LEARNING MADE EASY

Venafi Special Edition

SSH Machine Identity Management

for dummies®
A Wiley Brand

Learn about SSH machine identities

Understand SSH machine identity risks

Proactively manage SSH machine identities

Brought to you by

VENAFI®

About Venafi

Venafi is the cybersecurity market leader in and the inventor of machine identity management, securing machine-to-machine connections and communications. Venafi protects machine identity types by orchestrating cryptographic keys and digital certificates for SSL/TLS, SSH, code signing, mobile, and IoT. Venafi provides global visibility of machine identities and the risks associated with them for the extended enterprise — on premises, mobile, virtual, cloud and IoT — at machine speed and scale. Venafi puts this intelligence into action with automated remediation that reduces the security and availability risks connected with weak or compromised machine identities while safeguarding the flow of information to trusted machines and preventing communication with machines that are not trusted.

With more than 30 patents, Venafi delivers innovative solutions for the world's most demanding, security-conscious Global 5000 organizations and government agencies.

For more information, visit http://venafi.com.

SSH Machine Identity Management

Venafi Special Edition

for dummies®
A Wiley Brand

SSH Machine Identity Management For Dummies®, Venafi Special Edition

Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, or how to create a custom *For Dummies* book for your business or organization, please contact our Business Development Department in the U.S. at 877-409-4177, contact info@dummies.biz, or visit www.wiley.com/go/custompub. For information about licensing the *For Dummies* brand for products or services, contact BrandedRights&Licenses@Wiley.com.

ISBN: 978-1-119-78786-0 (pbk); ISBN: 978-1-119-78787-7 (ebk). Some blank pages in the print version may not be included in the ePDF version.

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Publisher's Acknowledgments

Some of the people who helped bring this book to market include the following:

Project Manager and Development Editor: Carrie Burchfield-Leighton

Sr. Managing Editor: Rev Mengle

Acquisitions Editor: Ashley Coffey

Business Development Representative: Molly Daugherty

Table of Contents

Introduction

Did you know that businesses spend billions of dollars each year on identity and access management? And almost all this money is spent on protecting the digital identities — usernames and passwords — of humans. Businesses spend almost nothing on managing machine identities such as Secure Shell (SSH) keys, even though their entire digital economies hinge on secure communications between machines. As businesses increasingly transform their operations to be primarily digital — a trend called *digital transformation* — the need for secure machine-to-machine communications and connections is even more critical.

About This Book

Welcome to *SSH Machine Identity Management For Dummies*, Venafi Special Edition. This book helps you understand where SSH machine identities are used in your network and what you need to do to keep these identities from being misused. You discover how SSH machine identities contribute to your security strategy and how to effectively manage the growing number of machine-to-machine connections that your infrastructure requires. This book makes clear why managing SSH machine identities should be a priority in your organization.

Feel free to explore the information in this book as you wish; immediately go to any part that interests you or read it from cover to cover. We wrote this book with a sequential logic, but if you want to jump to a specific topic, you can start in any chapter to extract good stuff.

Foolish Assumptions

When writing this book, we knew that the information would be useful to many people, but we have to admit that we made a few assumptions about who we think you are:

- You want to learn more about the weakest areas of your organization's security program.

- You're responsible for managing your organization's encryption assets, or you manage this function within your organization's security or operations group.
- You're somewhat familiar with encryption and security.
- You want to discover the easiest, most effective, and direct way to manage and protect your SSH machine identities.

Icons Used in This Book

We occasionally use special icons to focus attention on important items. Here's what you'll find in this book:

REMEMBER

The Remember icon highlights important facts about SSH machine identities and their effective management. So sip your drink and read on.

TIP

The Tip icon gives you the best ways to lower SSH machine identity risks. This content helps you get the most out of your management efforts.

WARNING

The Warning icon flags risky situations that, if not dealt with, can leave your organization more vulnerable to cybercriminal attacks. The information in these sections helps you prioritize your SSH machine identity management program tasks.

TECHNICAL STUFF

The Technical Stuff icon notes when the book goes a little deeper into the nitty gritty of SSH machine identity management. You don't need this information to understand the rest of the book, but this gives you techie types the details you crave.

Beyond the Book

This book can help you discover more about SSH machine identity management, but there's also only so much we can cover in these pages. If you want an accurate and prioritized view of your enterprise SSH risks and further mitigation recommendations, sign up for a free confidential SSH risk assessment at `www.venafi.com/ssh/risk-assessment`.

IN THIS CHAPTER

- **Explaining SSH machine identities**
- **Looking into how SSH machine identities are used**
- **Understanding why SSH machine identities are at risk**

Chapter 1
Defining SSH Machine Identities and How They're Used

IT and security teams use Secure Shell (SSH) to safeguard administrative access and automated processes for machines in their organizations. However, recent trends in digital transformation have spurred tremendous growth in the number of machines that organizations rely on. These new developments push IT system administrators to aspire to new levels of productivity via automation. Many administrators achieve this productivity by creating and deploying SSH keys, which establish fast, secure, automated connections to critical assets.

These SSH keys serve as machine identities, identifying and authenticating administrators and machines for critical business functions. But history shows how easy it is for organizations to lose track of SSH keys, which can lead to the misuse of privileged access on sensitive internal systems. Poor SSH configuration and management practices have left many organizations vulnerable to cybercriminals, insider threats, and failed audits — leaving IT and security teams without a clear understanding as to what went wrong.

Although SSH keys are a vital machine identity type used to identify and authenticate administrators and machines for critical business functions, these critical security tools are routinely left untracked, unmanaged, and unmonitored. This chapter identifies what SSH machine identities are and how they're used to secure your organization.

Defining SSH Machine Identities

SSH is a cryptographic network protocol that gives users — particularly system administrators — a secure way to access machines over an unsecured network. SSH provides strong authentication and encrypts data communications between two machines connecting over an open network such as the internet. System administrators use SSH to perform remote administration for nearly all systems, to automate operations, and to transfer files between systems.

The SSH protocol comes in these versions:

- **SSHv1:** The first version uses private Rivest–Shamir–Adleman (RSA) keys to decrypt challenges encrypted with the corresponding public key. But SSH protocol version 1 is limited in its support of message authorization codes, compression algorithms, and the algorithms necessary for key exchanges.

  TECHNICAL STUFF

  Since 1998, risks with SSHv1 have been known. And even after all this time, you'd think SSHv1 wouldn't persist in enterprises, but it does. This may be a result of administrators not upgrading to the latest version of OpenSSH, not disabling SSHv1, or possibly believing that legacy applications "need" it to function.
- **SSHv2:** Version 2 of the protocol requires that the client sign a message and transmit the signature (not the message) with the public key used. The server then recreates the message and verifies the server. SSHv2 is also not a monolithic protocol — it's made up of a series of protocols that includes improved public key certification, encryption standards, and even support for public key certificates.
- **OpenSSH:** This open source implementation of the SSH protocol is freely available and is delivered as a source code.

In all versions, SSH keys serve a crucial function in protecting the information that your organization values most. Therefore, it's in your best interest to effectively manage SSH keys.

Recognizing How SSH Machine Identities Work

SSH encrypts data exchanged between two parties by using a client-server model. The server listens to a designated port for connections, while the client is responsible for the Transmission Control Protocol (TCP) handshake with the server. That initial connection sets the stage for the server and client negotiating the encryption of the session based on what protocols they support.

During the connection process, SSH leverages two types of SSH keys as machine identities:

- **Host keys:** SSH uses host keys to guarantee the authenticity of the server and create the encrypted tunnel.
- **Authorized keys:** SSH users can place authorized keys on the server to grant them access without using passwords, which simplifies day-to-day work. This method is commonly referred to as *public key authentication*.

REMEMBER

Both host and authorized keys work in public-private pairs and must be administered together with a slew of security options (config files) such as the encryption algorithm, access levels, port forwarding, key length, and passphrase.

When a system administrator executes an SSH command, the SSH client and server engage in six steps:

1. **Send a connection request.**
2. **Authenticate back to the client by using the unique server host SSH key.**
3. **Set up an encrypted channel.**
4. **Authorize session access by using passwords or public key authentication.**
5. **Submit a session info request by executing shell commands.**
6. **Return session info when the SSH server sends the command return data back to the client.**

Seeing How SSH Machine Identities Are Used

Built into most operating and network systems, SSH has become the de facto interface standard for remote system access. SSH popularity is due largely to its security features, versatile usage, and baked-in automation. After SSH keys are put in place to enable client authentication, they can enable ongoing, automatic connections from one system to another, without needing to enter a password. Today, this broadly adopted cryptographic protocol is used by a majority of system administrators as well as many automated processes.

In this section, we cover the ways that system administrators may use SSH in your organization.

Securing privileged access

SSH is often used to safeguard administrative access for organizations — securing system-administrator-to-machine access for routine tasks. SSH keys ensure that only trusted users and machines have access to sensitive network systems and data. Administrators rely on SSH as an encrypted protocol to authenticate privileged users, establish trusted access, and connect administrators and machines.

TECHNICAL STUFF

You may be surprised how many systems in your organization rely on SSH keys for privileged administrative access and secure machine-to-machine automation. The short list includes application servers, routers, firewalls, virtual machines, cloud instances, and other devices and systems that leverage SSH. Like most large organizations, you're probably using SSH with thousands of systems.

Automating routine processes

The SSH protocol is multifaceted and encompasses many functions, which have been adopted by a wide variety of automation tools. SSH is also used to secure the machine-to-machine automation of critical business functions, such as automatically triggering operations and routine file transfers. SSH keys ensure that only trusted users and machines have access to sensitive network systems and data. For more on automation, check out Chapter 5.

Controlling cloud access

Cloud servers often have SSH enabled for remote administration and maintenance of servers. SSH is widely used in cloud environments because of the encrypted communication channel it provides the client and server. An SSH keypair may be generated by the cloud provider, delivering the private key to the user and retaining a copy of the public key. The public SSH key is used to generate an administrator password or used directly for logging into a virtual machine.

WARNING

Private SSH keys can be exposed when they're accidentally, insecurely, or publicly stored in the cloud.

Securing DevOps

Development operations (DevOps) teams focus on speeding up the delivery of products and services. To do this, developers need access to cloud-based, self-contained runtime environments, known as *containers*, to run individual modules called *microservices.* This access is often secured by SSH machine identities.

WARNING

In the fast-paced world of DevOps, traditional SSH controls may not be able to cope with new environments and can slow the delivery of IT services. The resulting frustration can cause developers to avoid encryption altogether, take shortcuts with SSH keys, or otherwise skimp on the security of machine identities. When this happens, it exposes your organization to unnecessary security vulnerabilities.

Why SSH Machine Identities Are at Risk

Even though SSH keys can grant root access and privileges to critical systems and data, most organizations don't know how widely SSH keys are used. Many organizations learn too late that they have hundreds of thousands of SSH private keys they were previously unaware of, and most of these keys aren't as tightly controlled as their level of privilege requires.

SSH keys and the connections they enable have gained in popularity significantly over the last several years. Yet, SSH deployment and its related configuration can leave organizations vulnerable if not done securely. In this section, you discover the most imminent threat risks.

SSH keys never expire

Unlike Secure Sockets Layer (SSL) or Transport Layer Security (TLS) certificates that include metadata like subject domain name, subject organization, issue date, expiry date, and more, SSH keys don't include that data, so they're difficult to track or manage, and they never expire. Because SSH keys never expire, when a system administrator leaves the organization or an IT automation process gets removed, related keys may still be located in various files and accessed by unauthorized users.

TIP

For these reasons, make sure to regularly rotate SSH machine identities to minimize the risks that can occur if these keys are left on your network indefinitely.

No visibility into where SSH keys reside

Most organizations have no insight into the number of SSH keys they're actively using. Visibility is, in most cases, the starting point for improving SSH key management. Without this visibility across your organization, cybercriminals have a broad attack surface to exploit thousands or even millions of untracked SSH keys in enterprises.

No automated way to remove unused SSH keys

Without automated rotation, the number of SSH keys in your organization can build over time. This happens when your users copy and share SSH keys to simplify administration across systems, or keys aren't removed after employees are terminated or reassigned.

No assigned responsibility for SSH key security

Most organizations allow their administrators to manage and configure their own SSH keys. When you entrust high levels of privileged access to folks who often have to prioritize speed and efficiency over security, you end up with inconsistent security controls, or worse, a compromise of privileged systems or data.

SSH mitigation difficult and time consuming

After an SSH key gets compromised and the attacker gains access, mitigating all SSH keys can be difficult. Most organizations don't have an inventory of their SSH keys, and revoking all keys will more than likely stop certain critical IT processes.

THE FUTURE OF SSH

To solve some pain points that come with standard SSH keys, many organizations want a better solution, so they're looking to SSH certificates to solve those issues in the future. Unlike SSH keys, SSH certificates offer the following benefits:

- They're digitally signed and include metadata, which allows you to track usage and assign expiration dates.
- They can be cryptographically verified and, like traditional SSH keys, are exchanged between client and host during the SSH handshake.
- SSH certificates are only valid for a specific period, and after that, they'll no longer be trusted. This passive revocation minimizes the exposure of a compromised private key.
- With SSH certificates, the onboarding process is as simple as issuing a new certificate for every new administrator. Because servers trust a certificate authority, instead of individual public keys, you don't need to configure each server.
- SSH host certificates allow multiple hosts to share a principal without sharing the same private key. With traditional SSH keys, you need to put the public keys of all your servers on all client devices, which can increase exposure to compromise.

IN THIS CHAPTER

- **Recognizing the challenges of SSH machine identity management**
- **Seeing how weak implementations play a part**

Chapter 2 Understanding SSH Machine Identity Challenges

Even though Secure Shell (SSH) is used for privileged access, most organizations have no inventory of the trust relationships enabled by SSH machine identities. Many IT organizations haven't changed or rotated SSH keys because they're wary of inadvertently increasing the risk of taking down an application if machine-to-machine connections fail. This concern is largely due to the fact that SSH works through tightly coupled private and public keys. If you update a private key but don't update all the corresponding authorized public keys, you may cause a critical automated process to stop working. It only takes one of those management meltdowns for IT teams to decide that they *never* change or touch SSH keys.

In this chapter, you explore some of the reasons that SSH machine identities can be challenging to manage.

Understanding Why SSH Machine Identity Management Can Be Challenging

Several factors drive the increased usage of SSH machine identities and the corresponding escalation of management challenges. When these factors aren't addressed, you may face severe consequences.

Defining the management challenges

In this section, we give you the reasons why SSH machine identity management can be so hard.

Difficulty of monitoring SSH usage

SSH is a valuable security asset because it allows strong encryption. However, once data is encrypted, it becomes expensive for security operations teams to monitor the data inside the encrypted channel. Some security administrators have given up on monitoring SSH completely, and it's not uncommon to hear comments such as "If it's encrypted, it must be good."

Setting up key life cycle isn't an IT priority

Generally speaking, security operations teams aren't big fans of elaborate approval processes. SSH is no exception. Unfortunately, that includes SSH key life cycles, including obtaining change approvals, assigning, auditing, approving, and rotating SSH keys. Without automated processes (see Chapter 5), all these life cycle functions can easily become part of an arduous workflow.

Vendor misconfigurations

It's not only your own IT teams that can create SSH vulnerabilities. Many vendors can inadvertently do it by leaving identical manufacturing root keys on the device to be installed. These keys are easy to get to by just opening a new box but can also be found on the internet.

Rapid adoption of DevOps and cloud

Development operations (DevOps) teams favor SSH access to enable rapid, frequent, and highly automated build and release processes. To accommodate this, cloud service providers offer fast and easy deployment of application services in cloud environments, but this rapid implementation opens more space for errors and misconfigurations by their users.

TECHNICAL STUFF

According to a recent report by Unit42, 32 percent of exposed hosts in public clouds have open SSH services, and 47 percent of the SSH servers on Azure virtual machines had password authentication enabled — making them potentially vulnerable to brute-force attacks.

Seeing the consequences of poor management

Management of SSH keys is most often left in the hands of IT administrators who manually manage them for the systems they control, skipping security best practices or using inconsistent policies. Despite the never-expiring and sweeping access they grant, SSH keys are left untracked, unmanaged, and unmonitored by most organizations. A continuously growing set of SSH machine identities creates several risks.

SSH key sprawl

A lack of governance in creation and management of SSH keys can lead to the reckless proliferation of keys, which can, in turn, lead to unauthorized access that's difficult to detect. For instance, SSH keys delivering privileged access can get duplicated or shared between users, making the connections less private and more prone to attacks.

Lost or stolen SSH keys

SSH credentials can be stolen or compromised in many ways, such as administrators getting tricked by a phishing attack or a malware using one-day vulnerability, which is slowly extracting data. SSH keys are defined in a file, easy to recognize, and stored on both sides of a connection. As a result, malware or malicious insiders can easily misappropriate keys, opening the door for an intruder to start a privileged SSH administrative session.

WARNING

After an SSH key has left your organization, you're challenged to limit the exposure, and responding can quickly become expensive.

Slow incident response processes

When a security incident occurs, responders need to take action and remove all potential access paths available to the intruder. A dense and uncontrolled mesh of SSH machine identities with hundreds of thousands — if not millions — of keys can be hard to clean up and consumes costly resources, which allows cybercriminals extra time to leverage the privileged access they've acquired.

Putting the Blame on Weak Implementations

As a result of poor SSH practices, SSH machine identities can become a security liability for information security teams, causing related assets and services to become more vulnerable to unauthorized access and misuse. After an SSH machine identity gets into the wrong hands, an attacker can gain unauthorized access to mission-critical systems, move laterally from system to system, and circumvent security controls.

In this section, we give you the common practices that can increase the vulnerability of SSH machine identities.

Unknown or unmanaged SSH keys

Throughout the years, administrators in any organization come and go, and there often isn't a formal process to clean up the SSH keys that they leave behind, or you can't track where all of them may be. This lack of process leads to a problem when old administrators leave and new administrators come onboard — likely not knowing what the old keys left behind are used for. So, out of fear of causing an SSH-related outage, new administrators don't remove old keys and instead generate new SSH machine identities — with the ssh-keygen command — to take care of their administrative tasks and build new keys into their own automated processes.

WARNING

The result of this faulty process is an unmanaged tangle of SSH trust relationships that can leave you and your company vulnerable. If an SSH key is compromised and regular rotation isn't enforced, your organization is at risk for repeated unauthorized access — indefinitely.

No usage restrictions

Because they have a simple file format, SSH keys can easily be copied or shared, making the automated connections they enable subject to compromise and misuse. Without implemented usage restrictions, it will be difficult to know who's using your SSH keys and for what purpose.

Weak or vulnerable SSH keys

WARNING

If you have the older SSHv1 present in your environment, you could potentially leave an open entry path for cybercriminals. Without visibility into your entire SSH environment, you won't know how big of a problem this could be. Of course, administrators may stumble across SSHv1 keys or spot check servers, but without a management solution to alert you where these vulnerable keys are, you may be flying blind when it comes to this risk. Check out Chapter 1 for more on the versions of SSH protocol.

Excessive root access

If there's any SSH usage at all in your enterprise, there's a high probability that root access is being granted for no other reason than it makes things easier for administrative tasks. While some applications may require root access to function, the majority don't. Root access makes it difficult to monitor who's doing what because the activities logged during a session are simply logged as root and not as the individual behind the keys. In an ideal scenario, root access would be disabled, forcing individuals to use their own keys to access a host for privilege elevation.

Keys replicated on new machines

For years the use of golden images — a simple, easy process that gets machines created in a repeatable manner — has existed in enterprises. All too often, however, during this process, unique SSH keys aren't created for each new machine coming online, leaving you with duplicate SSH keys for every device built off this golden image. This creates the potential for a man-in-the-middle attack should one of these duplicate keys become compromised.

Ideally, the golden image has no SSH keys built into it and has unique keys generated on being built or there's an automated process in place to rotate the key immediately when the device is spun up. Without a solution for SSH machine identity management, administrators are left with no easy way to check for duplicate keys throughout the enterprise.

SSH policy not followed

Many organizations find it hard to define policies for SSH, let alone enforce them. As a result, even SSH policies that may be in place aren't being followed throughout the enterprise. Some of this is due to keys that potentially predate the policy, while other instances represent a blatant disregard for the in-place policy. Regardless, policy non-compliance can leave administrators holding the bag if something were to happen with their SSH keys.

IN THIS CHAPTER

- » Checking out the SSH security risks
- » Understanding how cybercriminals misuse SSH

Chapter 3 Revealing How SSH Contributes to Security Risks

Although Secure Shell (SSH) is the most broadly used security protocol for remotely managing Unix/Linux, routers, firewalls, and other systems, most organizations have limited or no formal SSH policies or management in place. Many security practitioners and managers outside of the Unix teams have only a cursory knowledge of SSH — some may not be able to tell you whether SSH uses certificates or public keys, let alone how broadly it's used or the risks it poses if not properly managed.

In this chapter, you find out how SSH machine identities can contribute to security risks if they aren't properly managed.

SSH Security Risks

Unlike other security tools, SSH machine identities generally aren't centrally managed. Instead, SSH is most often managed by individual administrators for the servers they control. Consequently, most organizations don't have a central view or one way of controlling the configuration of SSH or the access it provides. This is surprising considering the level of privileged access SSH enables for many mission critical systems. Because of this lack of central oversight, administrators are left to their own devices, which results in significant security risks.

In this section, you discover the factors that increase security risks for SSH machine identities. For more on how automation can mitigate these risks, check out Chapter 5.

Unapproved SSH servers

You need to be particularly careful when activating an SSH server because these types of assets enable remote login. Attackers could abuse this facet in a poorly controlled SSH environment using free implementations like OpenSSH (see Chapter 1 for more info) to surreptitiously enable SSH on critical assets. With SSH set up, attackers could then gain remote access to an asset and thereafter do whatever they want with it.

WARNING

If you have users and administrators enabling SSH server access on systems where it isn't required, you're expanding your attack surface because attackers will have a greater possibility of remotely gaining access to those systems.

Unpatched SSH software

For systems where SSH use is justified, if SSH server and client software isn't kept up to date with fixes and updates, it can expose the systems and data it's designed to protect and make them vulnerable to compromise.

Vulnerable SSH configuration

Most SSH server and client implementations (such as OpenSSH; we cover this in Chapter 1) include a significant number of configuration parameters that impact operation and security. Most administrators choose secure defaults. However, a couple of these default configurations, such as port forwarding and the location of authorized key files, aren't optimal for security. In addition, if your users and administrators arbitrarily change those configurations without considering the security implications, they can open those systems to broader attacks.

SSH port forwarding

Dating back to the days where encryption wasn't available for all protocols, SSH features the ability to forward traffic sent to a local port on an SSH client. The traffic is forwarded through the encrypted SSH session to the SSH server or even beyond. If local port forwarding is enabled on an SSH client that's been granted SSH access to a server on the other side of a firewall, an attacker may be able to use it to bypass firewalls.

Private key compromise

When you configure SSH for public key authentication, private keys enable access to accounts. If a private key gets compromised, an attacker can authenticate into the account(s) where the private key is trusted. SSH private keys can be compromised in the following ways:

- **Careless users:** When administrators are authorized to use SSH public key authentication, they can be careless in their handling of their private keys by placing them in insecure locations, copying them to multiple computers, and not protecting them with strong passwords.
- **Administrator turnover:** When public key authentication is used for automated processes, one or more administrators is responsible for managing the private key for the process. Administrators can make copies of those private keys and, if they're reassigned or terminated, can still use the key(s) to authenticate to the target servers.
- **Weak keys:** Because many SSH keys haven't been changed in years, smaller length keys may still be in use, making it possible for a sophisticated attacker to derive the value of the private key. In addition, we have seen bugs in cryptographic libraries that have resulted in weak, easily breakable keys being generated.

Unauthorized SSH access

Cybercriminals can abuse unprotected SSH keys to gain unauthorized access to privileged accounts. For instance, if your organization adheres to default SSH configurations, users can manage their own authorized SSH keys, and that may leave you vulnerable. Attackers could exploit this setting by compromising a privileged user and setting up a backdoor key. Alternatively, they could leverage poorly protected private keys to gain illegitimate access to sensitive business accounts. The following list gives you examples of unauthorized SSH access:

- **Untracked trust relationships:** With administrators coming and going over time, many organizations have accumulated large numbers of SSH keys but don't have visibility into the trust relationships they establish between systems and accounts.
- **Terminated employees:** If SSH users — whether they're employees or outside contractors — change roles or are terminated and their access to SSH servers isn't properly updated or terminated, these individuals can have ongoing (yet unauthorized) access to mission critical systems.
- **Backdoor keys:** By default, most SSH implementations (such as, OpenSSH; we cover this more in Chapter 1) allow users to configure their own authorized key files. This means that administrators can place a public key in an account so they can access it using a private key. If your organization doesn't keep an up-to-date inventory of authorized keys and regularly review it, users — or even attackers — may place authorized keys in unexpected places for future access through a backdoor.

Privilege elevation

SSH is generally integrated with other components to enable privileged access — including operating system permissions, sudo (which allows one user to run a program as another user), and privileged access management (PAM) solutions. It's difficult to centrally orchestrate the secure configuration of all these components to prevent cybercriminals from successfully elevating privileges during an attack. What's even more challenging is when you have multiple individual administrators each making decisions on the implementation of SSH without any central oversight or review. Without this oversight, you face greater potential for privilege elevation, especially because attackers remotely accessing systems over SSH will have an encrypted session within which to hide their actions.

Rogue known host keys

If users or administrators who initially establish a connection from an SSH client to an SSH server don't check the authenticity of the public key for that server, they may inadvertently accept an attacker's public key and enable a man-in-the-middle attack.

Lateral movement

After attackers gain an initial entry into your network, their next goal is typically to get onto other systems, which is called *lateral movement.* When cybercriminals' jump from system to system, they can easily pivot on your network by abusing persistent SSH trust relationships to their advantage. That's especially the case if administrators don't review those keys often or maintain strong oversight over them.

WARNING

When SSH machine identities fall into the wrong hands, this is very dangerous for your organization because SSH users and automated process are typically granted elevated privileges.

Circumventing security controls

While cybercriminals are busy moving laterally within your network, they could come across firewalls and other security technologies designed to block malicious network activity. Unfortunately, if you don't properly control your SSH environment, attackers could bypass these safeguards by configuring SSH for port forwarding or other privileges. Doing so would allow the attackers to communicate with other systems that leverage authorized connections via firewalls and thereby find an alternate yet nonetheless "approved" route through the network.

Obscured and exfiltrated data

Attackers like to hide within the infrastructure by using readily available tools, like the SSH protocol, to redirect and exfiltrate data without being detected by traditional controls. SSH enables traffic redirects and allows its users to set up a listening port on a client and tunnel data through an encrypted channel to an exit server port or vice versa. As a result, encrypted SSH connections can also be abused by attackers to exfiltrate data without being detected.

How Cybercriminals Misuse SSH in Attacks

Cybercriminals are constantly looking for new ways to exploit systems and execute their attacks. Attackers diligently look for misconfigurations and weak authentication methods in public-facing remote services, and the number of attacks in the cloud that abuse SSH password-based authentication continues to grow.

Whether a threat is designed to gain initial access to a target machine through SSH, insert attacker-owned keys for persistence, or collect SSH keys to laterally move like a "worm" across the network, malware is developed with machine identity in mind. In many cases, lack of oversight and controls have led to violations of corporate access policies. These in turn can result in dangerous backdoors that can facilitate the launch of successful attacks through the otherwise trusted encrypted tunnels.

This section covers the tactics that misuse SSH machine identities in attacks.

Exposed services

Exposing an application service to the internet is a common misconfiguration that allows access to an internal system from anywhere and acts as a common attack vector. Attackers can leverage external-facing remote services as a point of entry to an application hosted in the cloud, aiming to compromise the underlying instance.

TECHNICAL STUFF

Another less reported attack vector on applications with exposed SSH services is for an attacker to use compromised SSH keys and credentials. Attackers can gather SSH keys and credentials from source control, public repositories or open buckets. They can also steal them from machines compromised in parallel or unrelated campaigns, or even purchase them on remote access markets where they are sold as-a-service.

APTs

Advanced persistent threat (APT) attacks typically use a combination of discovered machine identity vulnerabilities and malware that exploit weak or improperly managed machine identities to achieve their goals. A primary goal of an APT attack is to remain persistent on the victim's network. SSH machine identities are extremely useful to attackers because they support and enable persistence, lateral movement, and defensive evasion.

For example, one APT group was able to use a feature that allowed any user to trigger an SSH connection from the cloud provider to the managed server, with the SSH agent forwarding feature enabled. This allowed the attacker to relay authentication to any other server within the same cloud, achieving remote code execution (RCE) with root privileges.

In another recent attack, cybercriminals brute-forced an exposed SSH service to infect the host with a cryptominer and used it as a launchpad for further large-scale attacks. In yet another incident, skilled attackers managed to get initial access on a Linux system through what appeared to be a brute-force attack on an exposed SSH service and moved from there to the on-premises network.

In APT attacks, cybercriminals use the following tools:

WARNING

- **SSH backdoors:** Unfortunately, SSH can be used by both developers and attackers to ensure their access to the server. Attackers who are able to compromise a machine can enable the SSH service to allow SSH communication and by that establish persistence on the target. This backdoor access allows attackers to blend into legitimate traffic, avoid detection, and pass through any firewalls that are in place.

  Another common technique to establish persistence on a target where SSH service is enabled is to insert an attacker-owned SSH public key to the authorized keys file on the server to create a backdoor that ensures remote connection to the server without notice.
- **Legitimate SSH services:** Attackers often use legitimate and preinstalled remote service with valid accounts on compromised machines to evade defense mechanisms. Attackers collect insecure machine identities from their targets and use them to establish SSH communication, bypass any access restrictions on traffic, and raise no suspicions or flag any security controls.
- **SSH keys and "wormlike" malware:** Many APT attacks are designed to steal and exfiltrate SSH keys and known hosts information to enable lateral movement to more and more

machines. They use a "wormlike" malware that replicates itself in order to spread to other computers. This malware collects SSH keys and information on known hosts to spread through the SSH remote service to other victims.

TECHNICAL STUFF

According to a recent report by Orca Security, 5.6 percent of publicly exposed assets in the cloud contain SSH keys that could potentially be used to access adjacent systems. Attackers can use these keys to move laterally across the network and log into remote machines to search for higher profile assets or infect further victims.

- **Encrypted channel or SSH tunneling:** SSH tunneling is a method of transporting arbitrary networking data over an encrypted SSH connection. SSH tunneling attacks may use legitimate and preinstalled remote access services such as SSH machine identities to establish an interactive command and control channel to target systems within networks, since these are normally allowed by application control within a target environment.

Malware

In recent years, an increasing number of commodity malware has integrated the misuse of SSH machine identities into attacks. Campaigns such as cryptomining, spam, adware, and banking trojans are now equipped with SSH capabilities for credential theft, persistence, and lateral movement.

In most cases, the malware is used to add the attacker's SSH key to the authorized_keys file on the victim's machine, enabling the attacker to remain persistent on the device. In other cases, the malware was able to brute-force weak SSH authentication on public-facing servers and gain access to the target, steal credentials, and host information to laterally move across the network and infect further machines.

IN THIS CHAPTER

- » Cataloging access to your SSH machine identities
- » Honing in on SSH intelligence

Chapter 4 Gathering SSH Machine Identity Intelligence

Organizations have been using Secure Shell (SSH) for more than 20 years — plenty of time to amass large numbers of SSH connections. Also, due to its usefulness and popularity, SSH has been embedded into the frameworks of many applications. Given its long history and extensive usage, it's easy to see how SSH could've spiraled out of control, especially because these keys don't expire.

You may wonder why administrators don't have a better handle on their SSH inventory. The easy answer is that most of them don't have the support of a central management solution for visibility and intelligence. Unknown or undermanaged legacy keys pose a substantial security risk and make risk analysis difficult if they aren't understood.

WARNING

If you don't have full visibility, policy enforcement, and rogue SSH key detection, you may be exposed to undue risk, and consequently, it can be difficult to prepare for a comprehensive review of your SSH environment.

In this chapter, you explore the types of visibility and intelligence you need to keep your SSH machine identities secure.

Getting Visibility Across All Usage

Before you even begin an SSH machine identity management program, you need an inventory of all the SSH machine identities used across your enterprise. To successfully gather this information, keep in mind the dynamic nature of machine identities and the different types of data necessary to manage and protect them. For SSH machine identities in particular, you need centralized visibility into all your SSH servers, private keys, and any SSH configurations that limit access.

To build a successful SSH machine identity management program, you need to gather and provide immediate access to a variety of information about your SSH machine identities.

Extensive, enterprise-wide discovery

The best place to begin an SSH key management program is to discover all your SSH servers, private keys, and authorized keys that grant SSH access. You may be surprised by how many SSH keys you have. Many large organizations end up with more than one million SSH keys spread throughout their network.

TIP

Your organization should ideally use an automated solution to make sure you discover SSH keys stored in user home directories as well as SSH configurations that limit access (learn more about automation in Chapter 5). After that, continue to actively manage this inventory as administrators add or decommission SSH-based assets.

Central repository for comprehensive inventory

To attain SSH visibility and control across your network, make sure you have a complete and accurate inventory of all your SSH machine identities, including who owns them and which systems they can access. This inventory can be a challenge for many organizations who have tens of thousands of untracked identity keys, authorized keys, and corresponding trust relationships granting access across a large number of mission critical systems.

When you're prepping for an SSH inventory, make sure to include information on

- All SSH servers
- Private keys
- Connections
- Any SSH configurations that limit access
- Host and account locations of all identity and authorized keys
- Authorized key restrictions

In addition to creating an inventory of the location of all existing SSH keys, map trust relationships and evaluate them against defined policies.

Identify vulnerabilities to be remediated

An important part of strong SSH key management is scanning an inventory for known vulnerabilities and issues. Your organization should use tools to identify threats such as

- SSH root access
- Weak keys
- Potential backdoor keys
- Duplicated private keys
- Port forwarding
- Insecure configurations

Those tools should also help automate the identification process so companies can respond to issues and vulnerabilities as soon as they detect them.

Audit-ready reporting and analytics

Auditing of SSH user keys serves risk analysis and ensures that the provisioning, life cycle management, and termination processes — as well as continuous monitoring — are working properly. A comprehensive audit of SSH user keys for risk analysis purposes should be performed by all organizations that use SSH protocols. Audits for processes, on the other hand, can use representative sampling and condition detection tests to gain sufficient confidence that the processes are functioning.

TIP

A system that can provide central reporting and analysis tools across the inventory can reduce the time necessary to prepare the data required for audits and make it easier for auditors to verify proper implementation and identify exceptions.

Focusing on Critical SSH Machine Identity Intelligence

After you've collected an inventory of keys, access rights, and related hosts, you need to apply that intelligence to find high-risk connections. A lack of visibility into orphaned, shared, weak, or root keys can lead to unauthorized access and must be immediately reported for analytic review. You should also be able to create your own rules to identify out-of-policy practices like cross-environment key usage, improper key lengths, or aged keys.

The key areas of focus for your intelligence efforts for SSH machine identities should include the ones we cover in this section.

Missing controls

As business-critical SSH connections expand, uncontrolled oversight of the SSH keys, owners, access levels, and authorized assets often get lost, which results in a chaotic mesh of trusted connections. One common risk is having an abundance of unnecessary SSH root keys that violate data privacy policies and generate unwanted exposure.

WARNING

Cybercriminals misappropriate poorly protected SSH keys to bypass security controls and gain privileged access to internal network resources and data. With SSH keys, attackers can appear to be legitimate administrators or trusted machines, enabling them to hide and move around on internal networks — often for extended periods of time without being detected.

Root access orphan keys

Lack of insight and intelligence into ownership of keys, as well as orphaned, shared, weak, or root keys, can lead to unauthorized access — and should be reported immediately for review and corrective action.

All too often, administrators are stuck with the insurmountable task of trying to track down the root access that has been granted in a given environment. What can make this task even more challenging is when you have no visibility as to where the private key resides, resulting in an access orphan. Access orphans are especially concerning when they're root access orphans, which grant superuser privileges.

Duplicate keys

Duplicating unique host keys isn't all that uncommon — especially when a specific virtual machine gets copied and used in identical ways. Identity keys shouldn't be duplicated (copied to other systems) for interactive users and automated processes. While it's not recommended, if you choose to allow duplication for interactive users, you should provide guidelines for the acceptable locations where keys can be copied and used.

Shared private keys

Often, administrators may share their private keys out of convenience — perhaps when they're on vacation or busy with other duties and need other staff members to assist them. If uncontrolled, this process could quickly spin out of control as private keys are shared, stored, and used without limit.

Lateral movements

Many organizations leave themselves open to SSH-based lateral movement because they have no inventory of deployed SSH keys that enable persistent access between systems. You need to be able to monitor usage to prevent cybercriminals moving around and expanding their access on compromised systems. Monitoring helps you prevent a dense and uncontrolled environment of SSH key-enabled connections that allows attackers to move from asset to asset, using keys found in various user accounts.

Additional intelligence

Information about who's using an SSH key and which systems it can access will prove extremely valuable — especially if there's a security event or audit. Some of the key usage details that you should discover about your SSH machine identities include

- **Location:** SSH configurations can restrict the locations from which each authorized SSH key can be used. But many organizations fail to limit SSH key use by location. When access is limited to the known locations of administrators and machine-to-machine access, it helps to prevent malicious access from other locations.
- **Owner:** One of the best ways to prevent the misuse of SSH machine identities is to understand who's using them. Furthermore, your organization should assign ownership of all access granting SSH keys and monitor and analyze key-based access usage.
- **Protocol versions:** Over time, SSH protocol has expanded its encryption algorithms. Also, support for public key certificates was added in SSH, version 2 (SSHv2). As vulnerabilities start to spike, especially for older unsupported versions like SSH, version 1 (SSHv1), and support is no longer guaranteed, information security teams will deal with many configuration variants and ensure exploitable deployments are rooted out. We cover these versions more in Chapter 1.

SSH MANDATES AND STANDARDS CONTINUE TO GROW

Because of the security threats and operational risks connected with poorly managed SSH keys, auditors are becoming increasingly focused on those risks and the visibility and management of SSH keys. And the audits themselves evolve over time, generally becoming more dynamic and stringent. In recent years, the number of security frameworks and standards that require close inspection of SSH key risks have grown. The most noteworthy include the following:

- National Institute of Standards and Technology (NIST)
- Payment Card Industry Data Security Standards (PCI DSS)
- Health Insurance Portability and Accountability Act (HIPPA)
- Sarbanes-Oxley (SOX)

Rather than wait for an auditor to check on the health of your organization's SSH practices, you can be proactive about securing your SSH keys.

The following checklist helps gauge the likely outcome of your next SSH audit. If you're lacking one or more of these controls to secure your SSH keys, the chance of passing your next SSH audit may be small. Check off these tasks:

- Put a comprehensive SSH governance program in place.
- Establish effective SSH key authorization and management.
- Evaluate vulnerable SSH protocols and weak SSH configuration settings.
- Automate SSH key generation, rotation, and removal.
- Establish ongoing monitoring of SSH key usage.
- Build an SSH control assurance program.

IN THIS CHAPTER

- » Enhancing your processes through automation
- » Incorporating SSH machine identities with technology
- » Resolving SSH machine identity risks

Chapter 5 Automating SSH Machine Identity Management

Most organizations rely on manual processes to manage Secure Shell (SSH) machine identities. When SSH keys are manually managed, key risks grow and become a security liability, which leaves you vulnerable to failed audits and exposed to threats.

As a result, standards bodies such as National Institute of Standards and Technology (NIST) recommend using automated processes and tools to assist users with generating, deploying, and managing SSH keys. Not only will the deployment of automated tools make SSH key management less error-prone, but also automation facilitates SSH audit programs that now check for this functionality.

In this chapter, you discover how automation helps you reduce risk, resolve SSH issues, and enforce security policies and controls that limit the accessibility and use of SSH keys.

Optimizing SSH Machine Identity Management with Automation

Many operational processes can be optimized by automating SSH user key setups and removals and related approval, documentation, monitoring, and audit processes. The automation of the processes involved in the management of SSH machine identities can significantly improve your security, efficiency, and availability.

REMEMBER

Automation involves several best practices in multiple important areas, all of which require a comprehensive SSH machine identity management solution that provides full visibility and leverages automation to manage and enforce policies. This section gives you the rundown of the areas to focus on.

Inventory

Manual discovery and inventory of all SSH identity and authorized keys (along with corresponding restrictions) and mapping of all resulting trust relationships is practically impossible. Automation is essential if you want to accurately complete and continuously monitor an inventory of SSH machine identities across your organization.

Provisioning

Your administrators are probably spending too much time configuring and managing SSH user keys. When you automate the provisioning of interactive or automated access, approved requests can be automatically picked up by a key management system and implemented on all affected hosts. Automating provisioning in this way does the following:

- Removes manual steps for setting up keys
- Eliminates the need for manual root access for installing the keys
- Reduces the amount of privileged administrative access that you need to audit and review
- Eliminates configuration errors due to incorrectly implemented requests
- Ensures that the approval template remains available for future reference and for use in continuous monitoring and audits

Life cycle automation

WARNING

Using manual processes to manage SSH machine identity life cycles — such as monitoring, rotating, and replacing SSH keys — is inherently error-prone and resource intensive. You're probably already finding it difficult to manually track the progress of complex, multi-step processes across multiple systems. Another shortcoming of manual management is that it gives your administrators direct access to private keys, which increases the possibility of private key compromise.

REMEMBER

Automating the life cycle of SSH keys to support a machine identity management strategy streamlines your SSH machine identity life cycles and the critical connections they enable. Plus, it helps you reduce operational cost of managing the SSH keys (and key pairs) used for trusted connectivity. But the bottom line is that automating SSH key (and key pair) life cycles helps you respond quickly to imminent threats that may impact your organization's critical assets.

Policy enforcement

Policies and procedures play a critical role in SSH security by establishing consistent baseline requirements across the diverse systems and environments where SSH machine identities are deployed, including rapidly evolving cloud environments. The definition of policies should clearly spell out roles and responsibilities in order to prevent misunderstandings that result in security lapses and to ensure accountability.

Automation is a critical capability that makes it possible to consistently enforce SSH machine identity policies and applicable regulatory requirements. When you leave compliance in the hands of the various administrators who manage SSH keys for the systems they control, you'll see inconsistent results for policy enforcement. That's why it's critical that you educate all SSH stakeholders on SSH security policies and processes — and have automation.

TIP

For the best results, automated policy enforcement should drive every aspect of your SSH machine identities, including ownership, usage, configuration, and storage. With these capabilities, you can automatically revoke and replace any SSH machine identities that don't conform to appropriate policies. Plus, you have the flexibility to enforce SSH machine identity policies in a variety of ways: globally, by logical group, or by individual identity.

Remediation

After you identify risks, you must make sure they're fixed quickly to prevent ongoing exposure to a potential breach. By automating responses to SSH issues, you can quickly remove unauthorized keys, rotate or replace weak and old keys, and remove SSH root access and duplicate private keys. Ultimately, your goal is to enforce security controls that limit the accessibility and use of SSH keys outside of their original purpose.

Automation also gives you the agility to rapidly respond to critical security events, such as a breach or other compromise. For example, if a large-scale security event occurs, automation is the only way you can quickly make bulk changes to all affected private keys. Automation is also the fastest way to remediate SSH key risks, such as replacing an orphaned, duplicated, or shared private key that's used across multiple machines.

Continuous monitoring

Automating your intelligence gathering (continuous monitoring) is the only way to continually monitor the security and health of your SSH machine identities. Plus, when your intelligence is automatically updated, you can generate alerts when anomalies or vulnerabilities are detected.

Automation is also a virtual necessity for any continuous monitoring process. To meet ongoing security and compliance requirements, you need continuous, automated monitoring and tracking of SSH keys. You should also implement SSH audit practices that regularly review SSH entitlements, assess risk, avoid compliance violations, and increase accountability for identity and access management.

REMEMBER

When you've set up your SSH machine identity protection program to continually capture the information you need, you can rely on that intelligence to drive automated actions. The more management and security processes that can be reliably automated, the more benefits you see — from fewer errors to a reduction in management resources and better security.

Integrating with Your Technology Ecosystem

SSH machine identities are used to access nearly all the technology solutions that are deployed across your expanded network and security infrastructure. As a result, you need to be prepared to integrate and orchestrate SSH machine identities across a multitude of enterprise IT systems.

CI/CD pipelines

Machine identities that enable automated access to an SSH server are often used by continuous integration and continuous delivery (CI/CD) pipelines and with concepts like Infrastructure as Code (IaC). Automating access to policy-compliant SSH keys in the deployment pipeline helps ensure security as developers log in to a server using SSH.

The SSH private key is a sensitive piece of data, because it's the entry ticket to a server. Traditionally, developers generated an SSH key on the host machine, authorized it on the server (that is, copy the public key to the server) in order to log in manually, and performed the deployment routine. Automating that traditional process within a developer's system of choice not only saves time but also increases security.

Key vaults

Key vaults are critical to effective SSH machine identity management because they help organizations protect, monitor, detect, alert, and manage privileged accounts and other credentials for applications, scripts, and other machine identities.

Automating access to privileged credentials in key vaults allows SSH machine identity management solutions to perform sensitive renewal, replacement, and re-key operations without administrator involvement or the need to store credentials outside of the key vault. This reduces time-consuming administrative tasks that can also increase the risk of unnecessarily exposing private keys to additional people. By automating this process, you can accelerate the speed of your security operations and increase agility to respond to incidents.

HSMs

Another component of your SSH machine identity management infrastructure may include a hardware security module (HSM), which is a physical device that you connect to your network.

REMEMBER

While creating software-generated SSH keys is a suitable method, a more secure way is to have the HSM create these keys. Having these keys created from hardware by the HSM gives the keys better entropy.

Overcoming SSH Machine Identity Risks with Automation

In Chapter 3, we outline the risks of weak SSH machine identity management. Intelligence-driven automation addresses these, and many other, machine identity risks.

TIP

To help avoid the impacts of these risks, follow these guidelines:

- **Prevent breaches** by automating the collection of risk intelligence required to quickly identify and respond to SSH machine identity risks, weaknesses, or security events. Automated policy-enforcement and life cycle management ensure orphan, duplicate, or shared SSH keys are decommissioned.
- **Accelerate incident response** by automating the identification of impacted SSH keys as well as the actions needed to remediate large groups of machine identities. Being armed with information on location and owner of SSH keys can dramatically increase the speed of your response to large-scale security events.
- **Streamline operations** by automating routine administrative tasks to eliminate manual, error-prone processes and reduce the expertise and resources needed to manage and protect the growing number of SSH machine identities.

- **Ensure compliance** by automating policy enforcement to improve audit readiness, offering automated validation of SSH machine identity management, and generating scheduled or on-demand compliance reports.

Automation also makes it easy to implement role-based access controls that allow or block access to machine identities. Implementing change management and role-based access controls ensures you can effectively manage machine identities and demonstrate this control for audits.

A COLLABORATIVE SSH SECURITY AUTOMATION SOLUTION

CyberArk and Venafi teamed up to offer an integrated solution for enterprise-wide governance and risk reduction by enabling easy and robust management of SSH keys. The integration with Venafi's SSH Protect solution is designed to provide

- Higher levels of automation for system administrators
- Better visibility for InfoSec teams
- Fast, successful audits for GRC teams

In this integrated solution, CyberArk provides Privileged Access Management (PAM) for interactive human-user accounts including key management, session isolation, and audit, while Venafi provides Machine Identity Protection for automated machine-to-machine connections. Together, encryption key governance is achieved across the entire enterprise, protecting the full life cycle of keys from creation to termination, including the storage and auditing of those keys.

Venafi's integration with CyberArk further secures the key life cycle by automatically placing private keys discovered by SSH Protect into CyberArk's Privileged Access Manager and continuously monitoring those SSH sessions.

Learn more at `marketplace.venafi.com/details/cyberark-privileged-access-security`.

IN THIS CHAPTER

- » Finding your SSH machine identities
- » Monitoring your usage
- » Enforcing policies
- » Choosing automation

Chapter 6 Ten Steps to SSH Machine Identity Management

Secure communications between machines are essential to the success of every enterprise. But how do you keep the identities of your machines safe when your administrators are adding machines and changing them every day? To build your own Secure Shell (SSH) machine identity management program, you need to take specific steps. We cover that process in this chapter, and together these steps enable your organization to protect all the SSH machine identities you're using today. This process also positions you to keep up with the growing number of machines your enterprise will need moving forward.

1. Discover All Your SSH Machine Identities

You need to control all SSH machine identities in your environment, including who they belong to and what they're used for. But first you have to find them. A mix of discovery mechanisms and flexible reporting capabilities helps make this task run as quickly and smoothly as needed to find SSH keys across the enterprise.

2. Map All Trust Relationships

By running solid discovery, creating an inventory, and mapping SSH keys pairs, you get a clear overview of all SSH keys and trusted relationships, including users, hosts, and configuration options. Automatically importing trust relationship data into an existing privilege access management system for review leverages existing processes and facilitates an accurate review and tracking of approvals for access granted through SSH keys.

3. Identify and Remove Any Orphaned and Duplicate Private Keys

Implement a continuous proactive approach that scans for orphaned keys, monitors for duplicate key usage, and frequently replaces keys. Mapping all trust relationships also helps you identify any orphaned, shared, or duplicate keys that need to be removed — ideally by using automation. This step can have a big effect on the overall security posture of your enterprise environment and prevent further damage.

4. Implement Clearly Defined SSH Key Management Policies

To prevent the compromise of a key used by an authorized user, you should configure access controls on identity keys to restrict access to the interactive user or to the automated process to which they've been assigned. For automated processes, policies should define guidelines for assigning access to administrative staff responsible for managing identity keys.

5. Assign Ownership and Monitor Usage

One of the best ways to prevent the misuse of SSH machine identities is to understand who's using them. That's why it's important that you assign ownership of all access granting keys and monitor and analyze key-based access usage.

TIP

Specifically, your organization should conduct SSH audits for compliance violations, assessing risk, and increasing accountability for identity and access management.

6. Control SSH Identities

Because SSH provides remote access into systems, it's critical that access be tracked and controlled. Many organizations don't have centralized oversight and control of SSH, so the risk of unauthorized access is increasing.

7. Control SSH Configuration and Known Host Files to Prevent Any Tampering

WARNING

If you aren't properly controlling and hardening the configuration of your SSH environment, cyber criminals can use SSH to bypass security mechanisms. An otherwise useful administrative short cut, such as enabling port forwarding, can leave your organization vulnerable. Malicious insiders or cyber criminals can use these authorized connections to bypass firewalls.

8. Enforce Inventory and Remediation Policies

Be prepared to respond quickly when there's an issue with your SSH environment — removing unauthorized keys, replacing old keys, or enforcing security controls that limit the accessibility and use of SSH keys. Ideally, your organization should automate these procedures and pair them with your identification tools. Doing so helps ensure consistent policy enforcement of SSH key life cycle management.

9. Establish Continuous Monitoring and Audit Process

Just as any part of the IT infrastructure, usage of SSH machine identities should be continuously monitored and reported on —monthly, weekly, or even daily. Building a set of metrics and sharing them with your peers on risk and information security teams help create a mindset and perhaps an incentive to building out stronger SSH policies. It also creates a trackable record of all changes to your organization's SSH assets.

10. Automate the Whole Process

Time and resources are precious commodities for security operations teams. Automated capabilities like "single-click" machine-assisted key rotation, scheduled bulk cleanup of out-of-policy keys, or self-service managed key generation for system admins should be put into place to improve efficiencies, tighten security, and reduce errors introduced by manual processes.

Manage and protect SSH machine identities

Every network has two actors: humans and machines. Humans rely on usernames and passwords to identify and authenticate themselves, but machines don't. Instead, they use machine identities such as Secure Shell (SSH) keys. Every year, businesses spend billions managing usernames and passwords but almost nothing managing SSH machine identities that secure access to nearly every critical business function. Learn why cybercriminals target SSH machine identities and how effective management can keep yours safe.

Inside…

- Where SSH machine identities are used
- How SSH machine identities can be compromised
- Tips on managing SSH machine identities
- How to prepare for an SSH audit
- Why policies and automation are essential

VENAFI®

Go to Dummies.com™
for videos, step-by-step photos, how-to articles, or to shop!

ISBN: 978-1-119-78786-0
Not For Resale

9 781119 787860

for dummies®
A Wiley Brand