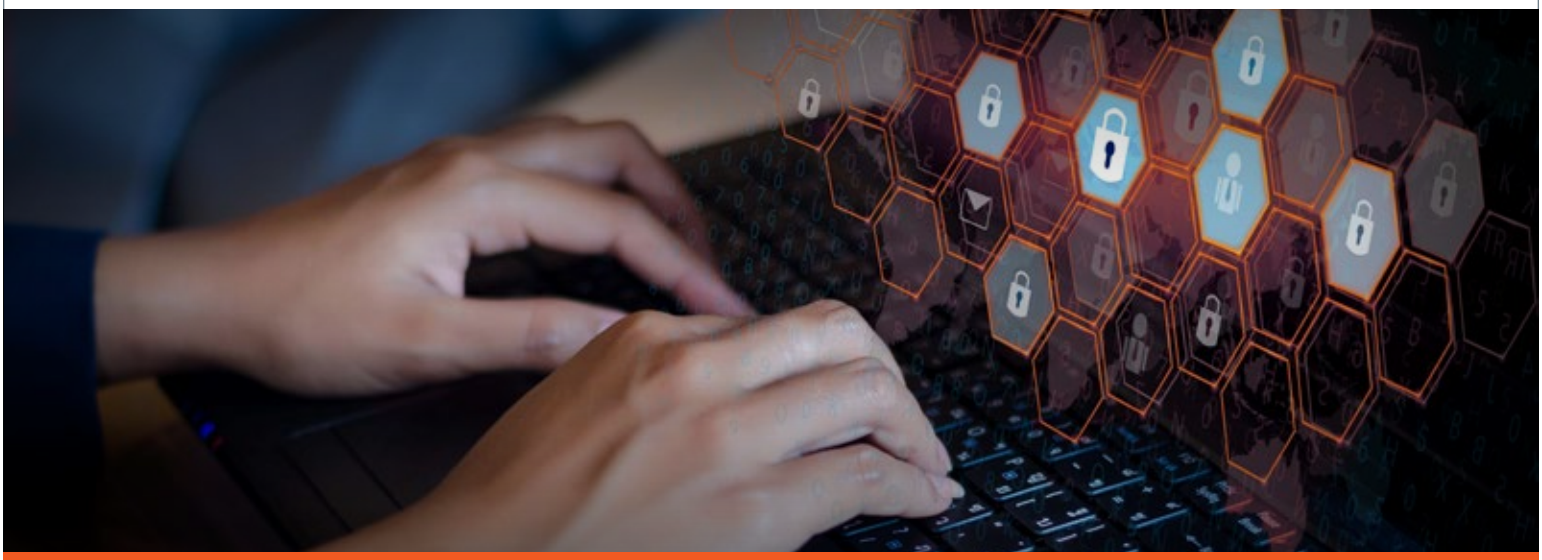
WHITE PAPER



PRIVILEGED PASSWORD MANAGEMENT EXPLAINED

BeyondTrust

TABLE OF CONTENTS

1 Introduction

Privileged password management, sometimes called privileged credential management or enterprise password management, refers to the practice and techniques of securely controlling credentials for privileged accounts, services, systems, applications, machines, and more. The ultimate goal of privileged password management is to reduce risk by identifying, securely storing, and centrally managing every credential that provides elevated access. Privileged password management works hand-in-hand with implementing least privilege and should be a foundational element of any organization's privileged access management (PAM) initiatives.

The ultimate goal of privileged password management is to reduce risk by identifying, securely storing, and centrally managing every credential that provides elevated access

Whereas in decades past an entire enterprise might be sufficiently managed through just a handful of credentials, today's environmental complexity means privileged credentials are needed for a multitude of different privileged account types (from Domain admin and sysadmin to workstations with admin rights), operating systems (Windows, Unix, Linux, etc.), directory services, databases, applications, cloud instances, networking hardware, internet of things (IoT), social media, and more.

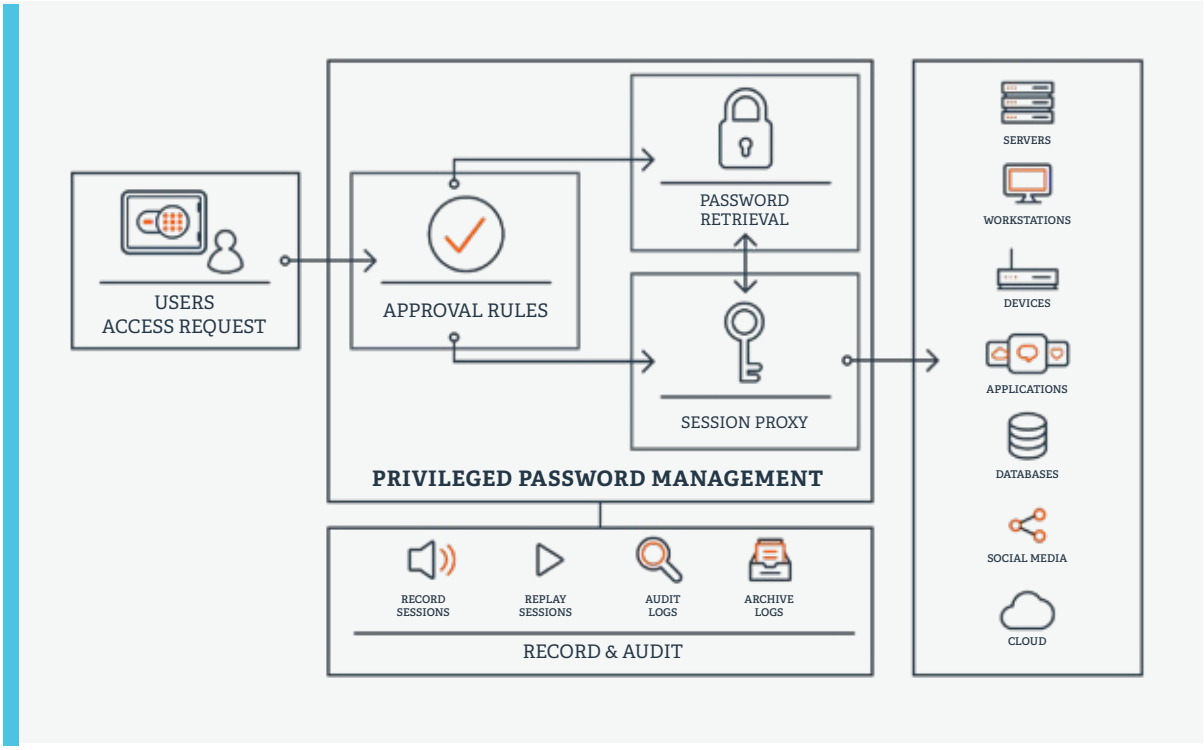


Figure 1: Representation of privileged password management

Most organizations rely on some variation of a privileged password management "solution." It could simply be an Excel spreadsheet for simple password tracking, or it could be an advanced enterprise password management solution that automates privilege account and credential discovery, onboarding, access control, centralized protection and storage, rotation, alerting, reporting, and oversight of all the enterprise's credentials that provide elevated privileged access rights.

While beyond the scope of this white paper, organizations can also benefit from team password management tools that control personal privileged credentials for local assets. These personal password managers ensure that important organizational credentials are secured locally and can even auto-login the user to the resources they use.

2 What is a Privileged Password?

A basic definition for a password is a word or phrase intended to differentiate an authorized user or process (for the purpose of permitting access) from an unauthorized user. Privileged passwords are a subset of credentials that provide elevated access and permissions across accounts, applications, and systems. Highly privileged account passwords—such as Root in Linux and Unix, and Administrator in Windows—are often referred to as "the keys to the IT kingdom," as they can provide the authenticated user with almost limitless privileged access rights across an organization's most critical systems and data. With so much power inherent of these privileges, they are ripe for abuse by insiders, and are highly coveted by hackers. Forrester Research estimates that at least 80% of all security breaches involve privileged credentials.

> *Privileged passwords are a subset of credentials that provide elevated access and permissions across accounts, applications, and systems*

3 What Makes for a Strong Password?

Before diving deeper into privilege management-specifics, here are some password management best practices universal to most types of passwords:

- Password length of at least 12 characters
- Passwords should be unique, complex, and nonsensical, comprised of a mix of non-repeating letters (upper and lower case), numbers, and symbols that do not contain dictionary words in any language, or have any other guessable context (employee ID, dates, etc.), or sequences from a keyboard like 'qwerty' or 'zxcvb'
- Prohibit password re-use; employees should be forbidden from using the same passwords across their personal and work accounts.
- If you ever need to share your password, change it when the other person is done with using it.
- Frequently change passwords—a process referred to as password rotation.

Password rotation remains a crucial best practice for protecting privileged credentials

When it comes to password rotation, current NIST guidance advises against forcing changes to personal passwords except in special cases such as suspected or known compromise of a password. However, password rotation remains a crucial best practice for protecting privileged credentials. The frequency of rotation should vary based on the password age, usage, and security importance. For instance, you should consider rotating superuser account (e.g., root, domain admin, etc.) and other highly privileged passwords, at more frequent intervals, including after each use—known as one-time-passwords, or (OTPs)—for your most sensitive accounts.

PITFALLS OF HUMAN-MANAGED PASSWORDS

While the password best practices cited above seem simple enough, what makes it difficult to translate these principles into practice?

Today, the number of passwords any employee may need to remember for access to various accounts, systems, and applications can range from dozens, to over one hundred. When it comes to an organization's privileged credentials, the number of passwords to manage can easily range in the tens of thousands, or even exceed a million.

The onus of having to manually remember so many passwords invariably trips up employees in ways that increase risk to credentials theft or misuse and cause downtime. Employees are prone to forget passwords from time-to-time, potentially locking them out of systems. To compensate, they may apply the same passwords for multiple accounts, select easy-to-guess passwords, or resort to recording passwords on paper or within electronic documents, such as MS Word or spreadsheets. Consequently, part of the danger is that hackers can correlate, along with email addresses and usernames, the password from one compromised account to other services that may be using the same password. So, for instance, using the same privileged credential on a server, application, switch, and social media account means that one compromised account also jeopardizes the other accounts.

In practice, privileged passwords are inadequately rotated and audited—leaving organizations susceptible to to exploitation of privileged credentials

While automated password management has become an essential measure to reduce the risk of compromise in the modern enterprise, many organizations still rely, to some degree, on manual/human password management practices. Consequently, in practice, privileged passwords are inadequately rotated and audited—leaving organizations susceptible to privileged credential exploits.

CHALLENGES TO MANAGING PRIVILEGED PASSWORDS

Now, let's examine some challenges more specific to an enterprise's privileged credentials.

Lack of Visibility & Awareness of All Privileged Accounts

Privileged accounts, many long forgotten, are sprawled across most IT environments. They may include accounts associated with both human and non-human identities, and credentials across an enterprise pose a monolithic challenge—especially for those companies that rely on manual processes and tools. 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 & Auditability

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 of time 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 also 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 thousands of concurrent privileged sessions running across an enterprise, how does IT expeditiously detect and halt inappropriate activity, whether malicious or accidental? 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, full coverage over privileged account usage will almost certainly require a third-party, enterprise-class solution.

Sharing of Privileged Accounts for Convenience

IT teams commonly share root, Windows Administrator, and many other privileged accounts so workloads and duties can be seamlessly shared as needed. However, with multiple people sharing an account, it may be impossible to trace actions performed to a single individual, complicating auditing and accountability.

Hard-Coded / Embedded Credentials

Applications, systems, and IoT devices, are commonly shipped, and often deployed, with embedded, default credentials that are easily guessable

Privileged credentials are needed to facilitate authentication for app-to-app (A2A) and application-to-database (A2D) communications and access, as well as for emerging areas, such as robotic process automation (RPA). 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. The secrets sprawled across DevOps tools, scripts, test servers, and production builds, are another frequent blind spot. All of these types of non-human privileged credentials/secrets 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 and secrets 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 manually enter log-in credentials. 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 backdoors 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. SSH keys are not necessarily tied to a single user—multiple people may share the private key and pass phrase 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 & the Cloud

The challenges of visibility and auditability are generally exacerbated in cloud and virtualized environments

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 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 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 and need access to networked systems. 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?

4 Some Common Password Attack Techniques

Password attacks come from all angles. Some programs, such as John the Ripper and L0phtCrack, can even crack complex passwords, while Pass the Hash (PtH) toolkits can be lethal without even cracking the password. Some common credential exploit tactics are outlined below.

Brute force attacks involve repeatedly testing a password, potentially generating millions of guesses per second, with combinations of characters (numbers, letters, and symbols) until one matches. The more mathematically complex a password, the more difficult to crack.

Password attacks come from all angles

Dictionary attacks, as opposed to a brute force type assault that computes random combos of characters, make password guesses based on words in a dictionary of any language.

Pass-the-Hash (PtH) attacks can occur on Linux, Unix, and other platforms, but are most prevalent on Windows systems. In Windows, PtH exploits Single Sign On (SSO) through NTLM, Kerberos, and other authentication protocols. When a password is created in Windows, it is hashed and stored in the Security Accounts Manager (SAM), Local Security Authority Subsystem (LSASS) process memory, The Credential Manager (CredMan) store, a ntds.dit database in Active Directory, or elsewhere. So, when a user logs onto a Windows workstation or server, they essentially leave behind their password credentials. In PtH attacks, an attacker doesn't need to decrypt the hash to obtain a plain text password, once captured, the hash can be passed through for access to lateral systems. A hacker could elevate privileges simply by stealing RDP credentials from a privileged user during an RDP session.

Pass-the-Ticket (PtT) and Golden Ticket attacks are similar to PtH but involve copying Kerberos tickets and passing them on for lateral access across systems. A Golden Ticket attack is a variation of Pass-the-Ticket, involving theft of the krbtgt account on a domain controller, which encrypts ticket granting tickets (TGT). With this prized foothold, a hacker could freely create his/her own access tickets for any level of access and duration of access. Rotation of privileged account passwords after every use and least privilege enforcement (such as separating different types of privileged and non-privileged accounts, and even better, removing admin rights from endpoints) are important security controls to thwart PtH, PtT, and Golden Ticket attacks.

Social engineering password attacks, such as phishing and spear phishing, involve tricking people into revealing information that can be used to gain access. Some prominent victims of a spear phishing attack include the Democratic National Committee (DNC), John Podesta, and others by the hacking group, Fancy Bear. In Podesta's case, he apparently received a spoof Gmail email alerting him that an imposter had tried to use his password, but that Google had detected the illicit activity and stopped it. Apparently, Podesta then clicked on an illegitimate link to change his password, and, thusly, his credentials were stolen, then later used to pilfer and dump embarrassing emails on WikiLeaks.

5 Privileged Password Management Best Practices

Ultimately, the mission of privileged password management is to effectively manage the lifecycle of privileged credentials to facilitate secure authentication for users, applications, machines, and tools to resources and to perform special processes. Most organizations lie somewhere on the continuum between manual and automated processes in their enterprise password management approach. With threats in the form of insiders who've honed internal knowledge of systems and resources, and from attackers armed with automated hacking toolkits—organizations relying on manual processes to manage passwords are at considerable disadvantage. To put it simply, if your attackers are fully embracing automated tools—why aren't you?

Achieving holistic enterprise password management will follow the course of a graduated approach

This section covers eight core areas of focus to improve your management of privileged accounts and credentials. Most likely, achieving holistic enterprise password management will follow the course of a graduated approach. By reading on, you will discover insights on where to start and how to proceed.

While you can tackle each of the eight areas piecemeal by applying a combination of manual and automated solutions, if you're ultimately interested in a completely automated approach, you shouldn't need eight different solutions. In fact, some privileged password management solutions will provide automated capabilities and a streamlined workflow across the entire password management lifecycle.

For holistic management of privileged accounts and credentials focus on these eight areas:

8 BEST PRACTICES

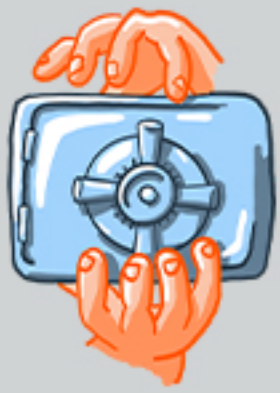
for Privileged Password Management

1 DISCOVER ALL PRIVILEGED ACCOUNTS

Identify where and how privileged credentials are being used and help reveal security blind spots across your on-premises and cloud infrastructure.

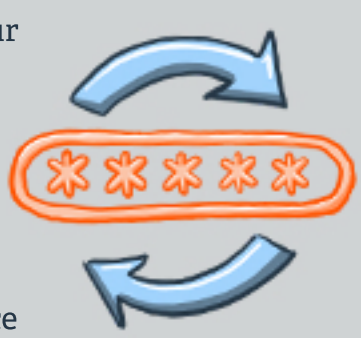
2 BRING PRIVILEGED CREDENTIALS UNDER CENTRALIZED MANAGEMENT

Automatically enforce your privileged password management policy by centrally storing and controlling all privileged credentials.



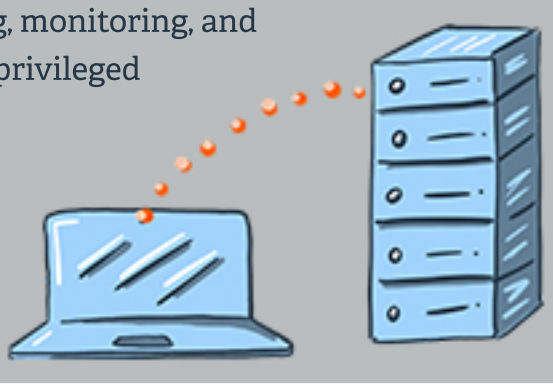
3 ENABLE PASSWORD ROTATION

Regularly rotate all of your privileged credentials at intervals set by your policy, and seamlessly synchronize the password changes to accounts, including service accounts and applications.



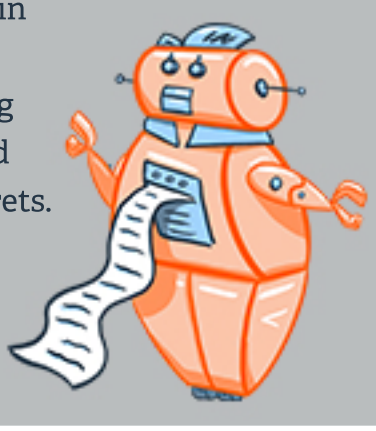
4 IMPLEMENT PRIVILEGED SESSION MANAGEMENT

Ensure complete oversight and accountability over privileged accounts and credentials by controlling, monitoring, and recording all privileged sessions.



5 ADDRESS NON-HUMAN & MACHINE CREDENTIALS

Implement API calls to gain control over scripts, files, code, and keys, eliminating hard-coded and embedded credentials and other secrets.



6 CONTROL SSH KEYS

Approach SSH keys as just another password — accompanied by a key pair that must also be managed — and regularly rotate private keys and passphrases to ensure each system has a unique key pair.

7 UTILIZE THREAT ANALYTICS

Continuously analyze privileged password, user, and account behavior to identify anomalies and potential threats, and accelerate awareness and mitigation.

8 AUTOMATE WORKFLOW MANAGEMENT

Streamline and optimize the entire credential management lifecycle and enable just-in-time (JIT) administration.

1. DISCOVER ALL PRIVILEGED ACCOUNTS

This includes shared admin, user, application, and service accounts, SSH keys, database accounts, cloud and social media accounts, and other privileged credentials – including those used by vendors– across your on-premise and cloud infrastructure. Discovery should include every platform (Windows, Unix, Linux, Cloud, on-prem, etc.), directory, hardware device, application, services / daemons, firewalls, routers etc. This process should also entail the gathering of user account details that will help assess risk, such as privilege level, password age, date logged on, and expired, and group membership and services with dependencies to the account.

Discovery should illuminate where and how privileged passwords are being used, and help reveal security blind spots and malpractice, such as:

- Long-forgotten orphaned accounts that could provide an attacker with a backdoor to your critical infrastructure
- Passwords with no expiration date
- Inappropriately use of privileged passwords—such as using the same Admin account across multiple service accounts
- SSH keys reused across multiple servers
- Service accounts, application accounts, etc., with hardcoded credentials

Findings from the discovery allow you to rethink your policies and retune the access permissions for the accounts. Since new systems and enterprise applications can sprout up at any time, you will need to perform periodic discoveries to ensure every privileged credential is secure, centralized, and under management.

Manual Approach vs Automated Discovery Solutions

Absent automation, comprehensive discovery is likely to be an inordinately time-consuming endeavor that relies on spreadsheets for recording, and draws on multiple scripting languages, APIs, etc. Even then, this approach will frequently result in missed credentials and security gaps (see more on applications password management below). With third-party solutions, you can automate scanning of IP addresses, ports, systems, services, applications, cloud, and even social media accounts. A process that could take eternity (as counted in human years) with a manual approach can be condensed into just minutes with an automated solution.

2. BRING PRIVILEGED CREDENTIALS UNDER CENTRALIZED MANAGEMENT

Optimally, the onboarding process happens at time of password creation, or otherwise, shortly thereafter during a routine discovery scan. Silos of individuals or teams (i.e. DevOps) independently managing their own passwords are a recipe for credential sprawl and human error. All privileged credentials should be centrally secured, controlled, and stored. Ideally, your password storage supports industry-standard encryption algorithms, such as AES 256.

Manual Approach vs Automated Solutions

You could centralize storage of privileged credentials in an encrypted database, and also record values in an Excel spreadsheet. An automated enterprise password safe solution will provide an encrypted database from which you can manage the password lifecycle. An enterprise password safe can automatically enforce your privileged password management policy, such as password complexity, uniqueness (different passwords per asset, account, etc.) expiration, rotation, check in and check out, elimination of default passwords, and other rules. This will drastically simplify the discovery and onboarding of credentials for new privileged accounts as they're created.

3. ENABLE PASSWORD ROTATION

Rotation policies should address every privileged account, system, networked hardware, and IoT device, application, service, etc. This reduces the threat window for password reuse attacks. As covered earlier, passwords should be unique, never reused or repeated, and randomized on a scheduled basis, upon check-in, or in response to specific threat or vulnerability.

Manual Approach vs Automated Password Rotation

You can rotate privileged credential values in an Excel spreadsheet and then manually log in to the associated accounts and systems. While not highly scalable, this can provide some password management coverage in simple environments, but management and rotation of some types of credentials (i.e. hard-coded passwords and keys) will likely prove impossible. An automated, third-party approach relying on an enterprise password safe means you can rest assured that all of your privileged credentials (thousands to millions) are regularly rotated at intervals set by your policy. Additionally, with an enterprise password safe, you can seamlessly synchronize the password changes in the directory where the account resides with the changes in the system/device/application/service where the password is used, to avoid any downtime.

4. IMPLEMENT PRIVILEGED SESSION MANAGEMENT

These solutions ensure complete oversight and accountability over privileged accounts and credentials. Privileged session management refers to the monitoring, recording, and control over privileged sessions. IT needs to be able to audit privileged activity for both security and to meet regulations from SOX, HIPAA, GLBA, PCI DSS, FDCC, FISMA, and more. Auditing activities can also include Dual Control (requiring two separate people to approve access or the execution of specific commands), the capturing of keystrokes and screens (allowing for live view and playback), the ability to record, lock, and document suspicious behavior without terminating sessions – or productivity, and other measures. You should be able to control, monitor and record every session across your privilege universe—whether initiated by employee or vendor, human or non-human.

Manual Approach vs. Third-Party Privilege Session Management Solutions

While you can certainly manually implement some processes, such as screen recording, automated solutions allow you to accomplish it seamlessly and at the scale of hundreds or thousands of concurrent sessions. Moreover, the best third-party solutions can provide automated workflows that give IT granular control over privileged sessions, such as allowing them to pinpoint an anomalous session and pause, lock, or terminate it until a determination is made that the activity is appropriate.

5. ADDRESS NON-HUMAN & MACHINE CREDENTIALS

Simply put, this requires deploying a third-party application password management or secrets management solution that forces applications and scripts to call (or request) use of the password from a centralized password safe. By implementing API calls, you can wrest control over scripts, files, code, and embedded keys, eliminating hard-coded and embedded credentials. Once this is accomplished, you can automate management of the password as often as policy dictates. And, by bringing the application password or DevOps secret under management and encrypting it in a tamper-proof safe, the credential and underlying applications/tools are vastly more secure than when the passwords remained static and stranded within code.

6. CONTROL SSH KEYS

NIST IR 7966 offers guidance for businesses, government organizations, and auditors on proper security governance for SSH implementations that include recommendations around SSH key discovery, rotation, usage, and monitoring. Approach SSH keys as just another password, albeit accompanied by a key pair that must also be managed. Regularly rotate private keys and pass phrases, and ensure each system has a unique key pair.

Manual Approach vs Automated SSH Key Management

To identify accounts set up to use SSH keys, you could manually pour through authorized keys file in the hidden .SSH user folder, but this still won't help you identify who has the private key matching any of the public keys in the file. While manual SSH Key management is better than no management at all, in even modestly complex environments, manual rotation of SSH keys is an unsustainable strategy. If this is the case, look to a third-party solution to generate unique key pairs for each system, and perform frequent rotation. Automated, third-party SSH key management solutions will substantially simplify the process of creating and rotating SSH keys, eliminating SSH key sprawl, and ensuring SSH keys enable productivity without compromising security.

7. UTILIZE THREAT ANALYTICS

To mitigate risk, and evolve your policy as needed, you should continuously analyze privileged password, user, and account behavior, and be able to identify anomalies and potential threats. The more integrated and centralized your password management, the more easily you will be able to generate reports on accounts, keys, and systems exposed to risk. A higher degree of automation can accelerate your awareness and orchestrated response to threats, such as enabling you to immediately lock an account or session, or change a password, such as when incorrect passwords (as with a brute force or dictionary attack) have repeatedly tried to gain access to a sensitive asset.

8. AUTOMATE WORKFLOW MANAGEMENT

While you can certainly build your own internal rule sets to trigger alerts, and apply some policies around password management, third-party solutions provide robust capabilities that can streamline and optimize the entire password management lifecycle.

Enterprise-class privileged password management solutions can also help automate:

- Grouping and management of assets in accordance to their profile and smart categories.
- Workflows for device access, including an approval process for when administrative access is required; consistent with least privileged access, you may want to implement context to workflow requests by considering, and potentially restricting access depending on the account, day, date, time, timeframe, and location (IP addresses) when a user accesses specific resources.
- Just-in-time administration (JIT), ensuring identities only have the appropriate privileges when necessary and for a limited amount of time, drastically reducing the window of vulnerability during which time a threat actor can exploit account privileges.
- Workflows to accommodate fire-call or break-glass requests to ensure access to password-managed systems afterhours, on weekends, or in other emergency situations.
- Password Check In and Check Out from the password safe and automated authentication or Single Sign On (SSO) for the user without any manual log-in requirements, so that privileged credentials, including for cloud administrative consoles, are never revealed to the user.
- Logon of users for RDP and SSH sessions, without revealing passwords.
- Triggers requesting a supervisor's approval in order to checkout highly sensitive credentials.
- Commencement of privileged session monitoring and alerting of any sensitive or suspicious activity.

6

Implementing Privileged Password Management

As with any IT security and governance project, start with a scope. Once you've completely discovered all of your privileged accounts and have a baseline of your privileged credential and asset risk, you can set priorities and flesh out a privileged credential policy. Incrementally implement automation across your privileged password management lifecycle to help scale your efforts to enforce best practices around all credentials.

Incrementally implement automation across your privileged password management lifecycle to help scale your efforts to enforce best practices around all credentials

Tackling privileged password management doesn't occur in a vacuum; you should also have a strong handle on the principle of least privilege and will need to implement it within an over-arching privileged access management framework.

7 Resources

BeyondTrust is a PAM technology leader and offers the broadest set of credential management capabilities in solutions tailored to address your specific risks and use cases. Contact BeyondTrust today to discuss how you can discover, manage, audit, and monitor privileged accounts of all types.

Additional resources:

- The Privileged Access Management Buyer's Guide
- Journey to Universal Privilege Management
- The Guide to Just-in-Time Privilege Management
- How to Access Privileged Passwords in 'Break Glass' Scenarios
- Enterprise Password Management (demo)

ABOUT PRIVILEGED PASSWORD MANAGEMENT

BeyondTrust Privileged Password Management solutions enable automated discovery and onboarding of all privileged accounts, secure access to privileged credentials and secrets, and auditing of all privileged activities. Security teams can instantly view any active privileged session, and if required, pause or terminate it. Leverage threat analytics that aggregate user and asset data to baseline and track behavior and alert on critical risks. Video recording, keystroke indexing, full text search, and other capabilities make it easy to pinpoint data. Reduce the risk of compromised privileged credentials for both human and non-human accounts while meeting compliance requirements.

ABOUT BEYONDTRUST

BeyondTrust is the worldwide leader in Privileged Access Management, offering the most seamless approach to preventing data breaches related to stolen credentials, misused privileges, and compromised remote access.

Our extensible platform empowers organizations to easily scale privilege security as threats evolve across endpoint, server, cloud, DevOps, and network device environments. BeyondTrust unifies the industry's broadest set of privileged access capabilities with centralized management, reporting, and analytics, enabling leaders to take decisive and informed actions to defeat attackers. Our holistic platform stands out for its flexible design that simplifies integrations, enhances user productivity, and maximizes IT and security investments.

BeyondTrust gives organizations the visibility and control they need to reduce risk,achieve compliance objectives, and boost operational performance. We are trusted by 20,000 customers, including half of the Fortune 500, and a global partner network.

beyondtrust.com