CYBERARK®

PAM Administration

PAM Self-Hosted Architecture

Agenda

In this session, we will look at:

- The **PAM Self-Hosted** system architecture
- The security controls protecting the **Vault** and encryption keys
- **Vault** encryption and key management
- How to locate and manage the local services, configuration files, and logs for the various **PAM Self-Hosted** components
- How to locate and manage the built-in **Safes** and users for the various **PAM Self-Hosted** components
- The internal integration and communication between the various **PAM Self-Hosted** components and the **Vault**

System Architecture Review

What is PAM Self-Hosted?

PAM Self-Hosted

PAM solution when all of its components are owned and operated by the customer

1. An entirely on-premise installation of the **Vault** and all the different components

2. An entirely cloud-based deployment where the **Vault** and components are deployed to one of the supported Cloud platforms

3. A hybrid deployment in which some components are in the Cloud and others, very often the **Vault**, are installed on-premise.

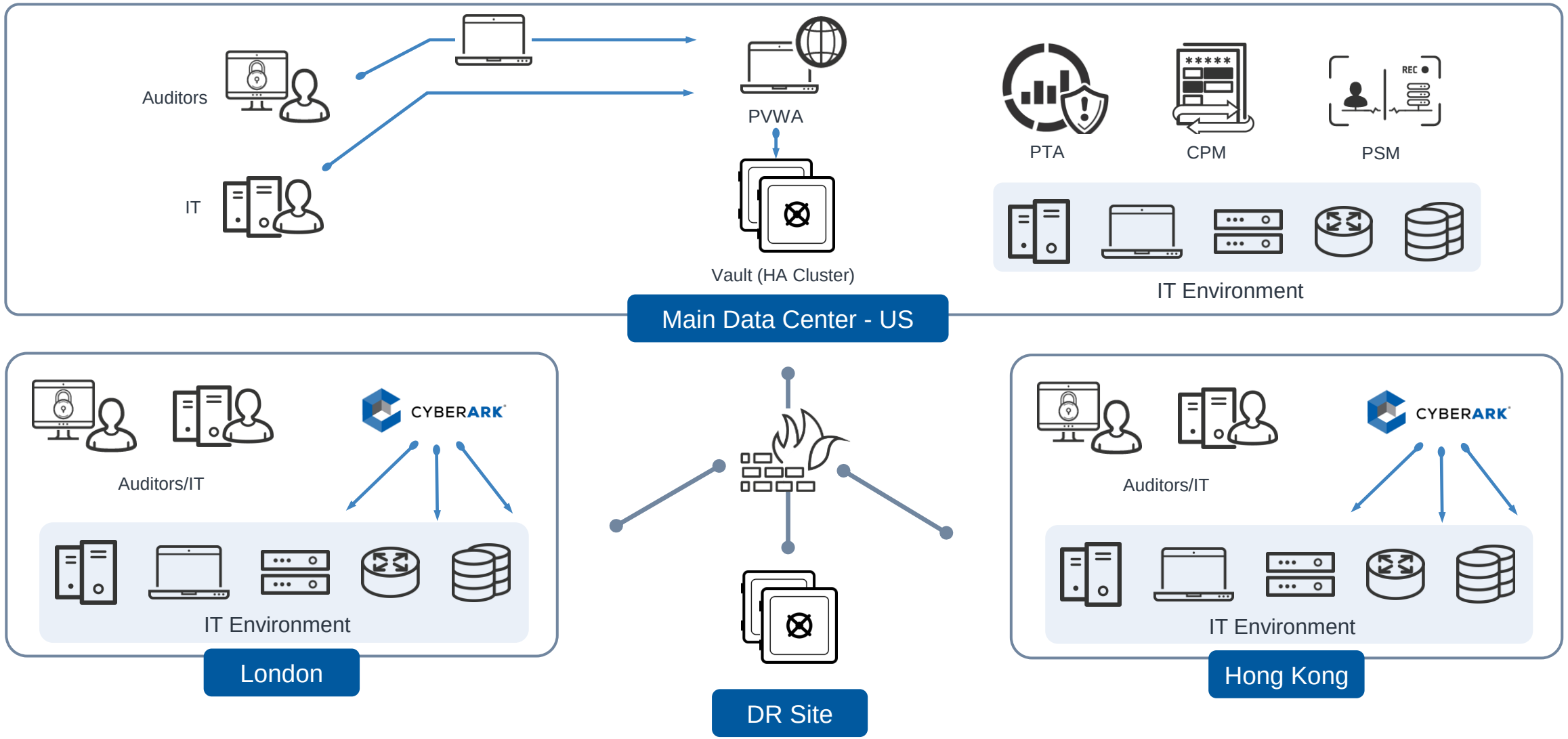
PAM SaaS

The **Privileged Access Manager** is delivered as Software as a Service

PAM Self-Hosted Components

Secure Digital Vault	• A secure server used to store privileged account information • Based on a hardened Windows server platform
Password Vault Web Access (**PVWA**)	• The web interface for users to gain access to privileged account information • Used by Vault administrators to configure policies
Central Policy Manager (**CPM**)	• Performs the password changes on devices • Scans the network for privileged accounts
Privileged Session Manager (**PSM**)	• Isolates and monitors privileged account activity. • Records privileged account sessions
Privilege Threat Analytics (**PTA**)	• Monitors and detects malicious privileged account behavior.

CyberArk's Scalable Architecture



Vault Security

In this section we will discuss the **Vault** security standards and encryption keys management

Vault Security Controls

The Vault: an Island of Security

Isolating the Server

- No domain membership or trusts
- No DNS or WINS
  - Uses a manually configured Host file

Hardening the Server

- Remove unnecessary services
- Secure configuration for remaining services
- Only Vault Server and PrivateArk Client are installed
- No additional applications

Documentation Resources

There are several pages that are key to successfully protecting your implementation

They include:

- Security Fundamentals
- Digital Vault Security Standard

Security Fundamentals

Details eight controls to protect your CyberArk deployment and, therefore, your privileged accounts

1. Isolate and Harden the Digital Vault Server
2. Use Two-Factor Authentication
3. Restrict Access to Component Servers
4. Limit Privileges and Points of Administration
5. Protect Sensitive Accounts and Encryption Keys
6. Use Secure Protocols
7. Monitor Logs for Irregularities
8. Create and Periodically Test a CyberArk Disaster Recovery Plan

CyberArk Digital Vault Security Standards

Securing your CyberArk implementation is CRITICAL!

The ***CyberArk Digital Vault Security Standard*** describes how to securely configure and maintain the digital vault. It details:

1. The Vault Security Layers
2. The Digital Vault Secure Platform and Enterprise Management Tools, including:
   - Backup/HA/DR
   - Monitoring the Vault
   - Remote Administration
   - External Storage
   - Virtualization of the Vault
   - Vault domain membership
   - Anti-virus

In almost all cases, installing third-party applications, virtualization, and external storage result in a relaxation of security.

All customers and partners should carefully read the Secure Platform document.

The Vault: End-to-End Security

Vault User

Stored Credential

Session Encryption → **Firewall** → **Authentication** → **Discretionary Access Control** → **Mandatory Access Control** → **Auditing** → **File Encryption**

- Proprietary Protocol
- OpenSSL Encryption

- Hardened built-in Windows Firewall

- Single or Two Factor Authentication (recommended)

- Granular Permissions
- Role Based Access Control

- Subnet Based Access Control
- Time Limits and Delays

- Tamperproof Audit Trail
- Event-based Alerts

- Hierarchical Encryption Model
- Every object has unique key

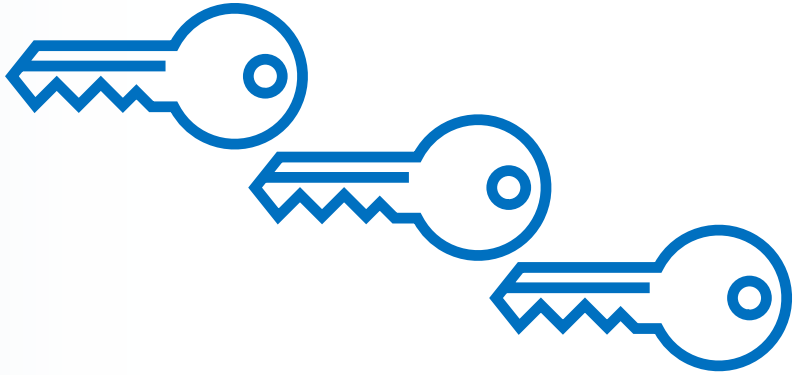
Vault Encryption and Key Management

Encryption Keys

There are three files that form the cornerstone of the CyberArk PAM solution encryption methodology.

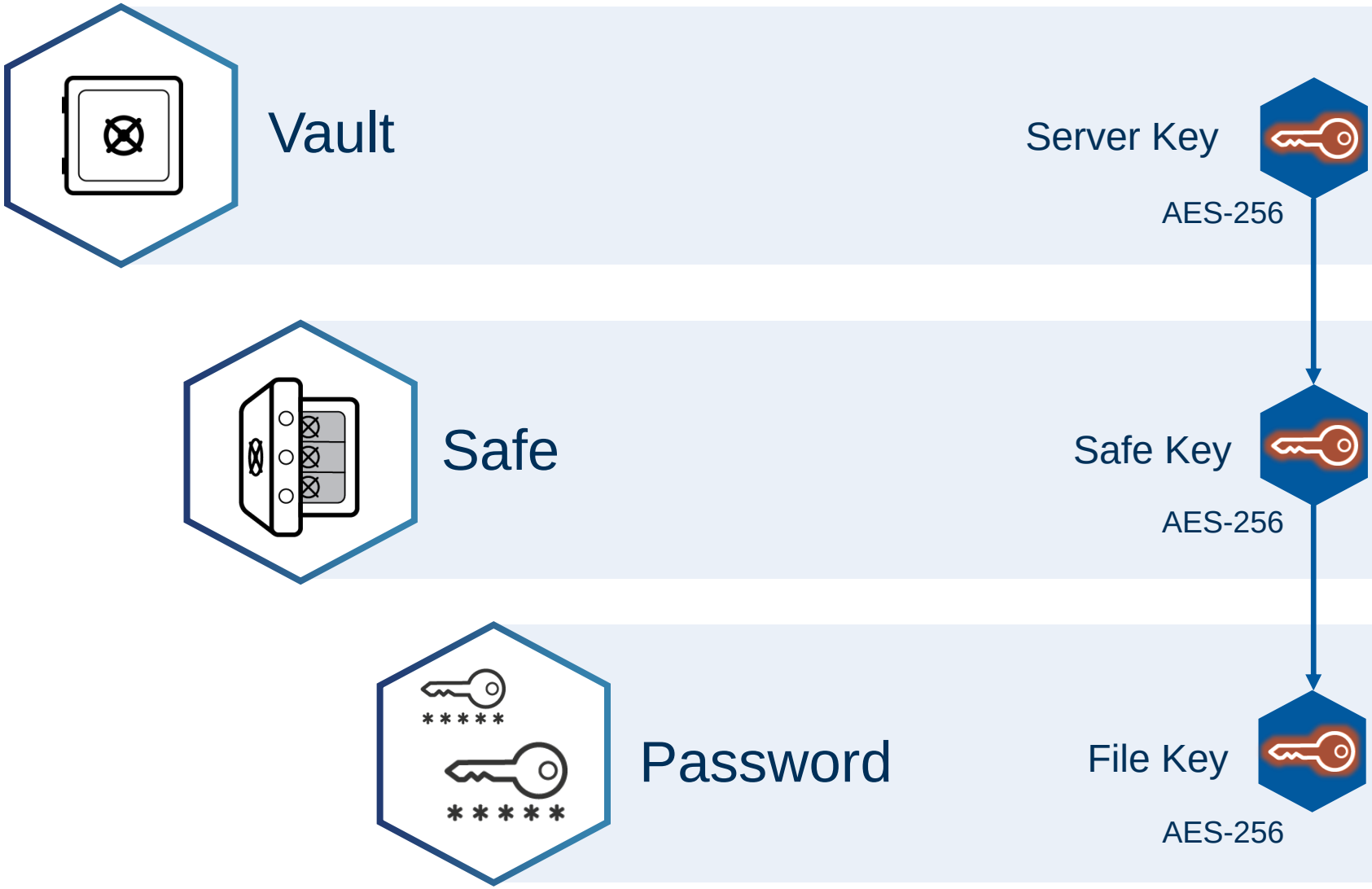
These encryption key files are required to install and operate CyberArk PAM. They are:

- Server Key
- Recovery Public Key
- Recovery Private Key

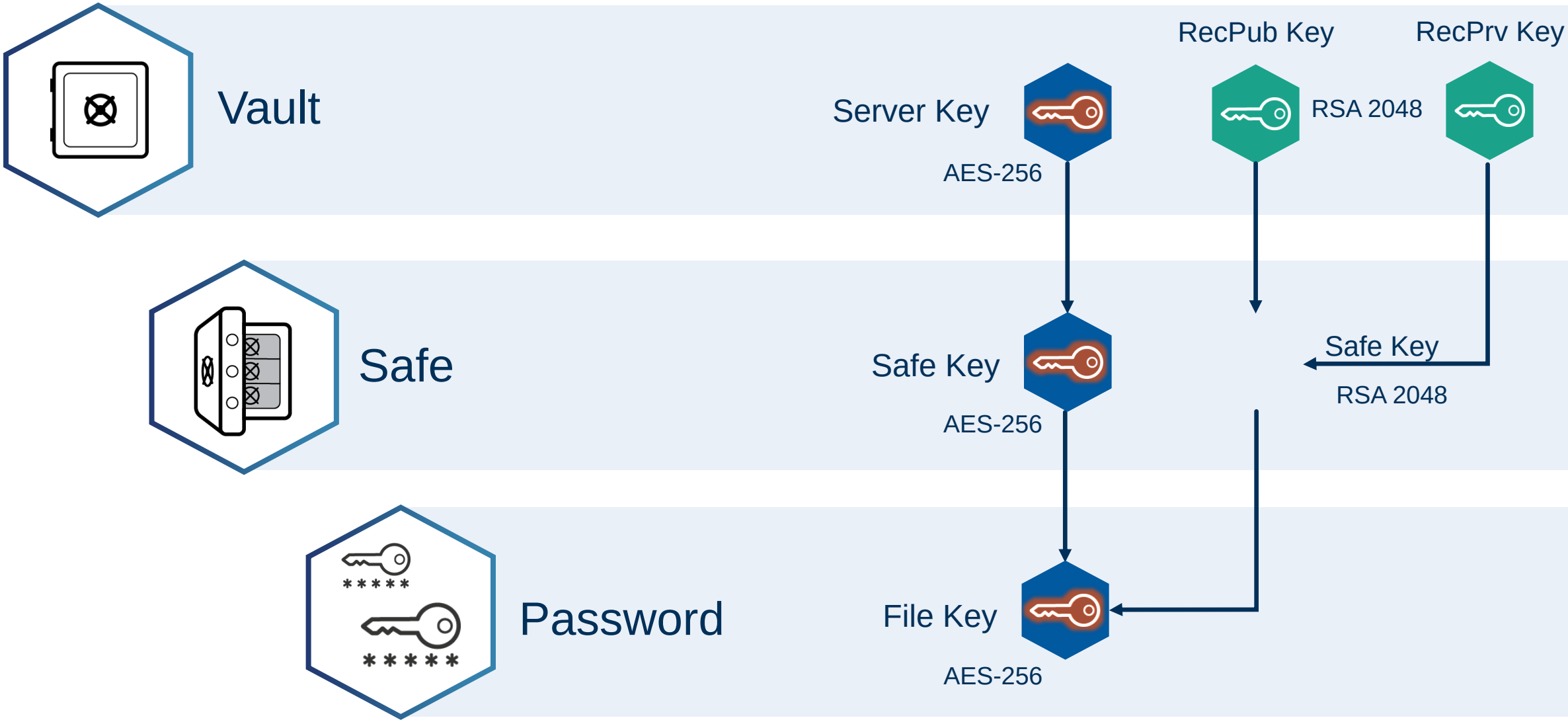


Let's have a look at how these keys are used to protect the keys to your kingdom.

Vault Object Encryption – Day-to-Day Operations

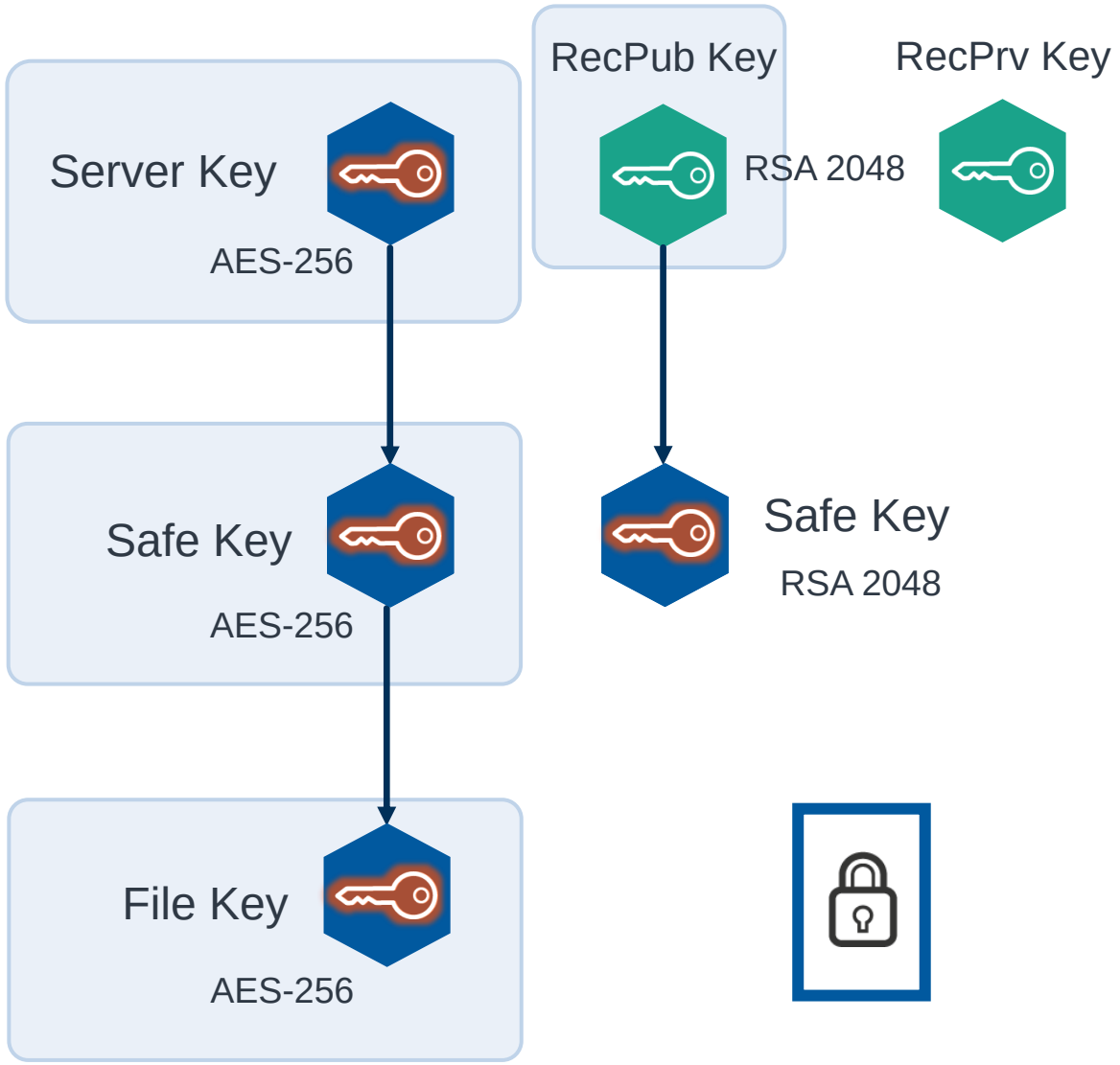


Vault Object Encryption – Emergency Measures



File Encryption Process

- Each Credential is stored as an encrypted file on the Vault
  - The **File key** is a unique symmetric key generated for each file
  - The File Key is then encrypted with the **Safe key**, which is a symmetric key unique to the Safe
  - The Safe key is then encrypted with the symmetric **Server key**, which is unique to the Vault
- **Server Key**
  - The Server Key is loaded into memory when the Vault starts
- **RecPub Key**
  - A copy of the relevant Safe Key is encrypted with the RecPub Key and stored with the Safe



How Encryption Keys are Distributed

Previously, the encryption keys required to install and operate the CyberArk PAM solution were physically delivered in the form of CDs containing the files.

As of March 2022, CyberArk now delivers these encryption key files via a secure email service.

You can go to the link below for more information on key delivery.

https://cyberark-customers.force.com/s/article/Digitized-Encryption-Keys-Delivery-End-User-Guide

Recovery Private Key Storage Strategies

The **Recovery Private Key*** must be copied to physical media and stored in at least two separate and secure locations:

One on the **Primary** site
and one on the **Disaster Recovery** site.

**** AKA the "Master Key"***

Server Key Storage Strategies

STRONG

Copy the key to external medium (USB drive, CD-ROM) and store it in a physical safe.

Insert the medium whenever starting/restarting the Vault.

Key in RAM

CONVENIENT

Copy the key to direct attached storage of the Vault server(s) and secure with NTFS permissions or by encrypting the key with a 3rd-party tool.

Always available.

Key in RAM

STRONG & CONVENIENT

Store the Server key in a Hardware Security Module (HSM).

Always available.

Key NOT in RAM

Component Local Environment

In this section we will look at the main services, configuration files, and logs for each of the following components:

- **Vault**
- **CPM**
- **PVWA**
- **PSM**

Inside the Vault

Vault Services

Services before Vault installation

Name	Description	Status
Base Filtering Engine	The Base Filtering Engine (BFE) is a service that m...	Started
Certificate Propagation	Copies user certificates and root certificates from ...	Started
COM+ Event System	Supports System Event Notification Service (SENS)...	Started
COM+ System Application	Manages the configuration and tracking of Compo...	Started
Cryptographic Services	Provides four management services: Catalog Data...	Started
DCOM Server Process Launcher	The DCOMLAUNCH service launches COM and DC...	Started
Desktop Window Manager Session...	Provides Desktop Window Manager startup and m...	Started
DHCP Client	Registers and updates IP addresses and DNS reco...	Started
Diagnostic Policy Service	The Diagnostic Policy Service enables problem dete...	Started
Diagnostic System Host	The Diagnostic System Host is used by the Diagnos...	Started
Distributed Link Tracking Client	Maintains links between NTFS files within a comput...	Started
Distributed Transaction Coordinator	Coordinates transactions that span multiple resour...	Started
DNS Client	The DNS Client service (dnscache) caches Domain ...	Started
Group Policy Client	The service is responsible for applying settings con...	Started
IP Helper	Provides tunnel connectivity using IPv6 transition t...	Started
Network Connections	Manages objects in the Network and Dial-Up Conn...	Started
Network List Service	Identifies the networks to which the computer has ...	Started
Network Location Awareness	Collects and stores configuration information for th...	Started
Network Store Interface Service	This service delivers network notifications (e.g. int...	Started
Plug and Play	Enables a computer to recognize and adapt to har...	Started
Power	Manages power policy and power policy notificatio...	Started
Print Spooler	Loads files to memory for later printing	Started
Remote Desktop Configuration	Remote Desktop Configuration service (RDCS) is r...	Started
Remote Desktop Services	Allows users to connect interactively to a remote c...	Started
Remote Desktop Services UserMo...	Allows the redirection of Printers/Drives/Ports for ...	Started
Remote Procedure Call (RPC)	The RPCSS service is the Service Control Manager ...	Started
Remote Registry	Enables remote users to modify registry settings o...	Started
RPC Endpoint Mapper	Resolves RPC interfaces identifiers to transport en...	Started
Security Accounts Manager	The startup of this service signals other services t...	Started

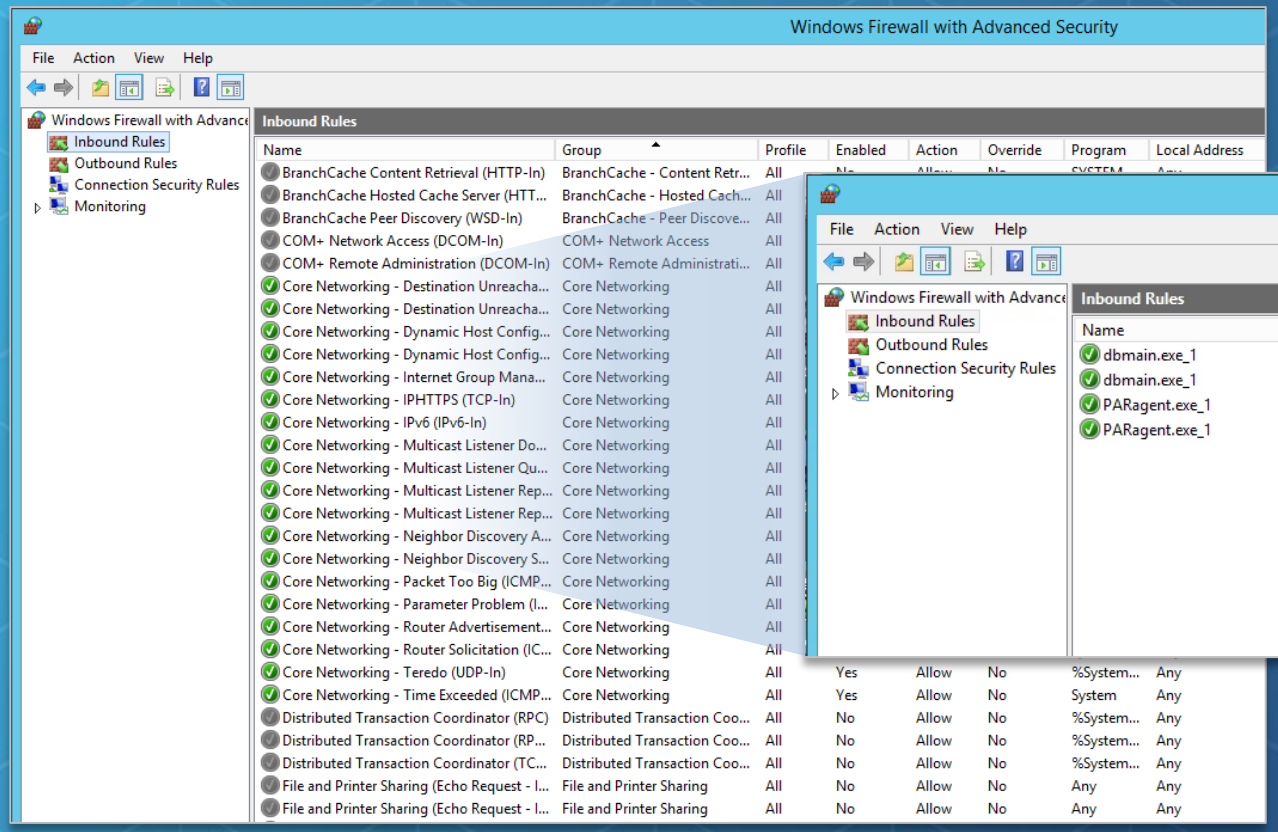
Services Post Installation and Hardening

Name	Description	Status
Cyber-Ark Event Notification Engine		Started
Cyber-Ark Hardened Windows Firewall	Windows Firewall helps prot...	Started
CyberArk Logic Container		Started
DCOM Server Process Launcher	The DCOMLAUNCH service l...	Started
DHCP Client	Registers and updates IP a...	Started
DNS Client	The DNS Client service (dns...	Started
Group Policy Client	The service is responsible f...	Started
Net.Pipe Listener Adapter	Receives activation request...	Started
Net.Tcp Listener Adapter	Receives activation request...	Started
Net.Tcp Port Sharing Service	Provides ability to share TC...	Started
Network Connections	Manages objects in the Net...	Started
Network List Service	Identifies the networks to ...	Started
Network Location Awareness	Collects and stores configur...	Started
Network Store Interface Service	This service delivers networ...	Started
Plug and Play	Enables a computer to reco...	Started
Power	Manages power policy and ...	Started
PrivateArk Database		Started
PrivateArk Remote Control Agent		Started
PrivateArk Server		Started
Remote Desktop Services	Allows users to connect inte...	Started
Remote Procedure Call (RPC)	The RPCSS service is the Se...	Started
RPC Endpoint Mapper	Resolves RPC interfaces ide...	Started

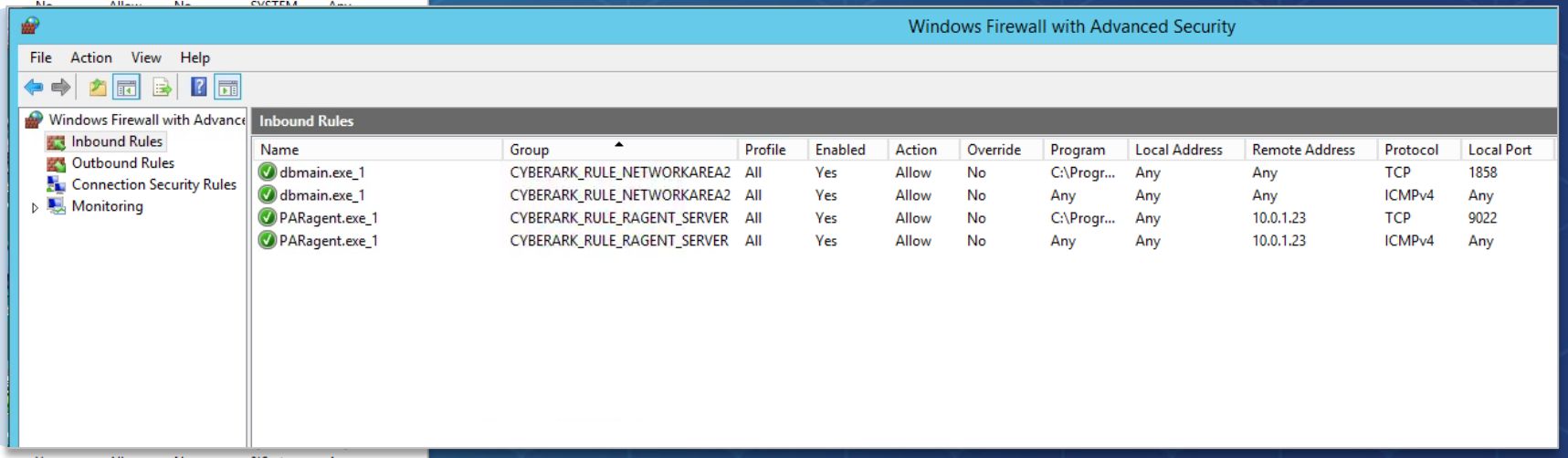
- Total number of previously running services has been reduced as part of the hardening process
- Vault installation has added 6 new services

Vault Firewall

Firewall before Vault installation



Firewall Post Hardening



All Firewall Rules that do not relate to CyberArk have been deleted, both inbound and outbound.

Vault Main Configuration Files

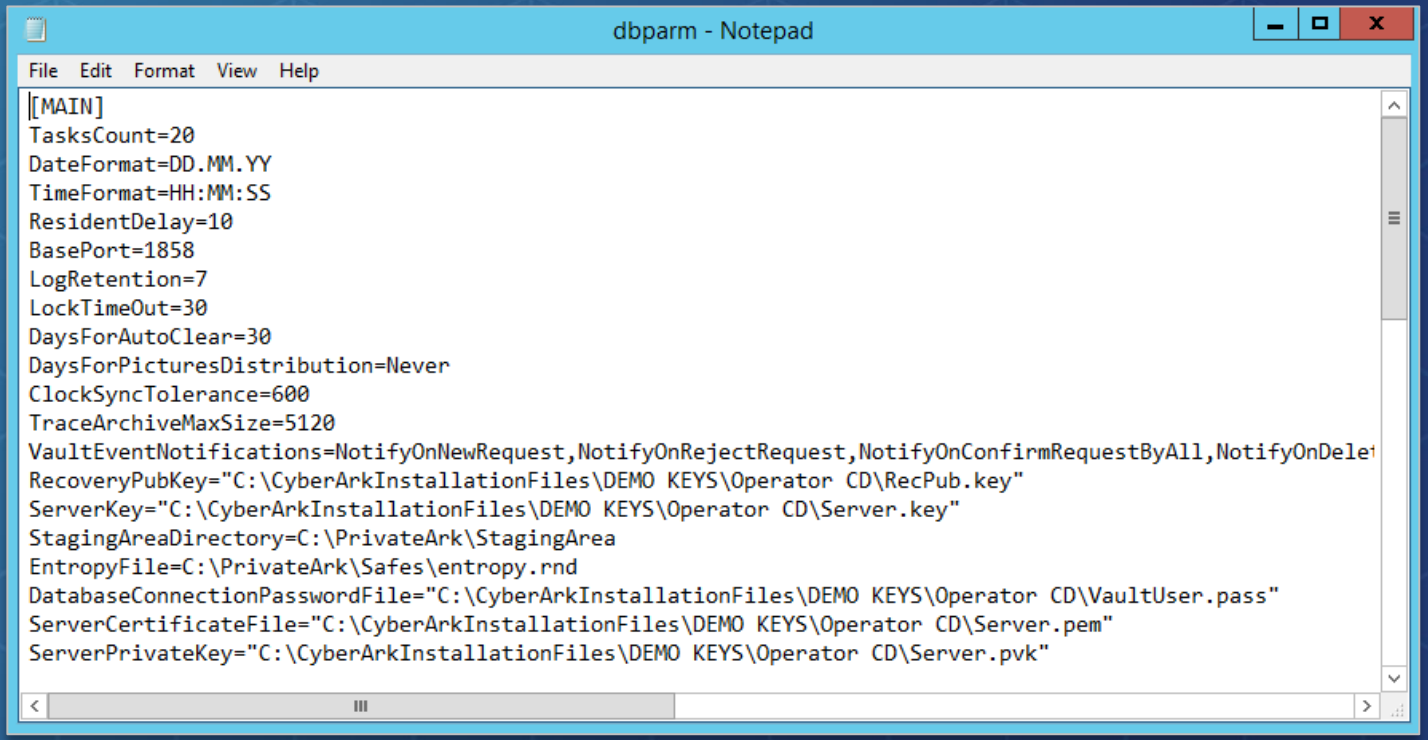
dbparm.ini	▶ Main configuration file of the Vault ▶ Any change requires a restart of the Vault service
passparm.ini	▶ Configure password policy for users of the Vault
PARagent.ini	▶ Configure Remote Control Agent in the Vault ▶ SNMP Configuration
tsparm.ini	▶ Configure the physical disks used to store Vault data

dbparm.ini

dbparm.ini:
Current Vault configuration file, contains parameters for Log Level, Server Key, Syslog, Timeouts, Recovery Key, etc.

DBPARM.sample.ini:
Contains all the possible configuration options. Full info on these parameters is contained in the PAM documentation.

dbparm.ini.good:
Contains the last known working configuration of the dbparm.ini file. Created automatically when the Vault server starts up.




```
[MAIN]
TasksCount=20
DateFormat=DD.MM.YY
TimeFormat=HH:MM:SS
ResidentDelay=10
BasePort=1858
LogRetention=7
LockTimeOut=30
DaysForAutoClear=30
DaysForPicturesDistribution=Never
ClockSyncTolerance=600
TraceArchiveMaxSize=5120
VaultEventNotifications=NotifyOnNewRequest,NotifyOnRejectRequest,NotifyOnConfirmRequestByAll,NotifyOnDelet
RecoveryPubKey="C:\CyberArkInstallationFiles\DEMO KEYS\Operator CD\RecPub.key"
ServerKey="C:\CyberArkInstallationFiles\DEMO KEYS\Operator CD\Server.key"
StagingAreaDirectory=C:\PrivateArk\StagingArea
EntropyFile=C:\PrivateArk\Safes\entropy.rnd
DatabaseConnectionPasswordFile="C:\CyberArkInstallationFiles\DEMO KEYS\Operator CD\VaultUser.pass"
ServerCertificateFile="C:\CyberArkInstallationFiles\DEMO KEYS\Operator CD\Server.pem"
ServerPrivateKey="C:\CyberArkInstallationFiles\DEMO KEYS\Operator CD\Server.pvk"
```

Vault Log Files

Italog.log	▶ Main log file of the Vault server.
Trace.d0	▶ Trace file of the Vault. ▶ It is detailed according to the debug level configured in the dbparm.ini.

Inside the PVWA

PVWA Service (IIS Services)

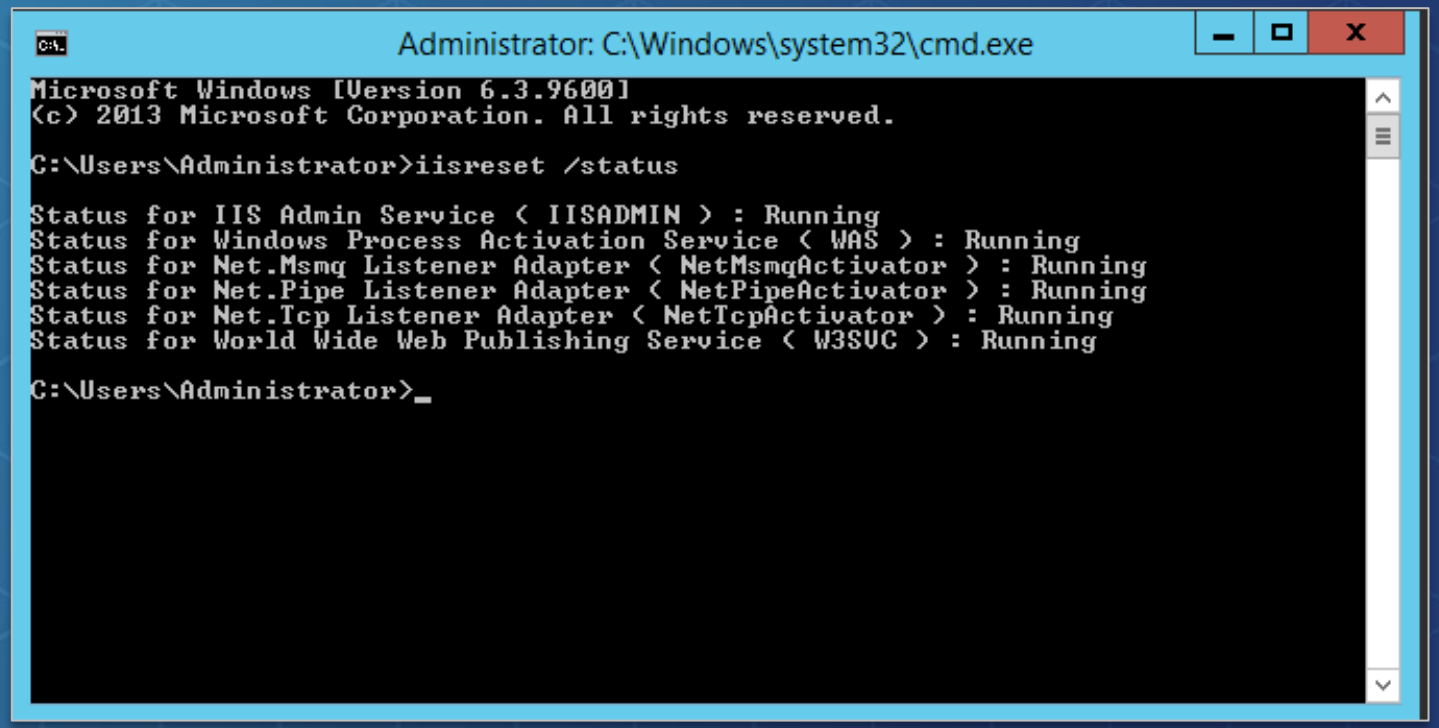
As the **PVWA** is a web application running on **IIS**, you can control it through the IIS Manager interface or use the command line by running:

iisreset /restart

or

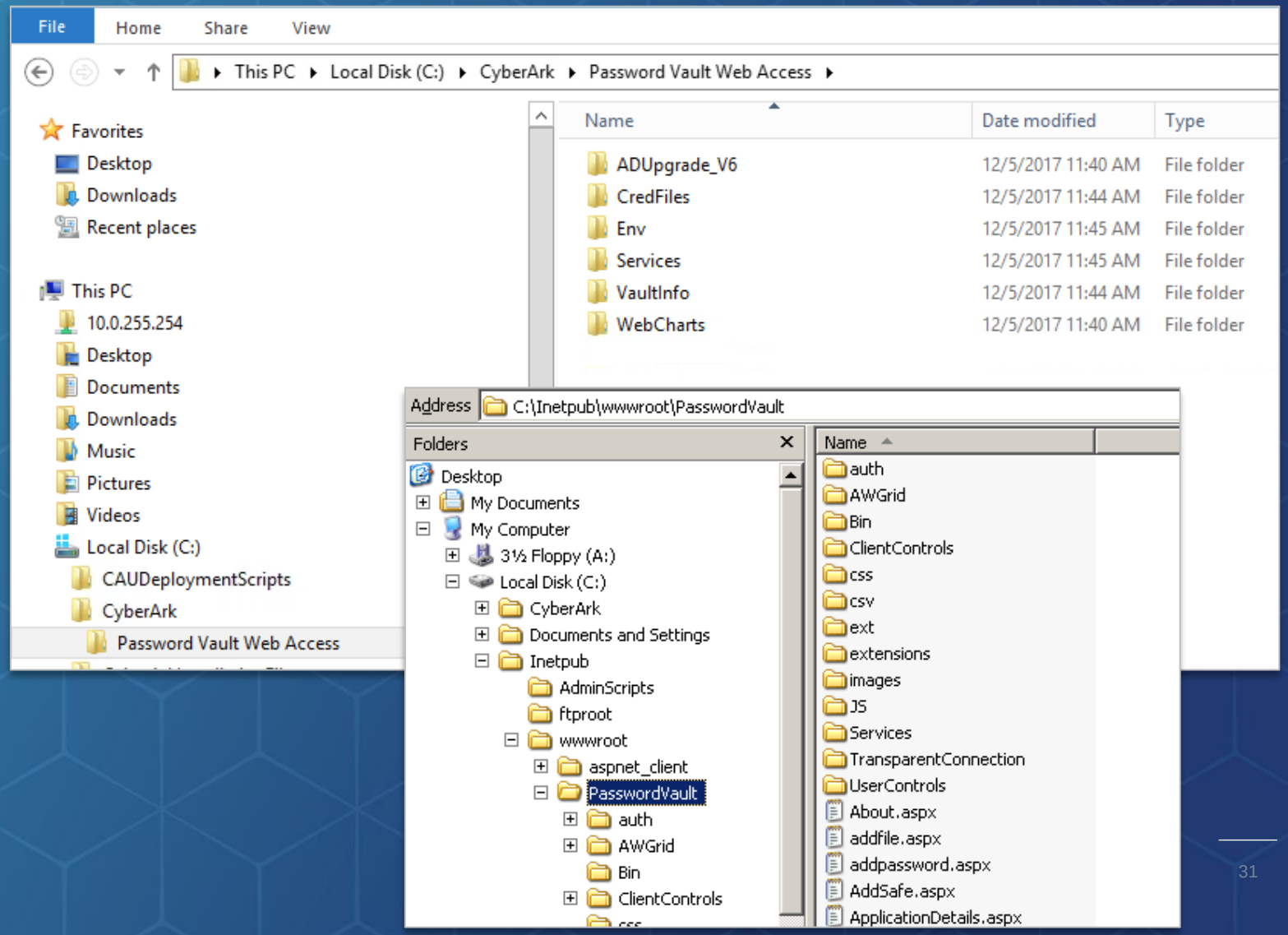
iisreset /status

to check status of website



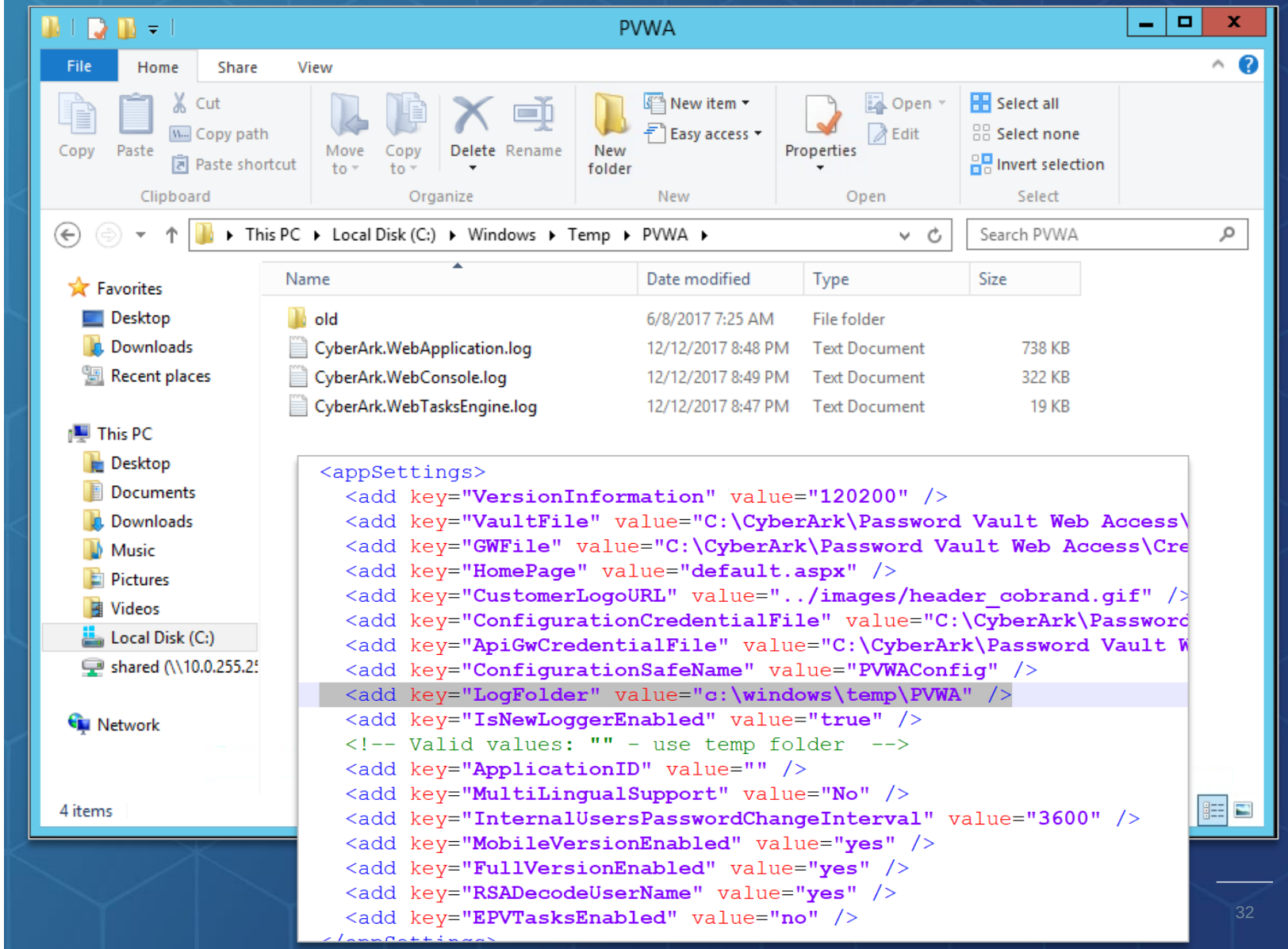
PVWA Directories (IIS Folder)

- **PVWA** application files are located at: ***C:\Cyberark\Password Vault Web Access***
- Web page: IIS Virtual Folder - ***PasswordVault***



PVWA Log Location

- Default log file location: ***%windir%\temp\PVWA\.***
- Can be changed by going to the ***PasswordVault*** folder under IIS, opening the file ***web.config***, and modifying the "LogFolder" parameter

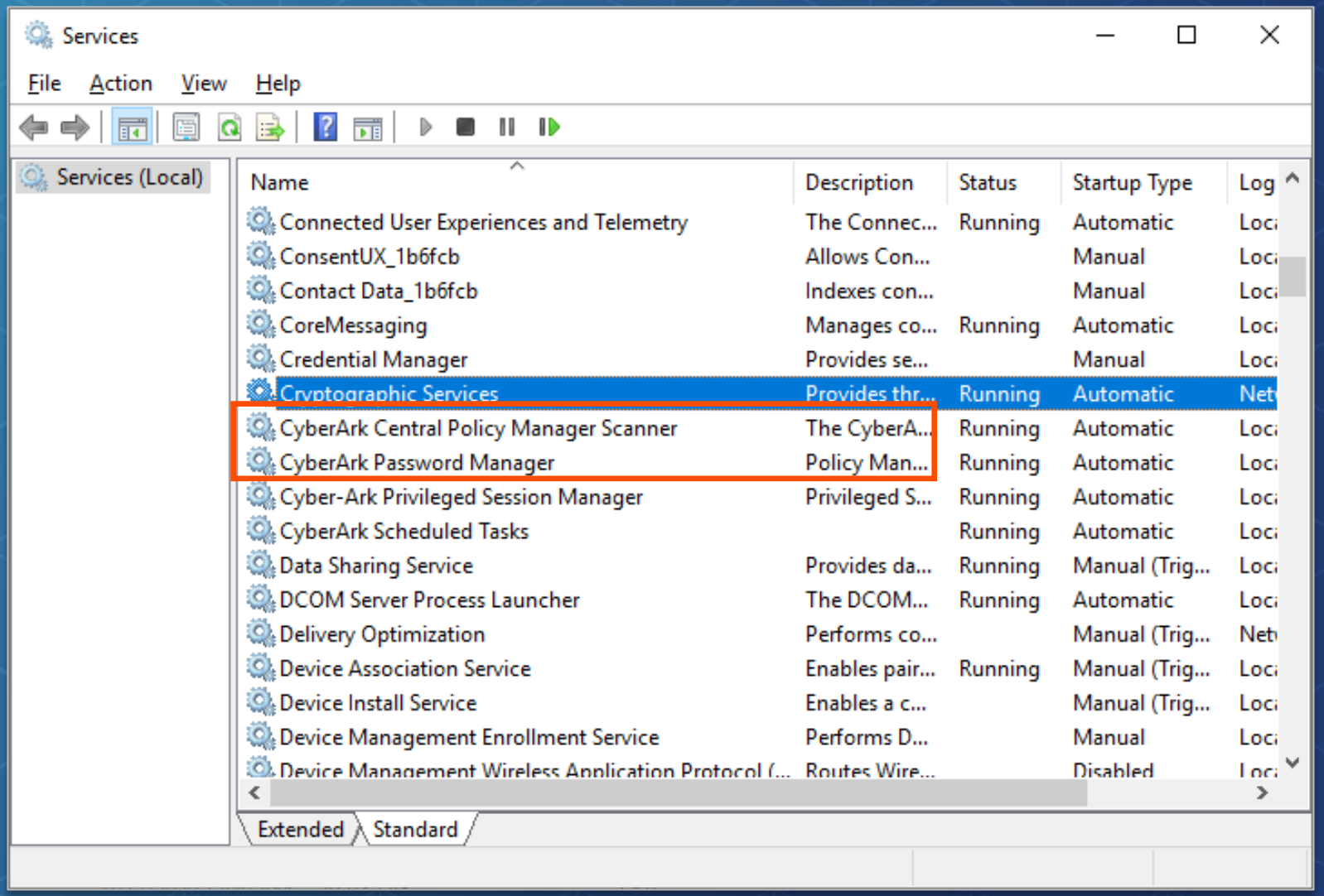


Inside the CPM

CPM Services

The **CPM** server has two main services:

- The ***CyberArk Password Manager*** service is a batch processor that connects to the **Vault** looking for work to do and kicks off the necessary processes to complete that work.
- The ***CyberArk Central Policy Manager Scanner*** is the scanner for the Accounts Feed workflow.



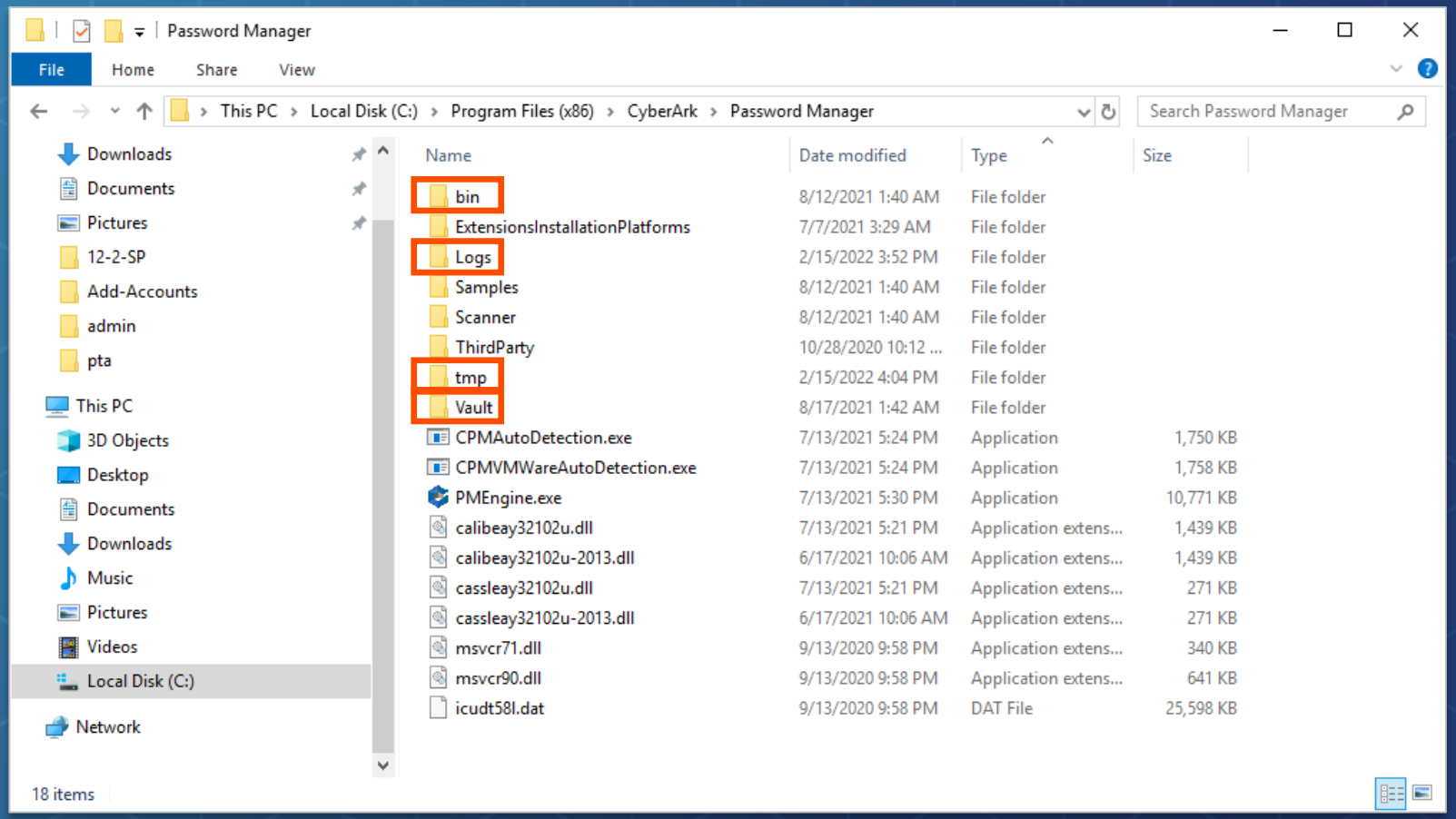
CPM Directories

bin –
Contains all the files required to run the **CPM** and the change password processes on target machines

Logs –
Contains **CPM** activity log files

tmp –
Contains files that are used by the **CPM** for internal processing

Vault –
Contains the configuration that tells the **CPM** where to find the vault and how to connect



Log Files

Activity Logs
(Logs folder)

- ***pm.log*** – contains all the log messages, including general and informative messages, errors, and warnings.
- ***pm_error.log*** – contains only warning and error messages.

Third-party Log Files
(Logs\ThirdParty folder)

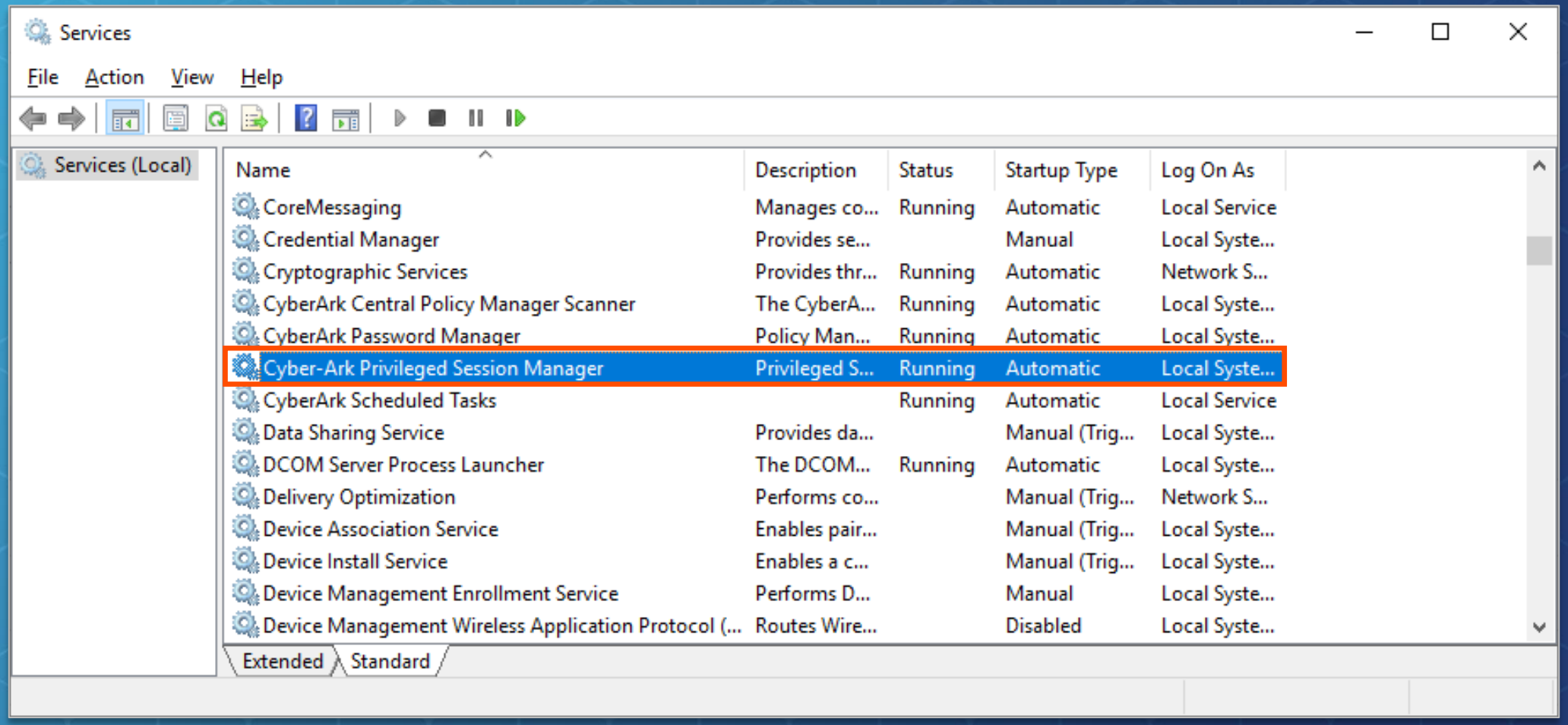
- Generated by the **CPM**'s password generation plug-ins when an error occurs
- Name of the log file:
  <type of password>-<Safe>-<folder>-<name of password object>.log
  E.g., ***Operating System-UnixSSH-1.1.1.250-Root.log***

History Log Files
(Logs\History folder)

- After a log file has been uploaded into the **Safe**, it is renamed and moved into the ***History*** subfolder.
- The file is marked with a time stamp and renamed as follows:
  <filename> (<date>-<time>).log

Inside the PSM

The PSM Service



PSM Directories

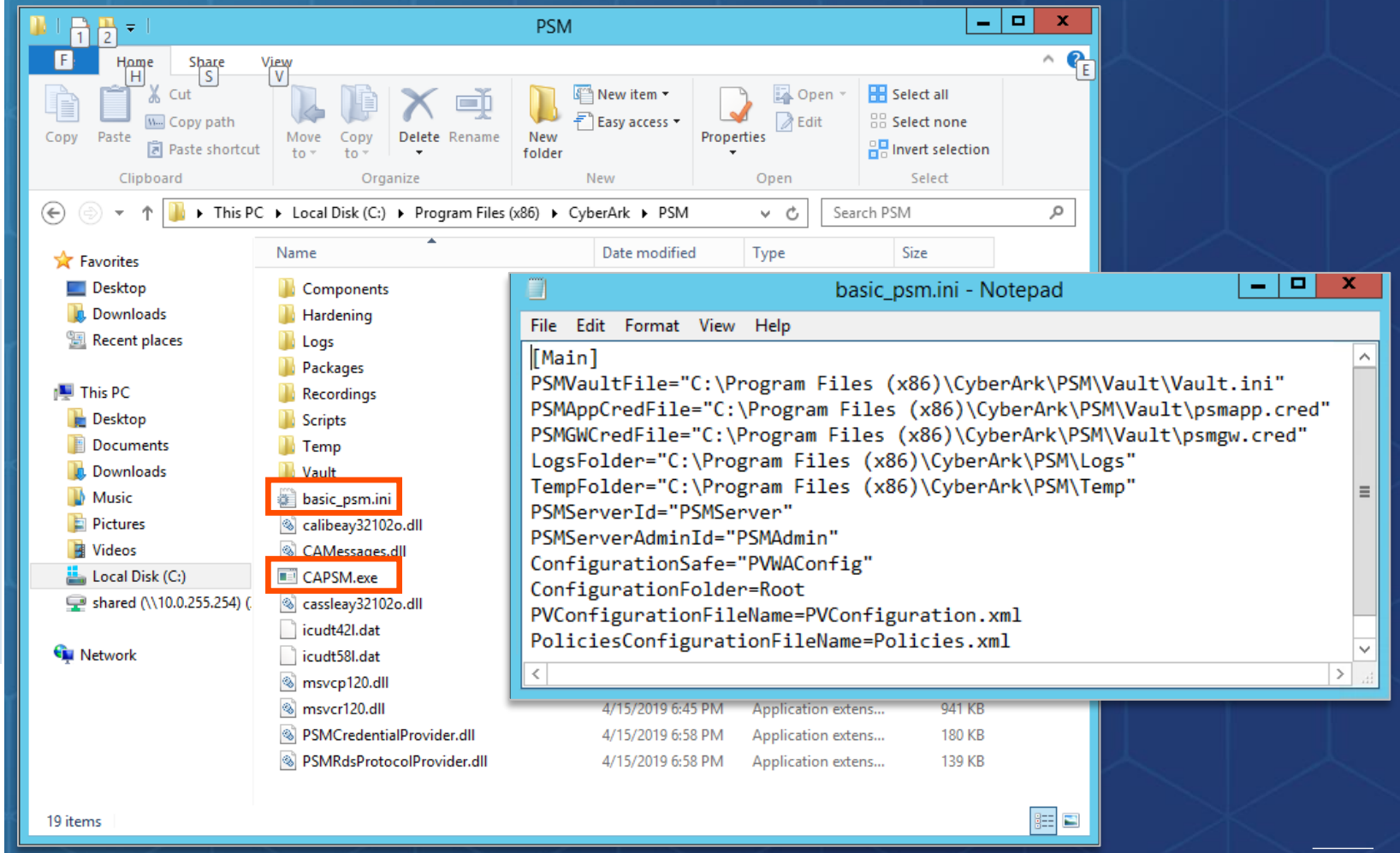
In the **PSM** directory you'll find all the configuration files, logs, and connectors that allow end users to connect to target systems.

Vault

Provides the PSM with the information required to log into the Vault

service user is given write permissions on this folder.

required to start the **PSM** (cred file locations, **Safe** names).



PSM Logs

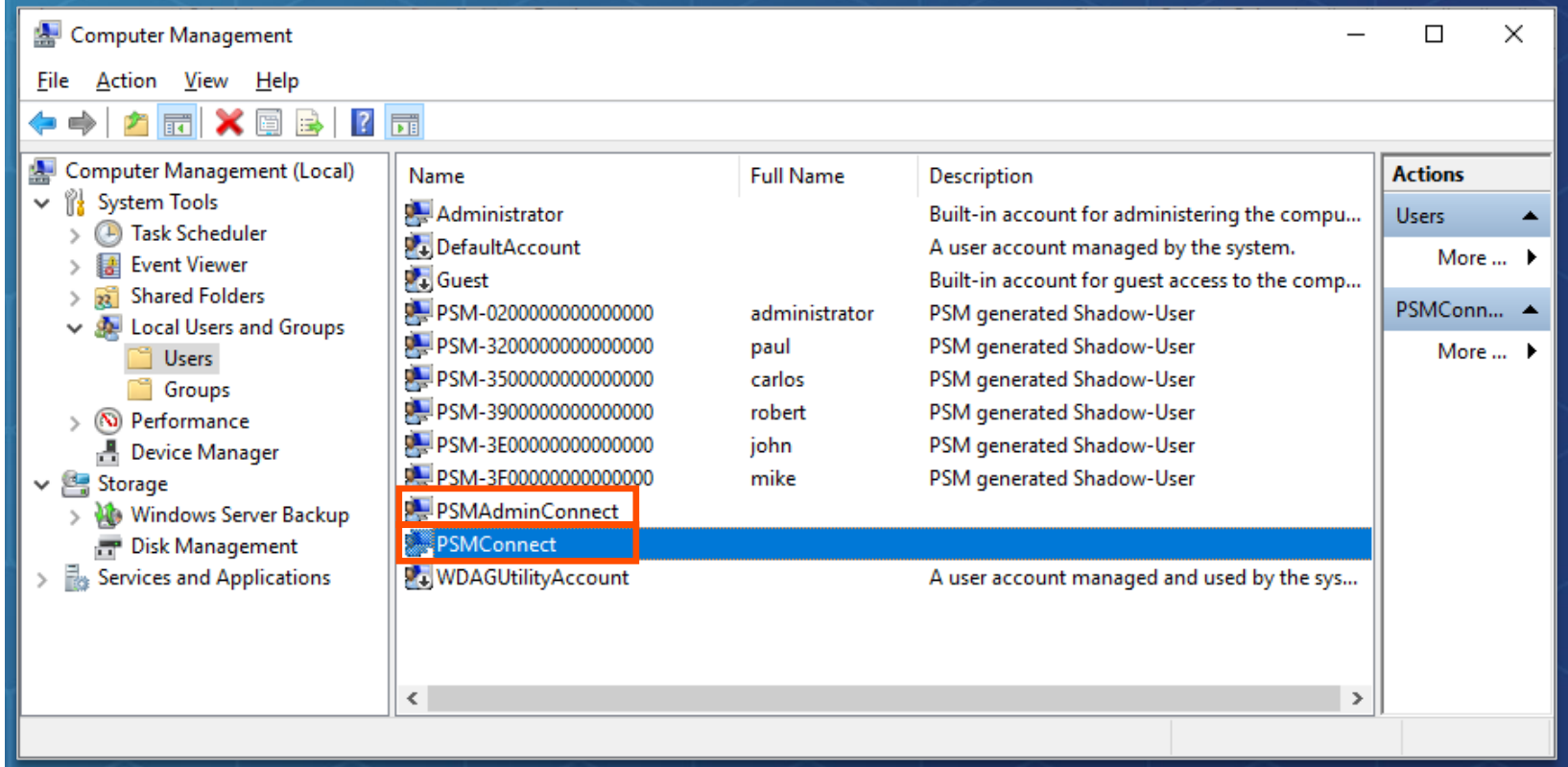
All activities that are carried out by the **PSM** are written to log files and stored in the ***Log*** subfolder of the **PSM** installation folder

PSMConsole.log	▶ Contains informational messages and errors that refer to **PSM** function.
<SessionID>.Recorder.log	▶ Contains errors and trace messages related to the **PSM** Recorder that can be used for troubleshooting with session video recordings. The types of messages that are included depend on the debug levels specified in the **Recorder** settings of the **PSM** configuration.
<SessionID>.<connection client >.log	▶ Contains errors and trace messages related to the connection client that can be used for troubleshooting.

PSMConnect and PSMAdminConnect Users

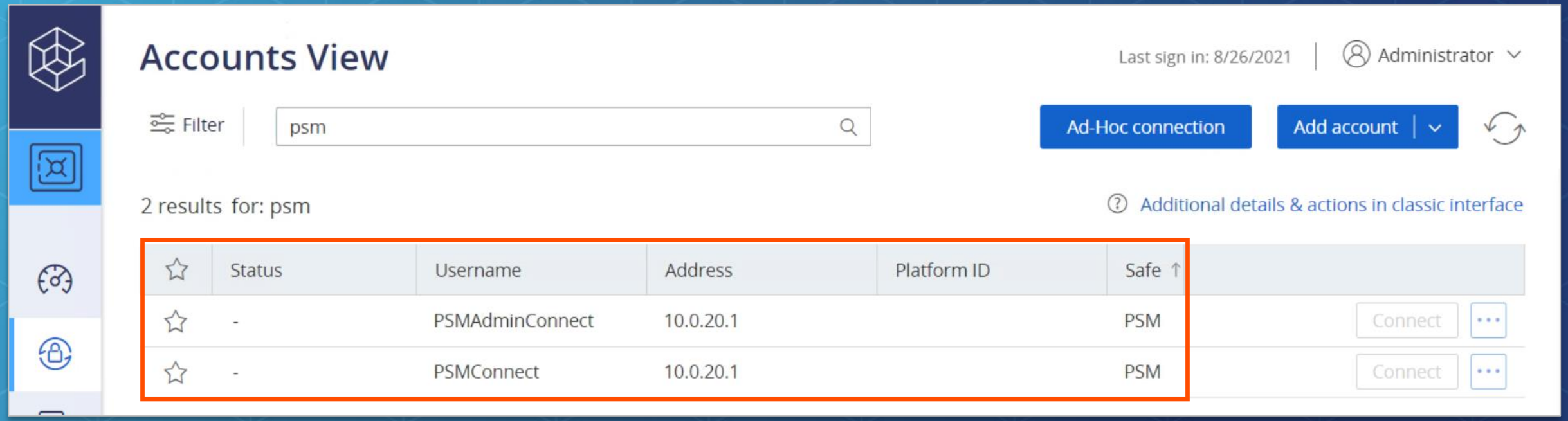
PSMConnect and ***PSMAdminConnec***t are local users on the **PSM** server.

- ***PSMConnect*** is used when an end user launches an RDP connection to a target system via **PSM**.
- ***PSMAdminConnect*** is used by Auditors when connecting via RDP to the **PSM** to monitor other users' RDP connections.



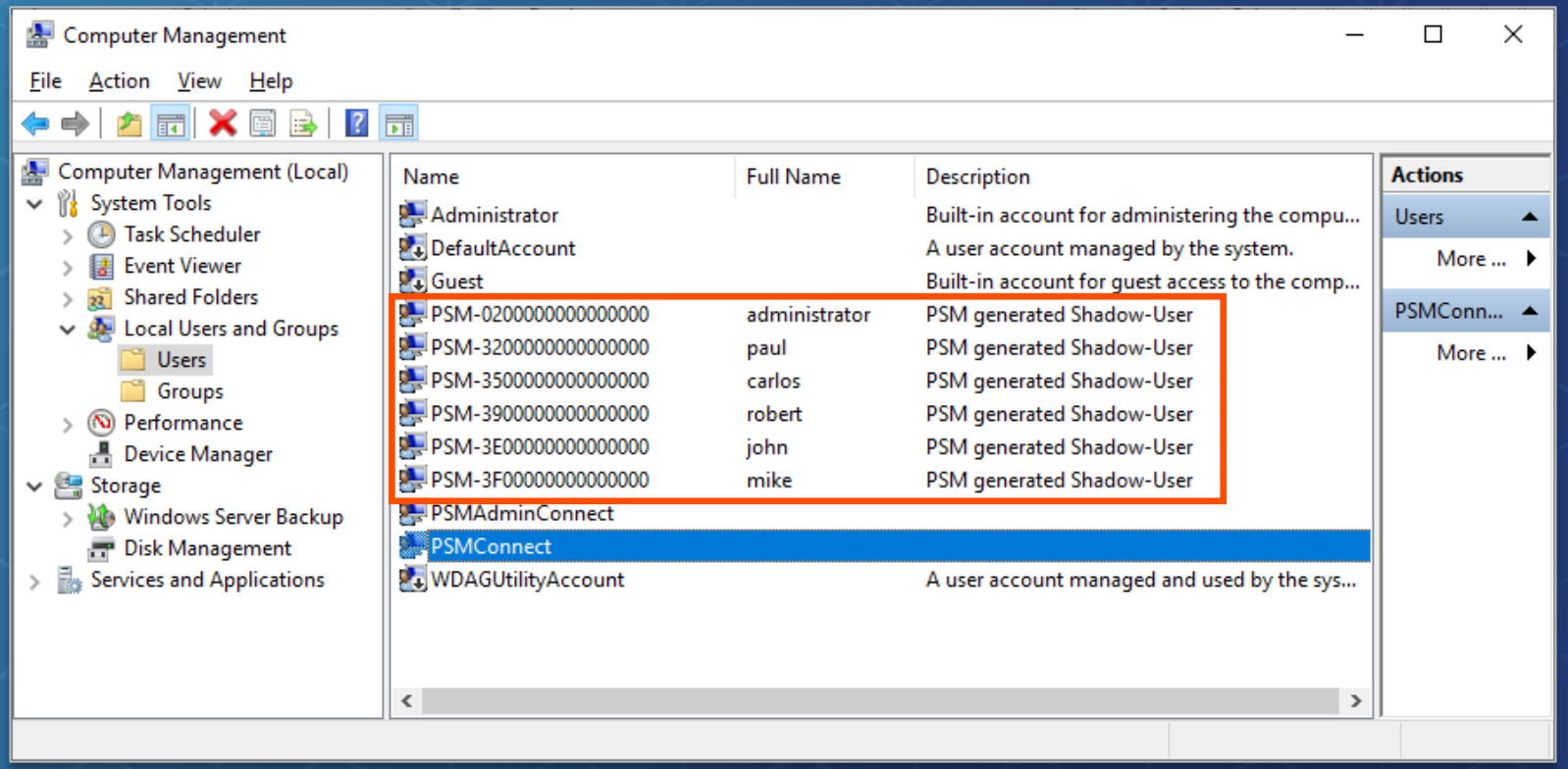
PSMConnect and PSMAdminConnect

The credentials for the ***PSMConnect*** and ***PSMAdminConnect*** users are stored as accounts in the **Vault** and should be managed in the same way any other account.

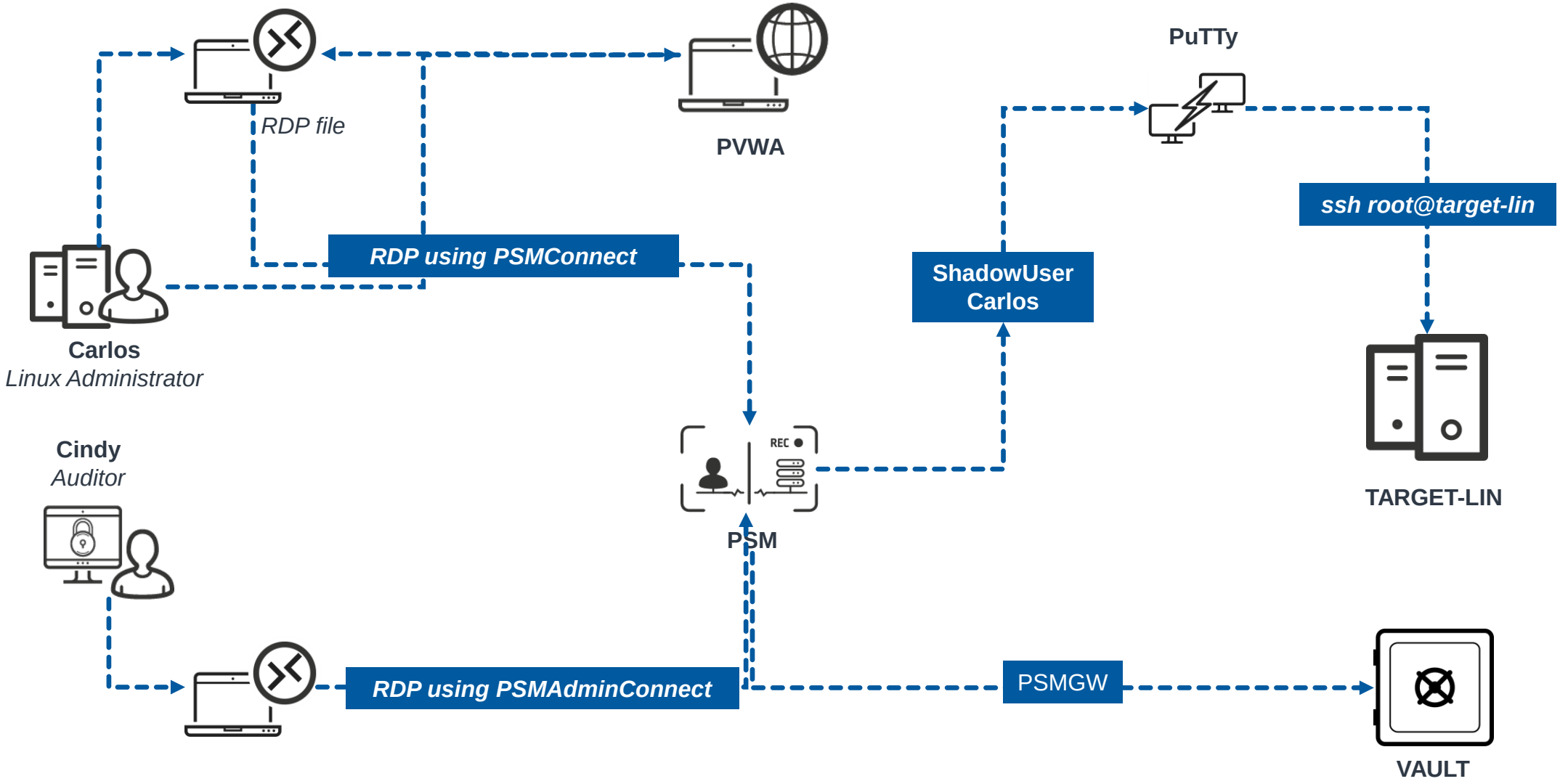


PSM
Shadow Users

- When a **Vault** user launches a session via the **PSM** for a **non-RDP** connection (e.g., **SSH**) for the first time, a shadow user is created for the user on the **PSM** server.
- This shadow user launches the application needed for the connection (**Putty** in the case of an SSH connection).
- The credentials for these users are reset with every connection.



PSM Users Summary



Internal Safes and Users

In this section we will look at the Internal safes and users created in the **Vault** for each component:

- **Vault**
- **CPM**
- **PVWA**
- **PSM**

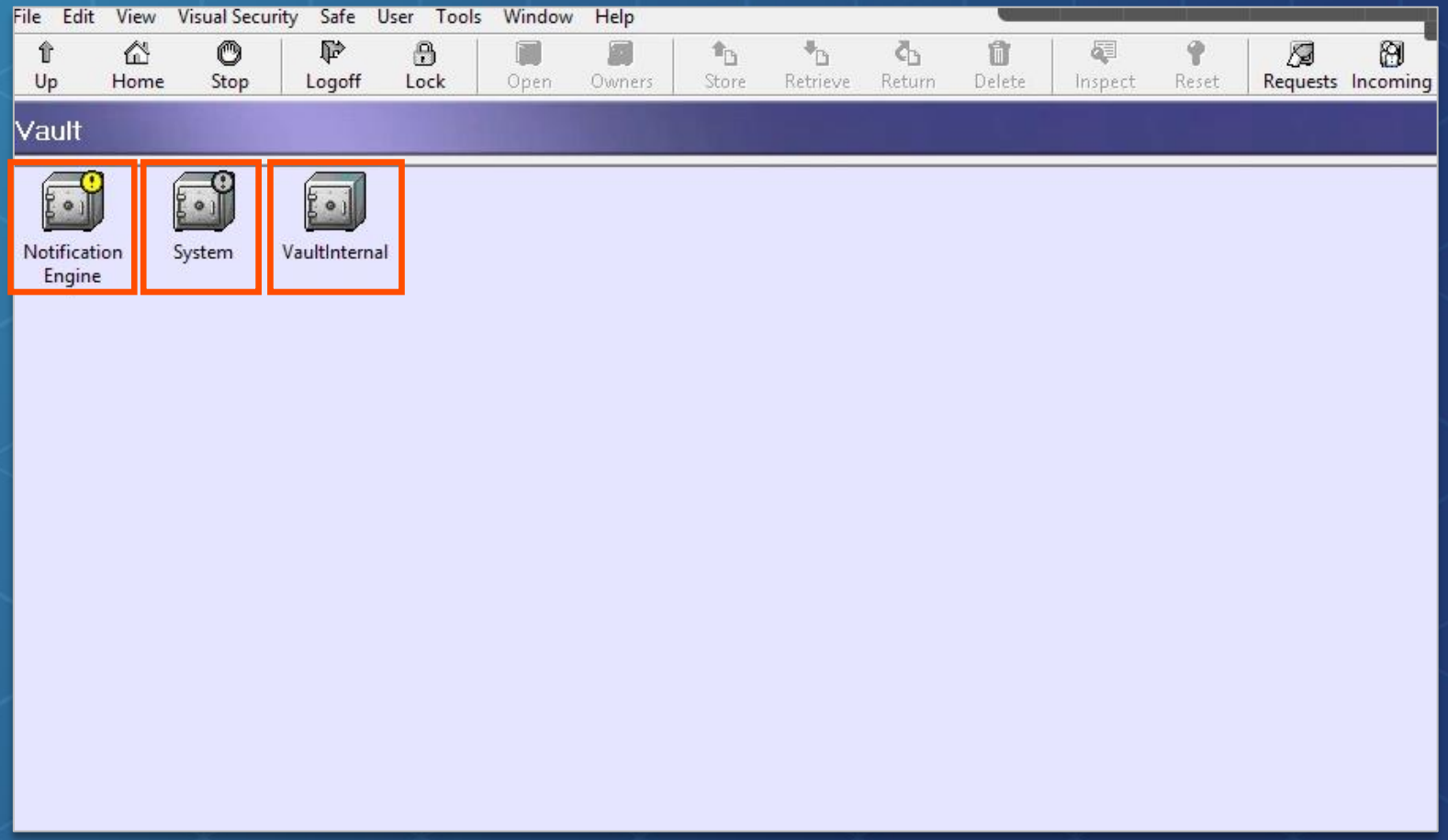
Vault Internal Safes

The three internal safes created during the **Vault** installation are:

- **Notification Engine:**
  used by the ENE service

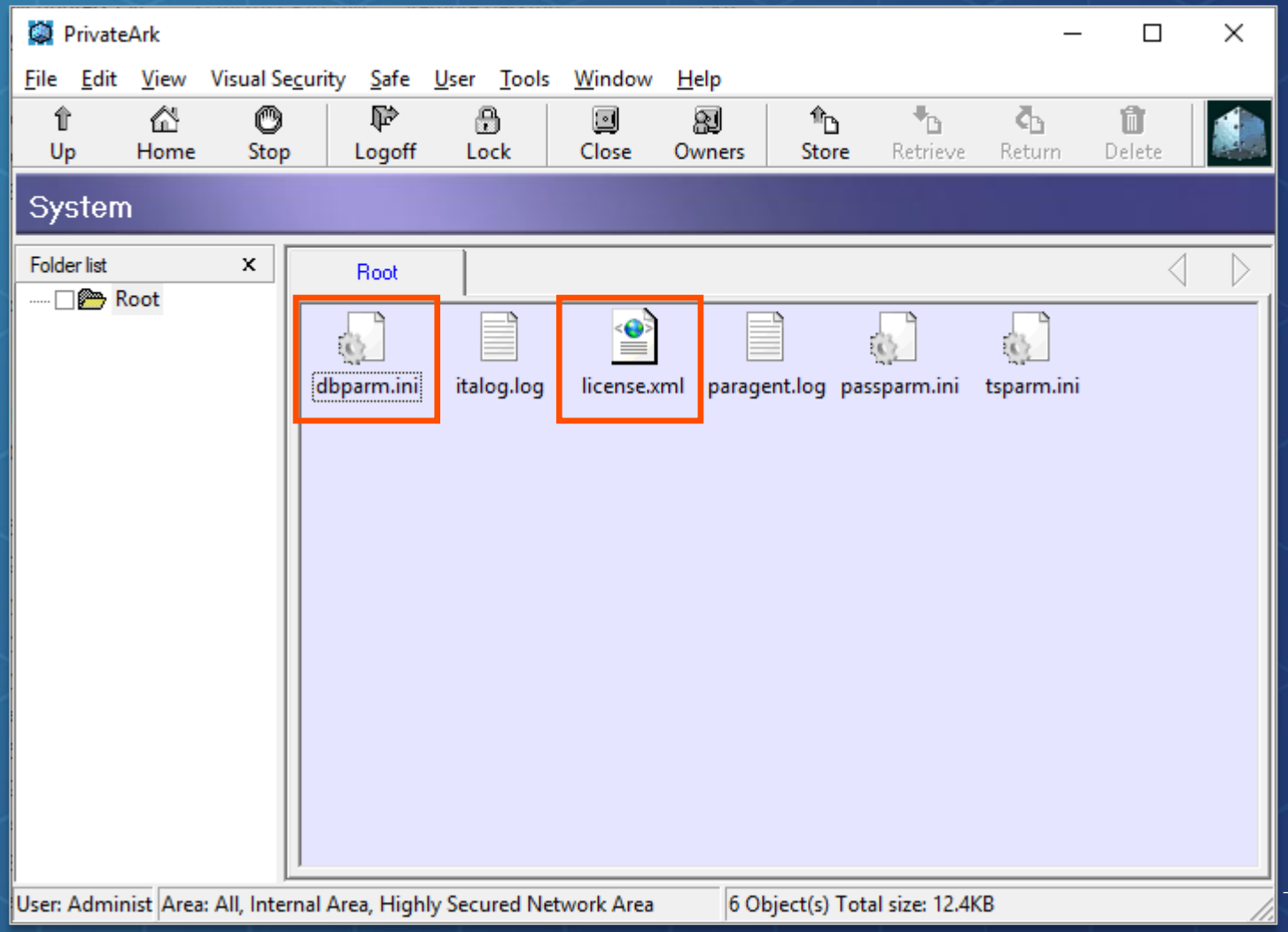
- **System:**
  contains the file links for ***dbparm.ini***, etc.

- **VaultInternal:**
  contains configuration data for **CyberArk** LDAP integration



The System Safe

- The **Vault**'s main configuration files and logs can also be accessed in the **System** Safe from remote stations using the **PrivateArk Client**
- A new ***License.xml*** file can be copied into this Safe to update the license without the need to restart the **Vault** service



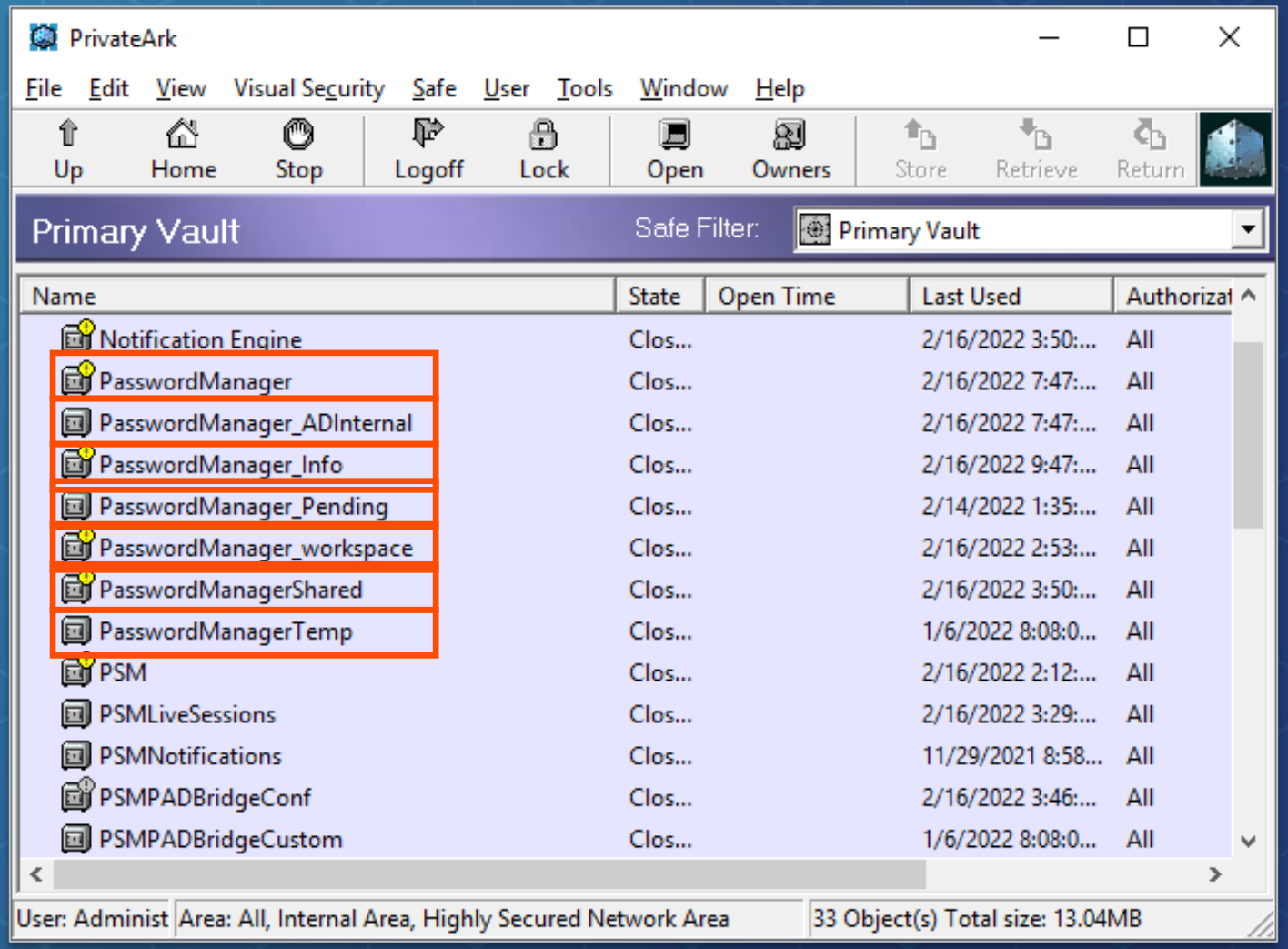
CPM Internal Safes

There are three safes shared by all **CPM** servers:

- ***PasswordManager_Pending***
- ***PasswordManagerShared***
- ***PasswordManagerTemp***

The remaining four safes will be duplicated for each CPM in the CyberArk environment and named after the user for that CPM, e.g.

- ***PasswordManager***
- ***PasswordManager_ADInternal***
- ***PasswordManager_info***
- ***PasswordManager_workspace***

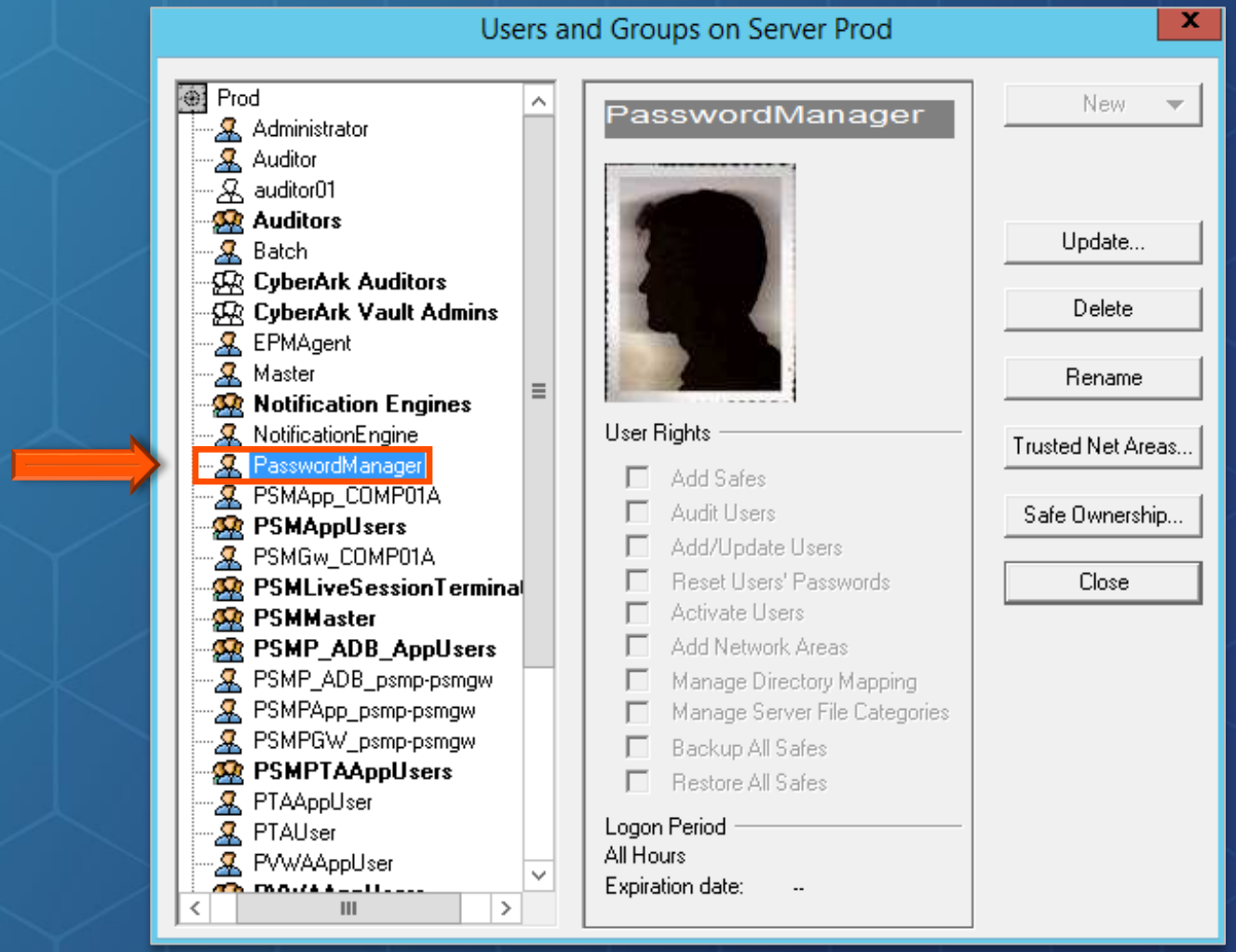


CPM Vault User

Tools > Administrative Tools > Users and Groups

By default, the first **CPM** user's name is ***PasswordManager***

- When creating a new **Safe** through the **PVWA**, the **CPM** user is automatically added to the **Safe**

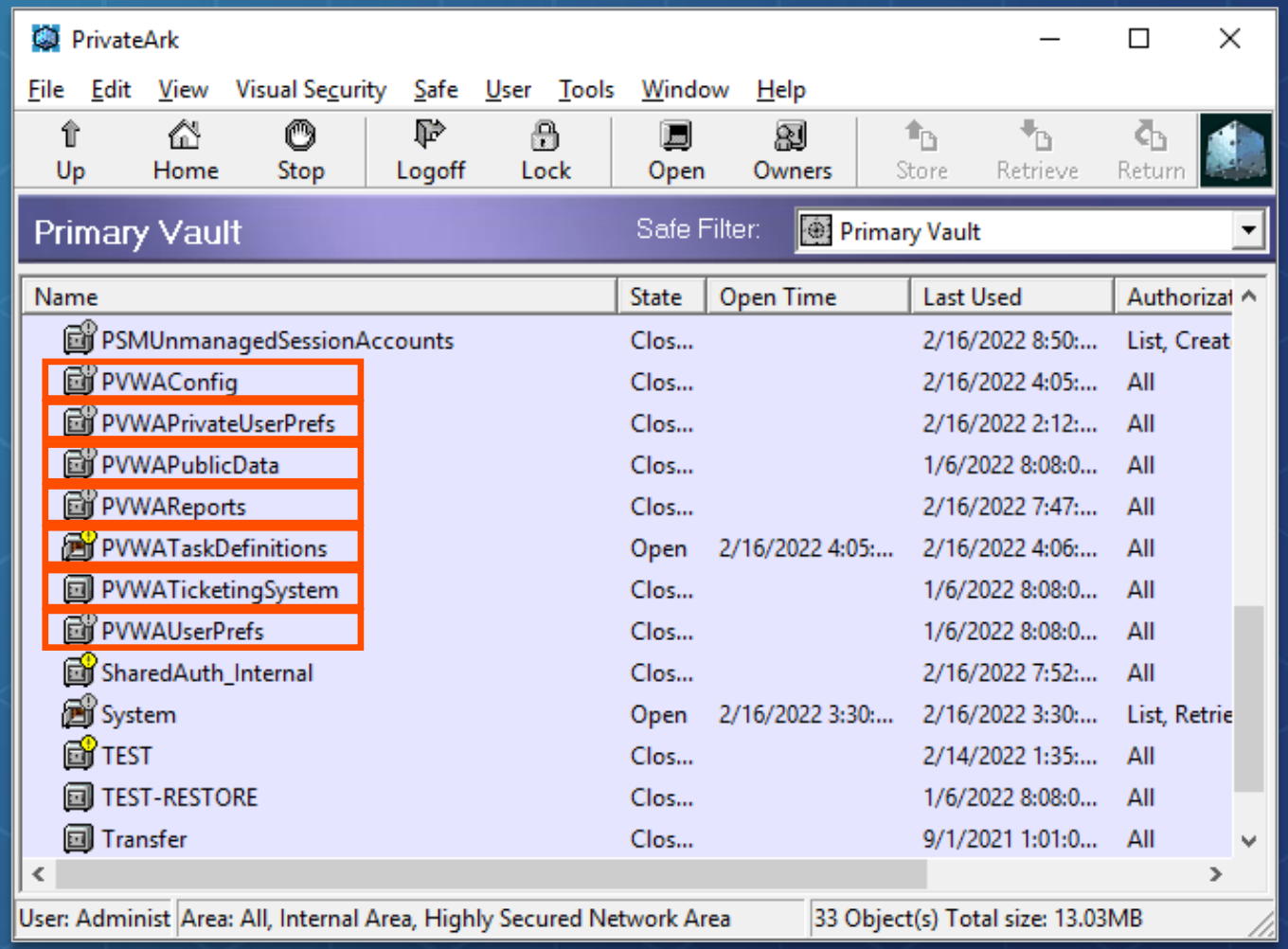


PVWA Safes

- **PVWAConfig** – configuration settings for **PVWA**
- **PVWAPrivateUserPrefs** – user preference settings

  ***Note**: The above two safes should not be accessed directly*

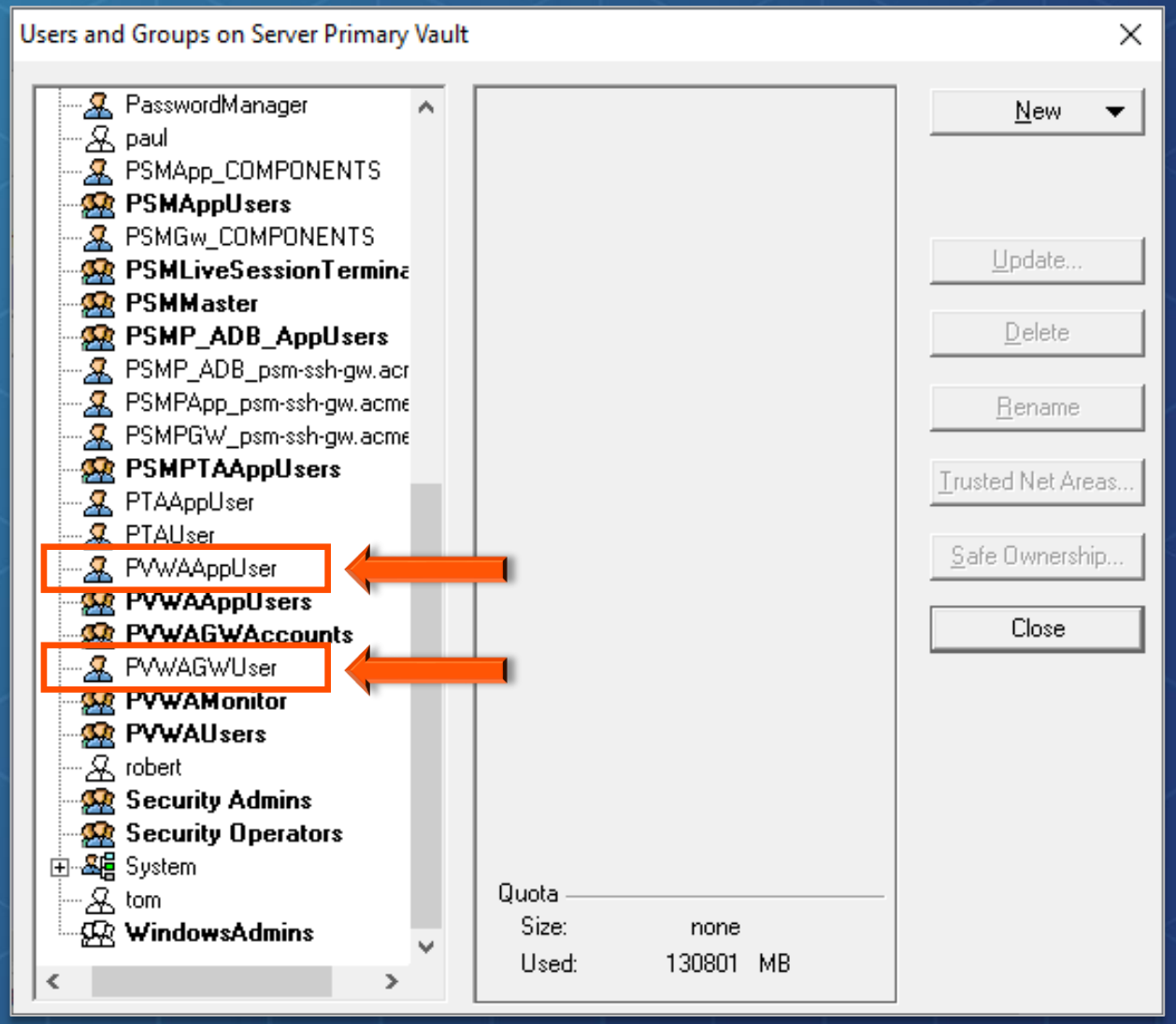
- **PVWAPublicDat**a – contains the help documents that can be accessed in the **PVWA**
- **PVWAReports** – completed reports
- **PVWATaskDefinitions** – report definitions
- **PVWATicketingSystem** – information on integrations with third-party ticketing systems
- **PVWAUserPrefs** – Changes to individual user preferences



PVWA Vault Users and Groups

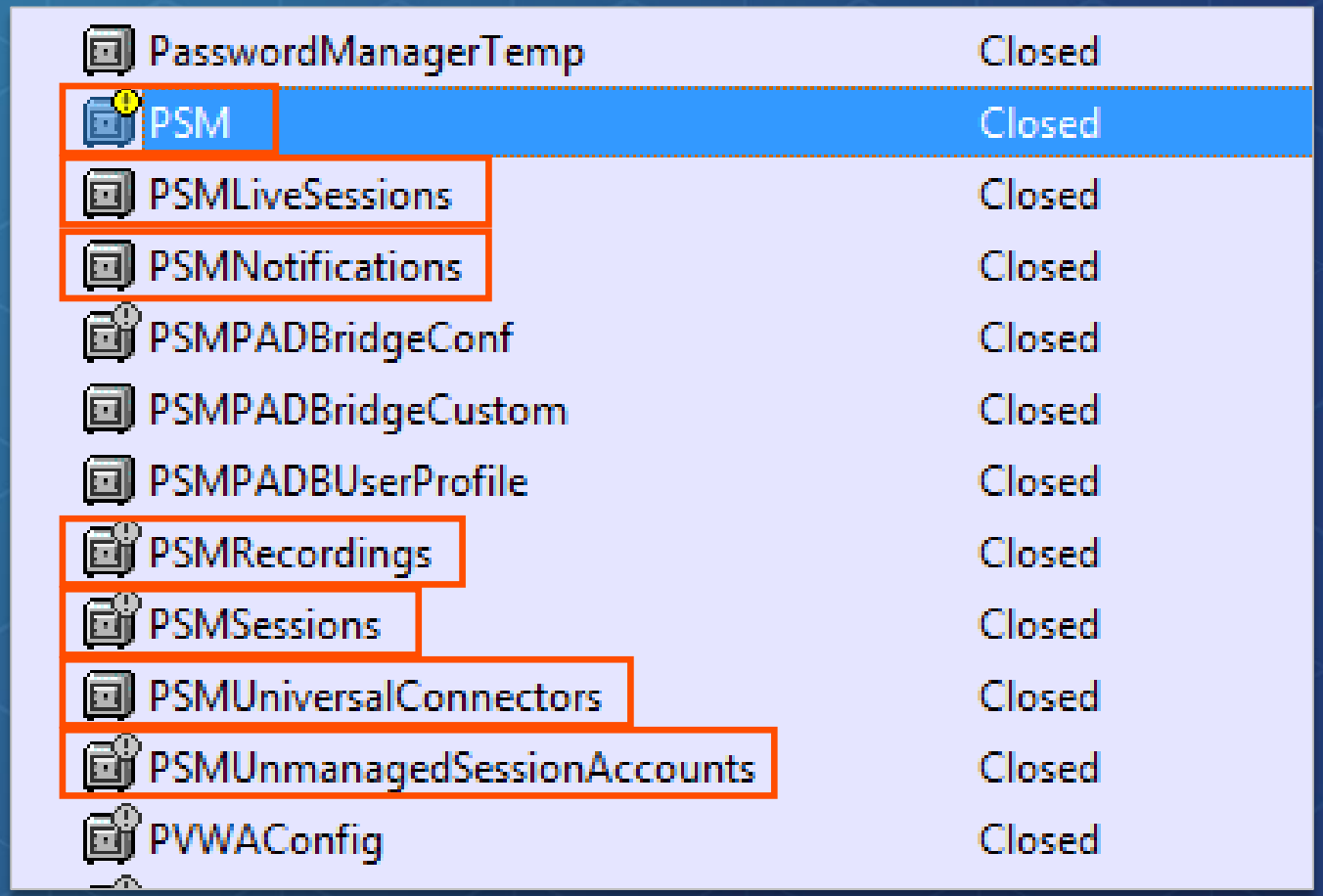
Tools->Administrative Tools->Users and Groups

- ***PVWAAppUser*** is used by the **Password Vault Web Access** for internal processing
- ***PVWAGWUser*** is the gateway user through which other users will access the **Vault**



PSM Safes

- **PSM** – contains the password objects for PSMConnect and PSMAdminConnect.
- **PSMLiveSessions** – allows users to monitor live sessions
- **PSMNotifications** – allows users to terminate, suspend, or resume sessions.
- **PSMRecordings** – default safe for storing recordings.
- **PSM Sessions** – allows users to launch sessions via PSM
- **PSMUniversalConnectors** – used in auto deployment for PSM connectors to multiple PSMs.
- **PSMUnmanagedSession**s – allows users to monitor live Ad-hoc sessions



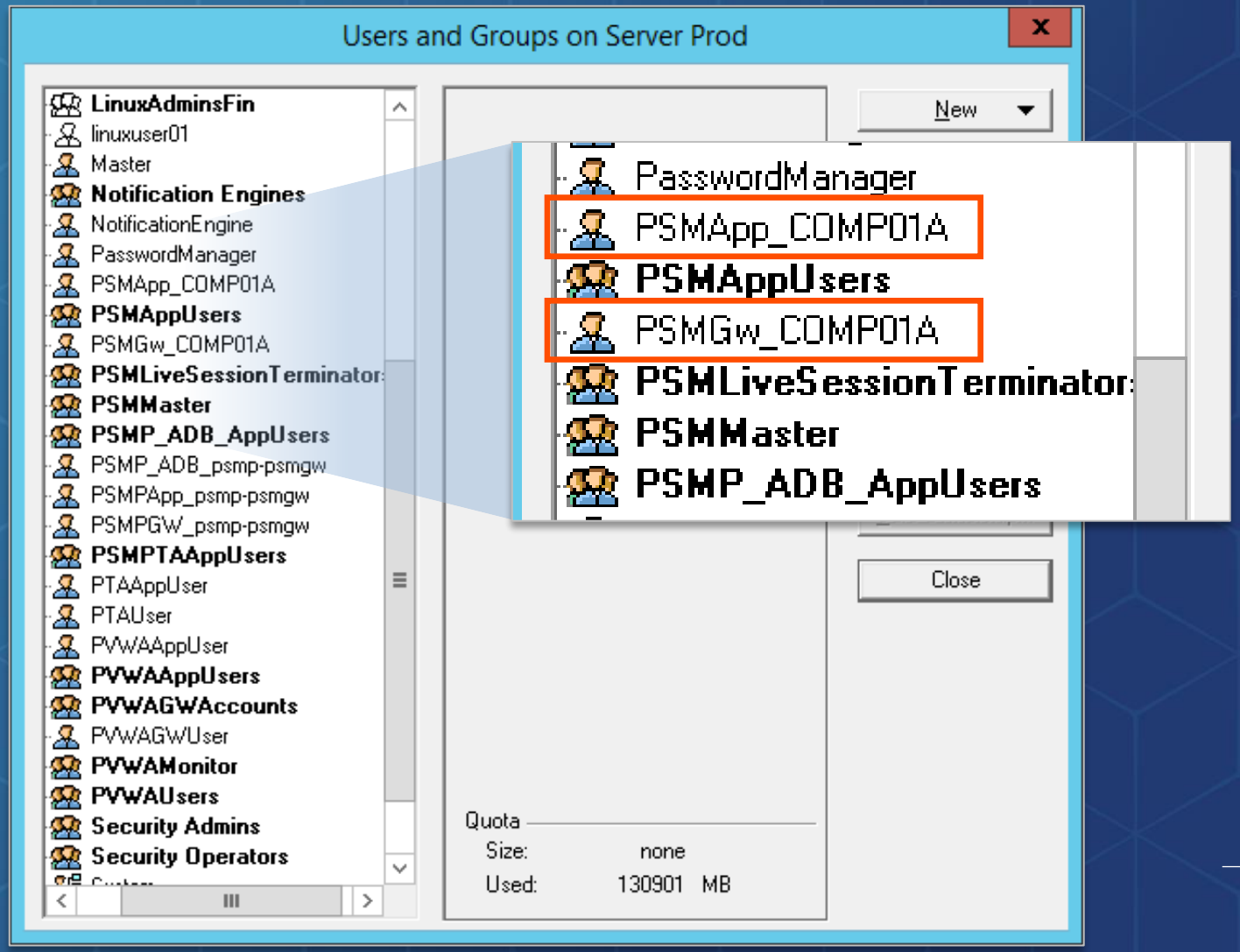
PSM Vault Users

PSMApp_<MachineName>

- Used by the **PSM** for internal processing
- The credential file for this user is stored on the **PSM** server in a file named psmapp.cred
- This user is added automatically to the PSMAppUsers group

PSMGW_<MachineName>

- This is the Gateway user through which the **PSM** will access the **Vault** to retrieve the target machine password
- The credential file for this user is stored on the **PSM** server in a file named ***psmgw.cred***
- This user is added automatically to ***PVWAGWAccounts*** group. Being a member of this group enables this user to access all password Safes



PSM Vault groups

PSMAppUsers

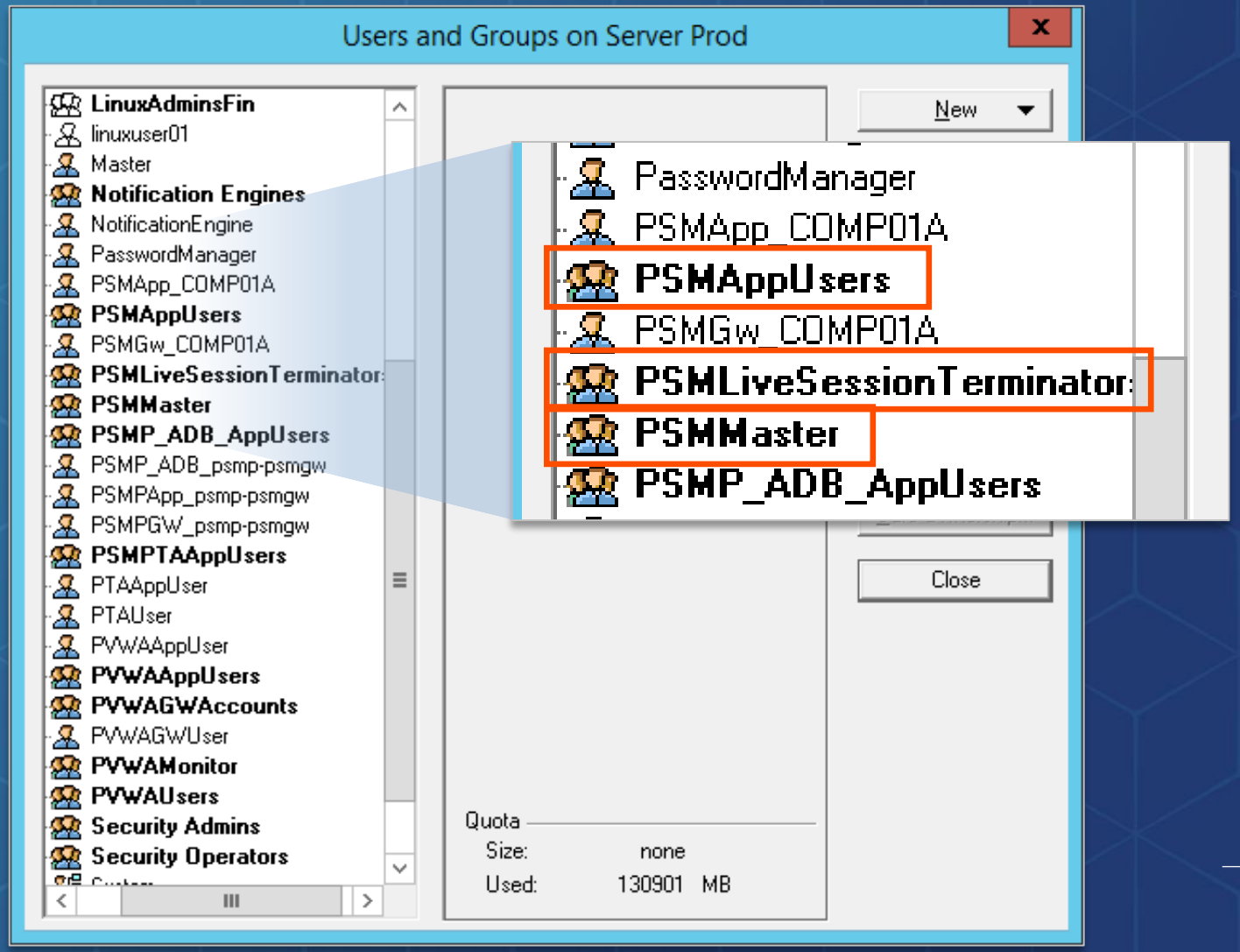
- This group is used to retrieve configuration data from the **Vault**, create Recording Safes, upload recordings, and perform other **PSM** activities

PSMLiveSession Terminators

- Members of this group can terminate, suspend, and resume live sessions

PSMMaster

- This group manages the Safes where recordings are stored.
- It is added to the Recordings Safes with all authorizations



Internal Communication

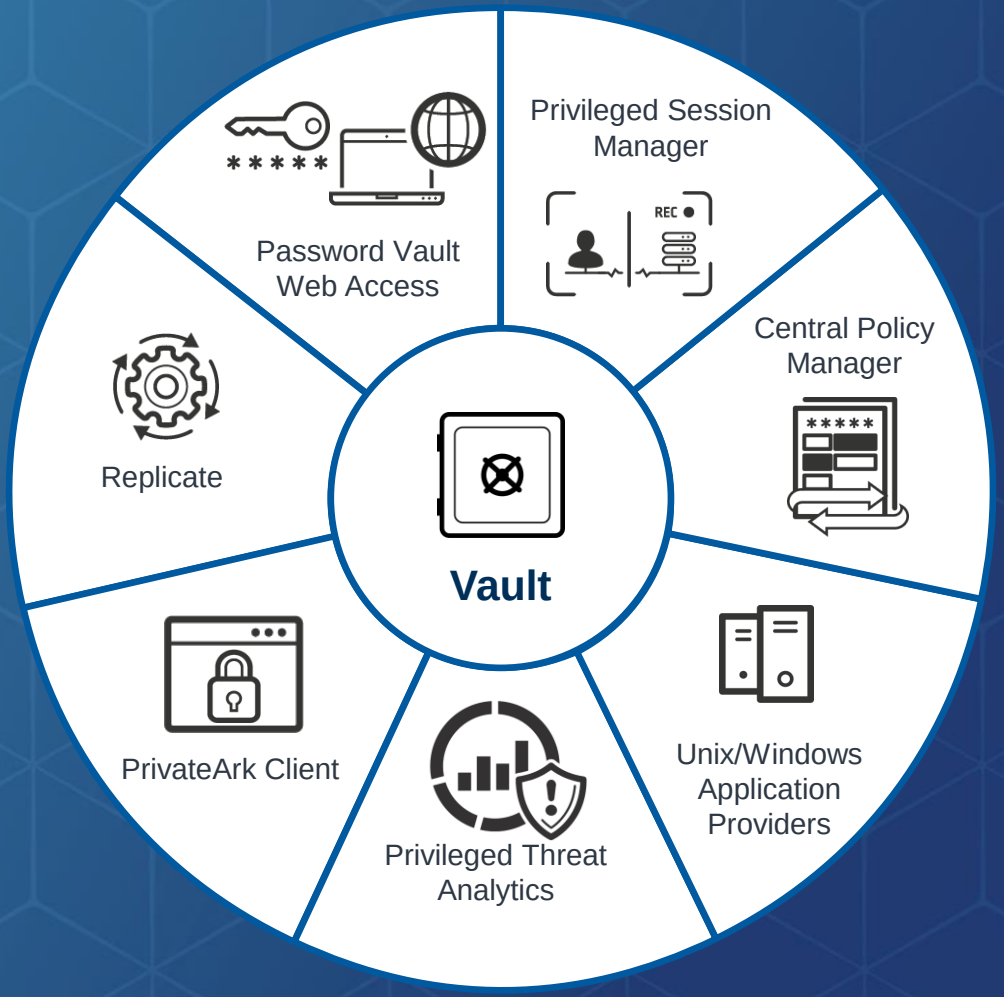
In this section we will look at how **Components** communicate with the **Vault** and each other:

- Direct communication with the **Vault**
- Communication with the **Vault** using **REST/API**

Direct Communication With the Vault

Connecting to the Vault

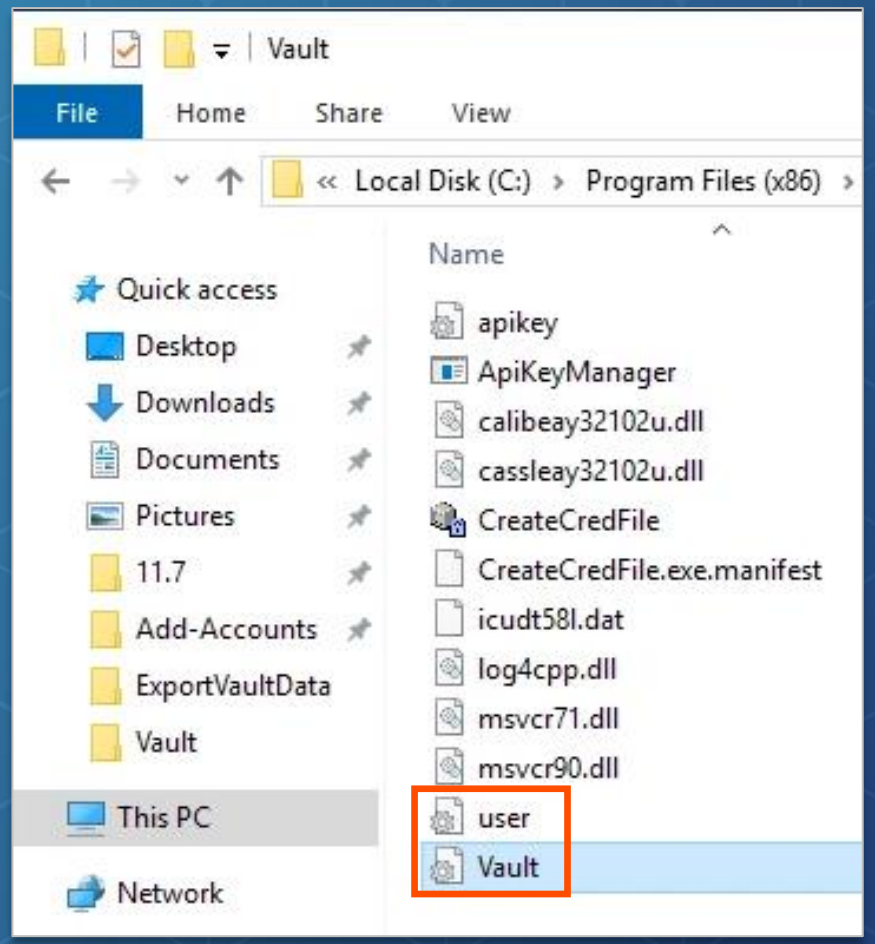
- Components communicate with the **Vault** using the **CyberArk** proprietary protocol on port **1858**
- Components must first authenticate to the **Vault** each time they are started
- Each Component has a User ID and password stored in a *“credential file”*

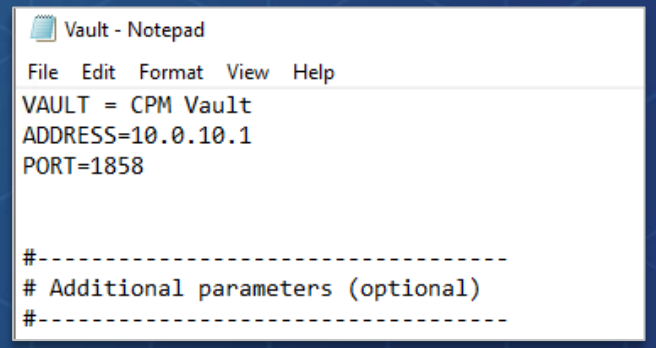


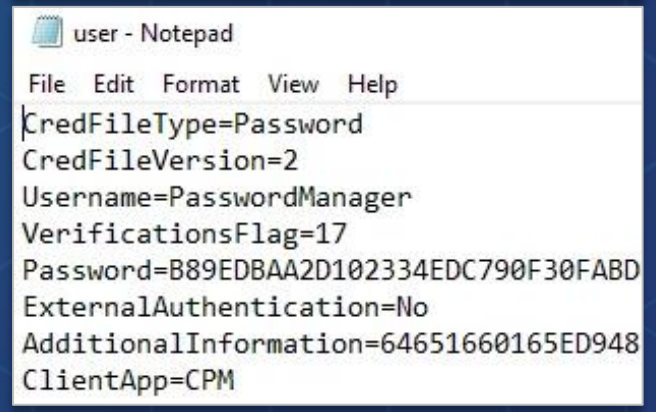
CPM Example

Vault Address and Credentials

- Each Component communicates with the **Vault** using the following configuration files:
  - *Vault.ini*
  - *Cred File*
- The **Vault.ini** file contains the **Vault** address and port
- The cred file contains the user name and a hash of the password used to authenticate to the **Vault**



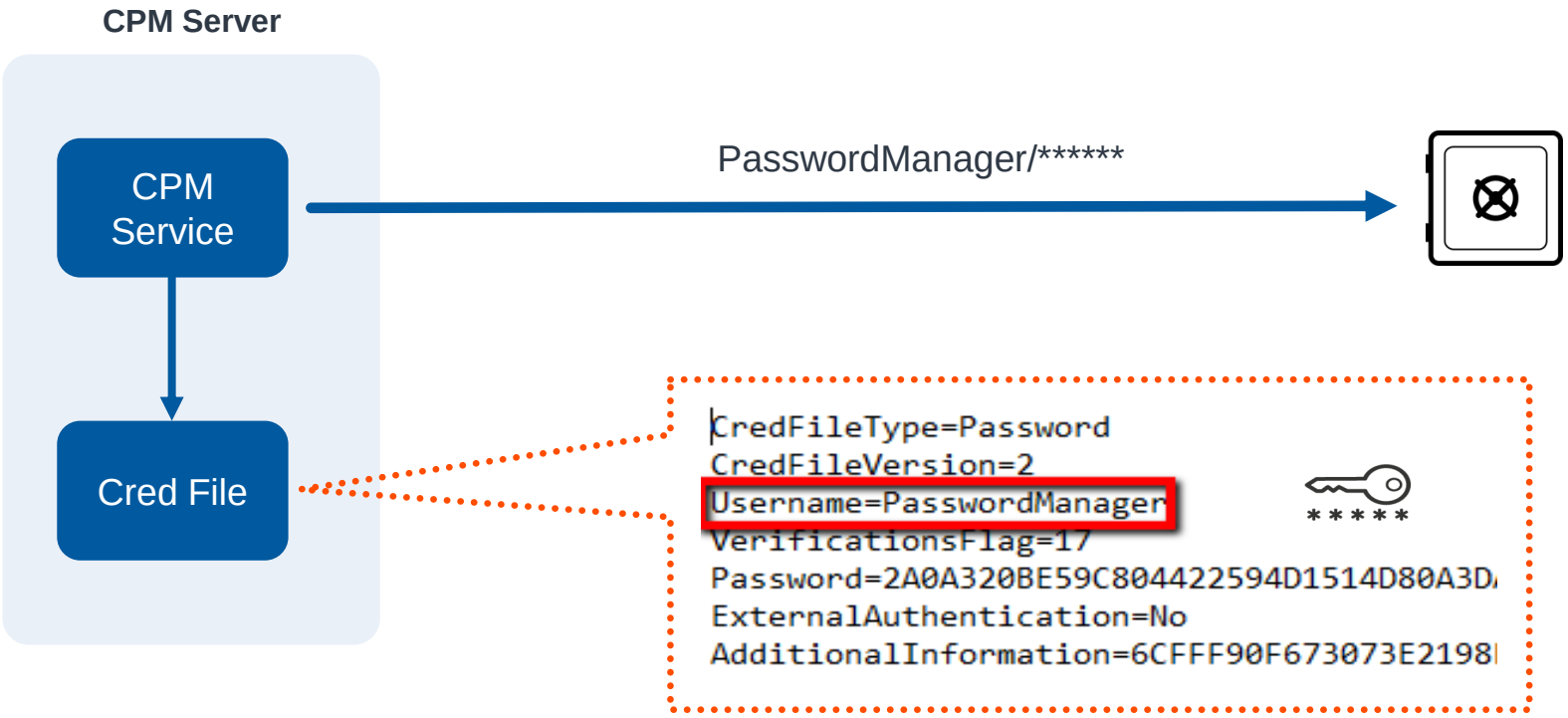




CPM Example

Vault Credential Files

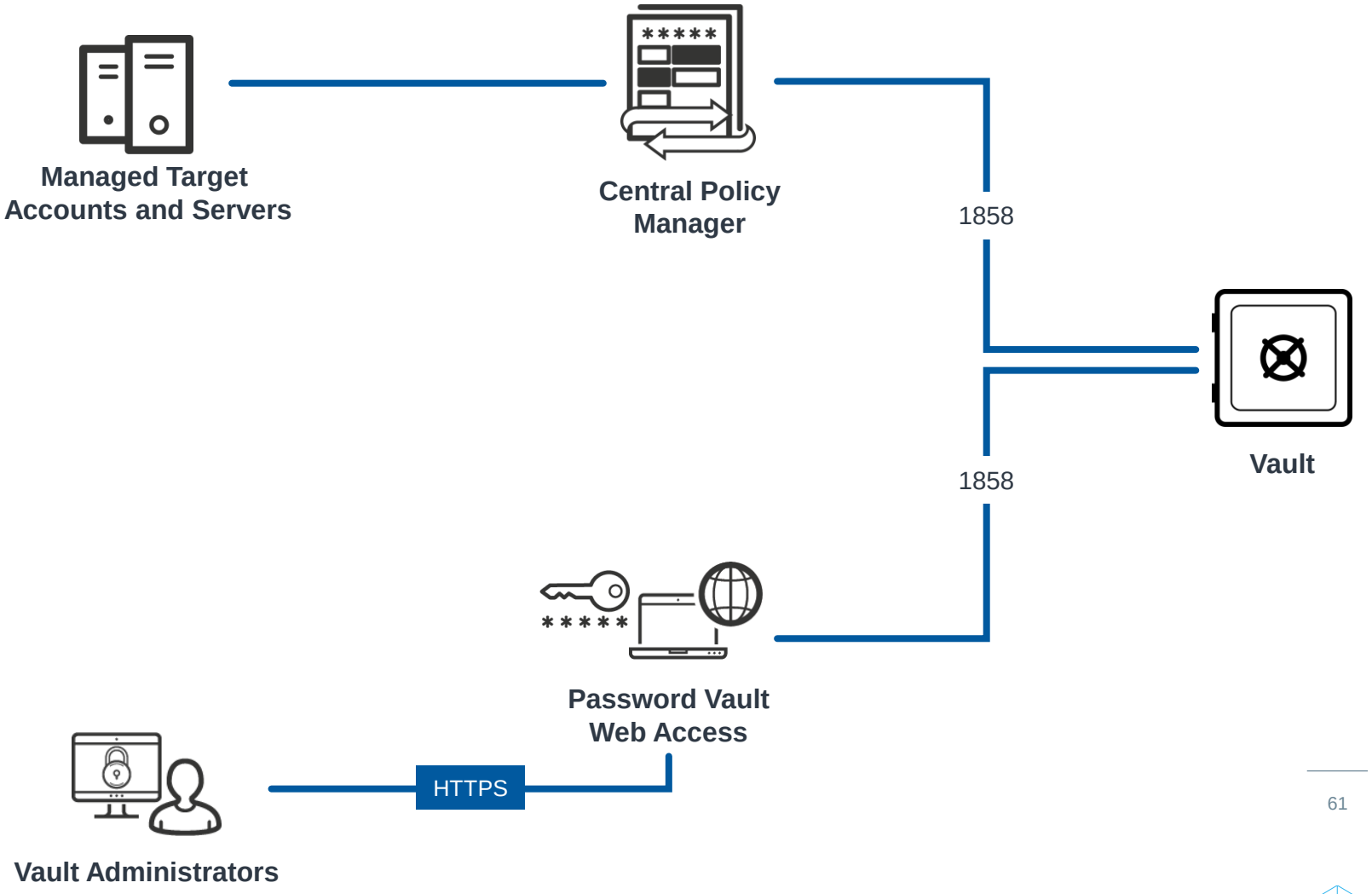
- When the **CPM** authenticates to the **Vault**, it uses the credentials stored in the file ***user.ini*** (the cred file):
  - *The CPM username*
  - *A hash of the password*
- After the **CPM** successfully authenticates, the password in the **Vault** and cred file are rotated



Communicating With the Vault Via REST

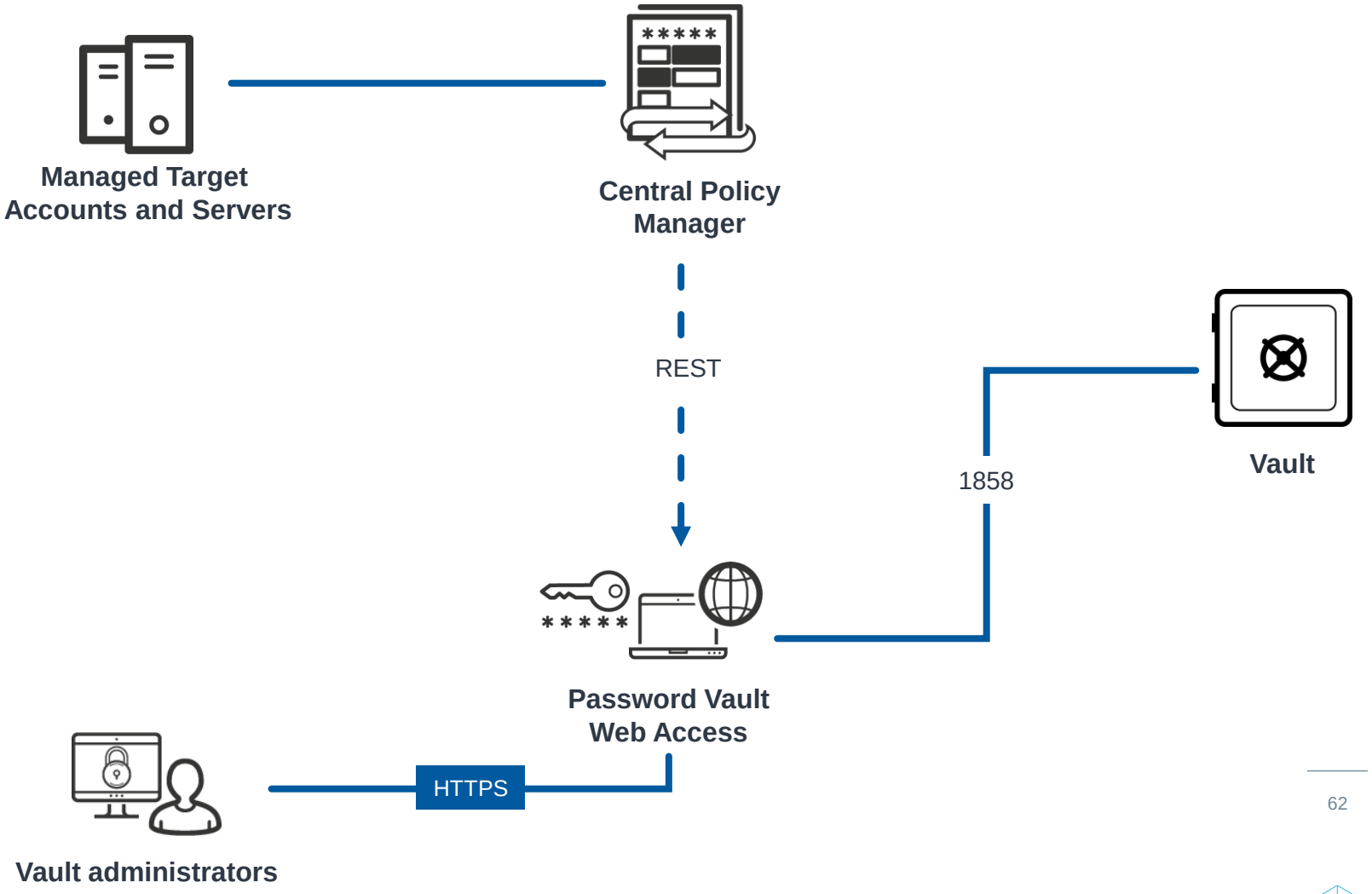
Component Internal Communication

Historically, components communicated directly with the **Vault** using the **CyberArk** proprietary protocol (over port 1858)



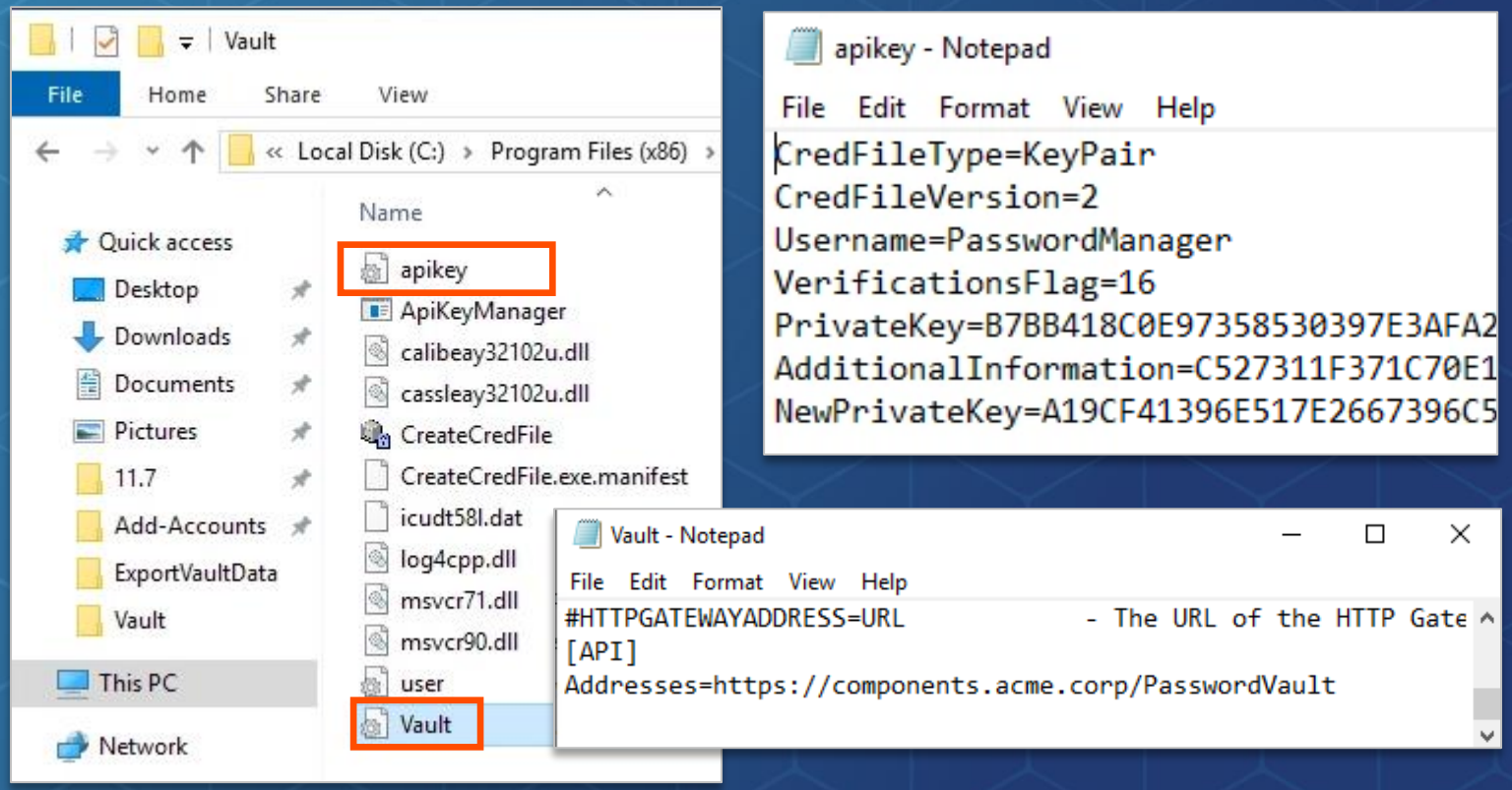
Component Communication – REST First

- As we move towards “REST first”, new functionalities use REST instead of the **CyberArk** proprietary protocol
- Components communicate with the **PVWA** over REST, and the **PVWA** performs the actions on the **Vault**



API Address and Keys

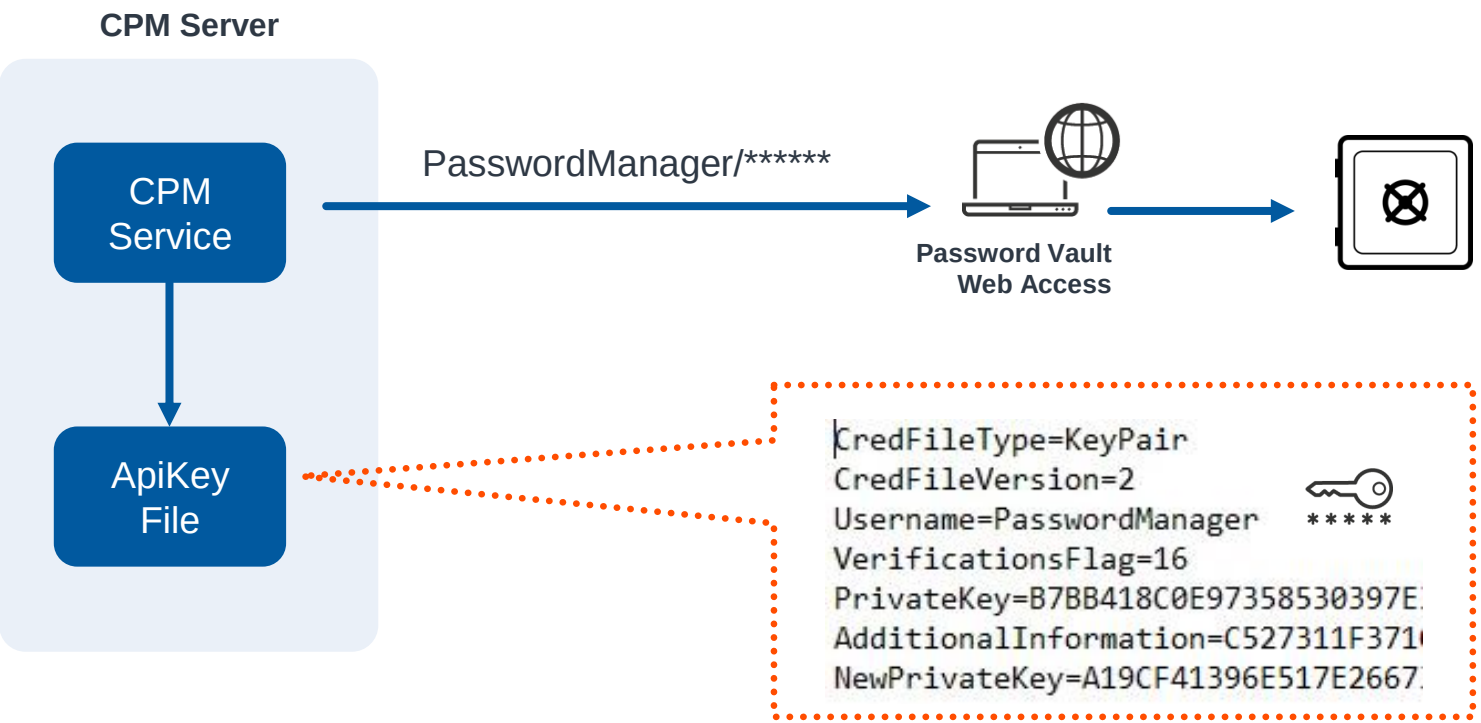
- When using REST to communicate with the **Vault**, components use the following configuration files:
  - ***Vault.ini***
  - ***ApiKey*** file
- The ***Vault.ini*** file contains the API address (**PVWA**)
- The ***ApiKey*** file contains the private key used to authenticate to the **Vault** via REST



CPM Example

API Keys

- An asymmetric key pair is used to provide a secure way for automated API calls and scripts, as well as **CyberArk** clients, to communicate with the **Vault**
- The private key is stored locally for use by the script or **CyberArk** client, while the public key is stored in the **Vault**
- Both keys are associated with a username that was previously created in the **Vault** and used for API authentication



Summary

Summary

In this session we discussed:

- The security controls protecting the **Vault** and encryption keys
- The local services, configuration files, and logs for the **PAM Self-Hosted** components
- The built-in **Safes** and users of the various components
- The internal integration and information flow among the **PAM Self-Hosted** components

Additional Resources

Documentation

CyberArk Digital Vault Security Standards

Security Fundamentals for PAM