CYBERARK PRIVILEGED ACCESS SECURITY SOLUTION

Integration Guide

Applicable Devices:
KMES Series 3

TABLE OF CONTENTS

[1] DOCUMENT INFORMATION

[1.1] DOCUMENT OVERVIEW

The purpose of this document is to provide information regarding the configuration of the Futurex KMES Series 3 with CyberArk's Privileged Access Security (PAS) solution using PKCS #11 libraries. For additional questions related to your KMES, see the relevant administrator's guide.

[1.2] APPLICATION DESCRIPTION

CyberArk's Privileged Access Security (PAS) solution is a full life-cycle solution for managing the most privileged accounts and SSH Keys in the enterprise. It enables organizations to secure, provision, manage, control and monitor all activities associated with all types of privileged identities, such as:

- Administrator on a Windows server
- Root on a UNIX server
- Cisco Enable on a Cisco device
- Embedded passwords found in applications and scripts

The Privileged Access Security solution provides a 'Safe Haven' within your enterprise where all your administrative passwords can be securely archived, transferred and shared by authorized users, such as IT staff, on-call administrators, and local administrators in remote locations.

The multiple security layers (including Firewall, VPN, Authentication, Access control, Encryption, and more) that are at the heart of the Privileged Access Security solution offer you the most secure solution available for storing and sharing passwords in an enterprise environment.

After the CyberArk Vault has been installed and has started successfully, you can generate a new Server key on the KMES Series 3.

The Server Key is the key used to "open" the Vault, much like the key of a physical Vault. The key is required to start the Vault, after which the Server key can be removed until the Server is restarted. When the Vault is stopped, the information stored in the Vault is completely inaccessible without that key.

[2] PREREQUISITES

Supported Hardware:

- KMES Series 3, version 6.3.1.3 and above, with the *PKCS11* license enabled

Supported Operating Systems:

- Windows Server 2012 and above

Other:

- OpenSSL

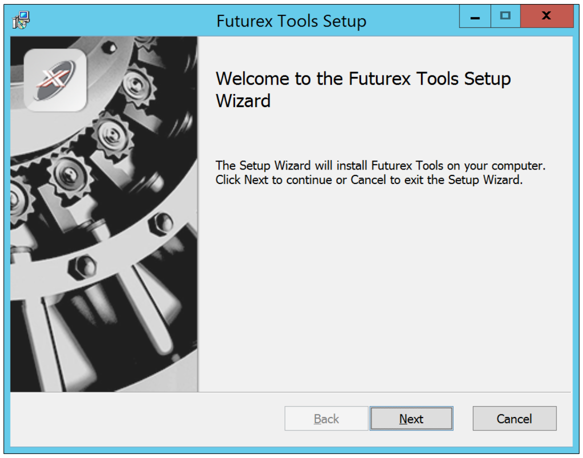
[3] INSTALL FUTUREX PKCS #11 (FXPKCS11)

In a Windows environment, the easiest way to install the Futurex PKCS #11 (FXPKCS11) module is through installing **FXTools**. FXTools can be downloaded from the Futurex Portal. In a Linux environment, you need to download a tarball of the PKCS #11 binaries from the Futurex Portal. Then, extract the *.tar* file locally where you want the application to be installed in your file system. Step by step installation instructions for both of these scenarios is provided in the following subsections.

NOTE: The Futurex PKCS #11 module needs to be installed on the computer/server where Apache HTTP Server will be installed.

[3.1] INSTRUCTIONS FOR INSTALLING THE FXPKCS11 MODULE USING FXTOOLS IN WINDOWS

- Run the FXTools installer as an administrator



Welcome to the Futurex Tools Setup Wizard

The Setup Wizard will install Futurex Tools on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

FIGURE: FUTUREX TOOLS SETUP WIZARD

By default, all tools are installed on the system. A user can overwrite and choose not to install certain modules.

- **Futurex Client Tools –**Command Line Interface (CLI) and associated SDK for both Java and C.
- **Futurex CNG Module –**The Microsoft Next Generation Cryptographic Library.
- **Futurex Cryptographic Service Provider (CSP) –**The legacy Microsoft cryptographic library.
- **Futurex EKM Module –**The Microsoft Enterprise Key Management library.
- **Futurex PKCS #11 Module –**The Futurex PKCS #11 library and associated tools.
- **Futurex Secure Access Client –**The client used to connect a Futurex Excrypt Touch to a local laptop, via USB, and a remote Futurex device.

After starting the installation, all noted services are installed. If the Futurex Secure Access Client was selected, the Futurex Excrypt Touch driver will also be installed (Note this sometimes will start minimized or in the background).

After installation is complete, all services are installed in the "*C:\Program Files\Futurex\*" directory. The CNG Module, CSP Module, EKM Module, and PKCS #11 Module all require configuration files, located in their corresponding directory with a *.cfg* extension.

NOTE: Only the HSM version of the PKCS #11 configuration file is installed. For KMES integrations, the `<HSM>` section needs to be replaced with a `<KMS>` section.

[3.2] INSTRUCTIONS FOR INSTALLING THE FXPKCS11 MODULE IN LINUX

Extract the appropriate tarball file for your specific Linux distribution in the desired working directory.

NOTE: For the Futurex PKCS #11 module to be accessible system-wide, it would need to be placed into */usr/local/bin* by an administrative user. If the module only needs to be utilized by the current user, then installing into *$HOME/bin* would be the appropriate location.

The extracted content of the *.tar* file is a single *fxpkcs11* directory. Inside of the *fxpkcs11* directory are the following files and directories (Only files/folders that are relevant to the installation process are included below):

- *fxpkcs11.cfg* -> PKCS #11 configuration file to use for HSM integrations
- *fxpkcs11-kms.cfg* -> PKCS #11 configuration file to use for KMES Series 3 integrations
- *x86/* - This folder contains the module files for 32-bit architecture
- *x64/* - This folder contains the module files for 64-bit architecture

Within the *x86* and *x64* directories are two directories. One named *OpenSSL-1.0.x* and the other named *OpenSSL-1.1.x*. Both of these OpenSSL directories contain the PKCS #11 module files, built with the respective OpenSSL versions. These files are listed below, with short descriptions of each:

- *configTest* -> Program to test configuration and connection to the HSM
- *libfxpkcs11.so* -> PKCS #11 Library File
- *PKCS11Manager* -> Program to test connection and manage the HSM through the PKCS #11 library

The *configTest* and *PKCS11Manager* programs look for the PKCS #11 configuration file in the */etc* directory. Because of this, it is necessary either to move the PKCS #11 configuration file from the */usr/local/bin/fxpkcs11* directory to the */etc* directory, or to set the FXPKCS11_CFG environment variable to point to the PKCS #11 configuration file.

NOTE: If using the KMES version of the PKCS #11 configuration file (i.e., *fxpkcs11-kms.cfg*), the file needs to be renamed to *fxpkcs11.cfg*.

[4] KMES SERIES 3 CONFIGURATION

The first half of this section covers general configurations users must make on the KMES to allow CyberArk Vault to integrate with the KMES to provide The Server Key. The second half of this section covers the steps required to configure TLS communication between the KMES and the Vault instance.

[4.1] CREATE A ROLE AND IDENTITY FOR CYBERARK WITH THE REQUIRED PERMISSIONS

A new role and identity need to be created for Vault on the KMES Series 3.

Note: In a later section, the name of this user will be configured inside of the Futurex PKCS #11 configuration file.

1. Log in to the KMES Series 3 application interface with the default Admin identities.
2. Go to the **Identity Management** menu, select **Roles,** and click the **[Add...]** button. This will pull up the **Role Editor** dialog.
3. Specify a name for the role, set the number of logins required to **1**, and navigate to the **Advanced** tab and allow authentication to the **Host API** port only. All other fields can be left as the default values.
4. Move to the **Permissions** tab and select the following permissions:
    - Cryptographic Operations > Sign, Verify, Encrypt, Decrypt
    - Keys > Add, Export
5. Click the **[OK]** button to finish creating the role.
6. Go to **Identities**, right-click anywhere on in the window and select **Add** > **Client Application**.
7. In the **Identity Editor** dialog:
    a. Under **Info**, select **Application** for the storage location, and specify a **name** for the identity.
    b. Under **Assigned Roles,** select the role you created.
    c. Under **Authentication**, configure the password.
    d. Leave all other fields as the default values and click the **[OK]** button to finish creating the identity.

[4.2] CREATE A KEY GROUP FOR CYBERARK VAULT KEYS

A key group needs to be created on the KMES Series 3 so Vault will have a place to store the encryption keys.

Note: In a later section, the name of the key group will be configured inside of the Futurex PKCS #11 configuration file.

1. Log in to the KMES Series 3 application interface with the default Admin identities.
2. Navigate to the **Key Management > Keys** menu, then right-click and select **Add > Key Group**.
3. Select **Symmetric** and **Trusted** in the Key Group Storage.
4. In the **Key Group Editor** dialog:

a. Specify a name for the key group.
b. In the **Owner group** dropdown, select the Vault role you created.
c. Click the **[Permissions]** button and give Vault role you created the **Use** permission. Click **[OK]** to save.
d. Click **[OK]** again to finish creating the key group.

[4.3] ENABLE THE HOST API COMMANDS REQUIRED FOR THE CYBERARK VAULT OPERATION

Because the Futurex PKCS #11 library will be connecting to the Host API port on the KMES, users must define which Host API commands will be enabled for execution by the FXPKCS11 library. To set the enabled commands, complete the following steps:

1. Log in to the KMES Series 3 application interface with the default Admin identities.
2. Go to **Administration > Configuration > Host API Options**, enable the commands listed below, then click **[Save]**.

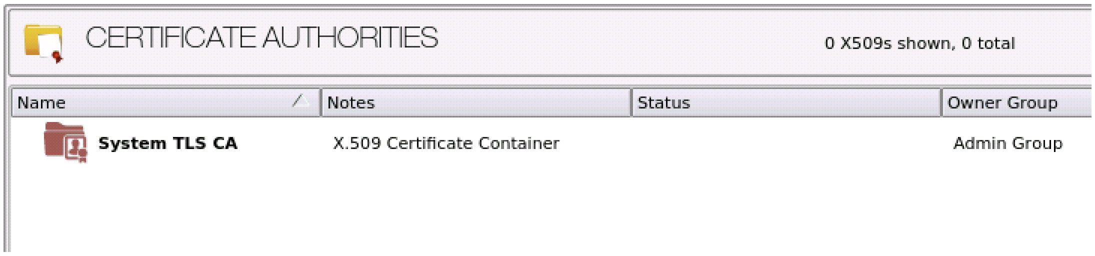
- **ECHO**: Communication Test/Retrieve Version
- **RAFA**: Filter Issuance Policy
- **RAND**: Generate Random Number
- **RKCK**: Create HSM Trusted Key
- **RKCP**: Get Command Permissions
- **RKCS**: Create Symmetric HSM Trusted Key Group
- **RKED**: Encrypt or Decrypt Data
- **RKHM**: HMAC Data
- **RKLN**: Lookup Objects
- **RKLO**: Login User
- **RKRC**: Get HSM Trusted Key

[4.4] CONFIGURE TLS COMMUNICATION BETWEEN THE KMES SERIES 3 AND THE VAULT INSTANCE

[4.4.1] Create a Certificate Authority (CA)

1. Log in to the KMES Series 3 application interface with the default Admin identities.
2. Select **PKI > Certificate Authorities** in the left menu, then click the **[Add CA...]** button at the bottom of the page.

3. In the **Certificate Authority** dialog, enter a name for the Certificate Container, leave all other fields as the default values, then click **[OK]**.
4. The Certificate Container that was just created will be listed now in the Certificate Authorities menu.

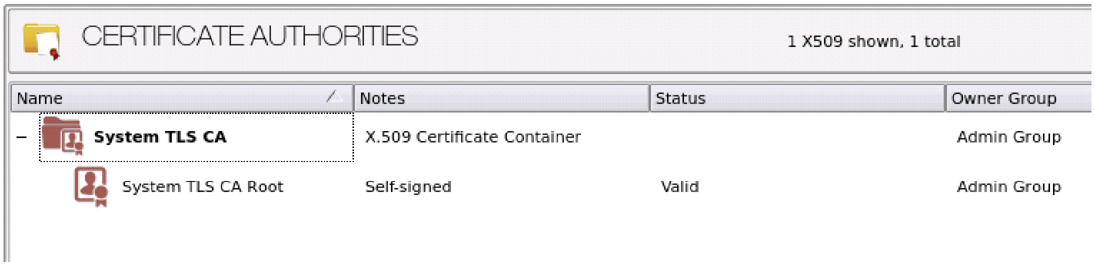


CERTIFICATE AUTHORITIES

0 X509s shown, 0 total

Name	Notes	Status	Owner Group
System TLS CA	X.509 Certificate Container		Admin Group

5. Right-click on the Certificate Container and select **Add Certificate > New Certificate...**
6. In the **Subject DN** tab, set a Common Name for the certificate, such as "System TLS CA Root".
7. In the **Basic Info** tab, leave all of the default values set
8. In the **V3 Extensions** tab, select the **Certificate Authority** profile, then click **[OK]**.
9. The root CA certificate will be listed now under the previously created Certificate Container.



CERTIFICATE AUTHORITIES

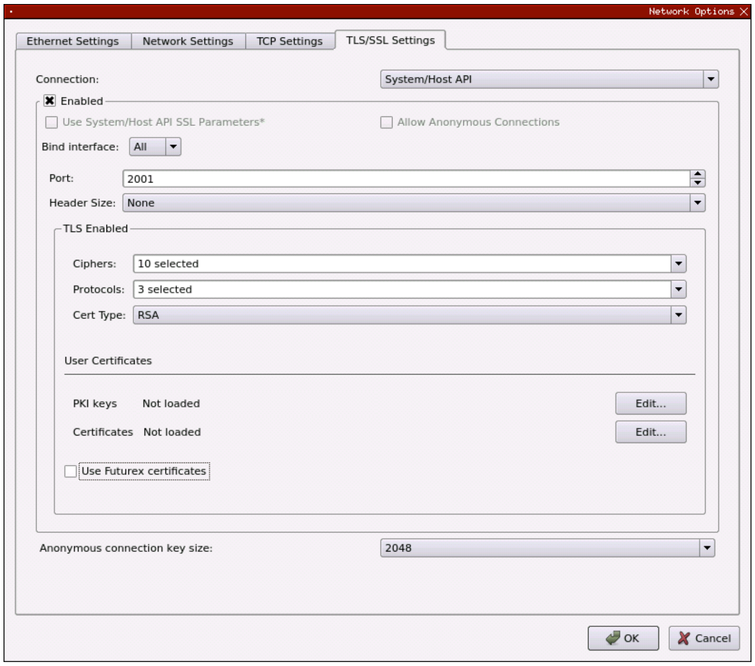
1 X509 shown, 1 total

Name	Notes	Status	Owner Group
– System TLS CA	X.509 Certificate Container		Admin Group
System TLS CA Root	Self-signed	Valid	Admin Group

[4.4.2] Generate a CSR for the System/Host API connection pair

1. Go to **Administration > Configuration > Network Options**.
2. In the **Network Options** dialog, select the **TLS/SSL Settings** tab.

3. Under the **System/Host API** connection pair, uncheck **Use Futurex certificates**, then click **[Edit...]** next to PKI keys in the User Certificates section.



Connection: System/Host API

☒ Enabled

☐ Use System/Host API SSL Parameters*

☐ Allow Anonymous Connections

Bind interface: All

Port: 2001

Header Size: None

TLS Enabled

Ciphers: 10 selected

Protocols: 3 selected

Cert Type: RSA

User Certificates

PKI keys Not loaded

Certificates Not loaded

☐ Use Futurex certificates

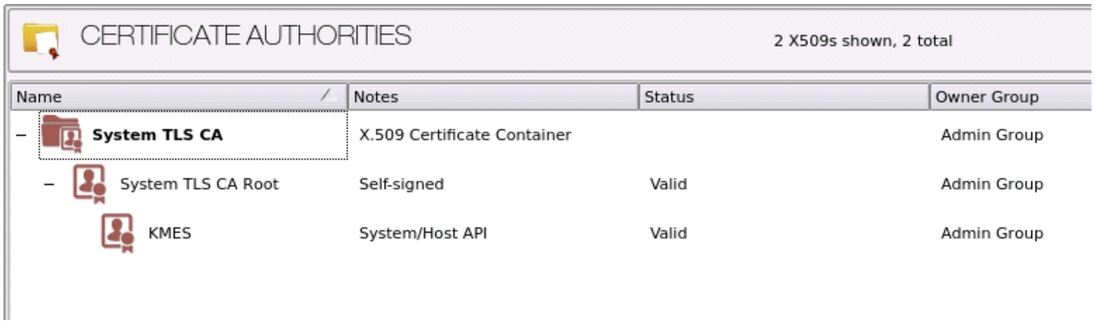
Anonymous connection key size: 2048

4. In the **Application Public Keys** dialog, click **[Generate...]**
5. There will be a warning stating that SSL will not be functional until new certificates are imported. Select **[Yes]** if you wish to continue.
6. In the **PKI Parameters** dialog, leave the default values set and click **[OK]**.
7. It should show that a PKI Key Pair is loaded now in the **Application Public Keys** dialog. If this is the case, click **[Request...]**
8. In the **Subject DN** tab, set a Common Name for the certificate, such as "KMES".
9. In the **V3 Extensions** tab, select the **TLS Server Certificate** profile.
10. In the **PKCS #10 Info** tab, select a save location for the CSR, then click **[OK]**.
11. There should be a message stating that the certificate signing request was successfully written to the file location that was selected. Click **[OK]**.
12. Click **OK** again to save the **Application Public Keys** settings.
13. In the main **Network Options** dialog, it should now show **Loaded** next to **PKI keys** for the System/Host API connection pair.

[4.4.3] Sign the System/Host API CSR

1. Go to **PKI > Certificate Authorities** menu.
2. Right-click on the root CA certificate created in section 2.1.1, then select **Add Certificate > From Request...**.

3. In the file browser, find and select the CSR that was generated for the System/Host API connection pair.
4. Once loaded, none of the settings need to be modified for the certificate. Click **[OK]**.
5. The signed System/Host API certificate should now show under the root CA certificate on the **Certificate Authorities** page.



CERTIFICATE AUTHORITIES

2 X509s shown, 2 total

Name	Notes	Status	Owner Group
– **System TLS CA**	X.509 Certificate Container		Admin Group
– System TLS CA Root	Self-signed	Valid	Admin Group
KMES	System/Host API	Valid	Admin Group

[4.4.4] Export the Root CA certificate

1. Go to **PKI > Certificate Authorities** menu.
2. Right-click on the **System TLS CA Root** certificate, then select **Export > Certificate(s)...**.
3. In the **Export Certificate** dialog, change the encoding to **PEM**, then click **[Browse...]**.
4. In the file browser, navigate to the location where you want to save the Root CA certificate. Specify "tls_ca.pem" as the name for the file, then click **[Open]**.
5. Click **[OK]**. A message box will pop up stating that the PEM file was successfully written to the location that you specified.

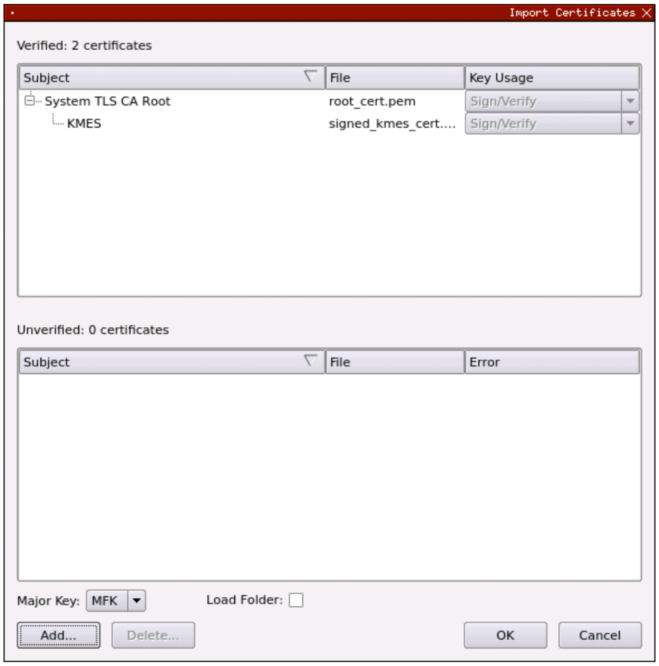
[4.4.5] Export the signed System/Host API certificate

1. Go to **PKI > Certificate Authorities** menu.
2. Right-click on the **KMES** certificate, then select **Export > Certificate(s)...**
3. In the **Export Certificate** dialog, change the encoding to **PEM**, then click **[Browse...]**
4. In the file browser, navigate to the location where you want to save the signed System/Host API certificate. Specify "tls_ca.pem" as the name for the file, then click **[Open]**.
5. Click **[OK]**. A message box will pop up stating that the PEM file was successfully written to the location that you specified.

[4.4.6] Load the exported certificates into the System/Host API connection pair

1. Go to **Administration > Configuration > Network Options**.
2. In the **Network Options** dialog, select the **TLS/SSL Settings** tab.
3. Click **[Edit...]** next to Certificates in the User Certificates section.

4. Right-click on the **System/Host API SSL CA** X.509 Certificate Container, then select **[Import...]**
5. Click **[Add...]** at the bottom of the **Import Certificates** dialog.
6. In the file browser, find and select both the root CA certificate and the signed System/Host API certificate, then click **[Open]**. The certificate chain should appear as shown below:



Import Certificates

Verified: 2 certificates

Subject	File	Key Usage
System TLS CA Root	root_cert.pem	Sign/Verify
KMES	signed_kmes_cert....	Sign/Verify

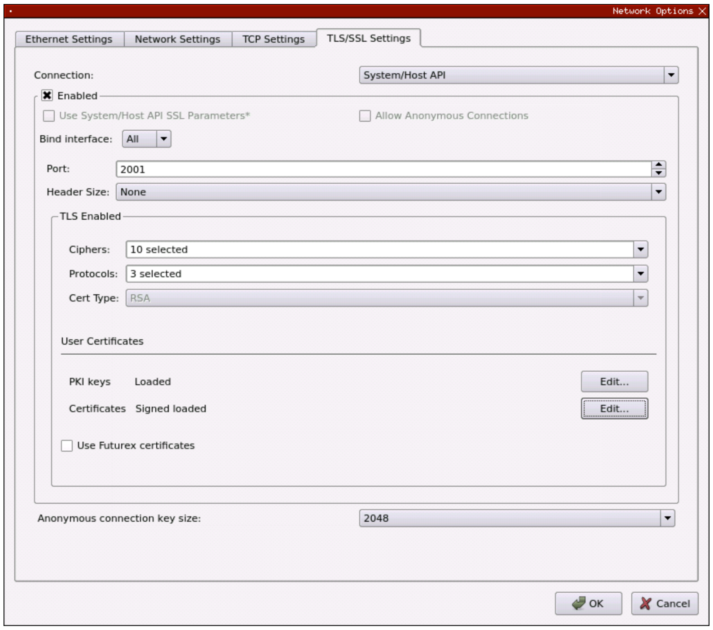
Unverified: 0 certificates

Subject	File	Error

Major Key: MFK

Load Folder: □

7. Click **[OK]** to save the changes. In the **Network Options** dialog, the System/Host API connection pair should show **Signed loaded** next to Certificates in the **User Certificates** section, as shown below:



Network Options

Connection: System/Host API

☒ Enabled

□ Use System/Host API SSL Parameters*

□ Allow Anonymous Connections

Bind interface: All

Port: 2001

Header Size: None

TLS Enabled

Ciphers: 10 selected

Protocols: 3 selected

Cert Type: RSA

User Certificates

PKI keys Loaded

Certificates Signed loaded

□ Use Futurex certificates

Anonymous connection key size: 2048

8. Click **[OK]** to save and exit the Network Options dialog.

[4.4.7] Issue a client certificate for Vault

Note: The client certificate that is being created for Vault will be configured inside of the Futurex PKCS #11 configuration file.

1. Go to **PKI > Certificate Authorities** menu.
2. Right-click on the **System TLS CA Root** certificate and select **Add Certificate > New Certificate...**.
3. In the **Subject DN** tab, set a Common Name for the certificate, such as "Vault".
4. All settings in the **Basic Info** tab should be left as the default values.
5. In the **V3 Extensions** tab, select the **TLS Client Certificate** profile, then click **[OK]**.
6. The Vault certificate will be listed now under the **System TLS CA Root** certificate.

[4.4.8] Export the Vault certificate as PKCS #12 file

Note: To be able to perform the steps below you must go to **Configuration > Options** and enable the **Allow export of certificates using passwords** option.

1. Go to **PKI > Certificate Authorities** menu.
2. Right-click on the Vault certificate, then select **Export > PKCS12...**
3. Make sure that the **Export Selected** option is selected, specify a unique name for the export file, then click **Next**.
4. Input a file password of your choosing, then click **Next**.
5. Click **[Finish]** to initiate the export.

Note: The **Vault** certificate and the Root CA certificate that was exported in section 4.4.4 both need to be moved to the computer that will be running the Vault instance. In a later section, they will be configured and used for TLS communication with the KMES Series 3.

[5] EDIT THE FUTUREX PKCS #11 CONFIGURATION FILE

[5.1] DEFINE CONNECTION INFORMATION

The *fxpkcs11.cfg* file allows the user to set the FXPKCS11 library to connect to the KMES Series 3. To edit, run a text editor as an Administrator on Windows or as root on Linux, and edit the configuration file accordingly. Most notably, the fields shown below must be set inside the **<KMS>** section (note that the entire *fxpkcs11.cfg* file is not included).

NOTE: Our PKCS #11 library expects the PKCS #11 config file to be in a certain location (*C:\Program Files\Futurex\fxpkcs11\fxpkcs11.cfg* for Windows and */etc/fxpkcs11.cfg* for Linux), but that location can be overwritten using an environment variable (FXPKCS11_CFG).

```
<KMS>
    # Which PKCS11 slot
    <SLOT>                  0                       </SLOT>

    # Login username
    <CRYPTO-OPR>            crypto1                 </CRYPTO-OPR>

    # Key group name
    #<KEYGROUP-NAME>        keygroup1               </KEYGROUP-NAME>

    # Asymmetric key group name
    <ASYM-KEYGROUP-NAME>    asymkeygroup1           </ASYM-KEYGROUP-NAME>

    # Connection information
    <ADDRESS>               10.0.8.20 </ADDRESS>
    <PROD-PORT>             2001                    </PROD-PORT>
    <PROD-TLS-ENABLED>      YES                     </PROD-TLS-ENABLED>
    <PROD-TLS-ANONYMOUS>    NO                      </PROD-TLS-ANONYMOUS>
    <PROD-TLS-CA>           /connection_certs/root_tls_cert.pem         </PROD-TLS-CA>
    <PROD-TLS-CERT>         /connection_certs/signed_fxpkcs11_tls_cert.pem       </PROD-TLS-CERT>
    <PROD-TLS-KEY>          /connection_certs/fxpkcs11_tls_privatekey.pem   </PROD-TLS-KEY>
#    <PROD-TLS-KEY-PASS>     safest                 </PROD-TLS-KEY-PASS>

    # YES = This is communicating through a Guardian
    <FX-LOAD-BALANCE>       NO                      </FX-LOAD-BALANCE>
</KMS>
```

The **<SLOT>** field can be left as the default value of 0.

In the **<CRYPTO-OPR>** field, specify the name of identity that was created on the KMES.

The **<KEYGROUP-NAME>** field can be used when an application needs to create symmetric keys on the KMES. For this integration, this field can be commented out because Apache only needs to create an asymmetric key pair on the KMES.

The **<ASYM-KEYGROUP-NAME>** field does need to be defined for this integration. The asymmetric key that Apache creates on the KMES will be added to a key group with the name specified here.

In the **<ADDRESS>** field, specify the IP of the KMES that the PKCS #11 library should connect to.

In the **<LOG-FILE>** field, set the path to the PKCS #11 log file.

In the **<PROD-PORT>** field, set the PKCS #11 library to connect to the default Host API port on the KMES, port 2001.

The **<PROD-TLS-ENABLED>** field needs to be set to "YES" because the only way to connect to the Host API port on the KMES is over TLS.

The **<PROD-TLS-ANONYMOUS>** field defines whether the PKCS #11 library will be authenticating to the KMES or not. Since we're connecting to the Host API port using mutual authentication, this value should be set to "NO".

The location of the CA certificate/s needs to be defined with one or more instances of the **<PROD-TLS-CA>** tag. In this example, there is only one CA certificate.

The location of the signed client certificate needs to be defined with the **<PROD-TLS-CERT>** tag.

The **<PROD-TLS-KEY>** tag defines the location of the client private key. Supported formats for the TLS private key are PKCS #1 clear private keys, PKCS #8 encrypted private keys, or a PKCS #12 file that contains the private key and certificates encrypted under the password specified in the **<PROD-TLS-KEY-PASS>** field.

If a Guardian is being used to manage KMES Series 3 devices in a cluster, the **<FX-LOAD-BALANCE>** field must be defined as “YES”. If a Guardian is not being used it should be set to “NO”.

Once the *fxpkcs11.cfg* file is edited, run the *PKCS11Manager* file to test the connection against the KMES, and check the *fxpkcs11.log* for errors and information. For more information, refer to the Futurex PKCS #11 technical reference found on the Futurex Portal.

[6] STEPS TO CONFIGURE THE FUTUREX PKCS #11 LIBRARY WITH CYBERARK VAULT

NOTE: Before proceeding with the steps that follow, the CyberArk PAS solution needs to be installed. For instructions on how to install the CyberArk PAS solution, please refer to CyberArk's online documentation at the following url: https://docs.cyberark.com/Product-Doc/OnlineHelp/PAS/Latest/en/Content/PAS%20INST/InstallationOverview.htm

After the CyberArk Vault has been installed and has started successfully, you can generate a new Server key on the KMES Series 3.

The Server Key is the key used to “open” the Vault, much like the key of a physical Vault. The key is required to start the Vault, after which the Server key can be removed until the Server is restarted. When the Vault is stopped, the information stored in the Vault is completely inaccessible without that key.

[6.1] INITIAL VAULT CONFIGURATIONS

1. To use an KMES that is attached to the network, configure the Firewall in order to allow communication to the KMES device. In *DBParm.ini*, configure the AllowNonStandardFWAddresses parameter to open the Firewall and allow access to the HSM device, as shown in the following example:

```
AllowNonStandardFWAddresses=[HSM-IP],Yes,1024:inbound/tcp,1024:outbound/tcp
```

NOTE: If utilizing a cloud KMES that is accessible through the internet (rather than a physical KMES connected to the local network), AllowNonStandardFWAddresses should **not** be defined in the *DBParm.ini* file.

2. Configure the PKCS#11 provider DLL and specify it in the PKCS11ProviderPath parameter in DBParm.ini, as shown in the following example:

```
PKCS11ProviderPath=<path to PKCS#11 provider dll>
```

3. Save DBParm.ini and close it.

4. Define the PIN/passphrase to be used by the Vault when accessing an KMES device: From a command line, run the following command, specifying your own PIN/passcode that will be used to access the Server key. The PIN/passcode cannot begin with "/":

NOTE: The "hsmpincode" is the password of the Identity created on the KMES Series 3 in step 7.6.

```
CAVaultManager SecureSecretFiles /SecretType HSM /Secret <hsmpincode>
```

Open DBParm.ini and make sure that the HSMPinCode parameter was added with the encrypted value of the PIN/passcode.

5. Restart the PrivateArk Server to apply the new firewall rules.

6. Shutdown the PrivateArk Server.

[6.2] LOAD THE SERVER KEY INTO THE KMES SERIES 3

The following process installs and stores the Server key on the KMES Series 3. Once this process is complete, the Server key is stored as non-exportable key on the KMES and will be used by the Vault.

Generate the server key in the KMES

1. Make sure that the Vault Server is not running.

2. Run the **CAVaultManager** command to generate the server key on the KMES:

```
CAVaultManager.exe GenerateKeyOnHSM /ServerKey
```

The above command will generate a new key for the Vault server and store it in the KMES device and will return the key generation keyword. For example: HSM#5

Each time a key generation is created, the keyword allocated is one number higher than the current server key generation specified in DBParm.ini. In order to create additional key generations successfully, users have to manually delete the first generation of the server key; otherwise, an error will be returned. If the ServerKey parameter in the CAVaultManager command specifies a path instead of an HSM keyword, the first key generation will be created, i.e., HSM#1.

3. Re-encrypt the Vault data and metadata with the newly generated keys on the KMES.

- Run the ChangeServerKeys command to change the encryption keys that will be used for the Vault server.

```
ChangeServerKeys PathToKeys PathToEmergencyFile HSMKeyword
```

For example, the following command will re-encrypt the Vault data and metadata with the encryption keys in ‘K:\PrivateArk\Keys’, and the ‘HSM#1’ key will be used as the server key.

```
ChangeServerKeys K:\PrivateArk\Keys K:\PrivateArk\Keys\VaultEmergency.pass HSM#1
```

4. Open DBParm.ini and in the ServerKey parameter specify the value of the key generation version that was generated and specified in the output of the CAVaultManager command above, as shown in the following example.

```
ServerKey=HSM#1
```

5. Start the Vault server and make sure you can log onto the Vault.

APPENDIX A: XCEPTIONAL SUPPORT

In today's high-paced environment, we know you are looking for timely and effective resolutions for your mission-critical needs. That is why our Xceptional Support Team does whatever it takes to ensure you have the best experience and support possible. Every time. Guaranteed.

- 24x7x365 mission critical support
- Level 1 to level 3 support
- Extremely knowledgeable subject matter experts

At Futurex, we strive to supply you with the latest data encryption innovations as well as our best-in-class support services. Our Xceptional Support Team goes above and beyond to meet your needs and provide you with exclusive services that you cannot find anywhere else in the industry.

- Technical Services
- Onsite Training
- Virtual Training
- Customized Consulting
- Customized Software Solutions
- Secure Key Generation, Printing, and Mailing
- Remote Key Injection
- Certificate Authority Services

Toll-Free: 1-800-251-5112

E-mail: support@futurex.com

ENGINEERING CAMPUS

864 Old Boerne Road

Bulverde, Texas, USA 78163

Phone: +1 830-980-9782

+1 830-438-8782

E-mail: info@futurex.com

XCEPTIONAL SUPPORT

24x7x365

Toll-Free: 1-800-251-5112

E-mail: support@futurex.com

SOLUTIONS ARCHITECT

E-mail: solutions@futurex.com