IBM Security Verify Privilege On-Premises
Version 10.9

Best Practices Guide

Table of Contents

Last updated: March 26, 2021

Getting Started

This document was developed after experiences with many successful customer deployments of Privilege Vault in their organizations. It covers the issues that most customers tackle as they consider which data to store, who needs access, what permissions to apply, and how to organize all their sensitive data. This document is not meant to cover everything; see the Further Resources section if your question is still unanswered.

Consider Privilege Vault as a platform for your organization to store all its passwords and sensitive data. It can be configured to work in many ways that depend on your industry, compliance requirements, and ultimate end goals. The trick is to know your objectives and then match the capabilities and best practices to your situation.

INSTALLATION

Before you install Privilege Vault, see the system requirements. The process for installing Privilege Vault is outlined in the installation guide by matching the version of Windows Server you use. If you have an active trial or purchased Privilege Vault licenses, find the installer, installation guides, and your licenses.

BASIC CONFIGURATION

When Privilege Vault is installed, see the Getting Started guide to begin setting up Privilege Vault right away.

- Adding your licenses
- Basic security settings
- Configuring automatic backups
- Setting up access for local and AD users
- Creating a Secret
- Heartbeat
- Remote Password Changing

SECURITY HARDENING

Security is a process - not a product. See the Security Hardening guide to ensure that your implementation of Privilege Vault has optimal security. The guide contains more in-depth recommendations for not only configuring the application in a secure manner, but also hardening the servers that Privilege Vault is hosted on.

TERMINOLOGY

Throughout this Best Practices Guide, certain terms refer to specific features or concepts within Privilege Vault.

Administrator

You can grant access to all the features within Privilege Vault to users by creating and assigning different roles. “Administrator” is one of the default roles that comes installed with Privilege Vault. By default, this role contains all role permissions, but it can be customized as well. In this guide, when “administrator” is used in the context of a Privilege Vault user, it is referring to the user who generally has the most permissions and manages the system. Administrators have control over the global security and configuration settings.

> *Administrators in Privilege Vault DO NOT automatically have access to all data stored in the system – access to data is still controlled by explicit permissions on that data.*

Secret

A Secret is any sensitive piece of information (typically a password) that you would like to manage within Privilege Vault. Typical Secrets include (but are not limited to) privileged passwords on routers, servers, applications, and devices. You can store files in Secrets, which allow storage of private key files, SSL certificates, license keys, network documentation information, or even a Microsoft Word or Excel document.

Privileged Account Management (PAM) Strategy

It is important to have a privileged account management strategy that helps you determine which types of features to use for your various accounts and sensitive data you are storing. See some of the suggested guidelines for creating a strategic plan. Read all sections of the guide for a comprehensive look at ways you can secure your Privilege Vault. However, these guidelines also link to other parts of the guide so you can choose to jump to a specific section for more information about a topic.

IDENTIFY DATA AT RISK

Consider all the types of sensitive data your team needs to be securely stored and managed. Where are the biggest risks and pain points in your current password management strategy? Data at risk also often includes more than just passwords.

To get started, consider these key accounts and principles:

- All shared privileged accounts: these are accounts that don't identify an individual (for example: administrator, root, enable, service accounts). These must have randomized passwords that are changed frequently.
- Do your users have individual privileged accounts? Maybe each user has a separate AD account for domain admin rights?
- Every password in your organization should be different.
- What passwords might an employee walk off with, if they ever quit or are fired?
- What passwords might be needed in an emergency, outside of regular business hours, or when someone is on vacation?

Typical account passwords and sensitive data that is being stored in Privilege Vault:

- Windows local administrator accounts
- UNIX, Linux, Mac root, and local user accounts
- Active Directory domain administrator accounts
- Active Directory service accounts
- Database accounts (MS SQL, Oracle, MySQL, and so on)
- Network equipment passwords (router, switches, firewalls, phones, appliances, and so on)
- Application passwords (SAP, custom apps, and so on)
- Website passwords (cloud services, DNS, Amazon AWS, vendors)
- Sensitive files (private key files, SSL certificates, network documentation information, and so on)
- Software license keys, serial numbers, personnel data, wifi passwords

WHO WILL ACCESS PRIVILEGE VAULT?

After determining the data you'll store in Privilege Vault, the next step is to decide who will use Privilege Vault to access and manage that data. A common approach is to begin by focusing on one group of users and the passwords they use regularly, later expanding to other teams after a good strategy has been put in place.

However, you might find it more beneficial to organize Privilege Vault for use by all your users or teams so that you can design an effective overall folder and policy structure that works well across all teams.

WHAT LEVELS OF PRIVILEGE ARE NECESSARY?

Giving a user access to an account in Privilege Vault can entail different levels of privilege. Do you want a user to be able to edit the user name, machine, or password of a Secret, or only view the Secret? Should they be able to share the Secret with other users? Once you incorporate use of the Launcher into your users' workflow for authenticating to an application, do they really need to know a password, or can you mask it? The Workflow Security section can help you determine and implement key measures to ensure that users have least privilege necessary.

WHAT ARE YOUR PASSWORD REQUIREMENTS?

It's unlikely that all your accounts have the same password complexity requirements and rotation schedule. In fact, for best security, you should have some variation. You can create sets of password requirements to control password length, characters, and complexity, then apply those to various account types by using Secret templates. Secret templates also allow you to set a default expiration period, which can convert to how often an account password will be changed automatically.

EVALUATE YOUR EXISTING SETUP

While transitioning to using a new tool for managing your passwords, it's important to consider how accounts are currently used in your environment. The following questions can help evaluate this:

- Do some of your users have their own, individual AD domain admin accounts, or are there only a few shared domain accounts?
- Do users use local administrator accounts or privileged domain accounts for admin access to systems?
- Are permissions to resources (servers, applications, and so on) controlled by using Active Directory group policy?

DEFINE YOUR CORE PAM STRATEGY

There are a few different strategies that typically work best in Privilege Vault. Other methods of password management might work but require a more significant amount of time and effort to configure and maintain. The most commonly used strategies are defined below.

Individual Privileged Domain Accounts

In this scenario, IT team members have their own domain admin accounts that are tied to their identity. They use these accounts to gain elevated privileges to resources such as production servers. Permissions to the various resources they're permitted to access are controlled by Active Directory.

To implement this in Privilege Vault, each account is stored as its own Active Directory account Secret. Only the user who is tied to that account is granted permissions to the Secret. A security setting such as Check Out

(one-time password) or Hide Launcher Password is enabled so the user depends upon Privilege Vault to use the account. Therefore, all access to that account is audited. When the IT admin needs elevated privilege to a box, they Check Out or view the Secret and then use the Launcher to access the machine.

A benefit of this strategy is that there isn't conflict with multiple users trying to use the same account for access to one machine. This strategy provides great accountability – the security team knows the exact user accessing an account and the machine being accessed. The password is not shared among multiple users, and all privileged access is audited by Privilege Vault.

A pitfall of this strategy can be that there is more management of permissions that are required in Active Directory. While machines might be white- or black-listed to force users to use the Privilege Vault Launcher, thus controlling machine access through Privilege Vault, this can be tedious.

It's more secure, less work, and simpler to organize permissions for access to domain resources in Active Directory. This strategy works best for organizations that already use Active Directory heavily to control permissions of individual privileged users to domain resources. Ongoing maintenance relies on updating permissions to resources in Active Directory and ensuring that all new individuals' privileged accounts are being added for management under Privilege Vault.

Shared Privileged Domain Accounts

You might choose to have your users use shared privileged accounts to access resources. This strategy involves creating a few service accounts that have permissions to OUs or groups of computers. In Privilege Vault, these accounts can be limited with the Launcher so they can only be used to Launch to certain computers. This means you can limit the number of domain accounts that are created and set permissions more broadly (such as at OU level). These passwords might be changed on a schedule or, where possible, used with Check Out to change the password after each use. Using this setup, accounts can be designated for team or function and can have varying Check Out intervals set to ensure that only one person at a time is using each account.

A benefit of this strategy is if individuals do not already have their own privileged domain accounts in Active Directory, then giving them access to shared accounts means less setup in AD while still maintaining accountability for who uses which account, and which machine they access.

A pitfall of this strategy can be that if the team (or function-specific accounts) cover a broad number of machines that can be accessed, it might be much work to set up Launcher white- and black-lists to control access through Privilege Vault. However, if these permissions are set only through Active Directory, it is difficult to have the visibility into these limitations for an auditor.

Hybrid of Individual and Shared Accounts

Sometimes, your employees' roles might require longer, more specialized access. For those accounts, you can have individual privileged domain accounts, and for the other regular users you can use a few shared privileged domain accounts. These can be stored in Privilege Vault, but with different settings governing their usage. For example, the shared accounts would still have Check Out enabled, while the individual privileged accounts will

simply have permissions limited to an individual user, possibly with the password hidden by using Hide Launcher Password.

WHAT IS THE HIGHEST RISK?

Implementing a comprehensive PAM policy should eventually cover all your privileged/shared accounts, but this can take some time. When looking at where to start, it's important to consider the areas of risk that your team must deal with. Where are the areas that need more immediate attention? Is it local Windows admin accounts all sharing the same password? Pass-the-Hash vulnerability? Protecting your network equipment passwords? Avoiding fines for not meeting compliance mandates? Maybe it is password misuse and auditing employee access to accounts is most important to you. Choose a starting point that will that gives your organization the most value, and then branch out from there.

Users

At minimum, the administrators who manage and use your organization's privileged passwords and data regularly needs to access your Privilege Vault. Privilege Vault users can be defined in two ways: as local Privilege Vault accounts or as Active Directory user accounts. Privilege Vault also has the concept of groups, which can be local (you create them in Privilege Vault) or AD-synced (security groups from Active Directory). Groups are a powerful tool for assigning and maintaining permissions to Secrets, and therefore should be given careful thought and planning.

LOCAL PRIVILEGE VAULT ACCOUNTS

Local users and groups must be created and managed manually in Privilege Vault, as they are not integrated with Active Directory. The first account that you create in Privilege Vault is an example of a local account. Local groups can include local users and Active Directory accounts and can have a user who is established as the group owner that is permitted to add or remove users to/from the group.

ACTIVE DIRECTORY ACCOUNTS

Active Directory accounts can be added for access to Privilege Vault either manually (one by one) or by AD security group. When adding users by security group, you choose which groups Privilege Vault synchronizes with Active Directory to update which users' access to Privilege Vault is enabled or disabled. Active Directory group synchronization happens on a regular, customizable interval to keep group membership changes that happen in Active Directory up-to-date in Privilege Vault as well.

LOCAL OR ACTIVE DIRECTORY ACCOUNTS?

It's recommended to use one of the following options:

- Only local users and groups (best security)
- Only Active Directory users and groups (most convenient)
- A hybrid of Active Directory users and local groups (balance of security and convenience)

You need to choose an option that provides the levels of security and convenience that are acceptable for your organization. Using the AD accounts option is easy for user maintenance, but it limits the security of Privilege Vault to the level of security of your Active Directory. This might be fine - be sure to consider the question of domain admin access to AD in combination with Privilege Vault permissions.

Only Local Users and Groups

Creating local users and groups within Privilege Vault provides much flexibility because you can tailor permission assignment by group to your exact needs. The major benefit of local users and groups is security: users and group membership can be controlled entirely by role-based access control (RBAC) within Privilege Vault. However, this approach requires more maintenance because creating/deleting users and managing group membership must be controlled in Privilege Vault.

Only AD Users and Groups

If you are considering the use of AD users and groups for Privilege Vault access and permissions assignment, review your teams that need access to Privilege Vault. Compare them to the corresponding groups in your Active Directory. If your AD groups map to ways you want to assign access to Secrets, you can synchronize your AD groups with Privilege Vault and start assigning permissions to Secrets (and levels of those permissions – View/Edit/Owner) by group. You can then effectively manage Privilege Vault access and Secret permissions completely from AD by changing AD group membership.

Many customers choose this option because they can maintain control in AD and do not worry about any user or group maintenance within Privilege Vault. If you want to use this option but your AD groups don't match the way that you want to assign Secret permissions, you need to create new AD groups to match this or might want to consider the hybrid approach (below), using local groups instead.

Hybrid of AD Users and Local Groups

A third option is to create local groups in Privilege Vault and add Active Directory users to those groups for organizing how permissions are assigned to Secrets. Many customers who use this setup create a single AD security group (for example, SecretServerUsers) to use to synchronize their AD users with Privilege Vault. This approach is more secure than using only AD groups and users, but it's important to keep in mind that an adversary might still reset an AD account password to gain access to your Privilege Vault.

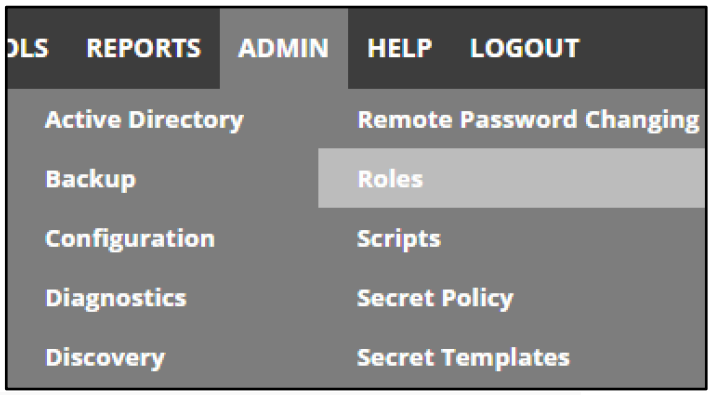
Strong Authentication

Protect the tool that you use to secure your privileged accounts by adding a second factor of authentication for users logging in to Privilege Vault. Two-factor authentication can be added whether users are logging in with local or AD accounts. For more information about using two-factor authentication with Privilege Vault, see the Security Hardening Guide.

Roles

Roles control which features of Privilege Vault a user can use, view, or administer. Existing roles can be customized, and new roles can be created as needed.

Privilege Vault comes with several roles by default, including Administrator, User, and Read Only. You must review the default roles and decide whether your organization needs further roles for various purposes such as third-party consultants or auditors.



Users with the default Administrator role (which contains all role permissions available) do not automatically have access to all data stored in your Privilege Vault. Secrets are only visible to a user based on the explicit Secret permissions that are assigned to them.

It is highly suggested to pull one or both role permissions pertaining to Unlimited Administration mode out of the default Administrator role. Unlimited Administrator mode is a “break-the-glass” feature that allows a user to view all Secrets in Privilege Vault. By splitting the Unlimited Administration permissions into separate roles, it ensures that no one user can both turn on the feature and operate as the unlimited administrator. For more information about how Unlimited Administration mode works and how to effectively control the relevant role permissions, see the Security Hardening Guide.

ROLE DEFINITION AND ASSIGNMENT

Once you have defined your roles, they will seldom need to be changed. Access to modify and assign roles should be tightly controlled.

GROUP ASSIGNMENT

If roles are assigned to groups, then assignment of the groups will also need to be controlled. Often very sensitive role permissions such as Unlimited Administrator are assigned at the user level to limit the risk of granting group assignment permissions.

Permissions

You have different sets of passwords that should only be viewed by administrators. You might also have certain passwords that should be read-only to some administrators, editable by others, and not even visible to other administrators. These options are possible to configure by using the permissions within Privilege Vault.

Permissions can be allocated at the individual user level but it tends to be easier to manage over time if you allocate your permissions at the group level. You need to decide whether your existing AD groups might work for these permissions or if you need to create new AD groups or if you want to create and manage local groups in Privilege Vault.

For more information about what each level of permissions entails, see the User Guide.

Folder Structure

USING FOLDERS TO CONTROL ACCESS (INHERIT PERMISSION)

You can apply permissions (View / Edit / Owner) at the Secret level. This allows you to apply very granular permissions on a single Secret if needed. Managing permissions on each Secret is powerful for situations where you need that flexibility, but it tends to be harder to manage over hundreds or thousands of Secrets. Instead, you should consider by using Folders to control permissions for most Secrets. This can be done by creating a folder structure that best represents your organization, teams, or data being stored; then, apply permissions (View / Edit / Owner) on the folders, by using inheritance across folders where appropriate. Secrets that are placed in a folder can then inherit the permissions of the folder.

DECIDING ON YOUR FOLDER STRUCTURE

The folder structure creates a hierarchy for organization and permissions. This means that folders near the root level need to break out access in high-level terms and then get more specific permissions (typically breaking inheritance) as you move down to the “leaf level” subfolders.

For example:

- Information Technology
    - Technical Services
        - Systems
            - Windows
            - UNIX
            - Network Infrastructure
        - Database
            - Oracle
            - SQL Server
    - Development Services
        - Programmers
- Vendors
- Human Resources
- Customers

The most typical configuration is to break out the folders based on the teams that need to use those folders with the most restrictive permissions at the outer most “leaf” folders of the tree.

An Oracle DBA might have the following permissions on the above folders:

- Information Technology (VIEW)
    - Technical Services (VIEW)
        - Database (VIEW)
            - Oracle (VIEW / EDIT / OWNER)

- SQL Server (VIEW / EDIT)

> ***Note***
>
> *A user will not be able to see the full folder structure unless they have View permissions on all the parent folders of a folder. For example, a user with View on the "Oracle" folder, would also need View on "Database", "Technical Services" and "Information Technology" to be able to see the full folder path.*

There are settings under **ADMIN** > **Configuration** > **Folders** to control whether inheritance on folders and Secrets should be turned on and whether users should always see all folders. There are many ways to configure this for your organization.

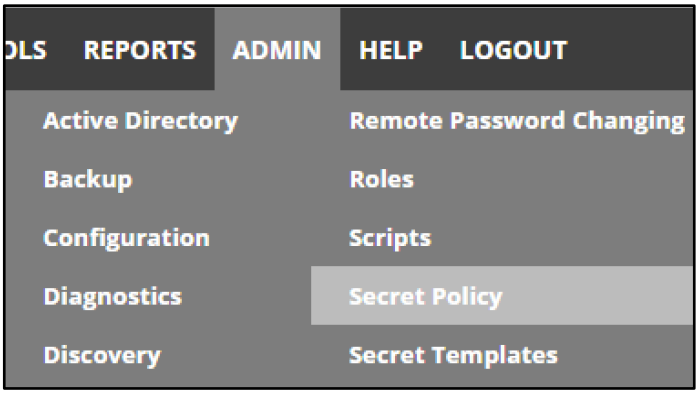
The most common approach is:

- Use inheritance.
- Don't allow users to see folders unless they explicitly have View permissions.
- Require all Secrets to have a Folder.

This allows different teams or even different departments within your organization to use the same Privilege Vault instance independently.

Secret Policy

A Secret Policy is a set of security and remote password changing settings that are normally applied to a Secret on the Security or Remote Password Changing tabs. The benefit of using a Secret Policy is not only that settings can be applied *in bulk* to Secrets (for example, by folder), but that these settings can also be *enforced*, preventing users from changing them.



Secret Policies should be established to apply settings to Secrets that are key to the workflow your organization is working toward. For example, if your primary concern is more detailed auditing around service account usage and you also have a requirement that all service account passwords change every 60 days at 2am on the next Tuesday, you can create a policy that includes these settings and apply it to the folder(s) that will contain all your service accounts. Whenever new accounts are added to the folder, such as when they're imported via Discovery, the settings will automatically be applied and enforced.

Secret Policies can also be updated after they've been assigned to folders. Therefore, if your password policy changes and you need your service account passwords to change every 30 days, you can update the policy. It will immediately apply to all Secrets the policy is assigned to.

Discovery

This section of the document discusses some key best practices around using Privilege Vault's Discovery feature to find and manage accounts in your environment. See Further Resources for a link to the comprehensive guide to configuring and by using Discovery.

DISCOVERY WORKFLOW

While it might be tempting to immediately get started by using Discovery to get your accounts under control, here are a few things that you can do ahead of time to make the enforcement of your organization's password policies more streamlined.

- Know which Secret template you want to import accounts to. This template can affect password change and Launcher settings that are applied to your imported accounts.
- Have a folder structure that is established so you have folders that are appropriated for each type or category of discovered accounts.
- Apply a Secret Policy to the folders you are importing to.

Having these settings in place can save you the considerable amount of time it might take must reorganize all your accounts and policies post-import.

LOCAL WINDOWS ACCOUNTS

How many local Windows accounts in your environment use the same password? Are they local admin accounts? Use Discovery to quickly mitigate the risk of pass-the-hash attacks by finding all your local Windows accounts and setting their passwords to unique, strong passwords managed by Privilege Vault. Where your admins previously had to remember one password to access all computers with local admin rights, they must now remember zero passwords because they can use Privilege Vault to find the computer and start an RDP session by using the local admin account without ever knowing, copying, or typing the password.

FIND BACKDOOR ACCOUNTS

Ensuring that users are not creating back door administrative accounts on Windows computers are important. These accounts can compromise general security and open the potential for a user to access a computer directly without being audited. By running Discovery on a regular interval and having Discovery Rules alerting you when new accounts are found, you can ensure that users any new local Windows account that is being created are identified and are either removed or brought into Privilege Vault.

SERVICE ACCOUNTS

Many organizations do not know where their Active Directory Service Accounts are being used across the network. By using Discovery to scan your network, you can find all the Windows services, application pools, and scheduled tasks that are run by Active Directory service accounts. When these accounts are found and brought into Privilege Vault, having Discovery run regularly can find any new locations where the account is

being used since they were added to Privilege Vault. With Discovery Rules, additional dependencies are automatically added to the existing Secrets. It is suggested that you ensure that the Service Account Discovery is run before you use Privilege Vault to change the service account password.

UNIX ACCOUNTS

When you scan for UNIX accounts, it is suggested to use SSH key validation, as discussed in the Security Hardening guide. This validation ensures that you are connecting only and trying to authenticate to UNIX servers that have a valid and trusted SSH key.

ESX/ESXI ACCOUNTS

Local accounts on ESX/ESXi systems must not change when the server is set up and configured. You want to consider creating Discovery Rules that monitor your ESX/ESXi servers and email the proper teams to inform them of any new account found. These accounts really must not be created, so it is important to monitor them and ensure that no one is creating them maliciously.

Workflow Security

Often you will have situations in which you want users to have access to accounts. However, you want these users to have access only under certain circumstances, such as on a specific day or after the approval of a manager. Maybe, compliance requires that you can monitor an active RDP, or that you use a one-time password for certain accounts. This section examines best practices around workflow security settings in Privilege Vault and scenarios when these settings are commonly used.

HIDE LAUNCHER PASSWORD

Many times, giving an employee access to a resource through Privilege Vault does not require that they have access to the actual password for the account used. When the application a user wants can be started by the Launcher, there's no reason they must copy, paste, or type the password. The Hide Launcher Password setting implements the following behaviors:

- Users with access to the Secret see only asterisks (********) in the password field.
- No copy-to-clipboard, field history, or unmask icons next to the field.

> *Users with Edit permissions to a Secret with Hide Launcher Password enabled can still view the password when you edit the Secret. To prevent all possible access to the password, limit users to View permission only.*

This practice can be a key way to reduce exposure of your privileged account passwords. Hide Launcher Password can be enabled for Secrets under the **Security** tab of a Secret or by applying a Secret Policy. You can also remove the ability for a user to see the password for any Secret with a Launcher by removing the **View Launcher Password** permission from their role.

REQUIRE APPROVAL

The Requires Approval for Access setting is typically employed in the following cases:

- When a user must request access to a Secret for a certain period
- When an administrator would like to approve a user's access to a Secret in advance for a period in the future (such as a maintenance period outside normal business hours)
- When a group of administrators would not like anyone to be able to access a Secret without the approval of another administrator.

This setting can be turned on under the **Security** tab for an individual Secret, but can also be applied with Secret Policy. When enabling Requires Approval, users must still have at least View permission to the Secret to request access to it. When access is granted to the Secret, they have whichever level of permission that was assigned to them for the Secret (View, Edit, or Owner). The approvers of the Secret are specified when you enable Requires Approval, and these individuals can modify the period that the requester originally submitted their access request for or deny the request altogether.

To require all approvers of a Secret to also request access from another approver, be sure to enable the **Owners and Approvers also Require Approval** setting.

CHECK OUT

In some scenarios, users must be able to access a password directly. However, you still want to have control over how long they can use the account without the need to approve access each time. In this case, Hide Launcher Password is not a possibility. However, concerns exist about the user knowing what the password is after they are done by using it. Another concern is often the risk that exists with the password hash that is stored locally on remote devices after each use and potential vulnerability to a "Pass-the-Hash" attack.

Check Out is a security setting that means:

- Only one user at a time has access to a Secret
- A user can only access the Secret for a predetermined Check Out interval, such as 30 minutes
- At the end of a Check Out interval (Check In), or when a user manually Checks In the account before the time is up, the Secret is available for Check Out by another user.
- When enabled, the password can also be changed automatically upon Check In

Domain Administrator accounts are a great example of a case in which using Check Out to change the password every time it is used can be beneficial. This behavior ensures that users are not copying the password to Notepad or writing it down for later use. This behavior also invalidates the hash that was stored on the remote computer after a remote desktop session.

Check Out can be turned on under the **Security** tab for an individual Secret but can also be applied through Secret Policy.

SESSION MONITORING

For critical systems and highly privileged accounts, sometimes by having an audit trail showing when someone viewed the account in Privilege Vault is not enough. Maybe the auditor also wants to be able to review what was done with the account on a remote session. For these critical Secrets, it is suggested to enable Session Recording for the Secret. When Session Recording is enabled, all Launcher sessions can be recorded for later viewing by the auditor or manager when they need to investigate the actions that are completed during a remote session.

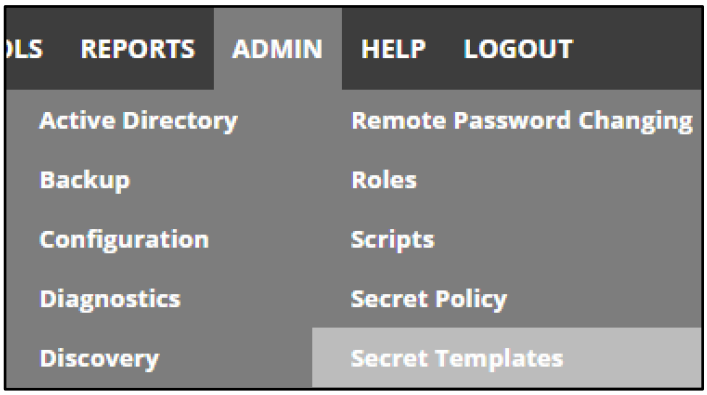
Before you enable Session Recording, you might want to evaluate your users' roles to know exactly who can monitor real-time sessions and view recordings. The permissions to look for are **Administer Session Monitoring**, **View Session Monitoring**, and **View Session Recording**.

Session Monitoring can be turned on under the **Security** tab for an individual Secret but can also be applied through Secret Policy.

Secret Templates

Secret templates in Privilege Vault define the types of data (Secrets) that can be stored, and the settings for that data. You can store just about any type of sensitive data in Privilege Vault.

It's important to review the available templates and decide which ones must be available to your users and where you would like to change the default templates included.



CONFIGURING TEMPLATES

You can customize existing templates or create new templates if necessary. Many templates are included by default that cover common account types. For example, the Active Directory Account template contains the following settings:

- Domain, Username, Password, and Notes fields
- Thirty-day expiration period, applying to the password field
- RDP launcher, requiring user input for computer to connect to
- Password Changing and Heartbeat enabled
- Active Directory password changer, with Default password requirements

These settings are typically sufficient for most organizations to use out-of-box. However, you might want to enable other settings or change settings such as enforcement of a naming convention or more complex password requirements. In this case, you have the flexibility to either modify the existing template, copy the existing template to use as a base for a new template, or create a new template from scratch. The following sections cover some fundamental template settings available for you to customize.

Naming Patterns

Privilege Vault supports enforcement of naming patterns for Secret names. Naming patterns let you maintain consistency for Secret names and can help ease both browsing and grouping Secrets by name. Naming patterns use Regular Expression and allow you to enter a descriptive error message to describe your naming standard to users. The most common naming standard that is used is RESOURCE\ACCOUNT (for example, server0001\administrator). Find this setting by clicking Edit from the template designer page.

Password Requirements

Password Requirements determine the password compliance rules (for example, 16 characters, 1 upper, 1 lower, 1 symbol, and 1 number). These requirements can be customized and applied to passwords at the Secret template level or per individual Secret (under the Security tab). This setting controls the complexity of passwords that are generated by Privilege Vault. Password requirements can also be enforced when users try

to edit or create new passwords, and can be viewed for password compliance in reports. This template lets you have different complexity rules for different types of passwords if needed (Oracle, SQL, Windows, UNIX, and so on). You can choose to have Privilege Vault enforce the password requirements on add or edit by turning on validation on the Secret template (click Edit from the template designer page).

Talk to your security management, auditors, and industry experts – find out the best password complexity settings for your environment. Do not be afraid to increase the settings that you use. For example, use 100-character random generated passwords with combinations of symbols, alphanumeric, and upper or lowercase characters. A platform like Privilege Vault makes it easy to work with passwords so length does not matte, as you can use Launchers, copy-to-clipboard, Auto Change. In fact, large passwords enhance security since administrators are far less likely to remember them, write them down, or want to type them.

Another thing to consider when you create password requirements is which character sets to use. Some systems might not work well with certain characters; for example, underscores can be problematic in certain mainframe environments. You can create your own character sets (ADMIN > Secret Templates > Character Sets) for different password requirements. Use these templates when passwords are generated by Privilege Vault.

Secret Expiration

Privilege Vault uses expiration to ensure that passwords change regularly. Secrets can be set to expire on an interval such as 30 days (or other intervals as needed). Expiration is often combined with automatic password change to control how often a password is changed. Whenever it expires, Privilege Vault can queue the Secret up for a password change.

You can also control which field is used for expiration (this field does not have to be the password field). You might use expiration on a license key and set expiration to when the license is going to expire. When a Secret expires, you can then update the expiration field (say license key) and it will no longer be expired. This is a generic way to ensure that a specific field on a Secret is changed regularly.

Password History

Privilege Vault automatically keeps all history on all fields on a Secret template. All previous values for computer, user name, password, and any other fields are kept. This is helpful in ensuring that previous passwords can be found if needed.

Don’t Forget Your Files

Don't forget files. You can have fields on your Secret template that are for file attachments. This can be used for storing license key files, private keys, SSL certificates, even Microsoft Word, or Excel documents that contain sensitive data.

Session Launcher

The Launcher can be configured on the Secret template to allow any tool to be started by using the Secret. For example, Remote Desktop, PuTTY, web launcher, or a custom launcher you configure for a particular .exe; , MS

SQL Management Studio, SSH clients, FTP tools, and more. You can also use the Launcher with the Hide Launcher Password setting to allow administrators to start tools without revealing the password.

TEMPLATE MANAGEMENT

It is worth spending time in the beginning to define your Secret templates the way you want them before users start adding data. Therefore, when a user creates a new Secret, it is clear which Secret template to use, instead of selecting the wrong one, and attempting to fit account information into an unsuitable template. You can use an option on the Secret that is called Convert Template to convert a Secret to another template later. However, it is much simpler to plan before your organization begins adding data.

The Basics

When you create new Secret templates, make sure that you configure Remote Password Changing, password requirements, Secret expiration, and the Launcher, and ensure that your Secret template names are descriptive and use terms that your users understand. For instance, if you have one template that expires and one that does not, make sure it is clear from the name. If your organization does not use the term Active Directory account, change it to match the organization's language.

Limit Secret Template Administrators

Changing Secret templates must be limited to only a small subset of your Privilege Vault admins. Create a separate role that has the Administer Secret Templates role permission. Remove it from Administrator if you have many administrators. When you have Secret templates that are configured, it is unlikely they must be changed frequently so few people require access.

Deactivate Unused or Retired Templates

Privilege Vault comes with many Secret templates preconfigured. You must decide which templates you want to use and deactivate the others. You can also retire Secret templates if your requirements change over time. Secrets remain when a Secret template is deactivated but no one can create new Secrets for that Secret template.

Privilege Vault uses soft deletes rather than hard deletes. For soft deletes, data is marked as inactive rather than being deleted. This is essential for auditing since data cannot be removed from the system that is causing all audit activity to be lost. Secrets and Secret templates can be deactivated but not deleted. This behavior is something to consider when you configure your Privilege Vault.

Override Settings at the Secret

Many of the settings at the Secret template can also be overridden at the Secret. For example, if you create a Secret for your AD service accounts with a 30-day expiration but need 90 days for an AD service account, you can set it to 90 days for that one Secret. This gives some flexibility for Secrets that need to behave differently than other Secrets by using the same Secret template.

Further Resources

Review these resources for more information:

- Privilege Vault Getting Started Guide
- Privilege Vault User Guide
- Privilege Vault Installation Guides
- IBM Security Verify Privilege On-Premises 10.9: System Requirements
- IBM Security Verify Privilege Vault (previously known as IBM Security Secret Server SaaS): System Requirements
- Support
- Discovery Guide
- Video Tutorials
- IBM Developer