only administrators (not even executives) should have access to data in mass. This prevents an insider from dumping large quantities of information, or an executive's account being hacked and leveraged against the organization.

- All users should never use administrative accounts for day-to-day usage like email. This includes administrators themselves, in case their accounts are compromised too. All users should have standard user permissions.
- All access to sensitive data should be valid employees only. Former employees, contractors, and even auditors should not have access on a daily basis. These accounts should be removed or deleted per your organization's policy.
- Employees come and go. If the passwords are the same as people leave and new hires are acclimated, the risk to sensitive data increases since former employees technically still have known passwords to the company's sensitive information.
- Monitoring privileged activity is critical. This includes logs, session monitoring, screen recording, keystroke logging, and even application monitoring. Why? If an Insider is accessing a sensitive system to steal information, session monitoring can document their access and how they extracted the information and when.

If you think that if you follow all of these steps to protect against Insider Threat you will be safe, you are wrong. This assumes the threat actor is coming in from the front door to steal information or conduct malicious activity. Insider Threats can also evolve from traditional vulnerabilities, poor configurations, malware, and exploits. A threat actor could install malicious data capturing software, leverage a system missing security patches, and access resources using back doors to conduct similar types of data-gathering activity. Insider Threats are about stealing information

and disrupting the business but depending on the sophistication of the threat actor, they can use tools that are traditionally associated with an external threat. Therefore, we need to realize Insider Threats come from, essentially, two sides: excessive privileges (covered above) and poor security hygiene (vulnerability and configuration management). To that end, all organizations should also regularly perform these tasks to keep their systems protected:

- Ensure anti-virus or endpoint protection solutions are installed, operating, and stay up to date.
- Allow Windows and third-party applications to auto update or deploy a patch management solution to deploy relevant security patches in a timely manner.
- Utilize a vulnerability assessment or management solution to determine where risks exist in the environment and correct them in a timely manner.
- Implement an application control solution to allow only authorized applications to execute with the proper privileges to mitigate the risk of rogue, surveillance, or data collection utilities.
- Where possible, segment users from systems and resources to reduce "line if site" risks.

While these seem very basic, the reality is that most businesses do not do a good job at even the most basic security. If they do, the risk of Insider Threats can be minimized by limiting administrative access and keeping information technology resources up to date with the latest defenses and security patches. Insider Threats are not going to go away. They have been around for hundreds of years, but the medium and techniques for stealing information have evolved with modern technology. The goal is the same: stop the data leakage and be aware that an Insider has multiple

attack vectors to achieve their goals. As security professionals, we need to mitigate the risks at the source. A briefcase of paper is still an Insider Threat but not as relevant as a USB stick with your entire database of client information. In the end, an insider still needs privileges to steal all this information.

CHAPTER 7

Threat Hunting

As a child, or even as an adult, have you ever played the game, "Where's Waldo?" If you have, you may already understand how this section relates to Threat Hunting. For those who have not heard of the game, the object is to find a picture of Waldo in a picture filled with other graphics and people. Spotting Waldo is difficult and identifying him from the crowd is downright frustrating in some of the illustrations. It is a game of patience, visual acuity, and methodical review of graphics. To that end, a modern spoof on the game has graphics with nearly every person being Waldo. The objective is to find everyone that is not Waldo. This is a common analogy for false positives when performing Threat Hunting and the reason the analogy is so important.

So, for new security professionals, what is Threat Hunting? Threat Hunting is the cybersecurity act of processing information and process-oriented searching through networks, assets, and infrastructure for advanced threats that are evading existing security solutions and defenses. Firewalls, Intrusion Prevention Solutions, and Log Management are all designed to detect and protect against threats – even if they are zero-day threats and never seen before. Threat Hunting is the layer below this. What threats are actively running in my network that I am missing and how I can find them? It assumes the basic premise that a threat is there and have already been compromised.

The simple solution for most companies is to provide better inspection of the data already being collected. That includes diving deeper into log files, looking at denied logon access, and processing application events

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_7

correlated from denied application control solutions. But that is not really what Threat Hunting is. Those steps are merely security best practices and adhering to the guidelines in many regulatory standards from PCI to NIST for log management and review.

Threat hunting can be an automated or manual process to find hidden threats. It assumes the threat is already there; you just need to find it. The process involves processing multiple sources of data simultaneously and correlating information with an inherent knowledge of the systems, mission, and infrastructure producing the information. While this may sound like a canned answer, it is not. Security Information Enterprise Managers (SIEM) are designed to ingest this information but only allow limited tagging of data by source and type to apply a business element. They fail, like many technologies, to apply the human element. To aid with this and provide data intuition, this process can be automated using behavioral analytics or machine learning. It raises the bar for identifying patterns as a repetitive process, but that is all that it does; it has no knowledge of what the meaning is for detected patterns. For Threat Hunting to succeed, security professionals need to start with a hypothesis. This hypothesis assumes a threat and maps the patterns and manual review of data to the conclusion (a threat is actively occurring). Common hypotheses include the following:

- Analytics Driven: Patterns in analytics automation can be assigned risk ratings and used to determine if a high-risk pattern is occurring.
- Situational: High-value targets are analyzed including data, assets, and employees for abnormalities.
- Intelligence: Correlation of threat patterns, intelligence, malware, and vulnerability information to draw a conclusion.

Therefore, for Threat Hunting to succeed, we need to meet the following requirements or our data and hunt will be flawed:

- Crown Jewels and Sensitive (Privileged) Accounts are properly identified for data modeling.
- Sources of information can be correlated by CVE, IP address, and hostname reliably. Changes due to DHCP and even time synchronization (poor NTP implementation) can jade Threat Hunting results. We need to trust the data nearly implicitly.
- Consolidation tools like an SIEM are collecting all applicable data sources for pattern recognition.
- Threats to the business, like a game-over breach event, are established and used to build a hypothesis.
- Tools for risk assessments, intrusion detection, and attack prevention are up to date and operating correctly. If these systems are faulty, your first lines of defense are in jeopardy.
- Documentation such as network maps, descriptions of business processes, asset management, etc., are critical. Threat Hunting relies on the human element to correlate information to the business. Without being able to map a transaction to its electronic workflow, a hypothesis is blind as to how the threat occurred and is remaining persistent.

Threat Hunting is much like "Where's Waldo?" You know the threat actor exists, you kind of know what he looks like, but you cannot find him. While Threat Hunting may not know what the threat actually is, it is a safe assumption they are doing something wrong or staging to do something malicious in the future. If you can find that hidden threat, you can find

Waldo. Think of the problem, puzzle, and game with clear objectives and leverage the tools you have and not just a correlated black box report or an alert. Threat Hunting requires you to dig in deep, use a magnifying glass, and rely on your senses to help find the threat. Having security best practices, to begin with, is an absolute requirement for success since everything you do for Threat Hunting depends on it. Also, good threat actors will leverage your existing security tools against you to remain hidden. This is yet another reason why best practices must be rock solid before you embark on Threat Hunting. After all, if a threat actor is in your environment, and current solutions cannot find him, you need to question the privileges they are executing with in order to remain hidden.

CHAPTER 8

Data-Centric Audit and Protection

Not so long ago, it was much easier to protect your data. Perimeter defenses were in place, and there were only so many ways to get to your data. Data came in from IT-approved, enterprise-controlled devices and applications. It lived on your servers and in storage arrays. It was protected by walling off the outsiders and trusting your insiders, but things have changed in a big way. Now, more data than ever is collected from more applications, users, devices, cloud services, and connected hardware, with dwindling amounts of it under enterprise control. New forms of doing business demand easy access from the *outside* world. With the emergence of the cloud, your data, users, and applications may not even be on the *inside* anymore. And 'insiders' with access to your data increasingly include third parties who don't work for your organization at all. The approach to managing the granularity of access to this data is called DCAP (Data-Centric Audit and Protection).

Traditional computing models (Open Systems Interconnection model – ISO) allow access to all components on a server, in the cloud, and data based on a user's authentication. An authenticated user, depending on privileges (compromised, legitimate, or threat actor), can access all the way down the stack to the file system and the platform's configuration if privileges allow (Figure 8-1).

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_8

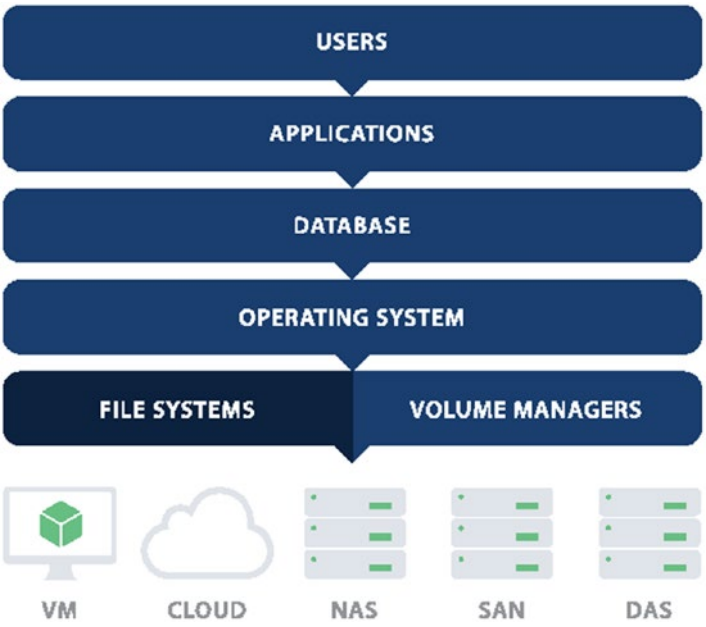

Figure 8-1. *DCAP Model Stack Model*

Restrictions and auditing are only governed by local access control lists and role-based access in applications, databases, and operating systems. An administrator can, therefore, have access to any file or volume simply by being an administrator. Users with permissions anywhere in between a standard user and administrator may need access to an application but limited (or no) access to the file system that supports it. This is the basis for client-server architecture or even a modern web application.

Unfortunately, for traditional operating system security controls across Unix, Linux, MacOS, and Windows, root or administrator allows access up and down the stack, and there is no native way to restrict access to it. You may be able to remove privileges, but as an administrator, you can always grant them back. Once an attacker has root or administrator, it is game over. There is always a way to circumvent security controls when you are an administrator. Privileged access management (PAM) can control the user's access but cannot necessarily control the file system and or

existing processes without taking ownership. File system and process control solutions can provide segmentation and encryption to files and directories (like DLP, DCAP, etc.) but cannot control the actual user being authenticated in the first place. Thus, if they are an administrator, there is probably a way to circumvent these technologies.

The solution to the problem utilizes privileged access management on the top of the stack to manage the operating system and applications, and a File Integrity Monitoring (FIM) and other control solutions to strategically block threats vertically along the traditional computing model. This includes managing privileges through all the layers from user authentication to FIM policies that grant or deny access: even as root or administrator. This requires the solutions to work together and not independently so any tampering can be correlated between the layers to prevent a compromise.

Therefore, when the concepts of DCAP are applied to PAM, the following use cases can be satisfied:

- User access is managed and monitored from authentication to file access.
- Applications are run with least privilege to mitigate elevated privilege risks without access to the supporting data structure.
- Databases and applications have passwords managed for automatic rotation and restricted access including in scripts and user utilities.
- Operating system access is restricted to standard users, commands, tasks, and scripts, and features are elevated on a need-to-use basis with specific privileges.
- Individual files associated with commands and scripts are protected separately from tampering but assigned or excluded to the same user privileges.

- User access in an attack chain can be monitored and mitigated along every horizontal plane in a traditional computing model.
- Only trusted and authorized users have access to an asset and its supporting data using privilege and file system integrity monitoring technology.
- The removal of privileges from the user to the application, and from user to the file system, can be supported in a trusted computing environment.

Data-Centric Audit and Protection is a natural extension of privileged access management. It applies the technical controls and policies for privileged use below the operating system to the file system and below access control lists. File integrity Monitoring (FIM) solutions that integrate with privileged access management provide this vehicle and provide a holistic approach to monitoring any layer a threat actor may use for exfiltration of information. This includes even blocking an administrator for accessing files and directories and relying on FIM as a security solution to enforce this segmentation.

CHAPTER 9

Privileged Monitoring

The primary risk for any privileged access activity is the activity itself. As an administrator or root, you must ask the following question: Was the activity appropriate, a mistake, or a threat actor behaving badly using elevated credentials? Unless you are sitting over someone's shoulder and have the expertise to monitor the activity, there are plenty of gaps in the traditional security model to review this activity and verify every session, every command, and all the information downloaded or displayed on the screen. Reviewing all activity is a daunting task, but luckily technology and automation exist to help address this significant challenge.

Session Recording

Session recording is the act of logging all visible activity that may appear on an end user's screen during a session (Figure 9-1). It can be done in the form of video recording, text logging, or rapid screen captures based on screen changes. Typical session recording solutions ensure that recordings are securely stored, allow for indexing, and provide advanced capabilities

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_9

for searching for details and understanding context by an auditor or via automation. Session recording can be implemented using a variety of technologies:

- An inline video capturing system that records monitor output before displaying on a screen. This technology typically also bundles OCR (Optical Character Recognition) to scrape the screen for keywords and text in the display. This technology requires hardware on the video side of servers and is normally not viable for cloud or virtualized technologies.
- An end-user agent or browser plug-in that captures the screen or session based on activity. The results are cached or streamed to a central server for review and processing. This approach requires agent technology to be deployed and does not manage out-of-band connectivity that can circumvent recording technologies.
- A proxy technology that is protocol aware to provide agentless screen recording of an active remote session. This approach supports segmentation and requires access to be routed through the proxy for a successful connection. All recordings are therefore recorded by the proxy, not stored on the end user's asset, and do not require hardware modifications except for the introduction of the proxy itself.

Regardless of the technology approach, the goal is the same: to review privileged session activity to sensitive data and systems. While this approach alone does not stop the activity of the threat actor, it documents their activity out of bounds of normal operations. The recording of privileged activity can be used for forensics and, when properly configured, can help identify a threat.

In addition, if the session recording system is advanced enough, automation can enable more proactive response to inappropriate behavior. For example advanced rules can be configured to trigger on screen output to perform mitigation activities such as sending an alert, locking or terminating the session itself, or disabling the associated user account. While this functionally requires a mature and advanced setup, it steps up the game if a threat actor tries to maintain a persistent presence by running specific commands or downloading information.

Finally, when discussing regulatory compliance with auditors, session recordings meet the requirements of documenting the privileged activity of appropriate use and privileged user attestation reports.

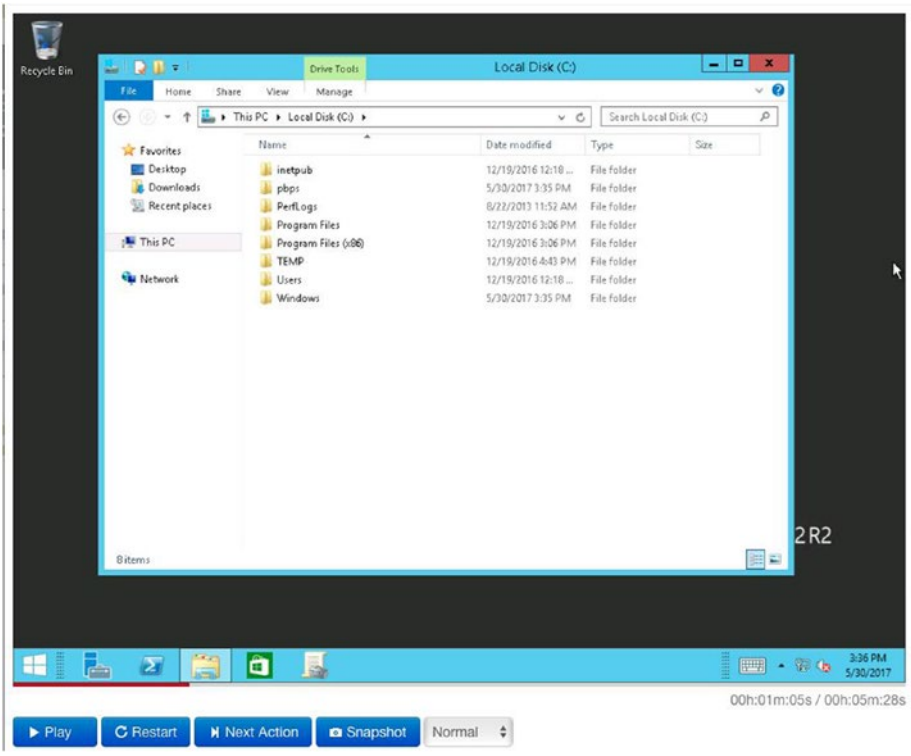

Figure 9-1. *Session Recording Playback*

Keystroke Logging

While session recording documents the screen itself, graphical or text based, it does not capture the end user's keystrokes from a keyboard: just the results if they show up on the screen. Shortcuts and keyboard commands may not be captured at all like copy (Ctrl-C). Based on the screen recording paradigms above, keystroke logging requires one of three methods as well to function and capture all user input:

- A physical inline device via USB or PS2 to capture keystrokes from a keyboard. These devices can store the information locally or have a software or network component to upload the capture information. There is no physical solution for wireless keyboards that connect via Bluetooth or proprietary dongle.
- An end-user agent that captures keystrokes. This is a common approach but needs to be whitelisted and not confused with malware that performs keystroke logging as well. This approach works with all wired and wireless keyboard technologies since the agent captures all input device data.
- Proxy technology that captures the difference between screen rendering and user input. This approach requires no physical hardware (outside of the proxy) and no local agent to capture explicit user keystrokes. Proxy technologies to capture keystrokes works with virtual form of keyboard or textual input technology.

The primary purpose of keystroke logging is to stop a threat actor at the command level. Specific commands to add a user, retrieve a database, or install malware are relatively standard across operating systems, applications, and databases. If the privileged monitoring system

is properly configured to monitor, alert, or terminate a session if these commands are issued, a breach can potentially be identified before valuable information is leaked. A threat actor must issue these commands in order to be successful in their attack. The commands themselves require privileged elevation via any of the methods we have previously discussed. Therefore, if we can identify and control authorized sessions successfully and flag for potentially malicious ones, we have another vehicle to mitigate privileges as an attack vector. See Figure 9-2.

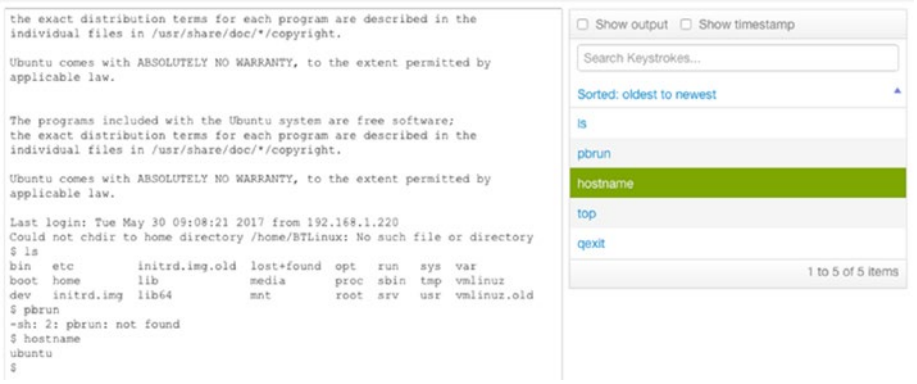

Figure 9-2. *Command-Line Filtering and Command Searching*

Application Monitoring

Applications represent a unique challenge for privileged monitoring. Every application is essentially different even if they share best practices for common menus, buttons, or depend on runtime engines from Oracle Java to Adobe Flash and even native compiled code. Session recording can capture mouse movement and screen recording but reviewing the sessions for a specific button, client utility, or dialog screen banner is labor intensive without additional technology. There is nothing in native session recording to capture application activity outside of a visual change since the primary input mechanisms are mouse clicks or using a touch screen. Also, keystroke logging can not capture mouse clicks outside of x-axis and

y-axis coordinates unless it is aware of the application itself. Due to these problems, the only solutions that work for application monitoring are to have local code present in the form of an agent, dissolvable (temporary) agent, or advanced OCR (optical character recognition) technology. OCR however requires post processing of the recording, may have trouble with fonts, cannot see file paths, and is not viable for real-time alerting. Therefore, the only viable method for application monitoring related to PAM is to use some form of agent technology.

Application monitoring agents, regardless of the delivery mechanism (persistent or dissolvable), monitor for API calls, mouse clicks, and screen changes based on user interaction. The application's title bar, button names, and menus are all exposed via Windows API's, for example. When a user interacts, they can be captured and documented on a time line with the session recording and keystrokes as well. This provides a complete audit trail for forensics or regulatory compliance attestations and potential malicious activity. Think about our Where's Waldo example for Threat Hunting.

For a threat actor, the final vector for data manipulation is under security management. Tools that allow you to graphically manipulate data and continue malicious activity are monitored even if they use the graphic user interface only for their attack. Buttons and dialogues are typically clearly labeled for data deletion, download, or querying for all programs. Therefore, similar automation techniques to keystroke logging can be used to look for keywords that contain indications of malicious activity. The results can alert security teams or terminate the session using the same proxy or agent technologies.

Application monitoring is a vital part of thwarting a threat actor. These applications need privileges even in the user interface to perform sensitive tasks, and monitoring the application itself as it interacts with the user and operating system allows for sensitive user-interface components to be monitored for inappropriate activity. See Figure 9-3.

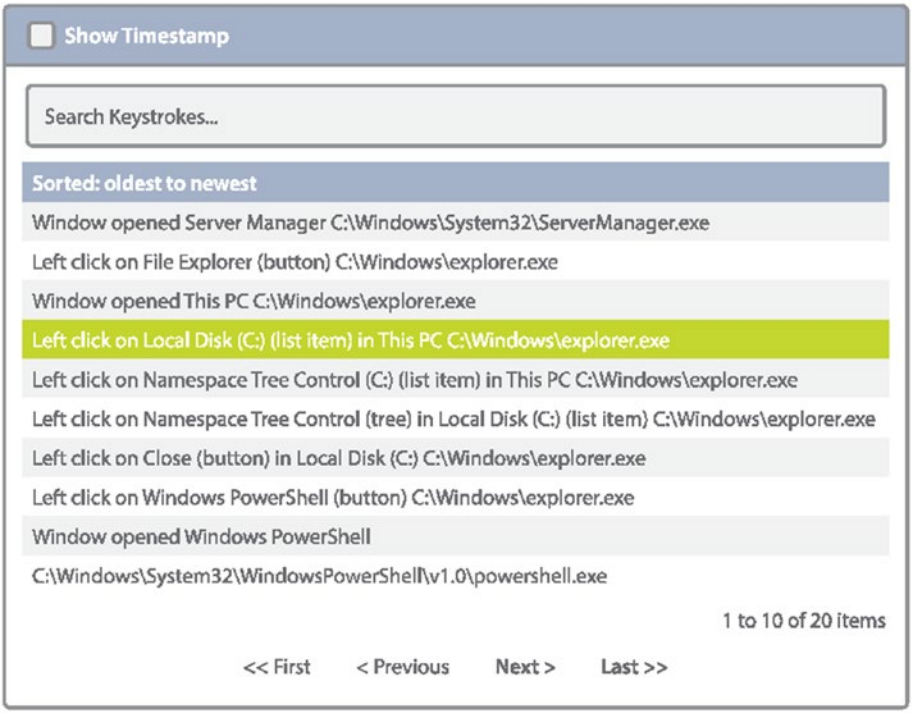

Figure 9-3. *Application Monitoring using Agent Technology*

CHAPTER 10

Privileged Access Management

Privileged Access Management (PAM) is often referred to as Privileged Account Management (also PAM) or Privileged Identity Management (PIM). It is considered a subset of the Identity Access Management (IAM) or Identity Access Governance (IAG) market as defined by leading analysts.

PAM's primary goal is to keep your organization safe from accidental or deliberate misuse of privileged credentials, the risks of which we have clearly defined. This threat is particularly relevant if your organization is evolving and experiencing change due to growth, new markets, and other business expansion initiatives. The larger and more complex your environment's information technology systems become, the more privileged users you have. These include employees, contractors, remote, or even automated users. This does not diminish the need for small organizations from embracing PAM but rather that security professionals have a more difficult time scoping the problem and conducting mitigating exercises at larger scales. Every business and every consumer is potentially at risk from privileges being used as an attack vector. This fact alone necessitates the need for PAM everywhere even though only portions may be needed to mitigate relevant risks. Therefore, successful privilege as an attack vector strategy does not require all of PAM's disciplines to be implemented in order to mitigate the risk –only the ones that are relevant to

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_10

your business. However, it does go hand in hand that the larger and more complex the business, the more PAM use cases you will need to implement.

A successful PAM strategy offers a secure and workflow optimized method to authorize and monitor all privileged users for any and all resources. This will provide your business with the following capabilities:

- Grant privileges to users only for resources on which they are authorized.
- Grant access only when appropriate and revoke access when the need expires.
- Eliminate the need for privileged users to have or need knowledge of system passwords.
- Ensure all privilege activities can be associated to an account and when accounts are shared enforce mappings to an identity.
- Centrally and quickly manage access of all physical and virtual resources, on-premise or in the cloud, accommodating any set of heterogeneous resources that require privileged access.
- Create a sustainable audit trail for any privileged usage via session recordings, keystroke logging, and application monitoring.
- Empower organizations to readily respond to breaches by logging privileged activity that provide indicators of compromise.

When you consider these benefits of Privileged Access Management, the threat actor's ability to gain privileged access and navigate undetected is greatly diminished, mitigating the privilege as an attack vector dilemma. Figure 10-1 illustrates the workflow for this entire process when using a Privileged Password Management solution as a component of PAM.

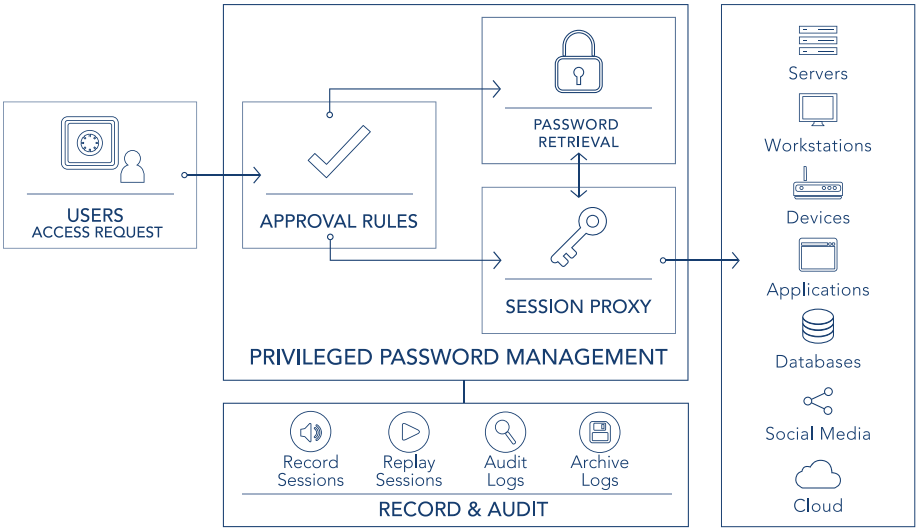

Figure 10-1. *PAM Access for Password Management*

PAM Challenges

As we undertake the challenge of managing privileges, we must be aware of some of the intrinsic problems without an efficient Privileged Access Management Strategy:

- **Lack of visibility and awareness** of all of the privileged accounts and credentials across an enterprise poses a monolithic challenge—especially for those companies that rely on manual processes and tools. Privileged accounts, many long forgotten, are sprawled across most organizations. Different teams may be separately managing—if managing at all—their own set of credentials, making it difficult to track all the passwords, let alone who has access to them and who uses them. An admin may have access to 100+ systems, possibly disposing them to take shortcuts in

maintaining the credentials. Beyond this, as elaborated in the sections below, some types of credentials are virtually impossible to find, let alone bring under management, without third-party tools.

- **Lack of privileged credential oversight and auditability**: Even if Information Technology (IT) successfully identifies all the privileged credentials strewn across the enterprise, this does not by default translate into knowing what specific activities are performed during a privileged session (i.e., the period during which elevated privileges are granted to an account, service, or process). Privileged access to a superuser account should not amount to ceding *carte blanche* to the user. Moreover, PCI, HIPAA, and other regulations require organizations to not just secure and protect data but be capable of proving the effectiveness of those measures. So, for both compliance and security reasons, IT needs visibility into the activities performed during the privileged session. Ideally, IT should also have the ability to seize control over a session should inappropriate use of the credentials occur. But, with potentially hundreds or concurrent privileged sessions running across an enterprise, how does IT expeditiously detect and halt malicious activity? While some applications and services (such as Active Directory), can log user actions, and while Windows servers using logon events within Event Log data can reveal some behavioral anomalies, expect full coverage of privileged account usage to require a third-party solution.

- **Sharing of privileged accounts for convenience**: IT teams commonly share root, Windows Administrator, and many other privileged passwords so workloads and duties can be seamlessly shared as needed. However, with multiple people sharing an account password, it may be impossible to trace actions performed with an account to a single individual, complicating auditing and accountability.
- **Hard-coded / embedded credentials**: Privileged credentials are needed to facilitate authentication for app-to-app (A2A) and application-to-database (A2D) communications and access. Applications, systems, and IoT devices, are commonly shipped, and often deployed, with embedded, default credentials that are easily guessable and pose formidable risk until they are brought under management. These privileged credentials are frequently stored in plain text – perhaps within a script, code, or a file. Unfortunately, there is no manual way to detect or centrally manage passwords stored within applications or scripts. Securing embedded passwords requires separating the password from the code so that when it's not in use, it's securely stored in a centralized password safe, as opposed to being constantly exposed as when in plain text.
- **SSH keys**: IT teams commonly rely on SSH keys to automate secure access to servers, bypassing the need to enter login credentials manually. SSH key sprawl presents a substantive risk for thousands of organizations, which may have upwards of a million SSH keys—many long dormant and forgotten, but still viable back doors for hackers to infiltrate critical

servers. SSH keys are standard, and more prevalent, in Unix and Linux environments, but are also used across Windows. Admins leverage SSH keys to manage operating systems, networks, file transfers, data tunneling, and more. As with other privileged credentials, SSH keys are not necessarily tied to a single user—multiple people may share the private key and passphrase to a server, which holds the public key. As with other types of privileged credentials, when organizations rely on manual processes, there is a pronounced tendency to reuse a passphrase across many SSH keys or to reuse the same public SSH key. This means that one compromised key can then be harnessed to infiltrate multiple servers.

- **Privileged credentials and the Cloud**: The challenges of visibility and auditability are generally exacerbated in cloud and virtualized environments. Cloud and virtualization administrator consoles (as with AWS, Office 365, etc.) provide vast superuser capabilities, enabling users to rapidly provision, configure, and delete servers at a massive scale. Within these consoles, users can spin-up and manage thousands of virtual machines (each with its own set of privileges and privileged accounts) with just a few clicks. One predicament then arises around how to onboard and manage all of these newly created privileged accounts and credentials. On top of this, cloud platforms frequently lack the native capability to audit user activity. And, even for those organizations that have implemented some degree of automation for their password management (either through in-house, or

third-party solutions), if not architected with the cloud in mind, there's no guarantee a password management solution will be able to adequately manage cloud credentials.

- **Third-party vendor accounts / remote access solutions**: Finally, another quandary for organizations is how to extend privileged access and credential management best practices to third-party users, such as consultants or other vendors that may perform a variety of activities. How do you ensure that the authorization provided via remote access or to a third party is appropriately used? How do you ensure that the third-party organization is not sharing credentials, or otherwise exercising poor password hygiene, such as by failing to terminate authorization credentials when an employee departs from the company?

Password Management

Password Management is a simple security function that helps a user store and organize passwords. Password storage solutions (commonly referred to as password managers, password safes, or password vaults) store passwords encrypted, requiring the user to create a master password. This assumes the solution is designed for direct end-user password management and potentially personal usage. The master password allows access to the password database or password keychain for retrieval of passwords for application usage.

Business-oriented password management takes these concepts to a different level. They add role-based access to the storage and retrieval of shared passwords, automatically rotate the passwords, provide API's for programmatic password access, and provide other enterprise auditing,

encryption, and, logging capabilities for multiple users and applications across an entire enterprise. These features cover everything from session recordings to password attestation reporting. These capabilities are necessary to mitigate privileged threats but also to demonstrate regulatory compliance.

Password Management solutions can also be implemented in a wide variety of formats based on an organization's needs. This can include software, appliances, virtual instances, or even hosted in the cloud. Regardless of the deployment philosophy, the goal is still the same: secure privileged account passwords, and most importantly, make sure the password manager itself does not become a liability to the business. For example, decrypting passwords and unrestricted access to the password manager's database itself, at any time, would be like finding the Rosetta stone for access to any resource managed by the business. Organizations are willing to trade off the risk of storing all of their sensitive passwords in one highly secure fault-tolerant location versus the threats posed by unmanaged privileged access. Businesses just need to be aware of the tier one critical system nature of a password manager and the policies and procedures necessary to facilitate its successful fault-tolerant deployment.

Least Privileged Management

The concept of least privilege has its foundation roots in mainframe security. Any user when first instantiated has absolutely no privileges to do anything. It is considered a fully closed security model. As a user needs to perform functions, privileges are added to their account to perform specific tasks. Hopefully, the permissions are the bare minimum required to perform the specific task, and nothing more that could lead to privileged abuse.

Least privilege on every other platform operates the same way regardless if it is Unix, Linux, Windows, or MacOS. Unfortunately the default model for Windows and MacOS is the opposite; default initial users are administrators. To facilitate least privilege, new or existing users are assigned

basic (reduced) login rights and the applications, tasks, and even operating system functions are granted on an as-needed basis. The basic account assigned in this model is considered a Standard User. The basic user rights allow for interaction with the operating system, limited applications, but not perform any changes that could be a liability for the environment.

The problem with this model is that many tasks, applications, and configurations need higher permissions than standard user, including administrator or root. Traditionally, users have been granted a secondary account as an administrator to perform these tasks, but that introduces privileged attack vector risk.

In a least privilege model, technology provides a solution. Via policy and rules, individual commands, applications, and operating system functions are granted the permissions they need to operate and nothing more. They have least privileged rights. The users are not granted the rights; this is critical in mitigating privilege risks that could breach the user's runtime. Only the application is elevated based on administrator-specified criteria. Thus, the application runs correctly, a user can interact with it, and excessive privileges are removed to prevent a threat actor for leveraging them.

Application to Application Privilege Automation

Application to Application (A2A) automation utilizes an Application Programming Interface (API) that allows stored credentials to be managed automatically from an on-premise or cloud-based implementation. If you are a commercial application developer or create custom applications for your business, the primary benefit allows applications to authenticate without an end user intervening or hard-coding credentials in a script, compiled code, or a file. Team members, like database administrators, never need administrator rights to access a database if the tools retrieve

stored credentials automatically. Applications can make database connections, communicate with other applications and instances, and perform their own functions with the current password without manual intervention as well.

Organizations and application developers will realize multiple benefits in using a Privileged Access Management API to secure credentials from a threat actor:

- Secure credential management: Instead of entering static credentials, developers call on a PAM API to retrieve the latest credentials for the user, application, infrastructure, cloud solution, or database to authenticate and then release the credentials at the end of the session. This triggers automatic, randomized cycling of the password. The end user is never exposed to the username or password. All authentication is performed silently behind the scenes with complete activity auditing if desired.
- Simplified developer access : Improve the agility and responsiveness of IT by never requiring the entry of a username and password for connectivity to create custom applications. End users, like database administrators, never need administrator credentials to access a database if the tools retrieve stored credentials automatically. Management tools for services, remote access, and infrastructure automatically recognize the logged-on user and the asset they are on, and seamlessly request and pass credentials for the application.
- Protection from password reuse attacks: Since credentials can be passed within the application itself, directly from the API, IT can secure runtime and avoid

hacking techniques like pass-the-hash and keystroke logging, making this approach far more secure than traditional single sign-on (SSO) technology.

- Programming flexibility: To enable developers to access the API and help secure their applications, PAM vendors offer samples and support for a wide variety of programming languages including C# (.NET), PowerShell, Ruby, Python, Java, and Bash shell.

The end result eliminates the need for static passwords and secures applications in the cloud or on-premise with the latest password (or key) for their current runtime. Common API functions include these:

- The retrieval of the current password for an asset or application.
- Force the rotation of a password change.
- Register a resource for password management including the technology owning the account (operating system, database, application, cloud resource, social media, etc.).
- Automate policy and criteria for password management including retrieval.
- Access session monitoring details.
- Define groups of users and resources for simplified management.

SSH Key Management

Enterprise IT, which often consists of from dozens to thousands of Unix servers and only a handful of Unix admins to manage them, rely on SSH keys to help them do their jobs efficiently. For what they offer in terms of

convenient access, SSH keys can also pose security risks that are like those of shared accounts:

1. SSH keys are tied to accounts on a Unix server, not to an individual. What happens when you need to prove that a specific user accessed a server using SSH keys for an audit?
2. Replacing and managing SSH keys require manual effort. As they're used on Unix servers, and there are typically a handful of Unix administrators, it can be easy to 'set it and forget it.' The big operational risk here is obvious – the older the key, the more it is shared, the greater the chance of unauthorized access and a breach.
3. As a result of risk #2, managing and rotating SSH keys manually typically results in IT teams reusing the same passphrase for different SSH keys. As a result, IT teams are unwittingly putting their enterprise security at risk – if the passphrase falls into the wrong hands, a threat actor has a way to move laterally through your environment.

Like passwords, organizations should automate the life cycle of SSH keys from discovery, to onboarding, rotating, distributing, managing; and finally destroying them.

Directory Bridging

Applications and operating systems can have local role-based access security models or integrate into directory services like Active Directory (AD) or LDAP. Unfortunately, many operating systems do not natively allow cross-directory authentication from *nix platforms to Microsoft

Windows. This means that a user account on Windows cannot be used to authenticate against Unix and Linux and that an Alias account needs to be created to provide authentication.

When dealing with complex environments, this can lead to thousands of accounts, across thousands of systems, all potentially slightly different aliases for the same user. This represents a management nightmare, a password headache, and an auditing disaster to link aliases with a single physical human user –their actual identity.

The solution is directory bridging. It provides a standard based solution for a non-Windows operating system to authenticate users based on accounts created in Active Directory. Therefore, the same account they use to log on to Windows can be used with the same password to authenticate against Unix, Linux, and MacOS. For a management perspective, you achieve the following benefits:

- A single account for all users regardless of platform with the same credentials or multifactor requirements.
- Minimizing the need for alias accounts, their management, and correlation of user accounts.
- Simplified attestation reporting for any single user across all platforms.
- Simplified account discovery and identity management for non-Windows platforms via Active Directory.

Directory Bridging is such a basic function with so many benefits; it can help minimize insider threats to rogue account usage, simply by eliminating all the additional accounts created for users on non-Windows systems. A threat actor has few account back-door options since all the aliases have been eliminated and they are forced to attack accounts that are used daily and potentially monitored. When this is combined with data analytics, user behavior analysis, and good old-fashioned logging, finding malicious activity is much easier.

Auditing and Reporting

Without the ability to audit changes, report on events and findings, and provide an actionable trail of activity, Privileged Access Management projects only succeed in mitigating privileged attack vector risks. While that is a huge accomplishment, it does nothing to help document regulatory compliance to auditors or identify intentional or unintentional mistakes that could lead to a data breach.

Therefore, in order to have a successful PAM deployment, consider components that help document the changes and processes along the way. These include the following:

- Documenting changes to directory services like Active Directory that can affect the entire runtime of your PAM initiative.
- File Integrity Monitoring (FIM) across all your operating systems to identify unauthorized privileged changes to sensitive operating system and application files.
- Document changes in key applications like Microsoft Exchange or SQL Server that could be used by a threat actor for surveillance or the exfiltration of data.
- Monitoring of event logs for critical events that could indicate potential abuse of privileges.
- Session monitoring of all interactive sessions, keystroke logging, and application monitoring.

Once these concepts are implemented, demonstrating privileged access management as a function of compliance becomes rather elementary. The output from reports, command filtering, privileged session review, etc., all become the components to document compliance

and more importantly provide the security needed to stop privileges from being used as an attack vector.

Privilege Threat Analytics

While reducing permissions and embracing the concept of least privilege will reduce both the attack surface and potential impact of a breach, these users will, at some point, require elevated access to perform their normal job functions. It is these accounts that pose a significant risk to organizations. These accounts have been authorized to perform certain tasks and to access certain data repositories. The control and detailed auditing of these accounts fall outside of the scope of typical Identity Management and User Provisioning solutions. How does one determine when an approved account is misusing their given permissions, or if these accounts have indeed been compromised? For this, we need to start at the bottom and work our way up.

One of the strangest words in the English language is datum. It is, by definition, the singular form of data, but is rarely used in conversation or written documentation. It generally refers to a single point of information or a fixed starting point of a scale or operation. When we review security or debugging information, we often refer to single entries in a log as data when it should be correctly referred to as datum. While the term may be considered obsolete when it comes to security, there are many times we make critical decisions on datum and not data. This is where discussions on analytics and user behavior become important. It would be a mistake to base a decision on user behavior strictly on datum. Analytics and user behavior require data.

Any analytics solution that makes a recommendation based on a single piece of information is more in tune with an event monitoring solution, or security information event manager than an analytics engine. For example, a single event based on user, time, date, and location is

not analytics – it's datum. That information correlated with other event datum, and processed via correlation is not analytics either. That is just a correlation engine reviewing multiple events in a logical order. This technology has been around for decades.

Therefore, if the events are unique, processed via machine learning, cluster analysis, adaptive correlation engines, etc., then we could potentially have analytics. It takes more than just a single event and event matching to create analytics based on variable event data. Being mindful of the analytics claim and data absorption model is key in understanding whether an analytics solution can really help you detect and resolve security anomalies.

A good threat actor attempts to erase or eliminate any traces of their movement, surveillance, or actions within an organization. The primary point of privilege as an attack vector is to document any time the threat actor tries or has access to privileged accounts. This produces data of their activities based on unusual behavior, and using data analytics provides a mathematical automation engine to detect even the best threat actors as they infiltrate an environment.

The current approach and trend in the market is to implement advanced threat and behavior analytics to identify suspect behavior for these accounts. However, many of these solutions require significant historical analysis, are not trusted given their "black box" approach, and only analyze high-level data elements such as logs or data forwarded to a SEIM. Furthermore, these solutions are focused on identification and not containment. This is an area in which integrated PAM capabilities can provide significant benefit. PAM is an inline solution that can grant or deny access for sensitive access. PAM is not restricted to rigid all or nothing access policies, but can rather dynamically adjust access policies, approval work to sensitive systems, applications, and data. This is an area that organizations and security professionals should continue to monitor. Vendors like McAfee have developed new standards such as OpenDXL to automate the response based on any correlated events and will close the loop with automation based on Threat Analytics results.

CHAPTER 11

PAM Architecture

Privileged Access Management (PAM) provides an automated password and session management solution that provides secure access control, auditing, alerting, and recording for any privileged account. The technology is designed to manage a local or domain shared administrator account; a user's personal admin account; service, operating system, network device, database (A2DB), and application (A2A) accounts; and even SSH keys, cloud, and social media. By improving the accountability and control over privileged passwords, IT organizations can minimize privileged threats and achieve compliance objectives.

However, the deployment of this technology depends on all the use cases listed above and the presence of the resources on premise, virtual, or in the cloud. In addition, environments need to consider high availability, disaster recovery, break glass, and time to recover once a fault occurs in the solution itself or any component in the supporting infrastructure from networks to Internet connectivity that could cause an outage.

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_11

Therefore, many different configurations need to be supported to scale from single site installations to multisite, geographically dispersed environments. These include the following:

- Active/Active - Sometimes called multi-active, this deployment type allows multiple nodes (distributed heads) to be active at one time. Each node is connected directly to the database.
    - Advantages
        - Unlimited scalability
        - Redundancy of components
        - Targeted password change events for specific locations
    - Disadvantages
        - Requires an external database
        - Redundant database configurations such as SQL Always On can be costly and require dedicated staff for administration. And, open source database solutions may not be suitable for a tier I application of this nature.
        - It is the responsibility of the customer to ensure that the database and supporting servers are securely hardened
- Active/Passive - Two installations are required for active/passive. The internal databases are replicated, and a heartbeat sent from the primary indicates to the secondary if it should take over operations.

    - Advantages
        - Easy to set up
        - All high availability is incorporated within the solution
    - Disadvantages
        - An external load balancer is required for auto-switching users to the active appliance
        - The failover process is not instantaneous and can take time to initiate
        - Cold Spare versions can have databases that are out of sync or in a split brain configuration if their age from initial backup is too large.
- Third Party Failover - For deployments where only one installation is desired, virtualization technology can be used to keep the installation continuously available via replication, even if the physical server running the instance goes offline for any reason.
    - Advantages
        - Cost-effective high availability with a single instance
        - Provides high availability and continuous operation during host server outages
    - Disadvantages
        - Relies on virtual replication technology to be licensed, set up, and configured correctly
        - Does not provide redundancy in the event of a software failure

Regardless of the selection for PAM availability and fault tolerance, the model needs to be adjusted depending on the deployment location; consider if a hybrid model is required as well. These will be addressed for PAM as on-premise deployment, cloud, Infrastructure as a Service (Iaas), and Software as a Server (SaaS). To that end, consider the PAM Maturity Model contained in Table 11-1. It will help you understand your journey in implementing PAM throughout your organization.

Table 11-1. *PAM Maturity Model*

The Privilege Maturity Model	Level 1 Absent	Level 2 Adhoc	Level 3 Standardized	Level 4 Managed	Level 5 Advanced
Shared Accounts	• Limited controls • No shared account password mgmt. • Lack of accountability	• Manual Controls & processes • Paper trail is not reliable	• Automated discovery, inventory & onboarding • Centralized Password Mgmt. with workflow approval and automated rotation • Privileged account usage reporting	• Password-less session access & management • Context-aware privileged access using RBAC and MFA	• Identity integrated (IAM, SSO, AD-Bridge, AD Audit & Recovery) • Advanced coverage (Cloud, SaaS, Apps) • HSM • User behavior analysis
Application & Service Accounts	• Unknown & Unmanaged	• Documented • Hard coded • Rarely changed, if ever	• Targeted AtoA management • Eliminated targeted hard coded passwords • API driven retrieval	• Centralized AtoA management • No hard coded passwords; ever	• DevOps Integrated • High Volume • HA and Caching for redundancy

(*continued*)

Table 11-1. (*continued*)

The Privilege Maturity Model	Level 1 Absent	Level 2 Adhoc	Level 3 Standardized	Level 4 Managed	Level 5 Advanced
Active Monitoring & Threat Detection	• No monitoring	• Distributed logs • Lack of tracking individuals use of shared accounts	• Centralized audit controls • Individual accountability over use of shared accounts • Deep Visibility with session and keystroke	• Advanced threat detection & UBA • SIEM integration • Automated keyword and activity indexing.	• Automated Privilege Active Response (Deny, Disable, Quarantine, Alert) • IGA integration • Platform independent

(*continued*)

Table 11-1. (*continued*)

The Privilege Maturity Model	Level 1 Absent	Level 2 Adhoc	Level 3 Standardized	Level 4 Managed	Level 5 Advanced
Fine-grained Desktop Management	• Unmanaged Users have Admin access	• Remove some administrator rights • Desktop tools for ad-hoc elevation	• Centralized Password Management • Limited whitelist \ blacklist proxy access • Reputation services	• Fine grained access • Controlled remote server sessions • FIM • Control latteral movement	• Context-aware access policy (user risk, asset risk, ITSM validation, MFA) • IGA integration with separation of duties • Desktop Asset and user policy independence

(*continued*)

Table 11-1. (*continued*)

The Privilege Maturity Model	Level 1 Absent	Level 2 Adhoc	Level 3 Standardized	Level 4 Managed	Level 5 Advanced
Fine-grained Server Management	• Unmanaged. Users have Root access	• Siloed • Open Source (SUDO)	• Centralized Password Management • Limited whitelist \ blacklist proxy access • Platform dependant	• Fine grained access • Privileged Shell • Controlled remote server sessions • FIM • Control latteral movement	• Context-aware access policy (user risk, asset risk, ITSM validation, MFA) • IGA integration with separation of duties • Server Asset and user policy independence
Fine-grained Infrastructure Management	• Unmanaged. Users have Root access	• Siloed • Vendor dependant	• Centralized Password Management • Limited whitelist \ blacklist proxy access	• Fine grained access • Controlled remote server sessions • Control latteral movement	• Context-aware access policy (user risk, asset risk, ITSM validation, MFA) • IGA integration with separation of duties

On-Premise

On-premise deployments of privileged access management operate within the confines of an organization's firewalled perimeter but can manage resources that allow outbound connectivity to the cloud from a data center. Essentially, using one of the paradigms above, software, appliance, or virtual appliances are deployed within the corporate data center to meet business objectives. The implementation can be air gapped (no Internet access) but must have a logical network route to target systems, or through remote management nodes, to conduct password changes remotely or through agent technologies.

This architecture is very similar to an on-premise email solution or anti-virus system with centralized management. The primary difference is that the PAM manager needs to resolve hostnames and route to each managed object for password changes, and each node needs to be able to resolve the server and provide a network route for any agent technology that may be a part of the PAM deployment.

If the network has stability issues with DNS, NTP, AD replication, routing, or performance, the integrity of any PAM deployment can be an issue. A well-architected and stable network is required since PAM relies on the infrastructure to onboard, manage, and change passwords efficiently with session monitoring.

For a threat actor, a poor infrastructure is a perfect place to get lost in the noise. Errors from DNS, AD replication, through poorly managed logs can conceal their identity even with a PAM deployment. Think of Waldo if he can hide behind infrastructure errors that would normally not be present in a properly functioning environment. Errors should be the exception and layering on security technology when the environment has poor cybersecurity hygiene will not make the infrastructure safer.

Cloud

Cloud-based deployments of Privileged Access Management can take on several different forms:

- Cloud to cloud for privileged management including application to application (IaaS).
- Cloud-based privileged storage and management for users (SaaS).
- Privilege management for on-premise resources (Hybrid).

If this was a multiple-choice question, your strategic business initiatives might require more than one of these categories. It is highly uncommon for privileged access management to be used in only one silo of the business without any plans to expand the technology to all sensitive systems and privileged accounts. While initial deployments may start out small, the cloud may be needed later for management everywhere. This is critical when selecting PAM on-premise, in the cloud, or a hybrid approach. For hybrid approaches, they can be a combination of IaaS, SaaS, or on-premise or a combination using remote management nodes to route and aggregate data securely.

Infrastructure as a Service (IaaS)

Whether your organization chooses to operate within a single cloud provider, multiple vendors, or has geographical requirements-based regulations or business model, cloud environments need to authenticate applications and users like any other information technology implementation. Cloud to Cloud privileged access management has unique requirements compared to an on-premise implementation:

- High-availability architectures may warrant additional cloud instances to provide high availability in case of a cloud or infrastructure outage that is out of the end user's control.
- Regulations may require separate but duplicate instances and filter data based on region or local laws.
- Environments may have public and private IP ranges to provide the required services and require special provisions to secure them.
- Vulnerability management due to public services takes a higher provider to mitigate threats.
- API access requires special attention for secure access and to limit exposure.
- Sensitive data in the cloud, such as passwords, requires additional database security such as HSM to protect information.

For organizations looking to perform PAM only in the cloud, there are multiple technology vehicles to implement a solution. The most common is to use black box technology based on PAM solutions hosted in cloud marketplaces (Amazon AWS, Microsoft Azure, Google Cloud, Oracle Cloud, or third party managed service provider). These allow for hardened PAM deployments based on a variety of licensing models and cloud runtime costs. Some PAM vendors also offer solutions that can be instantiated as a software implementation in a cloud operating system template. These provide the most flexibility for a client, but security, hardening, and operating system configuration are the responsibility of the client - not the cloud provider or PAM vendor. The risks are higher for these types of implementations due to any internal laxes the environment may have in basic cybersecurity hygiene but can be highly customized to meet unique requirements.

Software as a Service (SaaS)

Privileged Access Management solutions deployed as a SaaS solution can operate solely in the cloud or require on-premise management nodes to route and aggregate policy and events. These implementations are completely managed by the PAM vendor and share cloud resources with other PAM clients in the vendor's multi-tenant installation. While there are currently very few PAM solutions in the cloud using SaaS, the trend suggests businesses are gaining confidence of storing passwords, policies, and management tools for PAM in the cloud. This trend is being led by individual vendors and Managed Service Providers (MSP) that are providing cost-effective services based on commercial PAM offerings with little to no expertise needed by end users.

CHAPTER 12

Break Glass

Break glass is a term used in computing to describe the act of checking out a system account password for use by a human user when an emergency situation arises, and traditional access methods have failed. The term derives from the act of breaking the glass on a fire alarm.

Access controls in an application or asset can be bypassed during a critical emergency by using break glass. A user performs a break glass checkout or release of the account and password (credentials) when he or she needs immediate access, even if the user is not authorized to manage the system. This method is customarily used for the highest-level system accounts, such as root accounts for Unix and Linux, SYS or SA for a database, or administrator for Windows (local or domain). These highly privileged accounts are not usually assigned to a specific person, so instead, break glass limits their utilization with various controls to reduce risk and enable only specific tasks. However, it is obvious that user access to break glass credentials is still restricted and not as accessible as a fire alarm.

Break glass scenarios can be caused by network outage, application fault, or natural disaster that disrupts the normal availability of your privileged access management solution. Therefore, factors like power source and network connectivity should be considered when designing your break glass policy. Also, a threat actor may also consider your break glass process a target since it does contain credentials to the crown jewels. Access and monitoring to credentials used in break glass access should be strictly monitored at all times.

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_12

Break glass scenarios are usually considered when information technology administrators are deploying critical infrastructure to secure system access. Here are three common break glass scenarios applicable to most organizations:

1. Requirement for emergency, direct access to managed systems using a password as an enabler.
2. Getting access outside of the standard operating process because mission-critical systems are down, or a required approver is unavailable.
3. Retrieving passwords or secrets from a physical safe or other offline backup on a physical device, such as USB drive/CD.

Break Glass Process

When developing a break glass policy, there are a few important considerations and potential processes to implement:

- For authorized break glass users (new or existing), consider creating pre-staged emergency user accounts that are managed and distributed in a way that can make them quickly available without administrative delay but have the appropriate restrictions from a threat actor. The break glass accounts and distribution procedures should be documented and tested as part of implementation, and carefully managed to provide timely access when needed. These can be stored in the password manager or a secure physical location, and have paper counterparts stored in another media or highly secure environment.

- To comply with auditing requirements, even if an approval is bypassed, the system should still fully log who has access and what actions were performed. Additionally, IT administrators should review the logs to ensure compliance with change management processes when a break glass process is used.
- Break glass processes that are implemented outside of the password management technology, such as a physical safe and storage of printed passwords, should be routinely updated and manually tested for effectiveness and change control. Only select users should have access to the combination or keys to the physical safe, and they should be treated like any other sensitive information within the organization.

Break Glass Using a Password Manager

Information technology (IT) organizations often utilize a password manager as a break glass solution to provide access to their environment when the established processes for log in or authentication fail. IT teams might authenticate with LDAP, AD, or multifactor, and the user would log in prior to using sudo or a least privilege solution to gain limited administrative privileges. When this method fails, the break glass process would require IT to provide a password for an account within established parameters (time frame, privileges, scope, etc.) to access the application or system.

During normal operation, users who need access to privileged passwords will access the tool to retrieve a password or establish a session so that they can perform whatever tasks or operations are assigned to their roles. This requires that the password management solution have the rights to fully manage, rotate, and keep the password current. Relying on

end users to diligently remember, rotate, and securely document all their passwords is invariably less reliable and riskier.

When using a password manager, consider these break glass use cases:

1. The person who needs a managed password cannot log in to the solution
    a. Repair user access to the password manager
    b. Reset the managed credentials
    c. Reset the password for the user accessing the solution
2. Fault authenticating to the password management solution
    a. Repair network connectivity for critical paths
    b. Restore password management connectivity to critical authentication services
    c. Repair authentication system
    d. Store a printed out copy of the passwords in a highly secure location
3. The password management solution is not available
    a. Repair network connectivity
    b. Access solution through fault-tolerant node
4. Managed passwords are invalid
    a. Refresh the password by using the solution to generate a new one automatically
    b. Use the password history feature of the password manager to determine the last valid password

5. Connectivity anomaly
    a. When critical services are not functioning, access may be required via iDrac, management networks, or crash carts
    b. When network connectivity does not allow access, lateral connectivity, not subject to segmentation, can provide break glass access
6. Processes and workflow prevent access
    a. No approver is available in the time period required
    b. User access is restricted due to system ownership, such as employee role, contractor, or vendor
    c. Time-of-day constraints or critical event requires immediate unrestricted access

Session Management

For a non Break Glass use case, the enterprise password management solution enforces connectivity through the session manager to document activity and enforce segmentation. By design, there is no alternate way to connect to the target network and system without first accessing the session manager. Break Glass has a requirement not to enforce this due to some form of outage. One option for achieving break glass access would be to drop security controls in order to restore availability. However, as with all risk-based decisions, it is important to review and document the risks and benefits, and get organizational alignment. This is true for any access granted outside of normal operating procedures. As a potential alternative, management networks controlling iDrac access or terminal servers

may provide a safer, alternate approach than reducing security controls in a break glass scenario, especially if the event is potentially security related. Access to management networks can therefore be monitored independently to provide similar controls and security assurances. Access to a break glass scenario should therefore include the following two ways to access the session manager in the event of an outage:

1. Controlling third-party access to managed systems
    a. Open alternate access into the environment via backup connections
    b. Disable session management access to the primary systems (not recommended)
2. Access session management in an alternate data center
    a. Open network path around the session management device (not recommended)
    b. Access session management device in an alternate data center or disaster recovery environment
    c. Operate session management independently for management networks to provide access

Stale Passwords

There are many situations where a password stored in the password manager may be stale through no fault of the technology. Such cases could arise due to restoration of backup images, rollback of virtual snapshots, or even the deployment of a new instance or system based on a template. In these use cases, the break glass password manager has automated the

rotation of passwords of human, service, or built-in accounts throughout the environment. Consequently, no one knows the correct password, and the password is not written down for manual retrieval. During normal operation, password managers will randomize and change the passwords, update managed systems, and store and test the new password.

So, what do you do when this process fails? Here are some recommendations:

1. If the tool cannot change a single or small number of passwords:
    a. Repair connectivity or retool the configuration of the system to make password changes based on the uniqueness of the targets
    b. Change password by hand using another account that has privilege. Most password management tools have an account assigned to perform such operational tasks, typically called the “Functional Account”.
2. If the tool cannot change any passwords:
    a. Repair network connectivity or system access
    b. Verify Functional Accounts have proper privileges to manage passwords remotely
3. If the password of a built-in account is not known:
    a. Randomize the password of the built-in account using the Functional Account
    b. Repair system by booting to single user mode and change password

4. If the password of a service account is not known, so a service will no longer start:
    a. Randomize the password of the service using the Functional Account
    b. Establish a privileged connection to the system using a stored credential and manually set the service account password before automating password management

Application-to-Application Passwords

In these use cases, IT administrators or developers have implemented a password manager to forgo hard-coded passwords in configuration files, scripts, or compiled applications. Instead, the application, script, or configuration file accesses the password manager via an Application Programming Interface (API) to retrieve the current password it needs to complete the processing operation. The application can potentially cache the password for continuous use, or release the password when it is complete. To do so, the environment must allow for password changes while applications are running. IT administrators must know the process for rotating and refreshing passwords midcycle. Here are some recommended steps:

1. If automation jobs develop a fault:
    a. Repair the password management solution
    b. Enable fault tolerance for the API

   c. Add caching to the scripts, configuration, or application to be fault tolerant for a network, connectivity, or password management outage

   d. Manually update jobs and resubmit; ensure that all dependencies have been met.

2. If automation jobs require change control for password changes:

   a. Schedule password changes during maintenance windows

   b. Develop applications that are fault tolerant or can be resumed in the event of an API query failure for any reason

Physical Password Storage

Your recovery plans should also include the ultimate break glass solution — retrieving physical copies of passwords. There are inherent risks with storing physical copies of privileged passwords. However, with the proper physical controls in place to securely store the credentials, physical storage of paper can serve as an option in break glass scenarios.

Recommendations for this use case include the following:

- Create a plain text copy of the credentials and automatically print them in a secure location or store them on reliable removable media. Regardless of the format, paper, or offline digital removable media, ensure that final storage is highly secure.

- If your processes require, re-encrypt the digital media with an offline encryption package before writing to a USB drive or CD. Remember to back up the password for the offline encryption in a secure location as well.
- Fully document the process for creating and storing break glass passwords. Passwords should be rotated and restored on a regular basis.

As with any disaster recovery process, the paper or removable media process must be tested periodically to ensure its reliability.

Context Aware

Credentials that must be accessed outside the organization can be challenging to lock down. To get it right, you need to apply ***context*** to the access, and all the runtime parameters of the request must be evaluated to enforce appropriate access. This will help mitigate the risk from an external threat actor attempting to compromise these credentials

- *Who is trying to log on?*
- *What system are they trying to access?*
- *Where are they logging in?*
- *What day of the week is it?*
- *What is the time of day?*

Applying context allows you to incorporate privileged access management best practices to better protect your organization from a breach. For example, if your break glass account is strictly for emergency use, only make it available during off hours. If it is expected that the account would be accessed via a remote employee working from home, verify that the request is coming in via VPN.

Architecture

If any component of a break glass process or password management system itself becomes unavailable (natural disaster or outage), multiple levels of redundancy mitigate the risk of data loss or degradation of access capabilities. Flexible high-availability deployment architectures ensure that passwords remain available whether everything is installed in a single data center, or across multiple geographic locations. This is traditionally the top priority of an architecture and defense before utilizing a break glass process. Physical copies of credentials should also be considered for disaster recovery locations as well.

Finally, for short-term outages of the entire on-premise infrastructure, passwords may be stored and retrieved via cloud environments. These would need to be configured to cache or replicate the information off premise and secured against external threats.

Break Glass Recovery

After a break glass event, the recovery to normal operations should consider a few security and operational events. While these may seem esoteric, the purpose of break glass process is to provide access in a worst-case scenario. If restoration is provided too quickly, or change control and checks and balances not verified, the break glass process could be used against the organization in a future attack or just lead to another similar event in the future. Therefore, consider the following before restoring normal services:

- What event occurred requiring the break glass process?
- Can this event be avoided in the future?
- Was the access to break glass credentials appropriate?

- Where there any resources in the break glass process that did not have coverage?
- Who was notified of execution of the break glass process?
- Was any additional risk (data loss, resource exposure, etc.) introduced by the process?

If these questions can be answered satisfactorily services can be resolved to normal operations. After they are, continue with the following queries:

- Was the restoration process of services accurate after a break glass event? If not, how can it be improved or fixed?
- Where all electronic credentials and passwords reset after the break glass event?
- Was all physical storage of credentials reinstated and codes to physical storage reset?
- Was all break glass session activity verified and audited for no inappropriate activity?

If break glass scenarios repeatedly occur, then the entire process should be evaluated in order to prevent their invocation in the first place. This could be anything from faulty hardware, network anomalies, to the unavailability of key personnel in a critical-need situation. The restoration of normal services should always include the complete postmortem of the break glass key event.

Break glass scenarios should be considered for any sensitive privileged account, even in the event of the stakeholder's death. Using the technology to support itself and physical access as a backup ensures that the controls recommended do not become a liability to the organization or a gold mine for a threat actor.

CHAPTER 13

Industrial Control Systems (ICS)

Critical Infrastructure systems that span manufacturing, transportation, water supply, and energy all depend heavily on information systems for their monitoring and control. Historically ICSs relied heavily on physical separation as the primary means for security. However, modern control system architectures, management processes, and cost control measures have resulted in increased integration of corporate and ICS environments. While these interconnections increase operational visibility and flexible control, it can also increase risks that previously did not occur with isolated ICS. Though an interconnected network, the ICS system can be exposed to threat actors that have already exploited and compromised on the Internet and corporate networking, or by insiders misusing their privileges. ICS-CERT[1] (Industrial Control Systems Cyber Emergency Response Team) provides ICS-CERT alerts[2] to assist owners and operators in monitoring threats and actions that could impact ICS systems.

[1] https://ics-cert.us-cert.gov/

[2] https://ics-cert.us-cert.gov/alerts

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_13

To address these risks ICS-CERT (Industrial Control Systems Cyber Emergency Response Team) encourages sound security practices using "defense-in-depth principles including but not limited to the following defensive in-depth measures and managing privileges in Table 13-1.

Table 13-1. *ICS Risk Matrix*

Risk Vector	ICS-CERT Recommendation	Privileged Access Management (PAM)
Secure Passwords	Remove, disable, or rename any default system accounts wherever possible,	Implementing an enterprise password management solution that supports enterprise password management, password rotation, active session management, and session recording is an effective method to eliminate many of these common challenges.
Strong Password Management	Establish and implement policies requiring the use of strong passwords.	Implement an automated password and privileged session management solution offering secure access control, auditing, alerting, and recording for any privileged account.
Reduce Risks of Brute force Attacks	Implement account lockout policies to reduce the risk from brute forcing attempts.	PAM strengthens the security of ICS and interconnected environments by: 1. Ensuring no device has a default password, 2. Guaranteeing each device has a unique complex password, 3. Automatically rotating passwords based on age and usage, 4. Limiting administrative access and communications.

(*continued*)

Table 13-1. (*continued*)

Risk Vector	ICS-CERT Recommendation	Privileged Access Management (PAM)
Minimize Network Exposure	This activity includes the implementation of firewalls and network segmentation. This can reduce the attack surface for bad actors and reduce the risks of lateral movement within a compromised environment.	Implement a PAM solution that can also be deployed as a secured enclave model to ensure all privileged accounts (employees, contractors, and third parties) do not have direct access to manage these devices. This model ensures that only approved devices and restricted network paths can be used to communicate with secured resources, which would include control system HMI computers (Human-Machine Interfaces). Using this best practice model for securing sensitive servers and networking devices ensures that all administrative activities are proxied through the management server to ensure that each session is approved, tied to a specific individual, is properly audited, and that passwords are automatically rotated after each session is complete. See the diagram below for a representation of the enclave model.

(*continued*)

Table 13-1. (*continued*)

Risk Vector	ICS-CERT Recommendation	Privileged Access Management (PAM)
Secure Remote Access	This activity includes deployment and appropriately updating remote access solutions, such as VPN, if required.	ICS Cert recognizes that remote access solutions such as a VPN is only as secure as the connected devices. PAM solutions can bulletproof your remote access infrastructure with complete control and audit access to privileged accounts such as shared administrative accounts, application accounts, local administrative accounts, service accounts, database accounts, cloud and social media accounts, devices, and SSH keys. Enabling Secure Remote Management: 1. Vendors should access ICS resources using PAM and existing remote access facilities. 2. Vendors authenticate via PAM and request a session to managed resources, which can include a system running ICS control software. Note that this session cannot only be restricted to a specific system but can also be restricted to a specific control system application, further reducing the risks of compromise and lateral movement.
Third-Party Vendors	Monitor the creation of administrator level accounts by third-party vendors.	

(*continued*)

Table 13-1. (*continued*)

Risk Vector	ICS-CERT Recommendation	Privileged Access Management (PAM)
		3. Vendor uses a native remote desktop tool (tool (MSTSC/PuTTY etc.) or an RDP/SSH session, which is proxied through PAM for session monitoring. 4. All vendor activities are logged and optionally recorded to comply with security and compliance policies.
Vulnerability Management	Apply patches in the ICS environment, when possible, to mitigate known vulnerabilities.	A vulnerability management process can proactively identify security exposures, analyze business impact, and plan to conduct remediation across network, web, mobile, cloud, virtual, and IoT infrastructure. 1. **Discover** network, web, mobile, cloud, virtual, IOT infrastructure; 2. **Profile asset configuration and risk potential;** 3. **Pinpoint** vulnerabilities, malware, and attacks; 4. **Analyze** threat potential, return on remediation, and more; 5. **Isolate** high-risk assets through advanced threat analytics; 6. **Remediate** vulnerabilities including default and weak passwords;

(*continued*)

Table 13-1. (*continued*)

Risk Vector	ICS-CERT Recommendation	Privileged Access Management (PAM)
		7. **Report** on vulnerabilities, compliance, benchmarks, etc.; 8. **Protect** approved and shadow devices from attack.
Threat Detection	ICS-Cert recommends that organizations monitor for suspect activities and to report their findings to ICS-CERT for incident response support and correlation with other similar incidents.	User behavior and risk analysis enable information technology and security professionals to identify the potential breaches and the seeds from incidents. Security Information Event Managers (SIEM)s and Threat Analytic solutions can set baselines for normal behavior, observes changes, and identifies anomalies that signal critical threats via the following steps: 1. Aggregate users and asset data to centrally baseline and track behavior; 2. Correlate diverse asset, user, and threat activity to reveal critical risks; 3. Measure normal behavior in asset and user changes to flag in-progress threats; 4. Isolate users and assets exhibiting deviant behavior; 5. Generate reports to inform and align security decisions. Any threat detection deployed by an organization must consider all the available security data and correlate the results. Threat detection should not rely on only one event and source.

While ICS represents a specific vertical targeted by PAM technology, the benefits for any implementation are easy to recognize:

- Discover all managed and unmanaged devices across your interconnected corporate and ICS infrastructure.
- Automatically discover and inventory privileged accounts used by third-party vendors.
- Provide central control by securely storing all passwords and SSH keys in a secure database.
- Reduce the risk of lost or stolen vendor credentials by systematically rotating passwords for all managed systems.
- Implement secure vendor enclaves to isolate ICS and vendor devices to reduce the risks of malware and attack.
- Provide verification that no default passwords exist on any managed system or device.
- Manage all managed devices automatically using Smart Rules and store a unique password per each device.
- Automatically rotate each device's password based on age or after each remote vendor session.
- Provide a complete workflow for device access including an approval process for when remote vendor access is required.
- Record all or select remote sessions with playback to document and review what occurs when a device is accessed.
- Provide detailed reporting of all credentials used and requested when remote activity occurs.

CHAPTER 14

Internet of Things (IoT)

The Internet of Things (IoT) introduces a unique set of threats based on privileges and attack vectors for a threat actor. By definition, they are single-purpose devices with embedded operating systems to perform a specific function. They possess a few required traits including the capability to be networked and provide a designated function. This includes everything from network-based cameras, digital video recorders, to digital personal assistants. These devices can be categorized for commercial use like biometric door looks to home use like Bluetooth door lock keypads and thermostats. While these types of devices have existed for years, they have only recently been grouped and labeled IoT based on their mass adoption and, more importantly, their mass identification of security risks and privileged attack vectors. Therefore, as IoT devices become more commonplace, there is a need to ensure that they do not represent an unnecessary security risk to standard business operations. Unfortunately, it has already been proven that many of these devices are insecure by design, have unresolvable flaws, and can be leveraged to compromise an entire organization with something as simple as a default credential. A simple gold mine for a threat actor. For any IoT deployment, consider

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_14

these five recommendations to mitigate security risks and keep privileged threats away from the sensitive information in a corporate environment:

1. **Segment networks**

   Using basic capabilities in modern network routers and switches, all IoT devices should be networked using separate wireless networks and VLANs. All communications from IoT networks should be explicitly blocked from critical servers, databases, and workstations that should not communicate directly with the devices. This helps ensure that if an IoT device is compromised, it cannot directly be leveraged to steal critical information. If possible, all IoT network communications should be monitored to the Internet and other trusted networks to identify any anomalous behavior.

2. **Change all passwords**

   Almost all IoT devices ship with default passwords for initial configuration. We understand based on previous chapters how much of a risk these can be. End users should change all usernames AND passwords on these devices to complex passwords and unique usernames, and consider changing at least the passwords on a periodic basis. This is where a password management solution can assist in mitigating any threats and keep the passwords on every device unique to avoid password reuse.

3. **Update firmware**

   Make sure that you maintain the latest firmware and security patches on all IoT devices to mitigate any emerging threats and identified vulnerabilities that could be leveraged against the devices.

4. **Don't place the device directly on the Internet**

   Never place IoT devices of any type directly on the Internet with public IP addresses. It is just a matter of time before they will be compromised or subject to a DDOS attack. IoT devices are based on very simple networking technology and not robust enough to thwart all the potential IP traffic that contains malicious code on the net.

5. **Prevent Shadow IT with discovery**

   Shadow IT is another buzzword for rogue devices and unsanctioned assets. Make sure any IoT devices placed on your network are approved and follow the steps above. Shadow IT based on IoT could easily violate many of your security policies and introduce a threat. Standard network discovery tools can find these rogue devices and help place them under proper management.

For any organization planning on introducing IoT devices on the corporate network, there are a few things that can be done to ensure their security. Consider wrapping these into your corporate security policies.

1. **Demand a vulnerability service-level agreement**

   Request from the manufacturer a service-level agreement for patching critical vulnerabilities once they are identified. This will help you ensure IoT devices selected for your organization will stand up to regulatory scrutiny and patch compliance initiatives. In addition, make sure these questions are asked during an RFP or procurement process to ensure the vendor has the proper maturity for managing risks.

2. **Perform security updates on a regular schedule**

   Document a process and ensure all IoT devices can be patched in a timely manner if a flaw is found and without extensive disruption to the business. Some devices are very difficult to patch and update and may have hidden labor costs to manage one at a time.

3. **Ensure role-based access**

   Any security model present within these devices is flexible enough to be integrated into an Active Directory or a Radius server. As a longer-term goal, all credentialed access to these devices should be centrally managed and properly organized within existing identity and access management solutions. If they cannot, these may present a new risk via rogue accounts and an easy target for a threat actor due to their limited management capabilities. Finally, if managed devices have no role based access model, or if they are not feasible to manage in this capacity due to operational reasons, consider a least privilege solution for IoT and network devices as an alternative solution.

IoT devices are just another piece of technology that businesses are enabling for convenience and unified sources of information and security. They are not mature compared to the server and desktop counterparts, and everything from default credentials to back doors present a real privileged risk to an environment. As immature as they are, they should be treated as young children. They need restrictions, governance, and should be monitored.

CHAPTER 15

The Cloud

The history of passwords dates back to the Roman military. Initially, they were carved into wood and soldiers passed them around via the active guard on duty. They were a shared resource. Today, the most common storage of a password is the human brain, and not physically documented and shared. We assign a password to a system or application, recall it when it needs to be used, and remember it each time we change it. Our brains are full of passwords and often we forget them, need to share them, and are forced to document them on post-it notes, spreadsheets, and even communicate them via email (a very poor security practice in itself). These insecure methods for sharing passwords have caused the press to report front page news articles on data breaches and educate organizations on the insecure methods for password storage, sharing, and phishing. Humans cannot be expected to verbally or typographically share a password each time we need it, nor is it safe to communicate them via email or text message to an authorized peer. Therefore, a better method to document passwords is needed that is highly secure, documents distributed access, and promotes sharing and collaboration with minimal risk no matter where the access occurs, and from virtually any medium. The cloud is ideal for this situation when passwords need to be available outside of the organization or shared (not preferred), and on-premise technology is incapable of meeting these requirements.

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_15

Technology professionals have embraced the cloud for many of the traits it possesses from sharing, storing, and securing information outside of the organization. Depending on the sensitivity of the information, extra steps are needed to ensure that the information is protected against modern attack vectors but still usable for a variety of use cases. For privileged access management and password storage in the cloud, several primary use cases stand out for cloud-based deployments:

- **Mobile Workforce:** The ability for remote team members to access current passwords and obtain policies and rule bases.
- **Distributed or Outsourced Information Technology Support:** The ability for outsourced, contracted, or remote information technology team members to access credentials and initiate sessions for resources they are responsible for using context-aware methodologies.
- **Information Technology Collaboration:** Team members often need to share passwords for assets and applications to perform a task or maintenance. A central repository for password storage allows collaboration without the risks of rogue document password storage.
- **Break Glass:** The technology independent storage of passwords for key systems and applications in case of a crisis or break glass incident.
- **Cloud Models**: The organizational responsibilities for securing cloud credentials vary depending on the selected cloud models – SaaS, Pass, or SaaS.

The Mobile Workforce

Most organizations today have some percentage of employees that are mobile or remote. They can range from sales, support, executives – all the way through to development. This remote workforce shares resources with brick and mortar employees through a variety of technologies from Virtual Private Networks (VPN) to cloud services. For some, they need access to systems that have shared or unique credentials that should not be documented in email or other media due to their sensitivity. This is where using the cloud to store passwords can assist with security and productivity due to its universal access.

In Figure 15-1, a trusted user (solo) accesses the cloud password storage solution to retrieve a password regardless of their location or connectivity. Then, when connected via VPN, accessing other cloud resources, or even managing their local system, they can apply the password to perform the tasks necessary to complete their job function. These passwords should never become stale and should follow established policies for password rotation and complexity. Automated password management solutions can assist with these requirements.

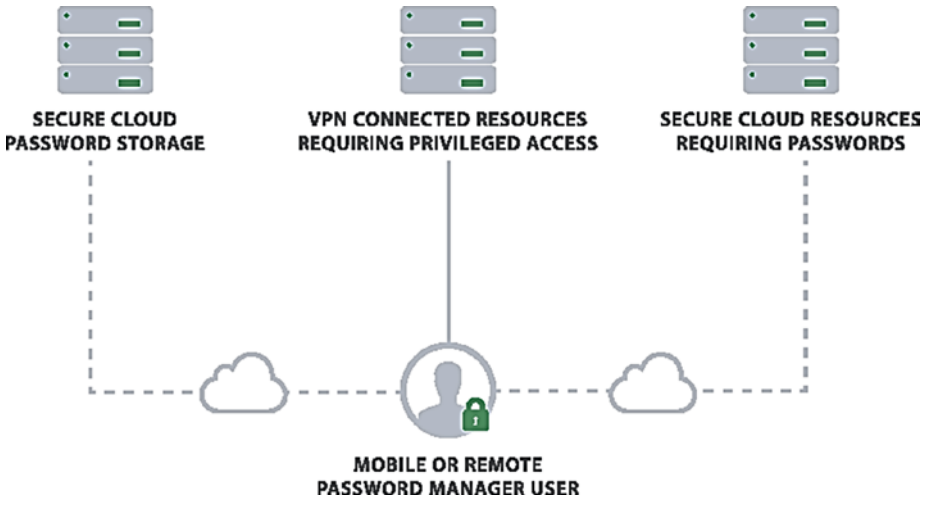

Figure 15-1. *A single user mobile password manager access*

Distributed Information Technology

Many businesses, management of systems, applications, and technology require a plethora of expertise from remote employees, contractors, and vendors. They access technology supporting the organization from a variety of locations, time zones, and device types. When automatic password management is not available or feasible, secure storage of the most recent valid passwords in the cloud makes perfect business and technical sense for all the resources to access the systems they need to complete their job functions.

In Figure 15-2, remote resources are connected to the cloud in a variety of ways (VPN, tablets, cellular, etc.). Based on the task, they retrieve a password and connect to the correct resource to complete their mission. It is up to security and information technology departments to routinely rotate these passwords, and it is encouraged that this form of collaboration is used on a limited scale. For large quantities of users, frequent access, and managed systems, this paradigm still works. However, password rotation should be an automated function and automatically synchronized with the cloud.

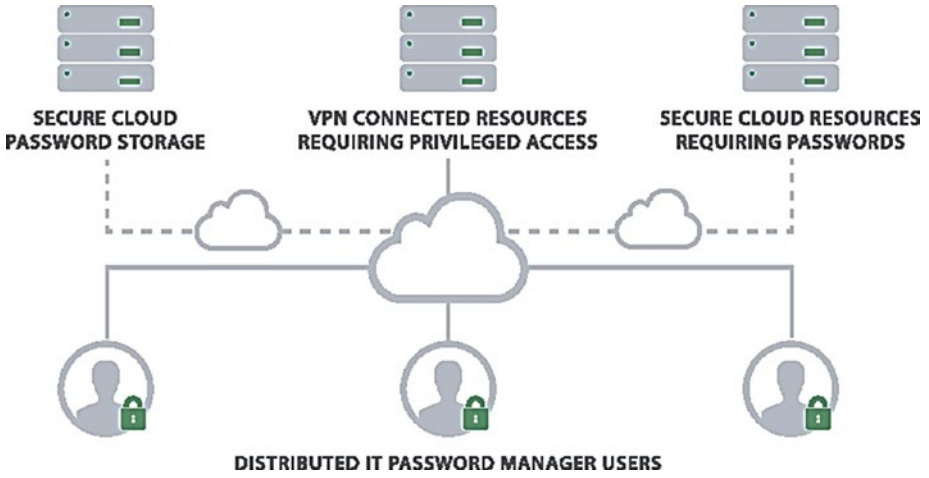

Figure 15-2. *Distributed IT Access to Password Managers*

Information Technology Collaboration

Modern technologies often require collaboration between teams to set up, configure, and maintain. The perspective of one privileged use account versus another should reveal the same logs and settings; however many times different accounts are needed for different components from the web server, middleware, to the database. For teams to successfully collaborate, access is sometimes needed from other perspectives to make the technology work and perform maintenance.

In Figure 15-3, technology teams are aligned and permitted to access passwords from other teams to perform tasks. The passwords are stored in the cloud-based password manager, and all team members are trusted to retrieve the credentials they need to complete the mission. While this access allows teams to user privileges associated with other roles, the organization needs to accept the risk, deviation, and monitor all lateral access. In addition, automated password management and session recording can document lateral access, but instead of these technologies being present, best practices for password complexity and rotation should be adhered to. Finally, the list of trusted users should be limited in order to maintain segregation of duties and the unfortunate risk of sharing passwords (which of course is never recommended but realistically still happens).

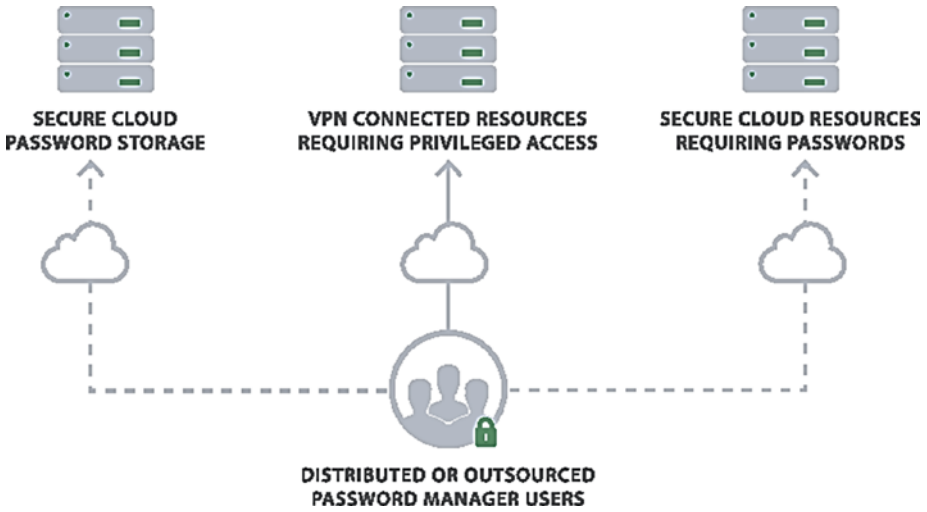

Figure 15-3. *Colaboration using a Password Manager*

Break Glass

In a crisis, access to systems for maintenance, upgrades, or due to a security event can mean the difference between a long-term outage or data breach. Critical passwords may not be available due to an employee not being available all the way through account lockouts. Traditionally, organizations have dealt with Break Glass use cases by documenting emergency passwords on paper, stored in a safe, or placed on secure removable media and locked away as well. These Break Glass scenarios assume someone has access to the physical safe or the decryption password for the secured file. Storing these critical emergency passwords in the cloud simplifies this practice and can manage additional risk vectors including natural disasters.

In Figure 15-4, information technology staff trusted to access the cloud-based password manager can retrieve Break Glass passwords, redistribute them as needed, and access key resources via whatever medium is necessary to complete the task. After any Break Glass usage,

all passwords accessed should be reset and recommitted to the password manager. Automated solutions can do this automatically and on a scheduled basis to ensure their security and proper utilization. For more details on privileged access management and break glass scenarios, please the dedicated section within this book on the topic (Chapter 12).

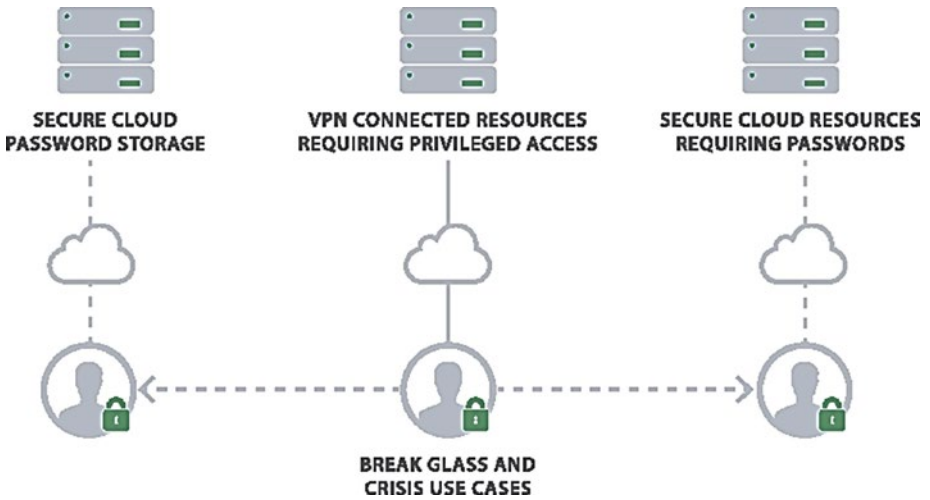

Figure 15-4. *Break Glass and cloud-based password managers*

It is important to remember, any time-sensitive information is stored in the cloud, the security of the information and the hosting application becomes a topic of security conversations. This includes data that could represent a "game-ending event" for the organization and regulatory compliance requirements between states, governments, and countries. If a breach were to occur to an individual account, or the entire system, the potential results could be devastating. When considering the cloud for privileged access management, security teams and operations should always assess the threats, risk surface, exposure, and personally identifiable information stored to determine if the benefits outweigh the risk. Based on this information, the selection of the proper cloud platform for privileged access management becomes the next logical step.

Cloud Models

Growing use of cloud environments for processing, storage, or application hosting and development has opened new avenues for hackers or malicious insiders to inappropriately access sensitive data and disrupt organizations. As cloud adoption continues to accelerate, organizations must secure access to these environments to mitigate security risks while meeting the cost and efficiency demands of hosting more applications and services in the cloud.

Like any on-premise asset, unmanaged cloud environments can create a significant security gap that opens networks to security breaches, data loss, intellectual property theft, and regulatory compliance issues. The first step in getting control over cloud assets is discovery and inventory that can span any cloud service.

Cloud-based deployments of Privileged Access Management can take on several different forms:

- Cloud to cloud for privileged management including application to application (IaaS).
- Cloud-based privileged storage and management for users (SaaS).
- Cloud-based Platform as a Service to deploy your own solution (PaaS).
- Privilege management for on-premise resources (Hybrid).

If this was a multiple-choice question, your strategic business initiatives might require more than one of these categories. It is highly uncommon for privileged access management to be used in only one silo of the business without plans to expand the technology to all sensitive systems and privileged accounts. While initial deployments may start our small, the cloud is also used for management everywhere. This is critical

when selecting PAM on-premise, in the cloud, or a hybrid approach. For hybrid approaches, they can be a combination of IaaS, SaaS, or on-premise or a combination using remote management nodes to route and aggregate data securely.

Infrastructure as a Service (IaaS)

Infrastructure as a service (IaaS) refers to the delivery of computing capacity and infrastructure as a service. In this model, another company operates the data center infrastructure and hardware, allowing customers to build from the operating system "up." Examples of IaaS infrastructures include Amazon AWS, Microsoft Azure, and Google Cloud Platform. Each of these platforms has its own permissions model to provide delegated access to users and groups. These permissions are typically banded together in built-in and\or custom defined roles that provide required access. Given the power and possible business impact if these accounts were to be compromised, proper security and control of these assets is paramount and must be included within the scope of an organization's multilayered security program – including the privilege access layer.

Whether your organization chooses to operate within a single cloud provider, multiple vendors, or has geographical requirements based on regulations or a business model, cloud environments need to authenticate applications and users like any other information technology implementation. Cloud to Cloud privileged access management has unique requirements compared to an on-premise implementation:

- High availability architectures may warrant additional cloud instances to provide high availability in case of a cloud or infrastructure outage that is out of the end user's control.
- Regulations may require separate but duplicate instances and filter data based on region or local laws.

- Environments may have public and private IP ranges to provide the required services and require special provisions to secure them.
- Vulnerability management due to public services may require coordination with the cloud provider to mitigate threats.
- API access requires special attention for secure access and limit exposure.
- Sensitive data in the cloud such as passwords requires additional database security such as HSM to protect information.

For organizations looking to perform PAM only in the cloud, there are multiple technology vehicles to implement a solution. The most common is to use black box technology based on PAM solutions hosted in cloud marketplaces. These allow for hardened PAM deployments based on a variety of licensing models and cloud runtime costs. Some PAM vendors also offer solutions that can be instantiated as a software implementation in a cloud operating system template. These provide the most flexibility for a client, but security, hardening, and operating system configuration are the responsibility of the client – not the cloud provider or PAM vendor. The risks are higher for these types of implementations but can be highly customized to meet requirements.

Software as a Service (SaaS)

Software as a service (SaaS) is a delivery model where a service is centrally hosted by the provider and licensed to customers on a subscription basis. Organizations and end users typically interact with these services via a web console or programming APIs. This allows you to consume a small part of an application without the cost and complexity of building servers and maintaining application software. Examples of corporate SaaS solutions

include SalesForce, Workday, Facebook, and LinkedIn. In a SaaS model, an organization's core security responsibility is the application itself. This includes who can access the application, what authentication is required, and what access users should have. Each application may have its own access model with varying levels of granular provisioning available based on the vendor. Some SaaS applications have traditional business services and may have fine-grained permission models to provide flexibility and permissions to specific groups of users based on tasks or use cases. These applications may also have built-in governance features such as separation of duties and fine-grained auditing to enable organizations to control and audit access to sensitive features and data. Other SaaS applications that have been traditionally consumer focused such as Facebook, LinkedIn, or Twitter, have minimal granularity in their permissioning models. In some cases, users share a common corporate account to manage the system on behalf of the company. While these SaaS applications may not have the same level of sensitive information such as customer lists or financial data, these accounts do represent a significant risk to an organization. Issues could include inconvenience; for example, if the sole user administrator is on leave, updates would come to a grinding halt. Another issue could be a disruptive one, such as if a hacker uses a compromised account to post or tweet inappropriate material that could impact the company reputation. In either case, proper management and control over these accounts should be considered when designing an overall security program.

In addition to securing and controlling access to cloud applications, this new privileged security layer can also be hosted. Privileged Access Management solutions, deployed as an SaaS solution, can operate solely in the cloud or require on-premise management nodes to route and aggregate policy and events. These implementations are completely managed by the PAM vendor, the end user's private cloud environment, or operating using shared cloud resources with other PAM clients in a vendor's multi-tenant installation. This could be hosted by the PAM vendor themselves or an MSP.

Platform as a Service (PaaS)

Platform as a Service (PaaS) is a category of cloud computing that provides an added level of abstraction and automation. These cloud services provide a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the infrastructure or automation framework. Examples of PaaS vendors include Oracle Cloud Platform, Cloud Foundry, and RedHat OpenShift. Given that the PaaS platforms are typically used to design and host an organization's critical applications and services, security needs to be built in from the ground up. A component of that design needs to include privilege access control and auditing.

To protect these cloud models, privilege management privilege solutions may be deployed to handle a variety of activities including the following:

1. Agent-based technologies deployed on the virtual machines running within an IaaS at the operating system or container layer to enable a least privilege access model and to provide detailed real-time activity logging of all privileged activity. Privileged roles could include server administrators, developers, database administrators, etc.
2. Password management functionalities to securely onboard, manage, automate, and randomize privileged accounts across operating systems and service layers provided by the virtual machines running within the IaaS environment. Privileged roles could include server administrators, service accounts, application scripts.

3. Password management functionalities to securely onboard, manage, automate, and randomize privileged accounts at the PaaS management layer. Privileged Roles could include Azure Cloud Administrators, Data Center Administrators, etc.
4. Session Management and Session Recording for all privileged activity at the PaaS management layer.
5. Password and Session management functionalities to securely manage, automate. and audit privileged account activities across corporate SaaS applications.

CHAPTER 16

Mobile Devices

Mobile devices represent a unique attack vector for a threat actor. They have accounts and credentials, but no role-based access, and there are generally only two permission types: user and root. In addition, there is typically only one account and the operating system does not provide provisions for more than one user account as a part of its design.

For a successful attack to occur, a threat actor needs to compromise the operating system or applications using malware or an exploit; this includes potentially malicious software that may be inappropriately hosted on a legitimate marketplace. The goal of the hacker is to leverage the device to do the following:

- Egress information from the device considered personally identifiable or organization sensitive.
- Enable surveillance via GPS, camera, or audio.
- Leverage the device using lateral movement to attack other corporate, home, public, or roaming assets.
- Establish a persistent presence for new or other advanced persistent attacks.

The important point to note here is the fact that the goal of the threat actor is the same regardless of a traditional corporate asset or other Internet of Things (IoT) device. Once privileged access is obtained, the offense by an attacker is the same. However, the defense is completely different

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_16

since there is no role-based access, there is no way to gain root on the vast majority of devices without a jailbreak, and some mobile devices do not even support traditional anti-malware solutions (like Apple iOS). Therefore, the best defense is to adapt to the models for security that are permitted:

- For businesses using mobile devices in a Bring Your Own Device (BYOD) or organizational supplied model, utilize a Mobile Device Manager (MDM) to provide application and data segmentation. This will allow the organization to enforce acceptable use policies and even block (uninstall) potentially malicious applications that could compromise the device.
- For non-Apple devices, there are a plethora of security solutions that can scan for malware, inappropriate permissions, and even poor configurations (like USB debugging) that could be used to compromise the device. Many of these agents are in the appropriate marketplace but also supplied by MDM solutions and traditional anti-virus vendors. It is recommended they be utilized to identify risks and mitigate any platform-specific threats for that mobile device.
- When possible, mobile devices should never have direct access to the data center and sensitive systems. Their connection should always be proxied or routed through a jump host for resource access. Virtual desktops and Remote Applications are ideal for mobile device segmentation to restrict access, enforce multi-factor authentication, and prevent lateral movement. This may include using password management to make the additional connections and session monitoring to capture that any potential roaming access is appropriate.

Mobile devices have provided the world with a vehicle to always stay connected. For a threat actor, they present a way to breach the perimeter of an organization even when the asset is not in the office. Gaining privileged access is not as critical, and as such these devices just do not have the same robust security models as traditional information technology resources. However, leveraging a mobile device to gain a foothold may be good enough for an exploit or malware to do the same amount of damage as root.

So how can a threat actor gain access needed to commit these crimes? It is easier than you think, and the security models for mobile devices are riddled with blatant best practice flaws. Consider these potential scenarios:

- The installation of new software from a trusted marketplace can contain malware. Vendors can only provide so much screening for applications, and repeatedly malware has bypassed detection and been published.
- Biometrics are used for authentication and authorization and used for device access to application credentials. A compromise of biometrics not only provides device access, but for applications like banking, uses the same mechanism to access financial information. Biometrics alone as credentials are just a bad idea because if the electronic form is compromised, they can never be changed. Biometrics should be used for multi-factor since the base credentials can always be changed while biometrics only proves your identity electronically. Unfortunately many mobile device manufacturers are blurring this line and have ignored security best practices by making this the only form of identification required to access a device during normal operations. This is gambling on

the strength of their biometric security module, and time will tell whether the designs will be robust enough to stop modern threats.

- Mobile devices (outside of Qi charging) require a corded connection for battery recharges on a daily basis. Also, they have various bidirectional communication systems from NFC, Bluetooth, and WiFi. The flaw is that there are very little controls around remote exploitation of these communication paths. These include USB chargers that include malware to "man in the middle" attacks that can compromise WiFi communications. These are just security flaws due to the nature of mobile devices, and represent a high risk with no real resolution outside of locking them down to known trusted sources. By default, they are blindly susceptible.
- For Android devices only, the operating system and hardware fragmentation represent unique security challenges per operating system version and device. The scope of the problems well exceeds the confines of this section, and in many cases, a flaw on one Android device may not be present on another, nor may the manufacturer choose to remediate the flaw. For businesses, allowing Android devices via BYOD or corporate purchased, minimum (or specific) versions and vendors should be considered. Not all manufacturers maintain the same service level agreement (SLA) for supplying patches, and others have been known to supply purposely built back doors for their own devices for targeted updates and monitoring - neither of which may be acceptable to a business with sensitive operations.

Despite these flaws, there exists technology to mitigate these risks and apply security best practices. For example:

- Never use biometrics for both device access and sensitive applications on a mobile device. Implementing this policy is good practice to make sure the privileges of one system (biometric access) cannot be used against another (application). In fairness, this is a perfect example of password reuse via biometrics.
- Using MDM technology or security best practices, lock down BYOD devices to trusted networks and disable USB charging from making Trusted Data Connections.
- Decide on what you can support and what you cannot. BYOD should not mean every device an employee may own, even if your MDM can support it. Having a finite list of manufacturers and OS versions will help mitigate risks, especially from outlier threats.

CHAPTER 17

Ransomware

Let me get this out right off the bat: **No one solution is 100% effective in mitigating the risk of ransomware.** Some technologies are claiming to have tested hundreds of samples, and that their tool can stop 100% of the samples. I'm sorry, but that is a falsehood. Why? If any single vendor had a solution that solved the problem completely, ransomware would not be such a problem.

Application control solutions, endpoint protection products, and even least privilege solutions have various degrees of success in mitigating ransomware, but none are 100% effective. Why? Modern ransomware can leverage privileges when available, does not always launch separate executables, and sometimes targets obscure devices like smart TVs. We have seen a spike in ransomware that uses Microsoft Office macros to propagate the threats and even versions that use JScript embedded in a document to conduct malicious activity. We have also seen ransomware like WannaCry leverage exploits across modern and end-of-life operating systems to devastate organizations. The attack vectors are growing as ransomware continues to mature and escalate as this decade's largest cybersecurity threat. Unfortunately, the delivery of the ransomware payload is equally as horrific to identify as the ransomware payment message. It can come from an exploitable vulnerability, an errant executable (the easiest to stop), PowerShell script, or embedded as a macro or script in a file or website. What makes this a little more disturbing is that many attacks combine methods and use a command control server

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_17

to hold encryption certificates versus locally based per infection that can be cured with a decryption solution. The privileges ransomware executes will help dictate how successful the malicious infiltration will be. And, modern ransomware may be just a Trojan Horse for other advanced threats designed to distract IT security teams.

This is why ransomware is so difficult to stop, and no one technology is 100% effective.

There are some actions you can perform with privileged access management to minimize the threat. Unfortunately, nothing will ever replace training users to not click on phishing links or select Run Macros when opening an unknown file. However, here are a few rules that are easy to implement that will block the vast majority of mistakes users can make, stop droppers from executing, and block vulnerable applications from being leveraged against your assets:

- Application Control to Block Untrusted Executables – Privileged Access Management solutions allow for application control and the ability for rules to elevate applications based on rules or policies. This will stop any non-authorized application from executing regardless of the source if it is not properly digitally signed or tries to execute a malicious child process as a dropper.
- Stopping Droppers – Unfortunately, trusted applications can launch other applications to perform their intended functions. This includes browsers, email programs, and even PDF readers. The consistent part of this problem is that these executables almost always launch from temporary file directories. Using Privileged Access Management to manage file integrity, administrators can track, alert, and block rogue dropper executables that appear in these directories or do not meet minimum reputation requirements.

- Vulnerable Applications – Privileged Access Management solutions typically have a reputation service engine or other technology to measure the risk of an application before its launch. This component allows for real-time assessment of an application's health for malware, vulnerabilities, permissions, and privacy. To that end, policies can be established to deny (or notify of) the launch of risky applications that could be leveraged in a ransomware attack. This helps ensure service-level agreements are being met for cybersecurity hygiene and no system is left out that could pose an unacceptable risk.

The lesson from ransomware is the same as privileges as an attack vector. Ransomware risk can be minimized using the same technology used for managing privileged accounts. While this approach is not 100% effective, it is a residual return on investment when organizations embrace this approach. Stop ransomware from running simply by not giving it the permissions it needs to execute in the first place.

CHAPTER 18

Secured DevOps (SDevOps)

As organizations continue to adopt more Agile development methodologies that require extensive integration and automation across operational tools, they often find that it becomes very difficult to effectively and securely manage the credentials required to support the end-to-end process. A typical DevOps process to automate, manage, and deploy code builds may include the following:

- Service Accounts that run various services (TFS, Builds, SQL).
- Scheduled tasks and Automation (Custom scripts, Git and GitHub, Jenkins, Puppet, and others).
- Third-party services (SMTP, Cloud services, SNTP, etc.).
- Certificates for SSL websites, automated code signing, and other processes that have security wrappers.

All these technologies that integrate and automate application development and deployment into a more streamlined process require ***credentials and have no identities*** since they are automated. In some cases, these credentials may be stored and shared in scripts, code, and configuration files. The risks of storing, sharing, and infrequently

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_18

changing credentials used to automate the DevOps processes make them susceptible to hacking and misuse especially if they are clear text. To reduce these risks, organizations should look to expand their privileged access programs and implement phases that include the following:

1. Eliminating hard-coded credentials in code (compiled), scripts, and service accounts. Most Enterprise Password Management vendors include service account and password APIs that can be implemented to address these items.

2. Implement a jump host and managed session facility to control when developers can access production servers. DevOps methodologies often require the pushing of code, compilation, and integration of post-compile workflows. The goal is to have developers safely and easily execute critical workflows, but without having direct access to the systems themselves. Implementing a jump host or session management solution makes this possible by controlling the secure connection into your continuous integration and continuous deployment environment by administrators, automation jobs, or developers.

3. Implementing the concept of least privilege across the application environment. Do the developers, development tools, or development processes need to have administrator or root access to the systems and databases supporting the application environment? A process should be developed so they should not. Implementing least privilege would ensure that these developers and processes only

have the privileges that they need to support their workflow in the end-to-end DevOps process. In addition, augmenting least privilege with session recording and keystroke logging would also help to identify compromised account activity and risks associated with privilege abuse and misuse.

4. To reduce the complexity of creating and managing local accounts across non-Windows systems in a dynamic cloud environment, designers should investigate methods to consolidate and centralize accounts.

Last, organizations should examine solutions to proactively protect containers and micro-services associated with enterprise applications. As organizations transform their traditional applications to the cloud and embrace new concepts including containers, they should consider how to mature the security basics and minimize risks associated with these dynamic environments. In addition to vulnerability scanning and integrity checks, continuous improvements require only approved containers are running in the environment, so organizations should also evaluate least privilege, access monitoring, segmentation, and file and service whitelisting at the container level, to protect the host and other containers that may be running in the environment. Moving all your source code and applications to the cloud is scary. Many of the controls that security professionals take for granted have alternative approaches and should not be ignored. For DevOps, security is the key and privilege management a must to protect the automation process!

CHAPTER 19

Regulatory Compliance

A threat actor does not care about the law, compliance, regulations, and security best practices. In fact, they are hopeful that your organization is lax on many of these specifications and frameworks to leverage them for malicious intent. While regulatory compliance is designed to provide legally binding guidelines for industries and governments, they do not provide the necessary means to stay secure. Compliance does not equal security. They are enforced guidance toward good cybersecurity hygiene, but implementing them without good processes, people, training, and diligence will leave you susceptible to a breach. Therefore, when reviewing leading regulatory compliance initiatives, consider the following:

- How they apply to your organization based on laws, sensitive information, contracts, industry, and geography.
- What overlaps exist between them and what processes can satisfy multiple requirements?
- Be sure to adopt the strictest guidance for your initiatives. The strictest and most comprehensive requirement should always win since it will exceed any looser requirements.

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_19

- Scoping is critical. Just applying the rules to sensitive systems is often not enough to provide good security. Consider the effort and cost of increasing the scope to mitigate risks through any connected system that could affect the legislatively required scope.

Therefore, keep in mind that any regulatory compliance requirements are the absolute minimum your organization should be doing. If you are not meeting the minimums or have lapses in the requirements, you are the low-hanging fruit a threat actor is seeking and slowest individual being pursued by the bear.

Payment Card Industry (PCI)

Initially developed in 2004 and currently on version 3.2, the Payment Card Industry Data Security Standard (PCI DSS) is an information security standard for every organization that accepts credit cards such as Visa, MasterCard, American Express, and others. The PCI standard:

- Was created to increase controls around cardholder data to reduce credit card fraud;
- Has become a de facto standard for protecting access to personally identifiable information (PII), especially in the retail industry;
- Is mandated by the card issuers; and
- Is administered by the Payment Card Industry Security Standards Council (PCI SSC).

Organizations face several challenges when working to prove their compliance with PCI DSS. The largest organizations are challenged with assessments that are conducted annually by a Qualified Security Assessor (QSA) who creates a Report on Compliance (ROC). And although compliance with PCI DSS is not required by federal law in the United States, the laws of some states either refer to PCI DSS directly or make

equivalent provisions. If an organization has been breached and was not in compliance with PCI, the card issuers can impose significant financial penalties on the merchant. Since it is the responsibility of the merchant to achieve, demonstrate, and maintain their compliance at all times during the annual assessment, best practice for PCI DSS compliance is to continually improve processes to ensure ongoing compliance, rather than treating compliance as a point in a time project. Naturally, this can create a tremendous resource drain on technology- and security-oriented teams.

As a part of this process, the primary mission is to protect cardholder data and the security of the transactions involved with this information. Privileged Access Management can assist with all 12 requirements for PCI DSS compliance in various forms from restricting access to command-line filtering. Figure 19-1 provides a high-level diagram of PCI DSS requirements. Based on the requirements, it is easy to see how PAM can impact privileges when implementing firewalls to restricting access to card holder data.

PCI Data Security Standard - High Level Overview	
Build and Maintain a Secure Network and System	1. Install and maintain a firewall configuration to protect cardholder data 2. Do not use vendor-supplied defaults for system passwords and other security parameters
Protect Cardholder Data	3. Protect stored cardholder data 4. Encrypt transmission of cardholder data across open, public networks
Maintain a Vulnerability Management Program	5. Protect all systems against malware and regularly update anti-virus software or programs 6. Develop and maintain secure systems and applications
Implement Strong Access Control Measures	7. Restrict access to cardholder data by business need to know 8. Identify and authenticate access to system components 9. Restrict physical access to cardholder data
Regularly Monitor and Test Networks	10. Track and monitor all access to network resources and cardholder data 11. Regularly test security systems and processes
Maintain an Information Security Policy	12. Maintain a policy that addresses information security for all personnel

Figure 19-1. *PCI DSS Requirements, high level*

HIPAA

Enacted by the United States Congress in 1996, the Health Insurance Portability and Accountability Act (HIPAA) provides provisions to protect health insurance coverage for workers and their families when they change

or lose their jobs, and require the establishment of national standards for electronic health care transactions and national identifiers for providers, health insurance plans, and employers. HIPAA has become a de facto standard for protecting the privacy and security of individually personally identifiable health information in the health care industry based on its initial mandates.

The Security Rule within HIPAA deals specifically with Electronic Protected Health Information (EPHI). It lays out three types of security safeguards required for compliance:

- Administrative Safeguards - Policies and procedures designed to clearly show how the entity will comply with the act.
- Physical Safeguards - Controlling physical access to protect against inappropriate access to protected data.
- Technical Safeguards - Controlling access to computer systems and enabling covered entities to protect communications containing PHI (Protected Health Information) transmitted electronically over open networks from being intercepted by anyone other than the intended recipient.

Based on these three safeguards, it is apparent that patient health information requires protection from a potential threat actor. While a single health care record is a viable target, especially for someone famous or of importance, bulk data is much more valuable on the dark web and for malicious data correlation. Accessing large quantities of data requires privileged access. A single doctor or health care provider does not have that level of privileges. Therefore, HIPAA requires privileged access management and as a vertical, can suffer from the same problems when privileges are used as an attack vector. Table 19-1 shows the sections in HIPAA solved by PAM:

Table 19-1. *HIPAA Requirements that can be addressed with PAM*

HIPAA STANDARD	REF
Security Management Process	164.308(a)(1)
Assigned Security Responsibility	164.308(a)(2)
Workforce Security	164.308(a)(3)
Information Access Management	164.308(a)(4)
Security Awareness and Training	164.308(a)(5)
Security Incident Procedures	164.308(a)(6)
Contingency Plans	164.308(a)(7)
Evaluation	164.308(a)(8)
Business Associate Contracts and Other Arrangements	164.308(b)(1)
Facility Access Controls	164.310(a)(1)
Workstation Use	164.310(b)
Workstation Security	164.310(c)
Device and Media Controls	164.310(d)(1)
Access Control	164.312(a)(1)
Audit Controls	164.312(b)
Integrity	164.312(c)(1)
Person or Entity Authentication	164.312(d)
Transmission Security	164.312(e)(1)
Business Associate Contracts or Other Arrangements	164.314(a)(1)
Requirements for Group Health Plans	164.314(b)(1)
Policies and Procedures	164.316(a)
Documentation	164.316(b)(1)

SOX

In July 2002, the United States Congress passed the Sarbanes-Oxley Act ("SOX"), which was primarily designed to restore investor confidence following well-publicized bankruptcies that brought chief executives, audit committees, and independent auditors under heavy scrutiny. The act applies to all publicly registered companies under the jurisdiction of the Securities and Exchange Commission (SEC) . Financial data and documentation are at the heart of the compliance issue, and within the legislation, SOX Section 404: Assessment of Internal Controls defines vulnerability and privileged access management as a business requirement. This helps a business understand the flow of transactions, including IT aspects, to identify points at which a misstatement could arise, and evaluate controls designed to prevent or detect fraud. The latter places privileges as an attack vector and session monitoring clearly in focus for fraud detection and prevention.

GLBA

The Gramm-Leach-Bliley Act (GLBA) was enacted to ensure protection over customers' records and information. To satisfy the rules and provisions of GLBA, financial institutions are required to perform security risk assessments; develop and implement security solutions that effectively detect, prevent, and allow timely incident response; and to perform auditing and monitoring of their security environment. Similar to SOX, a complete section covers risk management. The primary portions of Section 508 relevant to privileges as an attack vector include these:

- Subtitle A: Disclosure of Nonpublic Personal Information - Constructing a thorough [risk management] on each department handling the nonpublic information.

- Subtitle B: Fraudulent Access to Financial Information - Social engineering occurs when someone tries to gain access to personal nonpublic information without proper authority.

NIST

NIST Special Publication 800-53, Security and Privacy Controls for Federal Information Systems and Organizations was developed by a joint task force comprised of representatives from NIST, the Department of Defense, the Intelligence Community, and the Committee on National Security Systems. This interagency partnership formed in 2009.

This guide delivers a *holistic* approach to information security and risk management by providing organizations with a comprehensive set of security controls essential to fundamentally strengthen their information systems, as well as the environments in which they operate. The resulting systems are more resilient in the face of threats and cyberattacks. NIST SP 800-53 outlines a "Build It Right" strategy combined with various security controls for Continuous Monitoring and strives to provide the senior leaders of organizations information in near real time to support making risk-based decisions related to their critical missions.

Controlling and monitoring privileged access is extremely important for mitigating the risks posed by insider threats, preventing data breaches, and meeting compliance requirements. With that being said, security and IT leaders should walk a fine line between protecting the organization's critical data to ensure business continuity, and enable users and administrators to be productive. Disparate, disjointed tools deployed and managed in silos leave gaps in coverage over privileged access. This legacy model is expensive, difficult to manage, and requires too much time to show any meaningful risk reduction.

The NIST publication recognizes this dilemma and formalizes separation of duties, change control, and privileged session auditing. This clearly defines how an organization should manage access and when. Unfortunately, the size and scope of actual PAM mappings to NIST 800-53 is enormous. If your organization has NIST requirements, please consider external consultants (or in-house expertise if you have the resources) to map your business requirements to contracts and actual deliverables. The scope may even include your supply chain and be completely outside of your control except for contractually based audits.

ISO

The International Organization for Standardization (ISO) has established guidelines and general principles for initiating, implementing, maintaining, and improving information security management in an organization. The objectives outlined in ISO 27002:2013(E) provide general guidance on the commonly accepted goals of information security management.

The control objectives and controls in ISO 27002 are intended to be implemented to meet the requirements identified by a risk assessment. ISO 27002 can serve as a practical guideline for developing organizational security standards and effective security management practices and to help build confidence in interorganizational activities.

For organizations that have adopted ISO 27002, it is important that all existing and new security solutions map into this framework. The standard contains 14 security control clauses, collectively containing a total of 35 main security categories and 114 controls. Whether an organization's objective is to achieve legislative compliance or to adopt security best practices, these controls apply to most organizations and in most environments. These clauses directly translate to privileged access management and privileged session monitoring. Table 19-2 shows the categories and controls influenced by ISO 27002 and PAM:

Table 19-2. *PAM mappings for ISO 27002:2013(E)*

6 ORGANIZATION OF INFORMATION SECURITY
6.1 INTERNAL ORGANIZATION
6.1.1 Information security roles and responsibilities
6.1.2 Segregation of duties
6.1.5 Information security in project management
6.2 MOBILE DEVICES AND TELEWORKING
6.2.2 Teleworking
8 ASSET MANAGEMENT
8.1 RESPONSIBILITY FOR ASSETS
8.1.3 Acceptable use of assets
8.2 INFORMATION CLASSIFICATION
8.2.3 Handling of assets
9 ACCESS CONTROL
9.1 BUSINESS REQUIREMENT OF ACCESS CONTROL
9.1.1 Access control policy
9.1.2 Access to networks and network services
9.2 USER ACCESS MANAGEMENT
9.2.1 User registration and de-registration
9.2.2 User access provisioning
9.2.3 Management of privilege access rights
9.2.4 Management of secret authentication information of users
9.2.5 Review of user access rights

(*continued*)

Table 19-2. (*continued*)

9.3 USER RESPONSIBILITIES

9.3.1 Use of secret authentication information

9.4 SYSTEM AND APPLICATION ACCESS CONTROL

9.4.1 Information access restriction

9.4.2 Secure log-on procedures

9.4.3 Password management system

9.4.4 Use of privileged utility programs

9.4.5 Access control program source code

10 CRYPTOGRAPHY

10.1 CRYPTOGRAPHIC CONTROLS

10.1.2 Key management

12 OPERATIONS SECURITY

12.1 OPERATIONAL PROCEDURES AND RESPONSIBILITIES

12.1.2 Change management

12.4 LOGGING AND MONITORING

12.4.1 Event logging

12.4.2 Protection of log information

12.4.3 Administrator and operator logs

12.5 CONTROL OF OPERATIONAL SOFTWARE

12.5.1 Installation of software on operational systems

12.7 INFORMATION SYSTEMS AUDIT CONSIDERATIONS

12.7.1 Information systems audit controls

(*continued*)

Table 19-2. (*continued*)

13 COMMUNICATIONS SECURITY

13.1 NETWORK SECURITY MANAGEMENT

13.1.1 Network controls

13.1.2 Security of network services

13.1.3 Segregation in networks

13.2 INFORMATION TRANSFER

13.2.1 Information transfer policies and procedures

14 SYSTEM ACQUISITION, DEVELOPMENT AND MAINTENANCE

14.1 SECURITY REQUIREMENTS OF INFORMATION SYSTEMS

14.1.1 Information security requirements analysis and specification

14.2 SECURITY IN DEVELOPMENT AND SUPPORT PROCESSES

14.2.1 Secure development policy

14.2.6 Secure development environment

14.3 TEST DATA

14.3.1 Protection of test data

16 INFORMATION SECURITY INCIDENT MANAGEMENT

16.1 MANAGEMENT OF INFORMATION SECURITY INCIDENTS AND IMPROVEMENTS

16.1.1 Responsibilities and procedures

16.1.2 Reporting information security events

16.1.3 Reporting information security weaknesses

(*continued*)

Table 19-2. (*continued*)

16.1.4 Assessment of and decision on information security events
16.1.7 Collection of evidence
17 INFORMATION SECURITY ASPECTS OF BUSINESS CONTINUITY MANAGEMENT
17.1 INFORMATION SECURITY CONTINUITY
17.1.1 Planning information security continuity
17.1.2 Implementing information security continuity
17.1.3 Verify, review and evaluate information security continuity
17.2 REDUNDANCIES
17.2.1 Availability of information processing facilities
18 COMPLIANCE
18.1 COMPLIANCE WITH LEGAL AND CONTRACTUAL REQUIREMENTS
18.1.2 Intellectual property rights
18.1.3 Protection of records
18.2 INFORMATION SECURITY REVIEWS
18.2.1 Independent review of information security
18.2.2 Compliance with security policies and standards
18.2.3 Technical compliance review

Security best practices have been adopted in almost every regulation and framework. ISO 27002 is no different when monitoring and managing privileges, and sessions form a fundamental part of managing the privileged attack vector and thwarting threat actors. Mapping these controls to your privileged access management deployment will help close off many of the attack vectors that we have discussed.

ASD

The Australian Signals Directorate (ASD) has developed a list of strategies to mitigate targeted cyber intrusions. The recommended mitigation strategies were developed through ASD's extensive experience in operational cybersecurity, including responding to serious cyber intrusions and performing vulnerability assessments and penetration testing for Australian Government Agencies in 2014.

In 2017, the ASD expanded the Top Four recommendations to contain the Essential Eight. The dynamic nature of cybersecurity required a course correction to address the latest threats like ransomware. Businesses and governments are accustomed to broad stroke changes occurring every few years, but rarely are recommendations made that are very precise to manage specific threats. The Essential Eight are the following:

Australian Signals Directorate Top 4 (original from 2014)

1. Application whitelisting of permitted/trusted programs, to prevent the execution of malicious or unapproved programs including executables, scripts, and installers.
2. Patch applications – for example, Java, PDF viewer, Flash, web browsers, and Microsoft Office. Patch/mitigate systems with "extreme risk" vulnerabilities within two days. Use the latest version of applications.
3. Patch operating system vulnerabilities. Patch/mitigate systems with "extreme risk" vulnerabilities within two days. Use the latest suitable operating system version. Avoid Microsoft Windows XP.

4. Restrict administrative privileges to operating systems and applications based on user duties. Such users should use a separate unprivileged account for email and web browsing.

Essential Eight (Amended in 2017)

A. Disable untrusted Microsoft Office Macros, so malware cannot run unauthorized routines.

B. Block Web browser access to Adobe Flash, web advertisements, and untrusted Java code on the Internet. If possible, uninstall all browser plug-ins that are not required.

C. Multi-factor authentication for all systems when possible to make it harder for an adversary to access a system and information.

D. Daily backup of important data securely and offline to ensure even if data is compromised, protected versions are available for recovery.

Based on a threat actor's methods to gain privileges, these recommendations are completely in line with the threats solved by privileged access management. The privileged attack vector is included in the Top 4 and represents a strategic mitigation needed worldwide to stop modern threats.

MAS

The Monetary Authority of Singapore (MAS) was founded in 1971 to oversee various monetary functions associated with financial and banking institutions. Throughout the years, their guidelines have been revised to manage emerging technologies and the evolving threat landscape. In June

2013, the MAS created a new set of guidelines for Internet Banking and Technology Risk Management (IBTRM). This addendum mandated certain requirements for Technology Risk Management (TRM) and contained a set of guidelines as well (TRM Guidelines), along with errata notices (TRM Notices).

The TRM Guidelines are statements of industry best practices to which Financial Institutions are expected to adhere. The guidance is not legally binding but is used by MAS in risk assessment audits of financial institutions.

Privileged as an attack vector considers four of these sections relevant when protecting privileges from a threat actor:

- Section 4: Technology Risk Framework
- Section 6: Acquisition and Development of Information Systems
- Section 9: Operational Infrastructure Security Management
- Section 11: Access Control

GDPR

The General Data Protection Regulation (GDPR) is one of the most important movements in the area of data protection in recent years. It was passed into European Union (EU) law on April 28, 2016, and will become enforceable on May 25, 2018. In summary, the GDPR defines controls around how organizations store and process the personal data of EU citizens, irrespective of where the organization is based, owned, or operating. Anyone storing or processing the personal data of an EU citizen must comply with the GDPR or face significant fines in the event of an audit or data breach. Those fines can be up to 4% of the organization's global turnover or €10m, whichever is greater. With this level of impact,

it is vital that all organizations understand their obligations under the GDPR and take appropriate measures to ensure they are compliant demonstrating that the proper controls are in place to protect information.

GDPR was designed to simplify the current requirements and not introduce a massive new burden on organizations. In fact, GDPR consolidates the 28 distinct implementations of the previous Data Protection Directive (95/46/EC) into one regulation for consistency, standardized version control, and reporting. It is important to note, it does not apply to law enforcement agencies that may need to exchange this information as a part of an investigation.

To that end, PAM solutions that can help organizations achieve GDPR compliance by developing a strong, yet simple, cybersecurity foundation based on security best practices for privileged access. This is done by addressing privacy and user obscurity through standard PAM features:

- Privileged Password Management can help control who has access to operating systems, applications, databases, infrastructure, and cloud resources, and provide attestation reporting on session activity for users that may have access to sensitive data and resources.
- Server Privileged Management can manage privileged access to commands and applications, eliminating the need for root access and Sudo that may expose sensitive user data.
- Endpoint Privileged Management can pseudonymize data collected around user and administrative activity ensuring data cannot be linked to individuals within a single data store. This is the most effective strategy to protect end-user activity and access to sensitive data.

SWIFT

SWIFT's Customer Security Controls Framework 1.0 published on March 31, 2017, describes a set of mandatory and advisory security controls for participating SWIFT financial organizations. The framework is divided into three objectives:

- Secure Your Environment
  - Restrict Internet Access
  - Protect Critical Systems from General IT Environment (Lateral Movement)
  - Reduce Attack Surface and Vulnerabilities
  - Physically Secure the Environment
- Know and Limit Access
  - Prevent Compromise of Credentials
  - Manage Identities and Segregate Privileges (PAM)
- Detect and Respond
  - Detect Anomalous Activity to Systems or Transaction Records
  - Plan for Incident Response and Information Sharing

SWIFT requires that users self-attest compliance against the mandatory security controls (it is optional for the advisory controls), with a deadline of January 1, 2018, to submit their self-attestations. PAM provides coverage for the following mandatory controls:

1.1 Operating System Privileged Account Control

2.1 Internal Data Flow Security

2.2 Security Updates

2.3 System Hardening

2.6 A Operator Session Confidentiality and Integrity

2.8 A Critical Activity Outsourcing

4.1 Password Policy

4.2 Multi-Factor Authentication

5.1 Logical Access Control

5.4 A Physical and Logical Password Storage

6.2 Software Integrity

6.4 Logging and Monitoring

Organizations can address their compliance and security requirements as defined in the SWIFT Customer Security Controls Framework by implementing PAM solutions. Please note, if your organization currently adheres to the NIST Cybersecurity Framework, ISO 27002, or PCI DSS, SWIFT provides mappings to other frameworks to expedite compliance verification and not duplicate efforts in attestation reporting.

CHAPTER 20

Sample PAM Use Cases

A threat actor thrives on the weakness of processes and the inability for an organization to establish best practices or even follow processes. To that end, Privileged Access Management can stymie a threat actor, even if security best practices are being followed. For it to succeed, consider these top three problems almost every organization faces and the use cases to resolve them:

1. Employees and other insiders have unnecessary access: Employees, vendors, and other insiders are often given excessive access to systems and data – and that access can go unmonitored.
2. Credentials are shared and unmanaged: Passwords are created and shared but aren't audited, monitored, or managed with discipline or accountability.
3. Information Technology (IT) assets communicate unchecked: Desktops, laptops, servers, and applications communicate and open paths to sensitive assets and data.

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_20

Even with security best practices, these three deficiencies can materialize in almost every enclave or implementation. Consider the use cases in Table 20-1 to address these problems in the form of "Use Cases" that PAM can solve for any organization:

Table 20-1. *PAM Use Cases*

Title			
Challenge	**Need**	**Solution**	**Benefit**
Tasks Require Administrative Credentials			
Applications require privileged credentials to operate correctly. Security policies do not provide administrative or root credentials to users to complete their assigned tasks.	Users need to execute applications that require privileges above Standard User.	Implement a least privilege solution to change the privileges of the application or seamlessly apply privileged credentials to the application.	Users can perform their intended tasks, and security policies are maintained by not providing privileged credentials.

(*continued*)

Table 20-1. (*continued*)

Title			
Challenge	**Need**	**Solution**	**Benefit**
Local Credentials Have Stale Passwords			
Local accounts have passwords that are reused, well known, or never been changed on servers, desktops, laptops, and tablets.	Security best practices and regulatory compliance requires privileged password management and that reused, well known, or non-managed passwords are mitigated.	Using a password management solution or agent technology, provide a method to identify credentials used for user logins and services, and place them under management.	Ensures security best practices for credential management and ensures even mobile devices can be managed against password reuse and stale password problems.
Correlation and Consolidation of Account Aliases			
Organizations have too many local and directory service aliases for the same identity making reconciliation difficult.	Organizations and regulations require reliable identification of a user's activity. With disjointed aliases, this mapping is difficult to maintain.	Utilize a directory-bridging technology across all Unix, Linux, and MacOS environments to centralize authentication via Active Directory.	Ensures that an Identities Active Directory account is the same authoritative account for all platforms and eliminates local aliases.

(*continued*)

Table 20-1. (*continued*)

Title			
Challenge	**Need**	**Solution**	**Benefit**
Correlation of Vulnerable Applications and Usage			
Threat Analysis and vulnerability management programs lack the correlation of vulnerable applications and real-world usage.	Organizations cannot prioritize vulnerabilities based on user behavior and application usage.	Track application usage with granular details and map the results to known vulnerabilities.	Control the application via whitelisting, blacklisting, and greylisting based on vulnerabilities, age, and risk.
Removal of End User Administrative Privileges			
Security best practices, threat reduction, and compliance regulations require the management of privileged rights.	Remove administrative rights from all end users while allowing them to maintain productivity.	Implement a least privilege solution that can target applications and operating system tasks for privileged rights without providing the end user administrative credentials.	Risk reduction by avoiding baseline drift, malware mitigation through the removal of rights, lower total cost of ownership, regulatory compliance, and fewer administrative accounts.

(*continued*)

Table 20-1. (*continued*)

Title			
Challenge	**Need**	**Solution**	**Benefit**
Removal of Server Administrative Rights			
Security best practices, threat reduction, and compliance regulations require the management of privileged rights and session activity monitoring when accessing servers.	Remove administrative or root privileges from administrators while allowing them to maintain productivity on server-based operating systems.	Implement a least privilege solution that can target applications, databases, and operating system tasks for privileged rights without providing the administrator real local or domain credentials.	Risk reduction by enforcing change control, malware mitigation through the removal of rights, regulatory compliance, and full session management.

(*continued*)

Table 20-1. (*continued*)

Title			
Challenge	**Need**	**Solution**	**Benefit**
Removal of Application to Application Passwords			
Applications, services, and databases need credentials or certificates in order operate correctly as their processes authentication against local or remote resources.	The ability to remove stale and static password assignments within applications and replace them with API calls or programmatic replacements.	Implement a password management solution capable of replacing passwords within applications or substituting API calls within applications to remove user-defined or hard-coded passwords or certificates.	Passwords or certificates used between applications are no longer hard-coded, stale, and can be managed by a password management solution.

(*continued*)

Table 20-1. (*continued*)

Title			
Challenge	**Need**	**Solution**	**Benefit**
Change Control Workflow Requires Approvals			
Change control requiring administrative or root privileges mandates approval from team members before execution	Instrument a workflow that contacts team members, requires approval, or denial or privileged access to a host in order to complete privileged tasks governed by change control.	Implement a password management or least privileged solution that has a workflow engine (internally or compatible with third-party solutions) that can track, report, and provide access once approvals have been granted.	Change management, security best practices, and workflow approval and requirements can be met for privileged access.

(*continued*)

Table 20-1. (*continued*)

Title			
Challenge	**Need**	**Solution**	**Benefit**
Reduction of Threats for Infrastructure Access			
Non-server based infrastructure such as routers, switches, firewalls, load balancers, cameras, security systems, iDracs, etc. typically have the same password across multiple devices (password reuse) or have stale passwords leading to unnecessary risk and exposure.	Provide a mechanism to manage infrastructure passwords, ensure they are all unique, and automatically rotate (manage) them on a periodic basis to ensure they do not become stale.	Implement a password management solution that is capable of discovering and classifying infrastructure devices and managing (rotating) passwords on a periodic basis for any managed account.	Risk reduction and security best practices for unique passwords per device and automatic rotation of passwords to prevent leakage or stale passwords from being compromised.

(*continued*)

Table 20-1. (*continued*)

Title			
Challenge	**Need**	**Solution**	**Benefit**
Automatic Login with No Credential Exposure			
Provide access to a resource without exposing the credentials. How to control what happens to a password once it has been released?	The ability to log on to a resource (application, operating system, database, etc.) without exposing the credentials and providing a malicious opportunity to the user or insider threat for coping and reusing the credentials.	Implement a password management and/or a least privilege solution that can automatically pass credentials to a resource for authentication without exposing them to the end user.	Users are logged in automatically, and the session can be monitored for potentially malicious activity.

(*continued*)

Table 20-1. (*continued*)

Title			
Challenge	**Need**	**Solution**	**Benefit**
Document Privileged Activity for Audits and Compliance			
Determine what a user did during a session and alert on any potential inappropriate activity, especially when using administrative or shared accounts.	A solution that can record video, keystroke log, and application activity in a reportable and indexed format for review by security teams and auditors.	Implement a technology that can provide this capability (session record, keystroke log, and application activity) in line with an active session or using agent or proxy technologies. The results should be stored in a database, encrypted, and protected so that they could be used for forensics or a court of law if required.	Session activity can be reviewed for mistakes, malicious activity, training, or even breach forensics.

(*continued*)

Table 20-1. (*continued*)

Title			
Challenge	**Need**	**Solution**	**Benefit**
Provide an Access Broker to Cloud Resources			
Limit risk exposure to cloud resources by restricting privileged access to only trusted users, resources, and locations.	Implement security processes and technology that can control privileged access to cloud resources ensuring they do not get compromised from remote threat actors.	Implement a cloud access service broker or remote session proxy that can manage connections via user, credentials, location, and even context-aware time of date.	This adds a layer of security for environments to properly access and control cloud resources while restricting potential lateral connectivity.
Manage Third-Party Access Risk			
Ensure partner, contractor, and authorized third-party access into the company, cloud, or other resources are used correctly by non-employees even on a temporary basis.	Provide complete context-aware access of users, location, and time and date access to resources. Document all activity for auditing and forensics.	Implement a password management solution that controls and monitors non-employee access with granularity needed to review any session activity.	Limit the exposure of non-employee access and mitigate risks from stolen credentials, rogue sessions, and lateral movement by unauthorized personnel.

(*continued*)

Table 20-1. (*continued*)

Title			
Challenge	**Need**	**Solution**	**Benefit**
Break Glass			
Provide out-of-band access to systems during a crisis. Note: This is covered in detail in a previous section.	Privileged access can be granted in the event of an emergency.	Implement a password management system capable of releasing emergency (break glass) credentials in the event of a crisis and document all activity and usage to ensure proper resolution.	Ensures that crisis situations can be resolved quickly even if key personnel are not available or in the event of a disaster.
Minimize Data Exposure			
Controlling access to sensitive data when users or administrators have been granted privileged rights to a system, application, or database.	Provide a vehicle to monitor commands, data displayed, and output for malicious activity that might expose sensitive data.	Implement a password manager and least privilege solution that can perform command-line filtering, alert on activity, and search for displayed results that might indicate excessive data exposure.	Users and administrators can be blocked from issuing sensitive commands and teams can be alerted if data is visible from sensitive sources.

(*continued*)

Table 20-1. (*continued*)

Title			
Challenge	**Need**	**Solution**	**Benefit**
Granular Role-Based Access			
Operating systems and applications may not contain granular permission controls to restrict inappropriate access.	When possible, restrict commands, child processes, applications, and operating system functions even when the user is executing with privileged rights.	Implement a technology that can monitor individual commands, child processes, scripts, and applications and perform an action if they are executing including blacklisting the task from executing.	The results minimize the attack surface for operations that may not inherently have role-based access built in.
Rogue Accounts			
Privileged users (or insider threats) may have the ability to create rogue local, domain, or application accounts against company policies and security best practices.	Prevent out-of-band access and potential malicious activity by preventing the creation of rogue accounts.	Implement a technology that can monitor local, domain, and application account creation and based on policy, even deny the accounts from being created in the first place.	Risk reduction by controlling account creation to authorized business processes only.

(*continued*)

Table 20-1. (*continued*)

Title			
Challenge	**Need**	**Solution**	**Benefit**
Service Accounts			
Service Accounts have privileged access on the local system and in some cases, such as in the case of Windows domain accounts, access to off system resources. Given the complexity of managing these credentials and potential impact on operations they are often configured with non-expiring passwords and are rarely changed.	An automated method to discover, rotate, and restart distributed service account passwords while minimizing the impact on dependent applications and processes.	Implement a password manager that can perform centralized discovery, password management, and intelligent restarting of services accounts across the enterprise.	Stored Passwords are no longer hard-coded and can be cycled on an ongoing and frequent basis, all while reducing the downtime of an application and related services. This reduces the risks associated with back-door access employees and contractors and reducing the risks associated with numerous password hacking techniques.

(*continued*)

Table 20-1. (*continued*)

Title			
Challenge	**Need**	**Solution**	**Benefit**
Controlling Access Availability			
Dynamic Access Control is not a specific use case but may be implemented to provide added security in any of the previously discussed scenarios. Organizations that want to control when a user should have access to specific resources and systems can be limited by the native access models. For example, third-party vendors should not be able to access their passwords after working hours, or	The bottom line is that many organizations have internal and external entities that need to access the network on a regular basis. There is an issue with this: how can you be sure that the credentials used for access are being properly managed? As seen all too often, hackers will leverage external company credentials to find a route in. After all – you are only as strong as your weakest link.	Implement a password management and\or session management solutions that provide dynamic access policy constructs. Dynamic Access models evaluate all the parameters at the point of the access request to make sure the appropriate decision is made regarding access. Evaluation criteria can include: Who is trying to log on?	Applying context to each reduces risk by enabling the organization to incorporate best practices to privileged access that can help protect your organization from a breach. For example, if we know that a break glass account is for emergency use only, let's only make it available out of hours.

(*continued*)

Table 20-1. (*continued*)

Title			
Challenge	**Need**	**Solution**	**Benefit**
server administrators should not have access to the financial application server during month-end payroll processing or from remote locations.	Organizations need the ability to overlay a more flexible and dynamic access model on top of the native access constructs of the underlying systems and applications.	What system are they trying to access? Where are, they logging in from? What level of access are they requesting? What day of the week is it? What is the time of day?	Also, if we would normally expect that account to be accessed via a remote worker working from home, let's also make sure the request is coming in via the VPN concentrator.
Incident Tracking			
Remote management and ticketing systems lack the visibility into incidents and unplanned resource allocation.	The ability for authoritative sources for change control and incident tracking to have awareness and approvals of out-of-band access and changes	Implement a privileged access solution that integrates activity with ticketing, help desk, and other call center solutions for workflow and documentation.	Any and all access is documented with tickets and a documented process for access can be achieved.

CHAPTER 21

Deployment Considerations

Anytime you embark on an enterprise project, the costs, return on investment, risks, benefits, threats, and workflow (to name a few) should be considered. When deploying a PAM solution, the realization that it may impact the entire organization needs to be considered. This means that not only administrators will be effected but also end users that may lose administrative rights from contractors, executives, through temporary employees (although I hope your business never gives a temporary employee admin rights; sadly it happens). Deciding where to start, how to deploy, how to educate, and the measurable outcome are challenges that must be addressed up-front. If they are not, internal politics, user resistance, and Shadow IT may completely circumvent the reasons for embracing PAM in the first place. This chapter covers some of the deployment considerations all executives, security professionals, and operational teams should consider, discuss, and address along the journey.

Prioritizing the Risk

Lack of visibility and awareness of all the privileged accounts and credentials across an enterprise pose a monolithic challenge — especially for those companies that rely on manual processes and tools. Privileged

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_21

accounts, many long forgotten, are sprawled across most organizations including desktops, servers, hypervisors, cloud platforms, cloud workloads, network devices, applications, IoT devices, SaaS applications, and more. Different teams may be separately managing — if managing at all — their own set of credentials, making it difficult to track all the passwords, let alone who has access to them and who uses them. An admin may have access to 100+ systems, possibly disposing them to take shortcuts in maintaining the credentials.

With this proliferation of privileges scattered throughout the environment – where do you start? In some cases, organizations will start with the end users and target desktops and remove administrator rights to mitigate threats like ransomware. In other cases, they will start by protecting the *nix server environment supporting critical business applications like trading floors or banking systems. In some they will need to adhere to third-party vendor monitoring as a compliance requirement. Perhaps they have a shorter-term need to focus on a subset of assets to respond to an audit finding, such as properly securing and managing assets connected to the secured PCI network segment. Whether you begin with servers, desktop, networking devices, and/or other connected devices, your decision is a function of risks, complexity, and cost. Ask yourself where the biggest pain is first, what is the risk of tackling it first, and can it be successful? Once you understand the risk and pain, you start by “ripping the Band-Aid off” or “picking the lowest-hanging fruit” to prove success and gain experience.

Privileged Credential Oversight

Even if IT successfully identifies all the privileged credentials strewn across the enterprise, this does not by default translate into knowing what specific activities are performed during a privileged session (i.e., the period during which elevated privileges are granted to an account, service, or process). Privileged access to a superuser account should not amount to ceding

carte blanche to the user. Moreover, PCI, HIPAA, and other regulations require organizations to not just secure and protect data, but be capable of proving the effectiveness of those measures. So, for both compliance and security reasons, IT needs visibility into the activities performed during the privileged session.

Ideally, IT should also have the ability to seize control over a session should inappropriate use of the credentials occur. But, with potentially hundreds or concurrent privileged sessions running across an enterprise, how does IT expeditiously detect and halt malicious activity? While some applications and services (such as Active Directory) can log user actions, and while Windows servers using logon events within Event Log data can reveal some behavioral anomalies, expect full coverage of privileged account usage to require a third-party solution. Consider the use cases needed to track oversight and auditability and the necessary infrastructure when designing your deployment and workflow.

Account Sharing

IT teams commonly share root, Windows Administrator, and many other privileged passwords so workloads and duties can be seamlessly shared as needed. However, with multiple people sharing an account password, it may be impossible to trace actions performed with an account to a single individual, complicating auditing and accountability. For a successful deployment, access how often this problem occurs and where it needs to be addressed with PAM.

Embedded Credentials

Privileged credentials are needed to facilitate authentication for app-to-app (A2A) and application-to-database (A2D) communications and access. Applications, systems, and IoT devices are commonly shipped,

and often deployed, with embedded, default credentials that are easily guessable and pose a formidable risk until they are brought under management. These privileged credentials are frequently stored in plain text – perhaps within a script, code, or a file. Unfortunately, there is no manual way to detect or centrally manage passwords stored within applications or scripts. Securing embedded passwords requires separating the password from the code so that when it's not in use, it's securely stored in a centralized password safe, as opposed to being constantly exposed as when in plain text. For a successful deployment, identification of all the embedded credentials is key and how you handle fault tolerance when they are removed in favor of a PAM A2A solution.

SSH Keys

IT teams commonly rely on SSH keys to automate secure access to servers, bypassing the need to enter login credentials manually. SSH key sprawl presents a substantive risk for thousands of organizations, which may have upwards of a million SSH keys — many long dormant and forgotten, but still viable back doors for hackers to infiltrate critical servers. So ask, where are they, how are they being managed, and what do I do when they expire? Realistically, PAM can manage SSH keys so environments never get in this situation.

Privileged Credentials in the Cloud

The challenges of visibility and auditability are generally exacerbated in cloud and virtualized environments. Cloud and virtualization administrator consoles (as with AWS, Office 365, Azure, etc.) provide vast superuser capabilities, enabling users to rapidly provision, configure, and delete servers at massive scale. Within these consoles, users can spin-up and manage thousands of virtual machines (each with its own set of privileges

and privileged accounts) with just a few clicks. One predicament then arises around how to onboard and manage all the newly created privileged accounts and credentials. On top of this, cloud platforms frequently lack native capability to audit user activity. And, even for those organizations that have implemented some degree of automation for their password management (either through in-house, or third-party solutions), if not architected with the cloud in mind, there's no guarantee a password management solution will be able to manage cloud credentials adequately. For a successful deployment, ask about how many cloud resources are your organization using, who has privileged access, and how is access being maintained and monitored.

Applications

Traditionally applications usually only had to store credentials for resources external to the application. Some examples are remote databases, file shares, or LDAP servers. Ensuring that developers securely store these credentials has always been a challenge. Unfortunately developers have created a large number of applications over the years that store these credentials in plain text within the configuration files of the application. With the explosion of cloud computing, SaaS and IaaS offerings over the last 5 years, applications are increasingly interacting with many platforms and not just a single external resource. It is therefore common for configuration files to have many API keys and credentials for various platforms. Often, API keys are not seen as the sensitive piece of information that they should be by developers. This is evident by the number of applications where effort is put forth to securely store credentials for databases but API keys for cloud resources are left in plain text. How many times have developers pushed code to GitHub with API keys included or accidentally exposed API keys while posting source code to Stack Overflow? The carelessness is outstanding.

As with traditional resources, when investing in the cloud we need to push developers to achieve the highest application goals but with the least amount of privileges. This philosophy is hard to abide by with most public API's. With traditional username and passwords, it is often possible to create role-based access with limited privileges. Developers need to be aware that API keys usually grant applications access to the entire environment. This is contrary to the principal of least privilege. Exposure of an API key cannot be contained to the minimal amount of functionality that the consuming application requires. SendGrid is one of the exceptions to this, and does an adequate job providing fine-grain control to limit the functionality that the API key is allowed to consume. For example, with CPM a developer can configure the SendGrid API key to only use the email delivery API. Sensitive functional areas such as the management interface API cannot be consumed by CPM. This limits exposure to the SendGrid account if the API key were to be exposed. As enterprises continue to migrate workloads to the cloud and advocate for more secure coding, API security and vendor platform security will continue to mature. Privileged Access Management has a place by ensuring that privileges are not Boolean and any programmatic application access also has a fine-grained privileged model.

Vendor Accounts and Remote Access

Finally, another quandary for organizations is how to extend privileged access and credential management best practices to third-party users, such as consultants or other vendors that may perform a variety of activities. How do you ensure that the authorization provided via remote access or to a third party is appropriately used? How do you ensure that the third-party organization is not sharing credentials, or otherwise exercising poor password hygiene, such as by failing to terminate authorization credentials when an employee departs from the company?

CHAPTER 22

Privileged Account Management Implementation

Organizations increasingly recognize that properly securing and controlling privileged credentials ranks as one of the best lines of defense against attacks from external hackers as well as from insiders. For optimal results, a privilege management solution should protect privileges at all stages of the cyber kill chain by implementing comprehensive layers of control and analysis. Overall objectives include the following:

- **Reduce** the attack surface by limiting the use of privileged accounts and by controlling access to shared privileged accounts across the enterprise.
- **Monitor** privileged user, session, and file activities for unauthorized access and/or changes to key files and directories.
- **Analyze** asset and user behavior to detect suspicious and/or malicious activities of insiders and/or compromised accounts.

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_22

For maximum adoption across an enterprise, a privileged access management solution must also **protect privileges** without obstructing productivity or overburdening operations.

Implementing an end-to-end privileged access management solution should follow a defined process to minimize costs and distractions, and speed results. When managing privileges as an attack vector, using a simple 12-step approach helps manage risk and provide predictable and documentable results. The result of this 12-step process is that you have greater control and accountability over the accounts, assets, users, systems, and activity that make up your privilege environment.

Throughout the process of selecting and deploying your privileged access management solution, keep in mind these business requirements, as they will help you justify the cost within your organization and risks in mitigating the threat:

- Minimize total cost of ownership;
- Provide a fast time to value;
- Deliver the best information to make the best risk-based decisions.

Remember these steps are just a guide and do not necessarily need to be followed in sequence.

Step 1: Improve Accountability for Privileged Passwords

The most logical starting point for gaining greater control over privileges is by improving accountability over privileged passwords. Not effectively managing shared accounts is a problem that has significant scale and risks. You don't have to look much further than recent breaches to understand the implications – or the challenges. Certain systems have embedded or

hard-coded passwords, leaving opportunities for misuse. In addition to supporting human interaction, passwords are needed for application-to-application and application-to-database access. Passwords are generally static, so there must be protections against passwords leaving the organization. Manual password rotation is unreliable and time consuming. Auditing and reporting on access are complex and error prone. Therefore, how do organizations ensure accountability of shared privileged accounts to meet compliance and security requirements without impacting administrator productivity?

The answer is automation – automating password and session management; providing secure access control; auditing, alerting, and recording for any privileged account – from local or domain shared administrator, to a user's personal admin account (in the case of dual accounts), to service, operating system, network device, API keys, database (A2DB) and application (A2A) accounts – even SSH keys. By improving the accountability and control over privileged access, IT organizations can reduce security risks and achieve compliance objectives. With this goal in mind, consider these 10 recommendations for every privileged access management solution:

1. Full network scanning, discovery, and profiling with auto-onboarding.
2. Build permission sets dynamically according to data from scans.
3. Automatically rotate SSH keys and cycle passwords according to a defined schedule.
4. Granular access control, workflow, and auditing.
5. A clean, uncluttered user interface for end users that speeds adoption.
6. Workflow-based and break glass options for requesting access.

7. Password and session management together in the same solution – no requirement for two different interfaces or to be charged separately for each.
8. No requirement for additional third-party tools for session management – utilize native tools and applications instead versus introducing a third-party requirement that may have its own risks.
9. Leverage an integrated data warehouse and threat analytics across the privileged landscape.
10. Flexible deployment options: hardware appliances, virtual appliances, cloud, or software for maximum coverage.

With these requirements, organizations can discover all the accounts in their environment, place those accounts under management, and satisfy auditor requests that accounts are now managed.

Step 2: Implement Least Privilege Desktops

Once accounts and assets have been discovered and are being consistently managed, the next step to complete privileged access management is implementing least privilege on end-user machines. As a security best practice, organizations should reduce the risk on desktops before servers (such as Windows, Unix or Linux as indicated in step 4) as the endpoint is typically the last mile of security. Secure the last mile first. Some organizations may choose to reverse this order, so depending on the specific business environment and risk, the priorities for these steps could be refined to match the risk level and appetite for the business. In other words, the order of these seven steps can vary but almost always step 1 is the most important and represents the highest privileged attack vector risk.

The process for IT to restrict or enable end-user privileges potentially can be complex and time consuming, but it must be done to support audit or compliance mandates. When environments have standardized desktop images and applications, the process is relatively trivial. If every machine is different, then other desktop priority management techniques might be best first. And although users should not be granted local administrator or power user privileges in the first place, sometimes certain applications require elevated privileges to run. How do IT organizations reduce the risk of users having excessive privileges and subjecting the organization to potential exploitation or compliance violations without obstructing their productivity or overburdening the help desk?

The answer is only through least privilege access for applications - rules-based technology to elevate application privileges without elevating user privileges. By eliminating end-user desktop administrator privileges, simplifying the enforcement of least privilege policies, maintaining application access control, and logging privileged activities, IT closes security gaps, improves operational efficiency, and achieves compliance objectives faster.

Therefore, the top 10 desktop least privilege capabilities should include the following:

1. Default all users to standard privileges while enabling elevated privileges for specific applications and tasks without requiring administrative credentials.
2. Enforce restrictions on software installation, usage, and OS configuration changes.
3. Eliminate the need for end users to require two accounts.

4. Make dynamic least privilege decisions for applications based on that application's vulnerability, risk, reputation, and compliance profile.
5. Match applications to rules automatically based on asset-based policies.
6. Report on privileged access to file systems for all users and document system changes during privileged sessions.
7. Monitor sessions and log keystrokes during privileged access.
8. Provide a technique for using real domain or local privileges when required, including multi-factor authentication.
9. Integrate with other privilege solutions to achieve comprehensive privileged access management.
10. Leverage an integrated data warehouse and data analytics across the privilege landscape. With this solution, customers gain the ability to efficiently eliminate local admin rights, and make intelligent application elevation decisions based on real-world privileged threats.

Step 3: Leverage Application Risk Levels

Now that shared credentials are under management and end users have the privileges they need to perform their jobs – and nothing more - organizations can move to a better understanding of risks to help make better-informed privilege elevation decisions. The challenge, though,

is that most risk assessment solutions do little to help security leaders put vulnerability, attack, malware, and risk information in the context of business. Saddled with volumes of rigid data and static reports, the security team is left to manually discern real threats and determine how to act upon them.

Therefore, consider expanding your vulnerability management and risk assessment programs to include privileged access and application control. If teams deem an application too dangerous to execute based on a real-world threat, ransomware, or missing security patch, they should adopt privilege access management policies to compensate for the risk. This is the same as reputation-based application control. This not only stops exploits from becoming a privileged attack vector but also drive-by social threats that can leverage vulnerabilities within the environment until mitigation or remediation steps are available.

Step 4: Implement Least Privilege on Servers

In current information technology environments, business critical, tier-1 applications are attractive targets for threat actors. They contain the sensitive data and applications they want. Accessing privileged user credentials for these resources can provide access to e-commerce data, ERP systems managing employee data, customer information, and sensitive financial data. Having root passwords, superuser status, or other elevated privileges is important for users to do their jobs. But unfortunately, this practice presents significant security risks stemming from intentional, accidental, or indirect misuse of those shared privileges – especially when those shared privileges have access to tier-1 systems

that impact the business such as those running on Unix or Linux servers. Traditional responses to this problem include the following:

- Are inefficient and incomplete (such as native OS options) lacking the ability to delegate authorization without disclosing passwords.
- Are not secure enough (such as open source sudo or local administrator accounts) to address risk or compliance requirements lacking the ability to record sessions and keystrokes for audits.
- Don't account for activity inside scripts and third-party applications, leaving a shortcut to unapproved applications.
- Don't offer an efficient migration path away from sudo or shared accounts if it is being used throughout the organization.

Therefore, how do IT organizations limit who has assess to root accounts to reduce the risk of compromises without hindering productivity?

Organizations must be able to efficiently delegate server privileges and authorization without disclosing passwords for root, local, or domain administrators, or other accounts. Recording all privileged sessions for audits, including keystroke information, helps to achieve privileged access control requirements without relying on native tools.

Top 10 server privilege management capabilities include the following:

1. Pluggable Authentication Module (PAM) support to enable utilization of industry-standard authentication systems.
2. Advanced control and audit over commands at the system level.

3. Powerful and flexible policy language to provide a migration path from native tools.
4. Extensive support for many Windows, Unix, and Linux platforms.
5. Record and index all sessions for quick discovery during audits.
6. Broker permissions transparently, ensuring user productivity and compliance.
7. Change management of all settings and policy configuration, allowing full audit of who has changed what, version control, and rollback of all existing configuration files.
8. REST API for easier integration with third-party products.
9. Integrate all policies, roles, and log data via a web-based console.
10. Leverage an integrated data warehouse and threat analytics across the privilege landscape.

With this capability, you gain complete control over root and administrator access on any type of server operating system.

Step 5: Network Devices

The most common username and passwords for network devices are not necessarily the defaults that come with the device even though we are now very much aware of the risk. Most administrators change them. Unfortunately, in some environments, they can be guessed or compromised using brute force password attacks. In addition, the second

most common privilege flaw is to use the same ones across the entire infrastructure (password reuse) and rarely, if ever, are they changed in mass, even if you have outsourced the management. This problem can lead to a variety of malicious activities, including recent vulnerabilities that can replace the device's bootstrap loader with a piece of custom malware.

The risks can stem from a simple lack of privileged account management on network devices include these:

- Default or common passwords that are not configured correctly;
- Shared credentials across multiple devices for management simplicity;
- Excessive password ages due to fear of changing or lack of management capabilities;
- Compromised or insider accounts making changes to allow exfiltration of data;
- Outsourced devices and infrastructure where changes in personnel, contracts, and tools;
- expose credentials to unaccountable individuals.

Anyone of these could lead to excessive risk for your infrastructure. As such, organizations should look beyond desktops and servers when planning their Privilege Account Management security program by including these devices. Additionally with newer privilege solutions organizations can move beyond the boolean "access" or "no access" authorization models commonly used in many network devices. Organization now have access to proxy gateways that can enforce command whitelisting\blacklisting, session monitoring, active alerting and more.

Step 6: Virtual and Cloud Data Centers

Growing use of virtualized data centers and cloud environments for processing, storage, or application hosting and development have opened up new avenues for would-be hackers or malicious insiders to access sensitive data and disrupt organizations inappropriately. Despite these risks, cloud adoption continues to accelerate. As such, organizations must secure access to these environments to mitigate security risks while meeting the cost and efficiency demands of hosting more applications and services in the cloud.

Like traditional desktops and servers, unknown or undermanaged virtualized and cloud environments can create a significant security gap that opens networks to security breaches, data loss, intellectual property theft, and regulatory compliance issues. The first step in getting control over these assets is discovery. There are several techniques used to discover assets in virtualized and cloud environments including the following:

- Performing standard network discovery or scanning from a host machine with "line of sight" access to the virtualized environment;
- Querying the Hypervisor or Cloud Management Platform to retrieve the inventory of virtualized assets, or configuring an active notification upon inventory updates;
- Using agents that are preinstalled on the base image library, or that are installed during the normal server provisioning process;
- Querying a third-party asset management solution.

Once cloud instances are found, they must be managed to limit exposure. From a privileged management perspective, the options to secure these assets are like that of traditional desktops and servers:

- Use a password vaulting solution to manage the passwords across all virtualized machines automatically;
- Use a session management solution to control and monitor virtual machines access;
- Use native delegation capabilities of the underlying O\S to reduce the privileges associated to users interacting with the system;
- Use a privilege management agent with least privilege architecture to reduce exposure to administrator, root, and privileged developer accounts.

Now that the virtualized machines are under control, what about the hypervisor and cloud management platform itself? Here again, inappropriate or malicious activates at this management level could have a devastating impact on the business. This includes administrators of your VMWare, Microsoft Hyper-V, Amazon AWS, and Microsoft Azure environments. To counteract this threat, organizations again have several options:

- Use a password vaulting solution to automatically manage the passwords across all hypervisor and cloud management platforms;
- Use a session management solution to control and monitor all cloud management activities;
- Use native or third-party delegation capabilities of the hypervisor and\or cloud management provider to reduce the privileges associated to users interacting with the system.

Step 7: IoT Devices

With a growing number and sophistication of software attacks, it has become significantly more challenging for organizations to protect their environments. Recently a new generation of distributed denial of service attacks has emerged that represents a significant risk to organizations and governments alike. Like a lot of IT terms, the definition of IoT is open for interpretation. Typically, we think of IoT devices as being DVR's, CCTV, microphones, webcams, home automation, etc. But in reality, it can mean anything connected to the Internet, including video conferencing equipment, network printers, and more.

The number one vulnerability with IoT devices is the use of hard-coded, default, and/or weak passwords. Even when administrators change default passwords, most credentials can be still guessed via brute force attacks, especially when weak or shared passwords are used across the IoT infrastructure.

Step 8: DevOps

DevOps is a compound acronym for Software **DEV**elopment and Information Technology **OP**erationS. It is a designation for the communication and synergy between software developers and information technology departments. The goal of DevOps is not typical software application development but focuses on the programmatic automation of infrastructure management, whether it is software delivery, instance management, or automation for rapid deployment of resources and their corresponding operations management.

For commercial application developers, or programmers that create custom DevOps applications for your business, consider how beneficial it would be for your end users, or other applications, to never require entering a username and password for connectivity. If the tools stored credentials automatically or queried a management solution to prove

authorization, end users like database administrators would never need administrator rights to access a database. Management tools for services, remote access, and infrastructure would automatically recognize the logged-on user, the asset they are on, be fully context aware, and seamlessly request and pass credentials. Privileged Access Management solutions for password management make this capability a potential reality using an Application Program Interface (API) to set, retrieve, and process credential and password requests. Some of the benefits of this approach for DevOps are the following:

- Secure Applications – Privileged Access Management (PAM) API's are designed to provide better security for all applications that require a user or application to enter static credentials for normal operations. Developers can call a PAM API and retrieve the latest credentials for the user, application, infrastructure, cloud solution, or database to authenticate and release the credentials upon termination of the session. This can trigger automatic, randomized cycling of the password or other automated processes to meet business objectives. Users never see, or know, the latest credentials for any given resource or application.
- Attack Vector Mitigation – Using a PAM API secures the runtime of applications and avoids hacking techniques like Pass-the-Hash. This approach is far more secure than Single Sign-on (SSO) since the password is constantly being rotated per session, user, or other criteria, even if it is shared.
- Developer Simplification – This approach improves the agility and responsiveness of IT by never requiring the entry of a username and password for connectivity to

create custom applications. End users, like database administrators, never need administrator rights to access a database if the tools retrieve stored credentials automatically.

Step 9: Unify Management

It is no secret that information technology and security professionals are overloaded with privilege, vulnerability, and attack information. Unfortunately, advanced persistent threats (APTs) often go undetected because traditional security analytics solutions are unable to correlate diverse data to discern hidden risks. Seemingly isolated events are written off as exceptions, filtered out, or lost in a sea of data. The threat actor continues to traverse the network, and the damage continues to multiply. How do security and IT operations teams gain an understanding of where threats are coming from, prioritize them, and quickly mitigate the risks?

Data analytics enables teams to identify the data breach threats typically missed by other security analytics solutions. Solutions pinpoint specific, high-risk users and assets by correlating low-level privilege, vulnerability and threat data from a variety of third-party solutions.

Therefore, any data analytics and unified management solution should contain the following top 10 capabilities:

1. Correlate low-level data from a variety of third-party solutions to uncover critical threats.
2. Correlate system activity against application risk data and malware.
3. Report on compliance, benchmarks, threat analytics, what-if scenarios, resource requirements, and more.

4. View, sort, and filter historical data for multiple perspectives.
5. Locate network (local and remote), web, mobile, cloud, and virtual assets, as well as privileged accounts.
6. Profile IP, DNS, OS, Mac address, users, accounts, password ages, ports, services, software, processes, hardware, event logs, and more.
7. Group, assess, and report on assets by IP range, naming convention, OS, domain, applications, business function, Active Directory, and more.
8. Import from Active Directory, LDAP, IAM, or set custom permissions.
9. Workflow, ticketing, and notification to coordinate IT and security teams.
10. Share data with leading SIEM, GRC, NMS, and help desk solutions.

By unifying privileged access management and other threat management solutions, IT and security teams have a single, contextual lens through which to view and address user and asset risk by activity, asset, user, and privilege.

Step 10: Privileged Account Integration

Please consider step 4 for a moment. Once you have greater control over privileged access in server environments, the next logical step is to bring those systems under consistent management, policy, and single sign-on. Unix, Linux, and Mac have traditionally been managed

as stand-alone systems – each 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 or cloud environment – leads to inconsistent administration for IT, unnecessary complexity for end users, and a vast sprawling of alias accounts. These are known threats and areas of interest for a threat actor.

Therefore, how do IT organizations achieve consistent policy configuration to achieve compliance requirements, a simpler experience for users and administrators, and less risk from an improperly managed system?

The ideal solution is to centralize authentication for Unix, Linux, and Mac environments by extending Microsoft Active Directory's Kerberos authentication and single sign-on capabilities to these platforms. By extending Group Policy to these non-Windows platforms you gain centralized configuration management, reducing the risk and complexity of managing a heterogeneous environment and stop the sprawl of alias accounts.

The Top 5 Active Directory bridge capabilities should include these:

1. No requirement to modify Active Directory schema to add Linux, Unix, or Mac OS X systems to the network. This provides stability as the technology evolves.
2. Provide a pluggable framework with an interface similar to Microsoft's Management Console on Linux or Mac OS X, and full support for Apple's Workgroup Manager application would allow for seamless management and control of Mac system settings,
3. Single sign-on for any enterprise application that supports Kerberos or LDAP,

4. Provide a single familiar toolset to manage both Windows and Unix systems (ex: Active Directory Users and Computers, ADUC),
5. Allow users to use their Active Directory credentials to gain access to Unix, Linux and Mac, consolidating various password files, NIS, and LDAP repositories into Active Directory and removing the need to manage user accounts separately.

These concepts will enable simplified configuration management and policy for non-Windows systems and will help improve security and the user experience. This approach will help your organization be more efficient by reducing the number of logins (and the accordant help desk calls when they are forgotten), and the number of different systems, configurations, and policies to manage. Thus, the lower number of accounts, the less to audit and lower the risk surface for a threat actor.

Step 11: Auditing and Recovery

Once you have your non-Windows systems integrated into Active Directory, the next step is to audit user activity to gain additional insight into AD changes that could impact the business. But trying to keep up with all the changes made manually in Active Directory is an extremely time-consuming and complex process, with delays in discovering and addressing changes possibly leading to business disruption, not to mention the security and compliance implications of such changes.

When you include other Microsoft technology in the mix, understanding the “Who, What, When, and Where” of changes across the Windows infrastructure is even more complex.

Therefore, how do IT organizations better understand changes, have the capability to roll them back if necessary, and establish the right entitlements in the first place across a complex Windows infrastructure so they can more effectively protect the business?

Organizations need centralized real-time change auditing for Active Directory, File Servers, Exchange, and SQL, as well as the ability to restore Active Directory objects or attributes and to establish and enforce entitlements across the Windows infrastructure. Through simpler administration, organizations can mitigate the risks of unwanted changes performed by threat actors, insiders, and better understand user activity to meet compliance requirements.

To perform these necessary forensics tasks, consider these top 3 auditing and protection capabilities:

1. Audit and identify who, what, where, and when changes were performed.
2. Provide a mechanism for Active Directory backups and recovery. Rolling back an accidental (or threat actors) permission change could level unforeseen gaps in security.
3. Audit and report across multiple Windows domains and trusted servers.

With this capability, you gain detailed, real-time auditing of AD environments, and the ability to restore unwanted changes when threats or even mistakes arise. If you know that a threat actor has granted themselves privileges, would you want to know too? Auditing and recovery of privileges in AD is a simple step to identify and mitigate this risk also.

Step 12: Integrate the Identity Stack

Identity and access management (IAM) plays a critical role in an organization's IT security strategy. As organizations grow, so do the number of applications, servers, and databases used. Access to the organization's resources is typically managed through IAM solutions, which offer capabilities like single sign-on, provisioning, user management, access control, and governance. But securing an organization's sensitive data and applications requires more. Provisioned users, regardless of privileges, can leave an organization exposed if activity of their usage is not monitored and documented properly. Identity and access management solutions help IT teams answer 'Who has access to what?' But, to achieve complete user visibility, privileged access management solutions address the remaining questions: 'Is that access appropriate?' and 'Is that access being used appropriately? That is, PAM solutions should be providing more visibility and deeper auditing of the access and use of privileged accounts. Many times, IAM solutions will add users to a system or applications group, but will not provide the details as to what access that group membership provides, or access to the detailed session log or keystrokes collected during the privileged session. As such, PAM extends the visibility of the IAM solution to tighten security and audit controls further.

CHAPTER 23

Key Takeaways

Privileges as an attack vector represent the lowest-hanging fruit for threat actors in today's next-generation digital economy. While architecting and securing an environment is still relatively complex, these Top 20 recommendations can help any security professional achieve their goals and minimize risks to the business.

1. Use Standard User Accounts – Enforce that all users have a standard user account. Administrators across all platforms should log in with their standard accounts as normal practice. They should only log in with administrative rights when they need to perform administrative tasks.
2. Never Share Passwords – The risks of a shared password from peer to contractor just elevate the risk of the password being misused and shared by a threat actor.
3. Never Reuse Password – If one resource is compromised, then every other resource with the same shared password is at risk.
4. Never Store Passwords in Clear Text – Passwords should be kept secret. They should never be in plain sight, no matter how they are stored.

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_23

5. Secure Passwords – If passwords need to be documented, they should be on an encrypted file, secured file system, and locked away in a physical safe as required.
6. Minimize the Number of Aliases – Making people trackable and not hackable is key to detecting privileges used as an attack vector.
7. Minimize the Number of Administrative Accounts – The lower the number of privileged users, the lower the privileged risk surface and less to monitor and audit for privileged activity.
8. Rotate Passwords Frequently – Passwords should be rotated after every use for privileged activity or on a regular schedule for standard accounts. This keeps them from becoming stale.
9. Ensure Passwords Are Exceptionally Complex – Privileged passwords should not be humanly readable. This keeps them for from being copied or verbally discussed easily. Every password should be complex, but some should be more complex than others to remove the human risk element out of the equation.
10. Require Multi-Factor Authentication – Implement multi-factor authentication for access to internal systems, applications, and even data. While implementing static multi-factor based on whether a system or application is good, getting too restrictive can become frustrating for users. Look for solutions that can also restrict access based on the risk associated with the environment or activity.

For example, if someone tries to launch a sensitive application after hours for the first time, or tries to run a sensitive command on the Unix server that is missing critical patches, step up the security and trigger to reauthenticate with multi-factor.

11. Implement Application Control (whitelisting, blacklisting, and greylisting) – Implement policy to allow known good applications and log all other applications and launch attempts. If possible, restrict launching of end-user applications with critical known security vulnerabilities.

12. Enforce the Principle of Least Privilege – If a user does not need access to systems, applications or data, remove it. Remove administrator rights on desktops for all users. (Consider augmenting least privilege with file integrity monitoring to ensure appropriate use and to more practively detect compromized account activity)

13. Automate Password Management – Control and audit requests for administrative passwords. Require unique passwords across all privileged systems and accounts.

14. Go Beyond Passwords – Eliminate hard-coded passwords in service accounts and scripts. Implement SSH key management tools.

15. Use Context-Based and Adaptive Access Controls – At some point, people need access to do their jobs, but continue to lock down when they have access, and from which location they have access. Restricting access based on static elements like

time of day or subnet is good, but restricting access dynamically based on risk (i.e., does a ticket exist for the access, does this request adhere to normal access patterns, have I received recent alerts from my threat detection layers, etc.) adds greater protections.

16. Monitor All Sensitive Privileged Session Activity (especially to Crown Jewels) – Any type of privileged activity to the crown jewels should be session recorded, keystroke logged, and application monitored to review for inappropriate activity.

17. Understand Obligations to Auditors and for Compliance – Security professionals perform all of these functions to secure a business. They should not do them as a checkbox for compliance. Understanding what is required and the best way to meet the mandates make everyone more secure and ultimately, auditors happy (if there is truly ever such a thing).

18. Implement Threat and Advanced Behavior Monitoring – Somewhere along the line, accounts have access to stuff. Implement base security event monitoring and advanced threat detection (including user behavior monitoring) to more accurately and quickly detect compromised account activity, as well as insider privilege misuse and abuse.

19. Segment Your Network – Group assets, including application and resource servers, into logical units that do not trust one another. Segmenting the network reduces the "line of sight" access attackers

must have into your internal systems. For access that needs to cross the trust zones, require a secured jump server with multi-factor authentication, adaptive access authorization, and session monitoring. Where possible, go beyond standard network segmentation. Segment based on the context of the user and privileges; and the resources, applications, and data that they are accessing. This is also known as micro-segmentation.

20. If You Are NOT Having Fun, You Should Get a Different Job – If a security professional is unhappy, they are not doing their job correctly. All the items above are potentially at risk, and so is the business. Security professionals need to be happy with their work, satisfied with the environment, and challenged on a regular basis. Security is ever changing, compliancy in security is death, and being unhappy will let the latest threat walk right past you. A threat actor does not care if you are happy or not, they just want your root accounts. Therefore, someone always needs to mind the store that cares and will respond.

CHAPTER 24

Conclusion

Surrounded by a team of professionals focused on privileged access management, I am constantly involved in what would be considered research activities that include ongoing outreach to customers, advisory council members, industry leaders, and analysts that are all motivated to solve real-world challenges for today's complex security demands. This outreach yields the following predictions about how PAM will evolve in the future:

1. PAM Is a Security Layer

Privilege Account Management enables a secured life cycle for privileged credentials to facilitate secure authentication for users and applications to resources with added layers of process, control, and auditing. Today most organizations lie somewhere on the continuum between manual and automated processes in the privilege access management approach. With threats in the form of well-armed hackers seeking to compromise privileged users and insiders who've honed internal knowledge of systems and resources, organizations granting excessive access or relying on manual processes to manage passwords are at a considerable disadvantage. Though still gaining awareness, PAM is a foundation security layer that must be incorporated into every organization's security program. As the market matures, PAM will become commonplace and will be more closely integrated with the broader identity and access management life cycle.

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0_24

2. Simplification of PAM

Driven by customer demand for simplification and consolidation, PAM vendors will continue to expand and simplify existing integration interfaces. More and more organizations are looking to simplify the toolsets used to manage their internal security and compliance environments. Historically, many organizations have purchased PAM solutions to address specific audit findings or challenges in isolation, but this has resulted in islands of tactical products, consoles, and processes not effectively being managed. Leading Analysts recognize each of these as components of a successful PAM strategy, and must be a unified offering and not a loose collection of products.

3. Compliance as a Driver

Traditionally, many organizations focused PAM deployments on a small subset of the most critical servers and applications. Moving forward, organizations will continue to expand their PAM footprints to meet regulatory requirements, tighten security, and streamline operations. Organizations will recognize PAM as a fundamental security layer and will continue working toward ensuring complete coverage across everything (Unix, Linux, Mac, Windows, virtual, cloud, critical infrastructure, applications, SaaS, IoT, and DevOps processes) and not just as a compliance checkbox. As organizations continue to make strategic business investments that rely on specific technologies, it is critical that these initiatives be protected from malicious insiders and external threats.

4. Dynamic Policy

Organizations will become more strategic with respect to PAM. While PAM data provides valuable context that can be used in broader security and fraud detection systems, adding additional context within the policy modules will help further tighten privilege policy based on environmental and other risk factors. Policies will become adaptive, dynamic, and change automatically based on context to meet modern threats.

5. Proactive Analytics

As PAM solutions begin to consolidate and correlate information, the volume of events, session logs, and data will continue to increase. While correlating reports across the various platforms and PAM modules will help, more sophisticated, automated analysis is required to enable organizations to "cut" through the noise to detect privilege misuse and abuse. PAM security programs will mature and implement behavioral and predictive analysis that includes dynamic baselining and threshold management to detect anomalies and generate alerts and reports automatically. Threat actors are evolving and PAM will not remain static to combat these threats.

In conclusion, PAM in the foreseeable future will not stop with identities, accounts, credentials, and passwords. It will continue to evolve and the next generation set of solutions will look more like a complete fraud or intrusion prevention system then just managing privileges and passwords. The lowest hanging fruit, or the slowest runner away from the bear, will be the one who does not embrace this as a strategy and potentially get breached.

Index

© Morey J. Haber and Brad Hibbert 2018
M. J. Haber and B. Hibbert, *Privileged Attack Vectors*,
https://doi.org/10.1007/978-1-4842-3048-0

D

E

F

G

H

I, J, K, L

M

N

O

P

Q

R

S

T, U

V, W, X, Y, Z