APPLICATION IDENTITY MANAGER

AIM INTEGRATION - TECHNICAL DOCUMENTATION

Name of Company: WorkFusion

Website: http://www.workfusion.com/

Name of Product: Intelligent Automation: Smart Process Automation (SPA), Chatbots, SmartCrowd

Version: Intelligent Automation 2017 Sunbird Release

Date: June 28, 2017

WORKFUSION SOLUTION OVERVIEW

Smart Process Automation (SPA), part of WorkFusion's Intelligent Automation products, empowers enterprise operations to digitize. Smart Process Automation (SPA) combines robotic process automation (RPA), AI-powered cognitive automation and workforce analytics to automate high-volume business processes.

Our latest release, Intelligent Automation 2017 Sunbird Release supports cloud, on-premises and hybrid deployment types.
Supported operation systems:

1. CentOS 6.5-6.8 or RHEL 6.5-6.8
2. CentOS 7.x or RHEL 7.x

KEY BENEFITS

WorkFusion Intelligent Automation products solve common enterprise automation challenges, allowing enterprises to:

- Automate large & reliable, but inflexible & fragile core applications
- Tackle unstructured data that raises the need for manual work
- Scale back the need for email, chat, and phone interactions while increasing quality of services
- Optimize and increase people's capacity to do high-value work

Key features:
Business process design and management. With drag-and-drop simplicity, business people can configure complex end-to-end processes using a library of pre-built machine tasks and customizable worker interface templates.
Robotic automation. Robotics Process Automation (RPA) enables rules-based automation which eliminates the manual work of operating the user interfaces of enterprise applications like SAP and Oracle and moving structured data from one system to another.
Data sourcing and digitization. The software has built-in optical character recognition (OCR) and scraping technology to turn PDFs, documents, email messages, images and web content into machine-readable data.
Cognitive automation. WorkFusion's machine learning trains on historical data and real-time human work to automate more subjective work, like categorizing and extracting unstructured information.
Workforce management. Tasks within business processes are automatically routed to the right human or "bot", each action is quality controlled to ensure accuracy, and workloads are balanced to ensure optimal capacity utilization.
Business activities monitoring (BAM). The software delivers real-time, granular analytics on automation progress and coverage as well as workforce performance at an individual and team level.
Operational benefits:

- Improve service delivery by increasing customization, shortening cycle and concept-to-marketing times, and attacking backlog of unsolved application issues.
- Reduce cost per transaction by increasing throughput while decreasing manual effort.

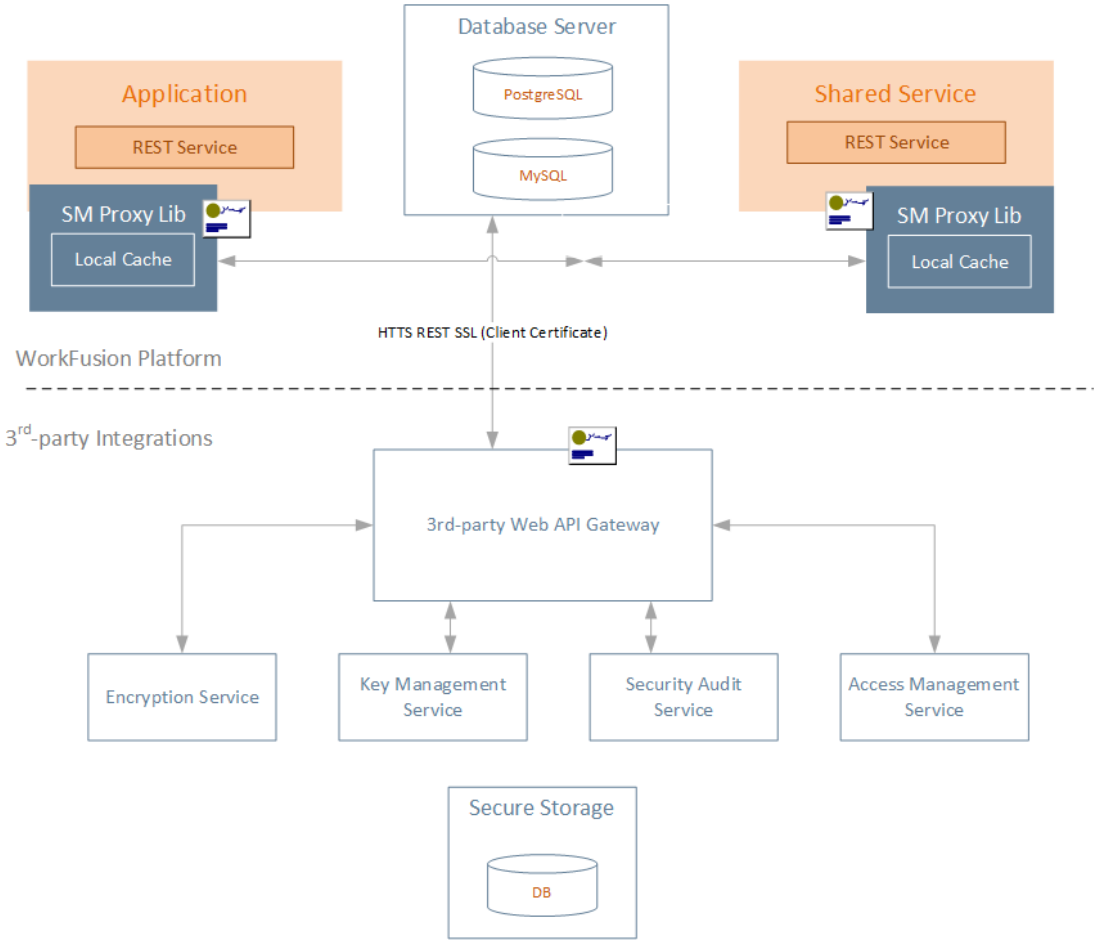
- Increase responsiveness to change through digitized work distribution, workload balancing, and process adaptability and flexibility.

Integration with AIM is targeted to enable WorkFusion customers with the option to store and manage the sensitive information (credentials) in a secure fashion (complied with enterprise information security standards). Two main types of credentials to be stored in CyberArk:

1. Robots' username and passwords

   WorkFusion's platform Robotic Process Automation (RPA) capability is used to automate the operations work in numerous customers' legacy and 3rd-party applications. The accounts credentials for robots to use will be stored and managed in CyberArk and WorkFusion will integrate with it seamlessly.
2. WorkFusion platform functional credentials. It includes databases and other functional components.

PRODUCT DIAGRAM & DESCRIPTION OF PRODUCT INTEGRATION



WorkFusion integrates with CyberArk deployment on customer side through CyberArk PAS (web API).Location of integration: End Point or Mid-Tier or Central Server. The setup may differ per customer. Default approach would be End Point.

Every WorkFusion installation on customer side will be integrated with CyberArk. Typically customer would have a separate WorkFusion installation per Line of Business and/or per geographical location.

The CyberArk REST API is utilized for credentials retrieval. Detailed description is in "Workfusion installation & integration configuration" section.

Two scenarios when credentials retrieval is triggered:

- Scheduled automation process (robot will request credentials from the vault for performing the planned action)
- Platform start up

Credential retrieval Frequency:

- Retrieve robots' username and passwords
  Every time robot needs to access customer's (as a rule legacy) or 3rd-party application – a call to the CyberArk vault is made to get the credential to perform the planned action. Volumes may differ per use case, and vary from about 100 to several million transactions a week.
- Retrieve WorkFusion platform functional credentials
  Retrieval is performed during WorkFusion platform start up.

AIM INSTALLATION

Refer to "Credential Credential Provider Implementation Guide" for CyberArk Credential Provider installation.

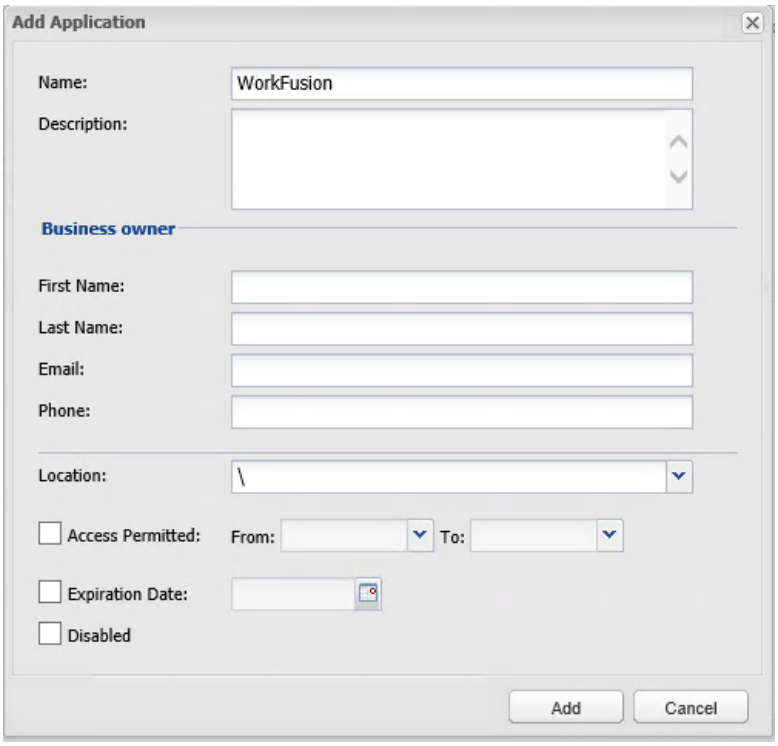
AIM CONFIGURATION

DEFINING THE APPLICATION ID (APPID) AND AUTHENTICATION DETAILS

To define the Application, here are the instructions to define it manually via CyberArk's PVWA (Password Vault Web Access) Interface:

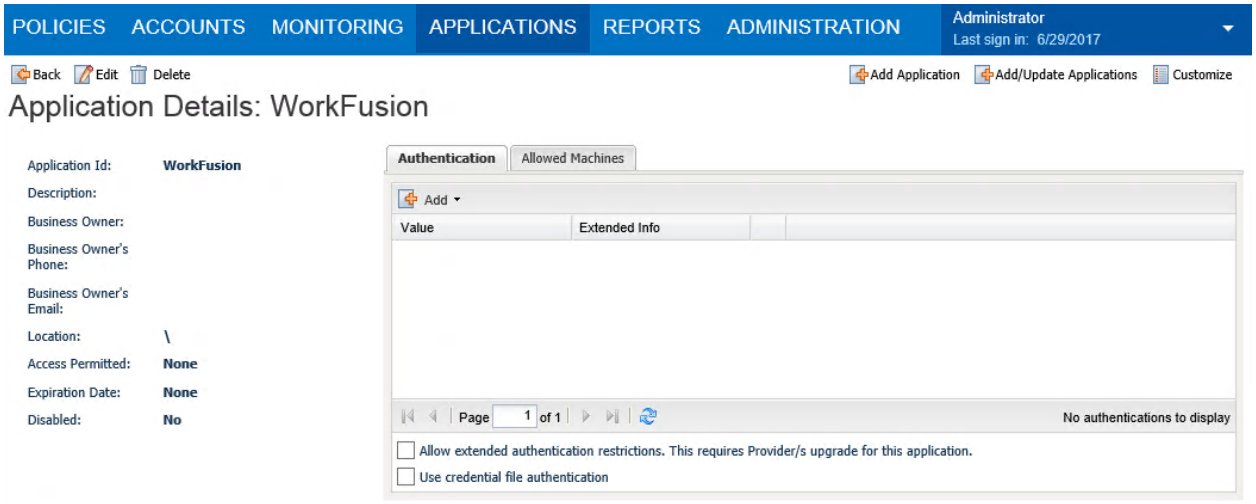
1. Logged in as user allowed to managed applications (it requires Manage Users authorization), in the Applications tab, click **Add Application**; the Add Application page appears.

>>Replace this screenshot with data filled in (i.e. with the pre-defined APPID the customer should use, specified in the “Name” field)



2. Specify the following information:
    - In the Name edit box, **WorkFusion**
    - In the Description, specify a short description of the application that will help you identify it.
    - In the Business owner section, specify contact information about the application’s Business owner.
    - In the lowest section, specify the Location of the application in the Vault hierarchy. If a Location is not selected, the application will be added in the same Location as the user who is creating this application.

3. Click **Add**; the application is added and is displayed in the Application Details page.



- **Allowing extended authentication restrictions**. This enables you to specify an unlimited number of machines and Windows domain OS users for a single application. Please check this box.

WorkFusion application uses Certificate Serial Number authentication.

PROVISIONING ACCOUNTS AND SETTING PERMISSIONS FOR APPLICATION ACCESS

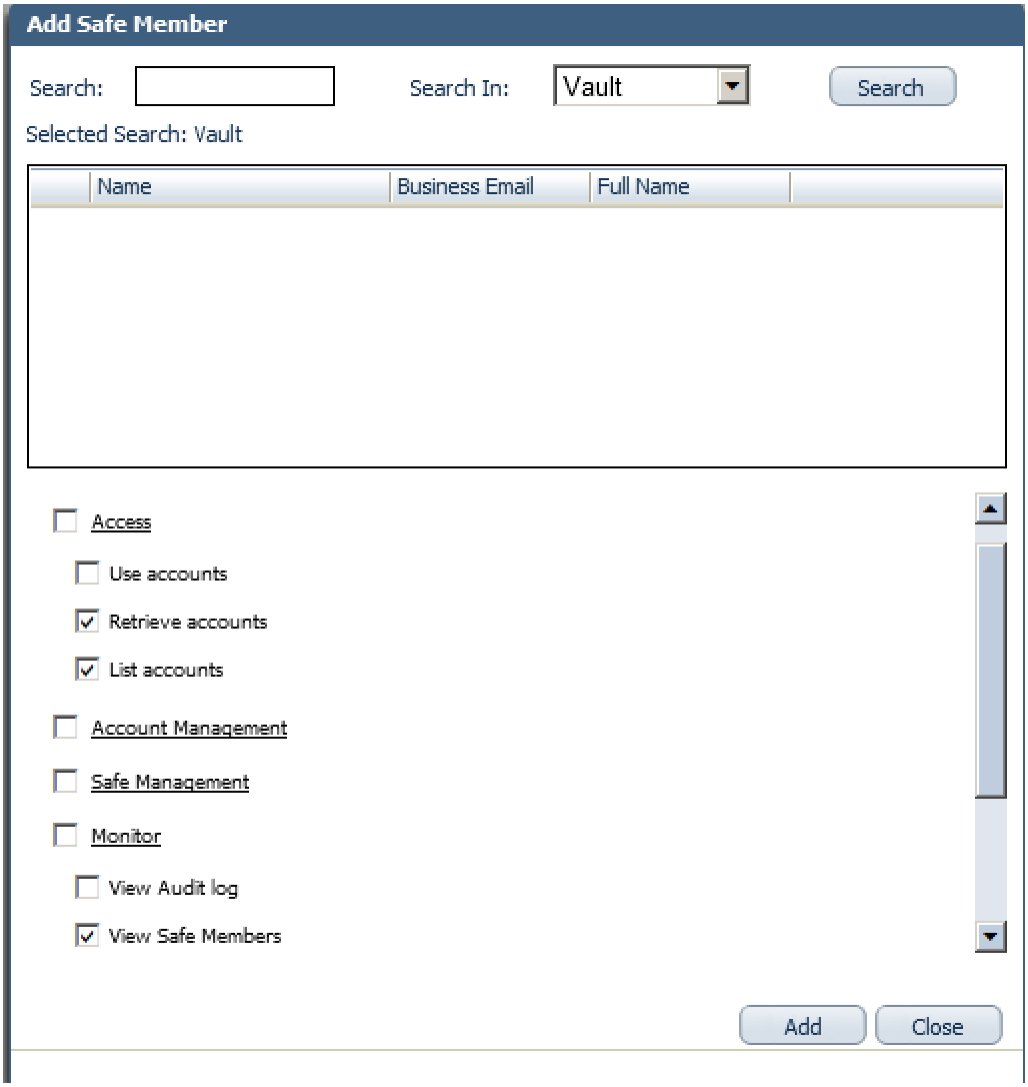
For the application to perform its functionality or tasks, the application must have access to particular existing accounts, or new accounts to be provisioned in CyberArk Vault (Step 1). Once the accounts are managed by CyberArk, make sure to setup the access to both the application and CyberArk Application Password Providers serving the Application (Step 2).

1. In the Password Safe, provision the privileged accounts that will be required by the application. You can do this in either of the following ways:
   - **Manually** – Add accounts manually one at a time, and specify all the account details.
   - **Automatically** – Add multiple accounts automatically using the Password Upload feature.

   For this step, you require the **Add accounts** authorization in the Password Safe.
   For more information about adding and managing privileged accounts, refer to **the Privileged Account Security Implementation Guide.**

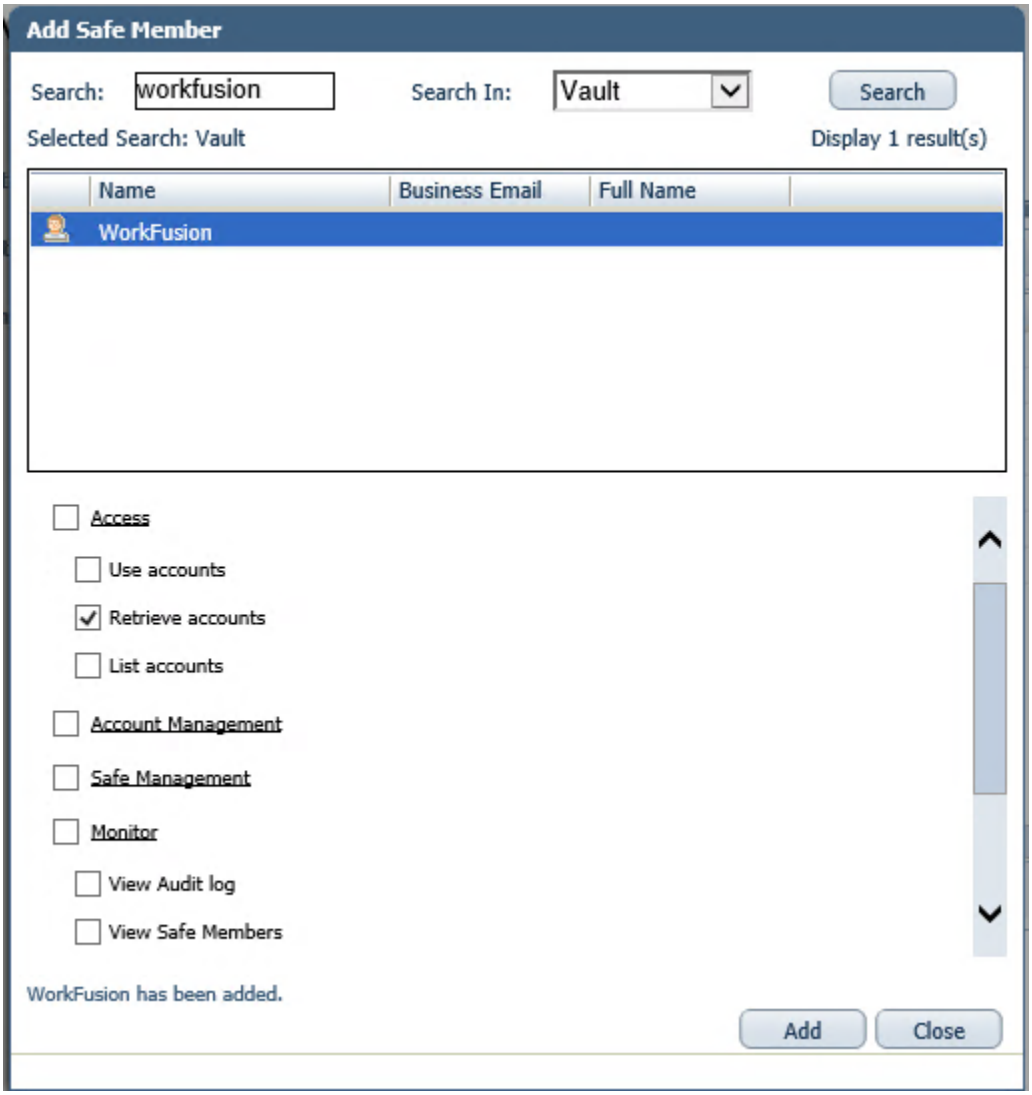
2. Add the Credential Provider and application users as members of the Password Safes where the application passwords are stored. This can either be done manually in the Safes tab, or by specifying the Safe names in the CSV file for adding multiple applications.
   i. Add the Provider user as a Safe Member with the following authorizations:
      - List accounts
      - Retrieve accounts

- View Safe Members
  Note: When installing multiple Providers for this integration, it is recommended to create a group for them, and add the group to the Safe once with the above authorization.



ii. Add the application(the APPID) as a Safe Member with the following authorizations:

- Retrieve accounts



iii. If your environment is configured for dual control:

- In PIM-PSM environments (v7.2 and lower), if the Safe is configured to require confirmation from authorized users before passwords can be retrieved, give the Provider user and the application the following permission:
  - Access Safe without Confirmation
- In Privileged Account Security solutions (v8.0 and higher), when working with dual control, the Provider user can always access without confirmation, thus, it is not necessary to set this permission.

iv. If the Safe is configured for object level access, make sure that both the Provider user and the application have access to the password(s) to retrieve.

For more information about configuring Safe Members, refer to the **Privileged Account Security Implementation Guide.**

WORKFUSION INSTALLATION & INTEGRATION CONFIGURATION

Refer to << WorkFusion 8.4 Installation Guide - 2017-06-08.pdf >> for Partner Product installation.
List specific steps to configuring Partner Product to work with AIM, including:

1. Install **CyberArk applications**:
   a. CyberArk EPV (with PVWA Rest API license)
   b. CyberArk PrivateArk Client
   c. CyberArk Central Credential Provider v9.8 or above

2. Setup **Certificate-based authentication** (mTLS) for IIS server.

2.1 Connect to the server where CyberArk PVWA was installed.
NOTE: This server have to have makecert.exe.
Guide for installing makecert.exe:
- If you use Windows Server 2012 R2 you can download and install Windows Software Development Kit (SDK) for Windows 8.1. (Try this link: Windows 8.1 SDK).
- You should install these 2 components only: [windows software development kit; net framework 4.5.1 software development kit].
Result: makecert.exe will be installed into: C:\Program Files (x86)\Windows Kits\8.1\bin\

2.2 Create **certificates using makecert.exe**

2.2.1 Create **Certificate Authority (CA)**

2.2.1.1 Create document by notepad and paste the following into it, where **-po changeit** → The password for the .pfx file:
Normal.dotm
"C:\Program Files (x86)\Windows Kits\8.1\Bin\x64\makecert.exe" -n "CN=CARoot" -r -pe -a sha512 -len 4096 -cy authority -sv CARoot.pvk CARoot.cer
"C:\Program Files (x86)\Windows Kits\8.1\Bin\x64\pvk2pfx.exe" -pvk CARoot.pvk -spc CARoot.cer -pfx CARoot.pfx **-po changeit**

2.2.1.2 Save the document (e.g. CreateCARoot.cmd) in your folder (e.g. C:\certificates).

2.2.1.3 Open Command Prompt -> go to your folder (C:\certificates) -> run CreateCARoot.cmd.

2.2.1.4 It should now prompt you to enter some passwords. (This is where we create and use the .pvk private key, so these need to match for success).
NOTE: You should now have 3 files: CARoot.cer, CARoot.pfx and CARoot.pvk in the folder where your batch files are.

2.2.1.5 Making CARoot.cer Trusted.

2.2.1.5.1 Open your new **CARoot.cer** file by double clicking it and see that isn't trusted.

2.2.1.5.2 To make it trusted on your machine open up the **Microsoft Management Console** (Find it by searching for **mmc** in start).

2.2.1.5.3 Go to the **File -> Add/Remove** Snap-in

2.2.1.5.4 Double-click **Certificates** in the list to the left

2.2.1.5.5 Choose **Computer account** and just go **next**, **finish** and **OK**.

2.2.1.5.6 Open the **Trusted Root Certification Authorities** -> **Certificates**.
Here you can see all of the currently trusted certificates that Windows trusts.

2.2.1.5.7 **Right-click** the **Certificates folder** -> **All tasks** -> **Import…**

2.2.1.5.8 The certificate **Import Wizard** will pop up.
Go **next** -> **Browse** to find the CARoot.cer file, what we created earlier. Keep going **next** until **finish** where a message box should appear saying "**The import was successful**."

2.2.1.5.9 Your CARoot certificate should now be in you Trusted Root Certification Authorities store.

2.2.1.5.10 Open the CARoot (double-click) and see that it is now trusted by your computer.

2.2.2 Create **Server Certificate**.

2.2.2.1 We need a certificate to handle SSL on the server. We will create this with a new command batch file in notepad just like before, this time with these parameters:

"C:\Program Files (x86)\Windows Kits\8.1\Bin\x64\makecert.exe" -n "CN=cbrk-api.workfusion.com" -iv CARoot.pvk -ic CARoot.cer -pe -a sha512 -len 4096 -b 02/03/2017 -e 01/03/2018 -sky exchange -eku 1.3.6.1.5.5.7.3.1 -sv %1.pvk %1.cer

"C:\Program Files (x86)\Windows Kits\8.1\Bin\x64\pvk2pfx.exe" -pvk %1.pvk -spc %1.cer -pfx %1.pfx -po changeit

NOTE: The **CN must match your domain** otherwise the browsers won't trust your SSL certificate and warn the end user not to proceed to your website.

2.2.2.2 Save the document (e.g. CreateSslServerCert.cmd) which will create a command batch file in your folder (e.g. C:\certificates).

2.2.2.3 Open **Command Prompt** → go to your folder (e.g. C:\certificates) → run CreateSslServerCert.cmd ServerSSL

2.2.2.4 Again it will ask you to create your private key password, use it to verify, also give the issuers password (which is the one you chose when creating your root CA) and lastly the private key password you choose in the first window.

2.2.2.5 You should now have 3 new files: ServerSSL.cer, ServerSSL.pfx and ServerSSL.pvk in the folder where your batch files are.

2.2.2.6 Import the **Personal Information Exchange (pfx)** certificate into your **Personal Certificates in the Microsoft Management Console**.

2.2.2.6.1 Open up the **Microsoft Management Console**. (Find it by searching for **mmc** in start)

2.2.2.6.2 Open the **Personal folder ➜ right-click Certificates ➜ Import…**

2.2.2.6.3 Again the **Certificate Import Wizard** pops up ➜ go **Next.** This time you will Browse for the **ServerSSL.pfx** file

2.2.2.6.4 Go **next ➜ Type in the password for your pfx file** (The -po parameter from the batch file) ➜ Continue going next until finish and the message box with "**The import was successful**" appears.

2.2.2.6.5 You should now see you newly imported certificate in your ➜ **Personal Certificates folder.** It is trusted automatically because your CARoot that signed it is trusted and has a private key corresponding to this certificate.

2.2.3 Create **Client Certificate**.

2.2.3.1 We will create this with a new command batch file in notepad just like before, this time with these parameters:

"C:\Program Files (x86)\Windows Kits\8.1\Bin\x64\makecert.exe" -n "CN=%1" -iv CARoot.pvk -ic CARoot.cer -pe -a sha512 -len 4096 -b 02/03/2017 -e 01/03/2018 -sky exchange -eku 1.3.6.1.5.5.7.3.2 -sv %1.pvk %1.cer

"C:\Program Files (x86)\Windows Kits\8.1\Bin\x64\pvk2pfx.exe" -pvk %1.pvk -spc %1.cer -pfx %1.pfx -po changeit

NOTE: need to change params to valid your params.

2.2.3.2 Save the document (e.g. CreateSslClientCert.cmd), what will create a command batch file in your folder (e.g. C:\certificates).

2.2.3.3 Open **Command Prompt** → go to your folder (e.g. C:\certificates) → run **CreateSslClientCert.cmd ClientCert**

2.2.3.4 Enter the passwords in the same pattern as the server certificate and you now have your client certificate.

2.2.3.5 You should now have 3 new files: ClientCert.cer, ClientCert.pfx and ClientCert.pvk in the folder where your batch files are.

2.2.3.6 Add it to your **Current User Personal Certificate store**:

2.2.3.6.1 Open up the **Microsoft Management Console**. (Find it by searching for **mmc** in start)

2.2.3.6.2 Click **File ➜ Add/Remove** Snap-in

2.2.3.6.3 Double-click **Certificates** again, but this time choose **My user account** and **Finish**.

2.2.3.6.4 Open the **Personal folder ➜ Right-click Certificates ➜ Import…**

2.2.3.6.5 Browse for your **ClientCert.pfx** file

2.2.3.6.6 Go **next ➜ Type in the password to your pfx file** (-po parameter from the batch file) ➜ Continue going next until finish and "**The import was successful**" message box appears.

2.2.3.6.7 You should now see you newly imported certificate in your **Personal ➜ Certificates** folder. Again the certificate is trusted because the CARoot is trusted by Windows.

2.3 **Configure IIS to use your certificates with your application including IIS client certificate mapping authentication**.

2.3.1 Open the **IIS Manager** (Find it by searching for **iis** in start)

2.3.2 In the IIS Manager go **to your server** (The top of the tree to the left) ➜ **Scroll down and double-click Server Certificates**.

2.3.3 Click **Import…**

2.3.4 Click the … and find your **ServerSSL.pfx** file, fill out the password (the -po parameter in your command batch file) and click **OK**

2.3.5 Double-click the newly added cert to verify that it is trusted

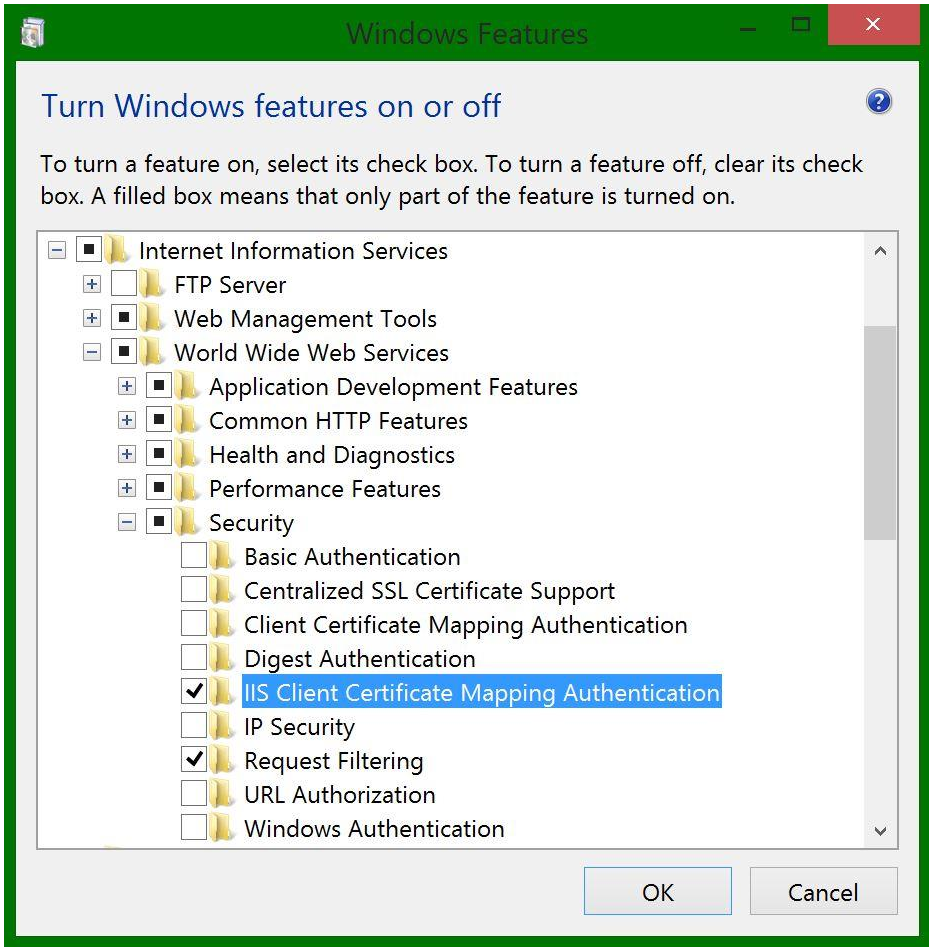
2.3.6 Go to **your server** (The top of the tree to the left) → **Sites** → **Default Web Site** → **Bindings...** (On right column actions).

2.3.7 **Add/Edit https bindin**g. (Setup ip address, port, host name and select **your ServerSSL** certificate from dropdown **SSL certificate**).

2.3.8 Close bindings window.

2.3.9 IIS Client Certificate Mapping Authentication for CyberArk apps (**PasswordVault** and **AIMWebService**)

2.3.9.1 First we need to install the feature, so bring up the "**Turn Windows features on or off**" again and install the following (Find it by searching for "Turn Windows features on or off" in start):



2.3.9.2 Go to **your server** (The top of the tree to the left) → **Sites** → **Default Web Site** → **PasswordVault(AIMWebService)** → **SSL Settings**

2.3.9.3 Select **Require SSL and Client Certificates - Accept** and click on the **Apply** (On right column Actions).

2.3.9.4 Go to **your server** (The top of the tree to the left) → **Sites** → **Default Web Site** → **PasswordVault(AIMWebService)** → **Configuration Editor**

2.3.9.5 Choose the **iisClientCertificateMappingAuthentication** section (you can also enter the path system.webServer/security/authentication/iisClientCertificateMappingAuthentication into the Section field)

2.3.9.6 **Enable** the **client certificate mapping authentication**

2.3.9.7 Add a mapping click the … of the **manyToOneMappings**

2.3.9.8 Add the users that you want to grant access. Click on the **Add** and fill out the properties for a mapping and repeat for each user you want to configure for access or denial. Remember that you need the client certificate and root CA certificate installed on all the user's mmc. **IMPORTANT**: In order for this to work you need to enter a valid username and password and since my computer is the server, **the credentials will be my Windows username and password**.

2.3.9.9 **Create some rules** to go with this mapping so the server can determine if a client is allowed in or not. It's a so click on the **rules property** and the **…** button

2.3.9.10 Add one rule as an example where the server will check the client certificate to see if it's signed by the correct CA root. Go ahead and add more rules for more safety, please visit the IIS Many-To-One Mapping reference for more documentation.
Example:
certificateField = Issuer
CertificateSubField = CN
compareCaseSensitive = True
matchCriteria = CARoot

2.3.9.11 Remember to **apply** the changes in the IIS Manager, so **close** the rules and mappings windows and click **Apply**

2.3.9.12 Check : Open a new incognito browser window to make sure to start from a clean slate cache and cookie-wise and enter your url with the /api/cats and see the browser prompting you for a certificate. Choose the ClientCert and click ok to gain access to the cats.
If this is not working, make sure that your client certificate is in your CurrentUser/Personal store as well as in your browser's certificate store. If yes, then go to Control Panel ➜ Internet Options ➜ Content and click Clear SSL state.

2.4 **Configure PasswordVault application for using mTLS authentication**.

2.4.1 Open the **IIS Manager** (Find it by searching for **iis** in start)

2.4.2 Go to **your server** (The top of the tree to the left) → **Sites** → **Default Web Site** → **PasswordVault** (Expand it) → **WebServices** → **auth** → **Shared** → **Select SSL Settings**

2.4.3 Select the following: [**Require SSL** and **Require for Client certificates**].

2.4.4 Click **Apply**.

2.4.5 Run **iisreset**.

2.5 **Configure AIMWebService application for using mTLS authentication**.

2.5.1 Open your **IIS Manager** (Find it by searching for iis in start)

2.5.2 Go to **your server** (The top of the tree to the left) → **Sites** → **Default Web Site** → **AIMWebService** → **Select SSL Settings**

2.5.3 Select the following:

Require SSL

Require for Client certificates.

2.5.4 Click **Apply**.

2.5.5 Run **iisreset**.

3. **Create specific CyberArk users for REST API**. (need to create 2 users: for SharedLogon and for checking permission on Secure Storage page).

3.1 Connect to the server where privateArk application was installed.

3.2 Open the **PrivateArk** application.

3.3 Click on the **Vault** and **logon** to the server vault using Administrator credentials.

3.4 Go to the **Tools** → **Administrative Tools** → **Users and Groups**.

3.5 Click on the **New** button and select **User**.

3.6 Fill the **User Name** input on the **General** tab. (e.g. wfRestApi).

3.7 Select **User type: EPVUser** and click on the **Authorized Interfaces...** button.
You should check that **Authorized Interfaces** column contains **IBVSDK** type. If not, please select it from **Available Interfaces** and click on the **<** button.

3.8 Click on the **OK** button.

3.9 Click on **Authentication** tab. Select **Password** for **Authentication method** input.
Enter the password into **Password** and **Confirm** inputs.
Uncheck **User Must Change Password at Next Logon** select-box.
Select **Password Never Expires** select-box.

3.10 Click on the **OK** button.

3.11 Click on the **Close** button.

3.12 Select **your user** and click on the **Trusted Net Areas...** button.

3.13 Click on the **Update** button and set 100 into field: **After: _ access violations**.

3.14 Click on the **OK** button and **Close**.

3.15 Go to the **User** → **Logoff**.

3.16 Close the **PrivateArk** app.

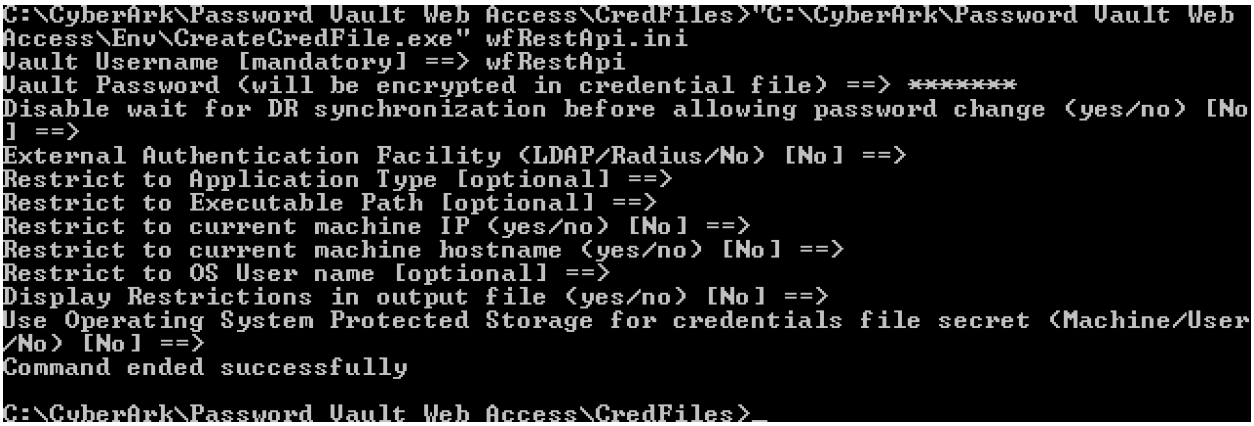
4. Configure shared logon for CyberArk PAS.

4.1 Go to the server where installed CyberArk PAS Web Services.

4.2 Open Command Prompt (**cmd**).

4.3 Run: **cd C:\CyberArk\Password Vault Web Access\CredFiles**

4.4 Run: **"C:\CyberArk\Password Vault Web Access\Env\CreateCredFile.exe" filename.ini** (e.g.: "C:\CyberArk\Password Vault Web Access\Env\CreateCredFile.exe" wfRestApi.ini)
fill correct info for :
Vault Username [mandatory] ==> need to write username, who we created in step 3 (e.g. wfRestApi)
Vault Password (will be encrypted in credential file) ==> password for user from step 3.
for **next steps** click **Enter** always.



4.5 **Open web config** file for PWA: "**C:\inetpub\wwwroot\PasswordVault\web.config**"

4.6 **Add** new **key** into <appSettings>:
<add key="WSCredentialFile" value="C:\CyberArk\Password Vault Web Access\CredFiles\wfRestApi.ini"/>
Save and **close** it.

4.7 Run **iisreset**.

5. **Import Application Platform plug-in** : DummyPlatform.zip. (Imported Dummy Application Platform).

5.1 Go to the: **https://your-domain/PasswordVault/logon.aspx**

5.2 **Sign In** with **Administrator credential**.

5.3 Click on the **Administration** tab.

5.4 Click on the **Platform Management** row.

5.5 Click on **Import Platform** button.

5.6 Select **ApplicationPlatform.zip**

5.7 Find platform by name: **No Management Plugin** and click on it.

5.8 Click on **Duplicate** button/icon (on right bar)

5.9 Enter the platform name - **WFApplication** and click **Save & Close** button.

5.10 Click on the **Policies** tab.

5.11 Click on the **Master Policy** item.

5.12 Select the "**Require users to specify reason for access**" under the "**Privileged Access Workflows**". Click on icon-button **Add Exception** (on right corner).

5.13 Select Platforms : **WFApplication** and click on the **Next**.

5.14 Select the **Inactive** buttons for "**Require users to specify reason for access**" and "**Allow users tof specify free text reason for access**" and click on the **Finish** button.

6. Check that internal application for AIMWebService is exists

6.1 Login to the instance where AIMWebService was installed.

6.2 Go to the folder where the Central Credential Provider Web Service was installed (default: C:\inetpub\wwwroot\AIMWebService) and open the **web.config** file.

6.3 Find the key "**AppID**", e.g. :

```
<appSettings>
  <add key="AppID" value ="AIMWebService"/>
</appSettings>
```

6.4 Login to the PVWA (https://your-domain/PasswordVault/logon.aspx) and open **Application** tab.

6.5 Input value from step "6.3" into **Search for**: and click on the **Search** button.

6.6 If the application doesn't exist, you should add it for **AIMWebService**. (refer "Central Credential Provider Implementation Guide.pdf").

NOTE: By default we are using this application with **Path** authentication. (default path: C:\inetpub\wwwroot\AIMWebService\bin\AIMWebService.dll).

7. Create specific application issuing the password request – WorkFusion

7.1 Login to the **PVWA** (https://your-domain/PasswordVault/logon.aspx) and open **Application** tab.

7.2 Click on the **Add Application**.

7.3 Fill inputs:

Name : WorkFusion

Description: WorkFusion application issuing the password request.

First Name: your first name, it is optional field

Last Name: your last name, it is optional field

Email: your email, it is optional field

Phone: your phone, it is optional field

Location: \

Click on the **Add** button.

7.4 You should see the **application details page**.

7.5 Click on the **Add** button in the **Authentication** grid.

7.6 Select **Certificate Serial Number**.

7.7 Insert Serial Number from client certificate (Step 2.2.3), what uses for mTLS authentication.
NOTE:
Extract the Serial Number value from the Client Certificate.
The following example shows how to extract the Serial Number in Windows Certificate Manager, although any management utility can be used.
Open the Windows Certificate Manager → find client certificate and click on it.
Select **Details** tab and select **Serial number** field.

7.8 Click on the **Add** button

8. Create specific safes.

8.1 Go to the: **https://your-domain/PasswordVault/logon.aspx**
8.2 **Sign In** with Administrator credential.
8.3 Click on the **Policies** tab.
8.4 Click on the **Access Control (Safes)**
8.5 Click on the **Add Safe** button.
8.6 Fill inputs:
Safe name - instance_domain_WFInternal for **internal** service accounts and
instance_domain_WFApp for **custom** accounts.
Click on the **Save** button.

Example for safe names for (demo32.workfusion.com host servers).
for internal safe - demo32.workfusion.com_WFInternal
for custom safe - demo32.workfusion.com_WFApp
8.7 You should see **Safe details** page with **Members grid**.
8.8 Click on the **Add Member** in **Members** grid.
8.9 In **Search** input fill your rest api username and click on the Search button.
8.10 **Select** your **rest api user** and add next Access for him:
Access
Use accounts
Retrieve accounts
List Accounts
Add accounts (includes update properties)
Update account content
Rename accounts
Delete accounts.
8.11 Click on the **Add** button, when you see "your_username has been added" message click on the **Close** button.
8.12 You should see your rest api username in the Members grid.
8.13 Add additional users which is used in **AIMWebService**.
Usernames: **WorkFusion** and username starts with **Prov_...**
Where : WorkFusion username was created in step 7.
Prov_.... username (e.g. Prov_WIN-PB5J4773KRG) was created when you install Central Credential Provider

These users should have next permissions (should be checked):
Access
Retrieve accounts
List accounts
Account Management
Initiate CPM account management operations

Specify next account content
Monitor
View Safe Members
Workflow
Access Safe without confirmation

8.14 Need to create **2 safes(_WFInternal and _WFApp)per instance (instance.domain_WFInternal and instance.domain_WFApp)**

9. Configure WorkFusion apps to integrate with CyberArk

9.1 Go to the Tomcat folder.

9.2 Go to the config folder and add/update workfusion.properties (workspace.properties, etc) next properties:

```
secure.storage.safe.customer.default=${instance.name}_WFApp
secure.storage.safe.internal=${instance.name}_WFInternal
secure.storage.type=CYBERARK
secure.storage.serverApi=https://cbrk-server-host.com
secure.storage.platformId=WFApplication
secure.storage.client.certificate=pathToClientCertificate (ClientCert.pfx)
secure.storage.client.keyPass=keypass
secure.storage.username.customer=username2
```

where
username2 - username from step 3 – second user. This user is custom validation that client can see secret from WorkFusion UI. WorkFusion UI ask client password for this user and check that this user has access to the safe and pass is correct. If pass is correct, client can see secret otherwise he can't see it.

WorkFusion get all internal secure properties and custom secure properties from CyberArk.

Summary:
Installed components:
CyberArk Vault
CyberArk PrivateArk
CyberArk PAS (web API)
CyberArk AIM (Central Credential Provider + Central Credential Provider Web Service)

Authentication:
WF <> CyberArk: Certificate-based Auth with Shared User (mTLS)
Configuration:
Create WorkFusion API user: EPV User Type w/ API permission (IBVSDK) should be used
Import Application Platform plug-in: DummyPlatform.zip
Create internal WF Safe: "${instance_name}_WFInternal" for secure configuration properties.

Create a Safe for custom secrets management: ""${instance_name}_WFApp" for custom secrets.
Please note for multi-tenant environment 2 Safes per instance would be required
Now WorkFusion support only 2 safes – internal for configuration properties and custom for specific custom secrets.

PARTNER CONTACT INFO

Business Contact	Name	Gautam Moorjani
	Email	gautam@workfusion.com
	Tel	+1 (917) 912-0344
Technical Contact	Name	Andrei Zhemaituk
	Email	azhemaituk@workfusion.com
	Tel	+1 (917) 909-9822
Support Contact	Name	WorkFusion Support
	Email	support@workfusion.com
	Tel	+1 646-453-7974