VENAFI TRUST PROTECTION PLATFORM

Integration Guide

Applicable Devices:

Vectera Plus

TABLE OF CONTENTS

[1] DOCUMENT INFORMATION

[1.1] DOCUMENT OVERVIEW

The purpose of this document is to provide information regarding the configuration of Futurex HSMs with Venafi's Trust Protection Platform using PKCS #11 libraries. For additional questions related to your HSM, see the relevant administrator's guide.

[1.2] APPLICATION DESCRIPTION

From Venafi's Trust Protection Platform datasheet: "Venafi's Trust Protection Platform manages, secures and protects keys and certificates, delivering an enterprise-grade platform that provides enterprise-wide security, operational efficiency and organizational compliance."

[1.3] GUARDIAN INTEGRATION

The Guardian Series 3 introduces mission-critical viability to core cryptographic infrastructure, including:

- Centralize device management
- Eliminates points of failure
- Distribute transaction loads
- Group-specific function blocking
- User-defined grouping systems

Please see applicable guide for configuring HSMs with the Guardian Series 3.

[2] PREREQUISITES

Supported Hardware:

- Vectera Plus, 6.7.x.x and above

Supported Operating Systems:

- Windows 7 and above

Other:

- OpenSSL

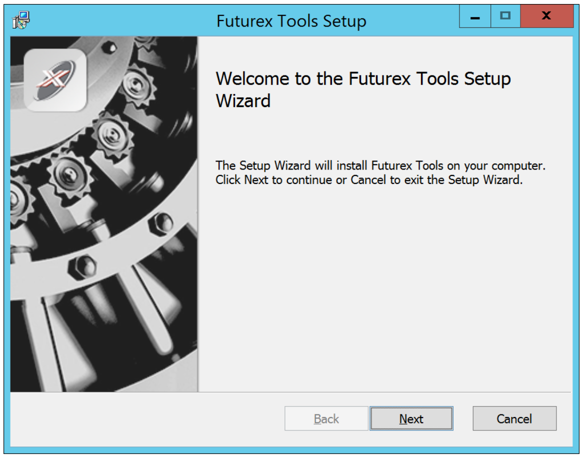
[3] INSTALL FUTUREX PKCS #11 (FXPKCS11)

In a Windows environment, the easiest way to install the **Futurex PKCS #11 (FXPKCS11)** module is with **Futurex Tools (FXTools)**. You can download FXTools from the Futurex Portal. Step-by-step installation instructions are provided below.

Note: Install FXPKCS11 on the same computer as the application integrating with the Vectera Plus HSM.

[3.1] INSTALLING THE FXPKCS11 MODULE USING FXTOOLS IN WINDOWS

Run the Futurex Tools installer as an administrator and follow the prompts in the setup wizard to complete the installation.



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

The Setup Wizard installs all tools on the system by default. You can override the defaults and choose not to install certain modules. The installation provides the following services:

- **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**- A client used to connect a Futurex Excrypt Touch to a local laptop through USB, which can then connect to a remote Futurex device.

If the Futurex Secure Access Client was selected, the process will also install the Futurex Excrypt Touch driver, which might start minimized or in the background.

After the installation completes, 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, which are located in their corresponding directory with a .cfg extension. In addition, the installation registers the CNG and CSP Modules in the Windows Registry (HKEY_LOCAL_ MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider), and installs them in the C:\Windows\System32\ directory.

[4] INSTALL EXCRYPT MANAGER (IF USING WINDOWS)

Sections 4 and 5 of this integration guide cover the installation of Excrypt Manager and FXCLI. Excrypt Manager is a Windows application that provides a GUI-based method for configuring the HSM, while FXCLI provides a command-line-based method for configuring the HSM and can be installed on all platforms.

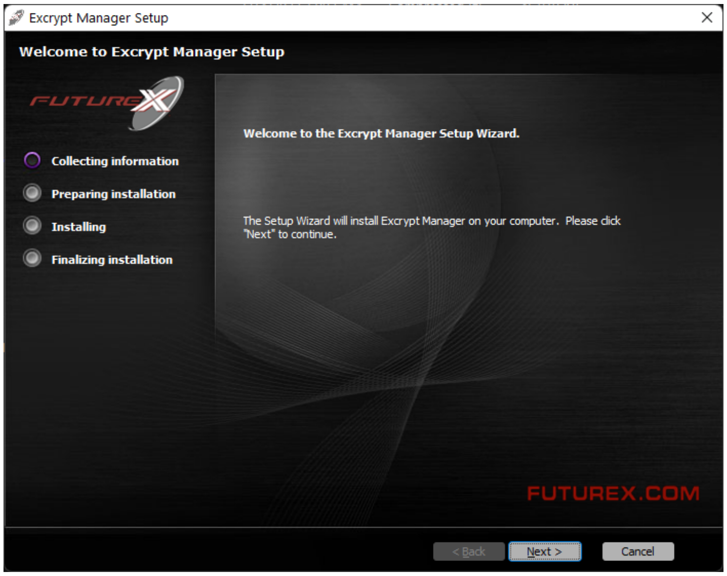
Note: If you will be configuring the Vectera Plus from a Linux computer, you can skip this section. If you will be configuring the Vectera Plus from a Windows computer, installing FXCLI in the next section is still required because FXCLI is the only method that can be used to configure TLS certificates in section 6.7.

Note: Install Excrypt Manager on the workstation you will use to configure the HSM.

Note: If you plan to use a Virtual HSM for the integration, all configurations will need to be performed using either FXCLI, the Excrypt Touch, or the Guardian Series 3.

Note: The Excrypt Manager version must be from the 4.4.x branch or later to be compatible with the HSM firmware, which must be 6.7.x.x or later.

To install Excrypt Manager, run the Excrypt Manager installer as an administrator and follow the prompts in the setup wizard to complete the installation.



Welcome to the Excrypt Manager Setup Wizard.

The Setup Wizard will install Excrypt Manager on your computer. Please click "Next" to continue.

FIGURE: EXCRYPT MANAGER SETUP WIZARD

The installation wizard prompts you to specify where you want to install Excrypt Manager. The default location is C:\Program Files\Futurex\Excrypt Manager\. After choosing a location, select **[Install]**.

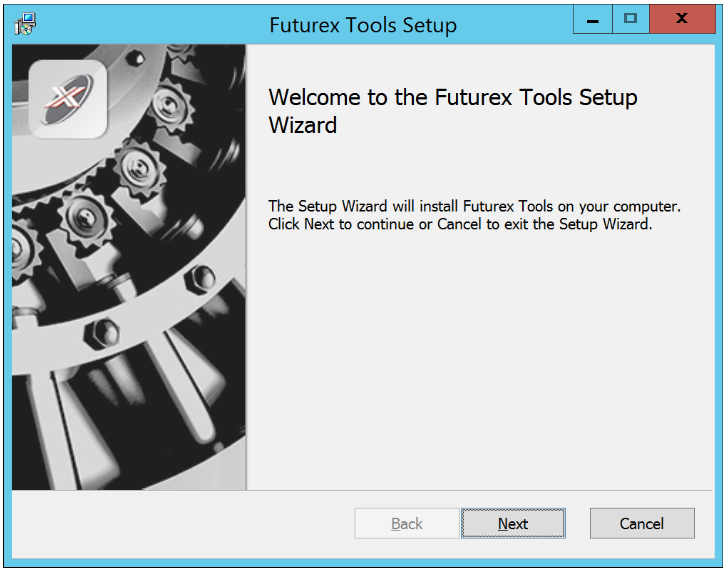
[5] INSTALL FUTUREX COMMAND LINE INTERFACE (FXCLI)

Note: Install FXCLI on the workstation you will use to configure the HSM.

[5.1] INSTALLING FXCLI IN WINDOWS

As mentioned in section 3, the FXTools installation package includes Futurex Client Tools (FXCLI). Similar to the Futurex PKCS #11 (FXPKCS11) module, the easiest way to install FXCLI on Windows is by installing FXTools. You can download FXTools from the Futurex Portal.

To install FXCLI, run the Futurex Tools installer as an administrator and follow the prompts in the setup wizard to complete the installation.



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

The setup wizard installs all tools on the system by default. You can override the defaults and choose not to install certain modules. The installation provides the following services:

- **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**:A client used to connect a Futurex Excrypt Touch to a local laptop through USB, which can then connect to a remote Futurex device.

[5.2] INSTALLING FXCLI IN LINUX

Download FXCLI

You can download the appropriate FXCLI package files for your system from the Futurex Portal.

If the system is **64-bit**, select from the files marked **amd64**. If the system is **32-bit**, select from the files marked **i386**.

If running an OpenSSL version in the **1.0.x** branch, select from the files marked **ssl1.0**. If running an OpenSSL version in the **1.1.x** branch, select from the files marked **ssl1.1**.

Futurex offers the following features for FXCLI:

- Java Software Development Kit (**java**)
- HSM command line interface (**cli-hsm**)
- KMES command line interface (**cli-kmes**)
- Software Development Kit headers (**devel**)
- YAML parser used to parse bash output (**cli-fxparse**)

Install FXCLI

To install an rpm package, run the following command in a terminal:

```
$ sudo rpm -ivh [fxcl-xxxx.rpm]
```

To install a deb package, run the following command in a terminal:

```
$ sudo dpkg -i [fxcl-xxxx.deb]
```

Running FXCLI

To enter the HSM FXCLI prompt, run the following command in a terminal:

```
$ fxcli-hsm
```

After entering the FXCLI prompt, you can run **help** to list all of the available FXCLI commands.

[6] CONFIGURE THE VECTERA PLUS

To establish a connection between the Futurex PKCS #11 library and the Vectera Plus, perform the following configuration steps:

Note: You can complete all of the steps in this section using either Excrypt Manager or FXCLI (except for section 6.7.2, which can only be completed using FXCLI). Optionally, you can complete steps 4 through 6 using the Guardian Series 3 (Please refer to the applicable guide for configuring HSMs for PKCS #11 integrations using the Guardian Series 3).

1. Connect to the HSM through the front USB port. (**Note:** If you are using a virtual HSM for the integration, you must connect to it over the network through FXCLI, the Excrypt Touch, or the Guardian Series 3):
    a. Connecting via Excrypt Manager
    b. Connecting via FXCLI
2. Validate that the correct features are enabled on the HSM.
3. Set up the network configuration.
4. Load the Futurex FTK.
5. Configure a Transaction Processing connection and create a new Application Partition.
6. Create a new identity that has access to the newly created Application Partition.
7. Configure TLS Authentication by using one of the following options:
    a. Enable server-side authentication.
    b. Create client certificates for mutual authentication.

Each of these action items is detailed in the following subsections.

[6.1] CONNECT TO THE HSM THROUGH THE FRONT USB PORT

Note: For both Excrypt Manager and FXCLI you need to connect your laptop to the front USB port on the HSM.

Connecting through Excrypt Manager

1. Open Excrypt Manager and click **[Refresh]** in the lower right-hand side of the Connection menu. Then, select **USB Connection** and click **[Connect]**.
2. Log in with both default Admin identities.
3. You must change the default Admin passwords for both of your default Admin identities (**Admin1** and **Admin2**) to load the major keys onto the HSM. To do so via Excrypt Manager, open the **Identity Management** menu, select the first default Admin identity (**Admin1**), and select **[Change Password…]**. Enter the old password and enter the new password twice. Select **[OK]**. Perform the same steps for the second default Admin identity (**Admin2**).

Connecting through FXCLI

1. Start the FXCLI application and run the following commands:

```
$ connect usb
$ login user
```

Note: The **login** command prompts for the username and password. You must run the command twice because you must login with both default Admin identities.

2. You must change the default Admin passwords for both of your default Admin Identities in order to load the major keys onto the HSM. Use the following FXCLI commands to change the passwords for each default Admin Identity.

```
$ user change-password -u Admin1
$ user change-password -u Admin2
```

Note: The preceding **user change-password** commands prompt you to enter the old and new passwords.

[6.2] REQUIRED FEATURES IN HSM

To establish a connection between the Futurex PKCS #11 Library and the Vectera Plus, the HSM must be configured with the following features:

- **PKCS #11** > *Enabled*.
- **Command Primary Mode** > *General Purpose* (GP).

Note: For additional information about how to update features on your HSM, refer to the **"Download Feature Request File"** section of the Vectera Plus user guide.

Note: Setting the **Command Primary Mode** on the HSM to *General Purpose (GP)* enables the option to create the FTK major key in the HSM. This key is required to be able to use the Futurex PKCS #11 library to communicate with the HSM. For detailed information about how to load major keys on the HSM, refer to the Vectera Plus user guide.

[6.3] NETWORK CONFIGURATION (SETTING THE HSM IP ADDRESS)

Note: For this step you need to be logged in with an identity that has a role with permissions **Communication:Network Settings**. You can use the default Administrator role and Admin identities.

Excrypt Manager

1. Navigate to the **Configuration** menu and modify the IP configuration as required.

FXCLI

1. Run the **network interface modify** FXCLI command to set an IP for the HSM. An example is provided below to show the command syntax:

```
$ network interface modify --interface Ethernet1 --ip 10.221.0.10 --netmask 255.255.255.0 --
gateway 10.221.0.1
```

Note: At this point during the HSM configuration, consider the following:

- You can complete the remaining HSM configurations in this section using the Guardian Series 3 (see the applicable guide for configuring HSMs for PKCS #11 integrations using the Guardian Series 3), except for the final subsection, which covers creating connection certificates for mutual authentication.
- If you are performing the configuration on the HSM directly right now, but plan to add the HSM to a Guardian later, you might have to synchronize the HSM after you add it to a Device Group on the Guardian.
- If your use-case requires configuration through a CLI, then you should manage the HSMs directly.

[6.4] LOAD FUTUREX KEY (FTK)

Note: For this step you need to be logged in with an identity that has a role with permissions **Major Keys:Load**. You can use the default Administrator role and Admin identities.

The FTK wraps all keys stored on the HSM used with PKCS #11. If using multiple HSMs in a cluster, you can use the same FTK for syncing HSMs. An HSM must have an FTK before you can use it with PKCS #11.

Excrypt Manager

1. Navigate to the **Key Management** menu, then select the **Load** button for the FTK in the Major Keys section. You can load keys loaded that are XOR'd together, M-of-N fragments, or generated. If this is the first HSM in a cluster, we recommend you generate the key and save to smart cards as M-of-N fragments.

FXCLI

1. Run the following **majorkey** FXCLI commands to load an FTK into an HSM. You must generate a random FTK if this is the first HSM you are setting up. Optionally, you can also load an FTK onto smart cards simultaneously with the **-m** and **-n** flags, as shown in the following example:

   ```
   $ majorkey random --ftk -m [number_from_2_to_9] -n [number_from_2_to_9]
   ```

   If it is a second HSM you're setting up in a cluster, load the FTK from smart cards with the following command:

   ```
   $ majorkey recombine --key ftk
   ```

[6.5] CONFIGURE A TRANSACTION PROCESSING CONNECTION AND CREATE AN APPLICATION PARTITION

Note: For this step you need to be logged in with an identity that has a role with permissions **Role:Add**, **Role:Assign All Permissions**, **Role:Modify**, **Keys:All Slots**, and **Command Settings:Excrypt**. You can use the default Administrator role and Admin identities.

Note: For the purposes of this integration guide, the terms *Application Partition* and *Role* are synonymous.

[6.5.1] Configure a Transaction Processing connection

Before an application logs in to the HSM with an authenticated user, it first connects through a Transaction Processing connection to the **Transaction Processing** Application Partition. For this reason, you must take steps to harden this Application Partition. The following items need to be configured for the Transaction Processing partition:

- It should not have access to the **All Slots** permissions.
- It should not have access to any key slots.
- Only the PKCS #11 communication commands should be enabled.

Excrypt Manager

1. Navigate to the **Application Partitions** menu, select the **Transaction Processing** Application Partition, and click **[Modify...]**.
2. In the **Permissions** tab, leave the top-level **Keys** permission checked, but uncheck the **All Slots** sub permission.
3. In the **Key Slots** tab, ensure that the settings do not specify key ranges. By default, the Transaction Processing Application Partition has access to the entire range of key slots on the HSM.
4. In the **Commands** tab, make sure that only the following PKCS #11 communication commands are enabled:
    - **ECHO**: Communication Test/Retrieve Version
    - **PRMD**: Retrieve HSM restrictions
    - **RAND**: Generate random data
    - **HASH**: Retrieve device serial
    - **GPKM**: Retrieve key table information
    - **GPKS**: General purpose key settings get/change
    - **GPKR**: General purpose key settings get (read-only)

FXCLI

1. Run the following **role modify** FXCLI commands to remove all permissions and key ranges that are currently assigned to the **Transaction Processing** role and enable only the PKCS #11 communication commands:

    Note: The **Transaction Processing** role was previously referred to as the **Anonymous** role. That is why *Anonymous* is specified in the name field in the commands below.

    ```
    $ role modify --name Anonymous --clear-perms --clear-key-ranges
    ```

    ```
    $ role modify --name Anonymous --add-perm "Keys" --add-perm Excrypt:ECHO --add-perm
    Excrypt:PRMD --add-perm Excrypt:RAND --add-perm Excrypt:HASH --add-perm Excrypt:GPKM --add-
    perm Excrypt:GPKS --add-perm Excrypt:GPKR
    ```

[6.5.2] Create an Application Partition

To segregate applications on the HSM, you must create an Application Partition specifically for your use case. Application partitions are used to segment the permissions and keys on an HSM between applications. The following steps outline the process for creating and configuring a new application partition.

Excrypt Manager

1. Navigate to the **Application Partitions** menu and select **[Add...]**.
2. In the **Basic Information** tab, configure all of the fields as follows:
    a. For **Role Name**, specify any name that you would like for this new Application Partition.
    b. Set **Logins Required** to *1*.
    c. Set **Ports** to *Prod*.
    d. Configure **Connection Sources** to *Ethernet*.
    e. Leave **Managed Roles** blank because you specify the exact Permissions, Key Slots, and Commands for this Application Partition or Role to have access to.
    f. Set **Use Dual Factor** to *Never*.
    g. Leave **Upgrade Permissions** unchecked.
3. In the **Permissions** tab, select the following key permissions:
    - **Keys**
    - **Authorized** (allows for keys that require login)
    - **Import PKI** (allows trusting an external PKI. Generally not recommended, but some applications use this to allow for PKI symmetric key wrapping.)
    - **No Usage Wrap** (allows for interoperable key wrapping without defining key usage as part of the wrapped key. Use this only if you want to exchange keys with external entities or use the HSM to wrap externally used keys.)
4. In the **Key Slots** tab, we recommend you create a range of 1000 total keys that do not overlap with another Application Partition. Within the specified range, you should have ranges for both symmetric and asymmetric keys. If the application requires more keys, configure accordingly.
5. Based on application requirements, particular functions need to be enabled on the Application Partition to use the HSMs functionality. The commands that need to be enabled for the Venafi TPP integration are listed below. These can be enabled in the **Commands** tab.

PKCS #11 Communication Commands

- **ECHO**: Communication Test/Retrieve Version
- **RAND**: Generate random data
- **HASH**: Retrieve device serial
- **GPKM**: Retrieve key table information
- **GPKR**: General purpose key settings get (read-only)
- **GPKS**: General purpose key settings get/change
- **TIME**: Get/set the HSM internal clock

Key Operations Commands

- **ASYL**: Load asymmetric key into key table
- **GECC**: Generate an ECC Key Pair
- **GPGS**: General purpose generate symmetric key
- **GRSA**: Generate RSA Private and Public Key
- **LRSA**: Load key into RSA Key Table
- **RSAC**: General purpose convert clear DER encoded RSA key to major key cryptogram

Interoperable Key Wrapping

- **GPKW**: General purpose key wrap (unrestricted)
- **GPWB**: General purpose key wrap with key strength bypass

Data Encryption Commands

- **GPSD**: General purpose symmetric decrypt
- **GPSE**: General purpose symmetric encrypt

Signing Commands

- **ASYS**: Generate a Signature Using a Private Key
- **GPSV**: General purpose data sign and verify

FXCLI

1. Run the following **role** FXCLI commands to create the new Application Partition and enable all required functions:

```
$ role add --name Role_Name --application --key-range (0,999) --perm "Keys:Authorized" --perm
"Keys:Import PKI" --perm "Keys:No Usage Wrap"
```

```
$ role modify --name [role_name] --clear-perms --add-perm Excrypt:ECHO --add-perm Excrypt:RAND
--add-perm Excrypt:HASH --add-perm Excrypt:GPKM --add-perm Excrypt:GPKR --add-perm
Excrypt:GPKS --add-perm Excrypt:TIME --add-perm Excrypt:ASYL --add-perm Excrypt:GECC --add-
perm Excrypt:GPGS --add-perm Excrypt:GRSA --add-perm Excrypt:LRSA --add-perm Excrypt:RSAC --
add-perm Excrypt:GPKW --add-perm Excrypt:GPWB --add-perm Excrypt:GPSD --add-perm Excrypt:GPSE
--add-perm Excrypt:ASYS --add-perm Excrypt:GPSV
```

[6.6] CREATE A NEW IDENTITY AND ASSOCIATE IT WITH THE NEWLY CREATED APPLICATION PARTITION

Note: For this step you need to be logged in with an identity that has a role with the **Identity:Add** permission. You can use the default Administrator role and Admin identities.

Excrypt Manager

1. Navigate to the **Identity Management** menu and select **[Add...]**.
2. Specify a name for the new identity and open the **Roles** drop-down menu to select the name of the previously created Application Partition. This associates the new identity with the Application Partition that you created.

FXCLI

1. Run the **identity add** FXCLI command to create a new identity and associate it with the Application Partition/Role that you created:

```
$ identity add --name Identity_Name --role Role_Name --password safest
```

You must set the name of this identity in the fxpkcs11.cfg file, in the following section:

```
#HSM crypto operator identity name
<CRYPTO-OPR>     [insert name of identity that you created]     </CRYPTO-OPR>

# Production connection
<PROD-ENABLED>    YES        </PROD-ENABLED>
<PROD-PORT>        9100        </PROD-PORT>
```

[6.7] CONFIGURE TLS AUTHENTICATION

Note: For this step you need to be logged in with an identity that has a role with permissions **Keys:All Slots**, **Management Commands:Certificates**, **Management Commands:Keys, Security:TLS Sign**, and **TLS Settings:Upload Key**. You can use the default Administrator role and Admin identities.

[6.7.1] Enable server-side authentication (option 1)

Futurex recommends mutually authenticating to the HSM using client certificates, but the Vectera Plus also supports server-side authentication. The following steps outline the process for enabling server-side authentication.

Excrypt Manager

1. Navigate to the **SSL/TLS Setup** menu. Then, select the **Excrypt Port** in the Connection Pair dropdown, check the **Allow Anonymous** box, and click **[Save]**.

FXCLI

1. Run the **tls-ports set** FXCLI command to enable server-side authentication with the **Allow Anonymous** SSL/TLS setting:

```
$ tls-ports set -p "Excrypt Port" --anon
```

[6.7.2] Create Connection Certificates for mutual authentication (option 2)

As mentioned previously, Futurex recommends mutually authenticating to the HSM using client certificates, and the system enforces mutual authentication by default. In the following example, FXCLI generates a CA which is used to sign the HSM server certificate and a client certificate. The client keys and CSR are generated using OpenSSL.

Note:

- For this example, you must connect the computer that is running FXCLI to the front USB port of the HSM.
- If you do not specify a file path for commands that create an output file, FXCLI saves the file to the current working directory.
- Using user-generated certificates requires you to load a PMK on the HSM.
- If you run **help** by itself, a full list of available commands displays. You can see all of the available options for any given command by running the command name followed by **help**.

1. Enter the FXCLI prompt by running **fxcli-hsm** in a terminal.
2. Perform the following steps to create connection certificates for mutual authentication:

```
# Connect your laptop to the HSM via the USB port on the front, then run this command.
$ connect usb
```

```
# Log in with both default Admin identities. This command will prompt for the username and
password. You will need to run this command twice.
$ login user
```

```
# Generate a TLS CA and store it in an available key slot on the HSM
$ generate --algo RSA --bits 2048 --usage mak --name TlsCaKeyPair --slot next
```

```
# Create a root certificate
$ x509 sign \
    --private-slot TlsCaKeyPair \
    --key-usage DigitalSignature --key-usage KeyCertSign \
    --ca true --pathlen 0 \
    --dn 'O=Futurex\CN=Root' \
    --out TlsCa.pem
```

```
# Generate the server keys for the HSM
$ tls-ports request --pair "Excrypt Port" --file production.csr --pki-algo RSA
```

```
# Sign the server CSR with the newly created TLS CA
$ x509 sign \
    --private-slot TlsCaKeyPair \
    --issuer TlsCa.pem \
    --csr production.csr \
    --eku Server --key-usage DigitalSignature --key-usage KeyAgreement \
    --ca false \
    --dn 'O=Futurex\CN=Production' \
    --out TlsProduction.pem
```

```
# Push the signed server PKI to the production port on the HSM
$ tls-ports set --pair "Excrypt Port" \
    --enable \
    --pki-source Generated \
    --clear-pki \
    --ca TlsCa.pem \
    --cert TlsProduction.pem \
    --no-anon
```

3. Run the following OpenSSL commands from Windows PowerShell rather than from the FXCLI program to generate client keys and CSR:

```
# Generate the client keys
$ openssl genrsa -out privatekey.pem 2048
```

```
# Generate a client CSR
$ openssl req -new -key privatekey.pem -out ClientPki.csr -days 365
```

4. Using FXCLI, sign the CSR that was just generated using OpenSSL.

```
# Sign the client CSR under the root certificate that was created
$ x509 sign  \
 --private-slot TlsCaKeyPair \
 --issuer TlsCa.pem \
 --csr ClientPki.csr \
 --eku Client --key-usage DigitalSignature --key-usage KeyAgreement \
 --dn 'O=Futurex\CN=Client' \
 --out SignedPki.pem
```

5. Run the remaining commands from Windows PowerShell:

```
# Use OpenSSL to create a PKCS #12 file that can be used to authenticate, as a client, using
the Futurex PKCS #11 library
$ openssl pkcs12 -export -inkey privatekey.pem -in SignedPki.pem -certfile TlsCa.pem -out
PKI.p12
```

[7] EDIT THE FUTUREX PKCS #11 CONFIGURATION FILE

The Futurex PKCS #11 configuration file (i.e., fxpkcs11.cfg) is used by the Futurex PKCS #11 library to connect to the HSM. It enables the user to modify certain configurations and set connection details. This section covers the **<HSM>** portion of the FXPKCS11 config file, where the connection details are set.

Note: By default, the FXPKCS11 library looks for the configuration file at C:\Program Files\Futurex\fxpkcs11\fxpkcs11.cfg for Windows and /etc/fxpkcs11.cfg for Linux. Alternatively, the FXPKCS11_ CFG environment variable can be set to the location of the fxpkcs11.cfg file.

Open the fxpkcs11.cfg file in a text editor as an administrator and edit it accordingly.

```
<HSM>
    # Which PKCS11 slot
    <SLOT>                  0                       </SLOT>
    <LABEL>                 Futurex                 </LABEL>

    # HSM crypto operator user name
    <CRYPTO-OPR>            [identity_name]            </CRYPTO-OPR>
    # Automatically login on session open
    #<CRYPTO-OPR-PASS>       [identity_password]        </CRYPTO-OPR-PASS>

    # Connection information
    <ADDRESS>               10.0.8.30     </ADDRESS>
    <PROD-PORT>             9100                    </PROD-PORT>
    <PROD-TLS-ENABLED>      YES                     </PROD-TLS-ENABLED>
    <PROD-TLS-ANONYMOUS>    NO                      </PROD-TLS-ANONYMOUS>
#    <PROD-TLS-CA>           /home/user/tls/root.pem        </PROD-TLS-CA>
#    <PROD-TLS-CA>           /home/user/tls/sub1.pem     </PROD-TLS-CA>
#    <PROD-TLS-CA>           /home/user/tls/sub2.pem     </PROD-TLS-CA>
    <PROD-TLS-KEY>          /home/user/tls/PKI.p12       </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>
</HSM>
```

The **<SLOT>** and **<LABEL>** fields specify PKCS11 slot 0 and the label *Futurex*.

The **<CRYPTO-OPR>** field specifies the name of the identity you created for the Application Partition.

The **<CRYPTO-OPR-PASS>** field specifies the password of the identity configured in the **<CRYPTO-OPR>** field. This can be used to log the application into the HSM automatically, if required.

The **<ADDRESS>** field specifies the IP address of the HSM that the FXPKCS11 library should connect to.

The **<PROD-PORT>** field specifies the port number of the HSM that the FXPKCS11 library should connect to.

The **<PROD-TLS-ANONYMOUS>** field defines whether the FXPKCS11 library authenticates to the server.

The **<PROD-TLS-KEY>** field 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.

Because a PKCS #12 file is defined in the **<PROD-TLS-KEY>** field in this example, the signed client cert does not need to be defined with the **<PROD-TLS-CERT>** tag, nor do the CA cert/s need to be defined with one or more instances of the **<PROD-TLS-CA>** tag.

If you use Guardian to manage HSMs in a cluster, define the **<FX-LOAD-BALANCE>**field as *YES*. Otherwise, set it to *NO*.

After you finish editing the fxpkcs11.cfg file, run the PKCS11Manager file to test the connection against the HSM 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.

[7.1] SPECIAL DEFINES REQUIRED FOR THIS INTEGRATION

For the Venafi integration, the following defines must be added to the **<CONFIG>** section of the FXPKCS11 configuration file:

```
# Override all key usage requests with specific values
<FORCED-SYMMETRIC-USAGE>   ENCRYPT | DECRYPT   </FORCED-SYMMETRIC-USAGE>
<FORCED-ASYMMETRIC-USAGE>  SIGN | VERIFY       </FORCED-ASYMMETRIC-USAGE>

# Specific for Venafi integration
<VENAFI-COMPAT>             YES                  </VENAFI-COMPAT>
<VENAFI-CONNECT-PING>       NO                  </VENAFI-CONNECT-PING>

# Generate keys with login requirement even if the application does not explicitly mark them as
"private"
<KEY-REQUIRE-LOGIN>         NO                 </KEY-REQUIRE-LOGIN>
```

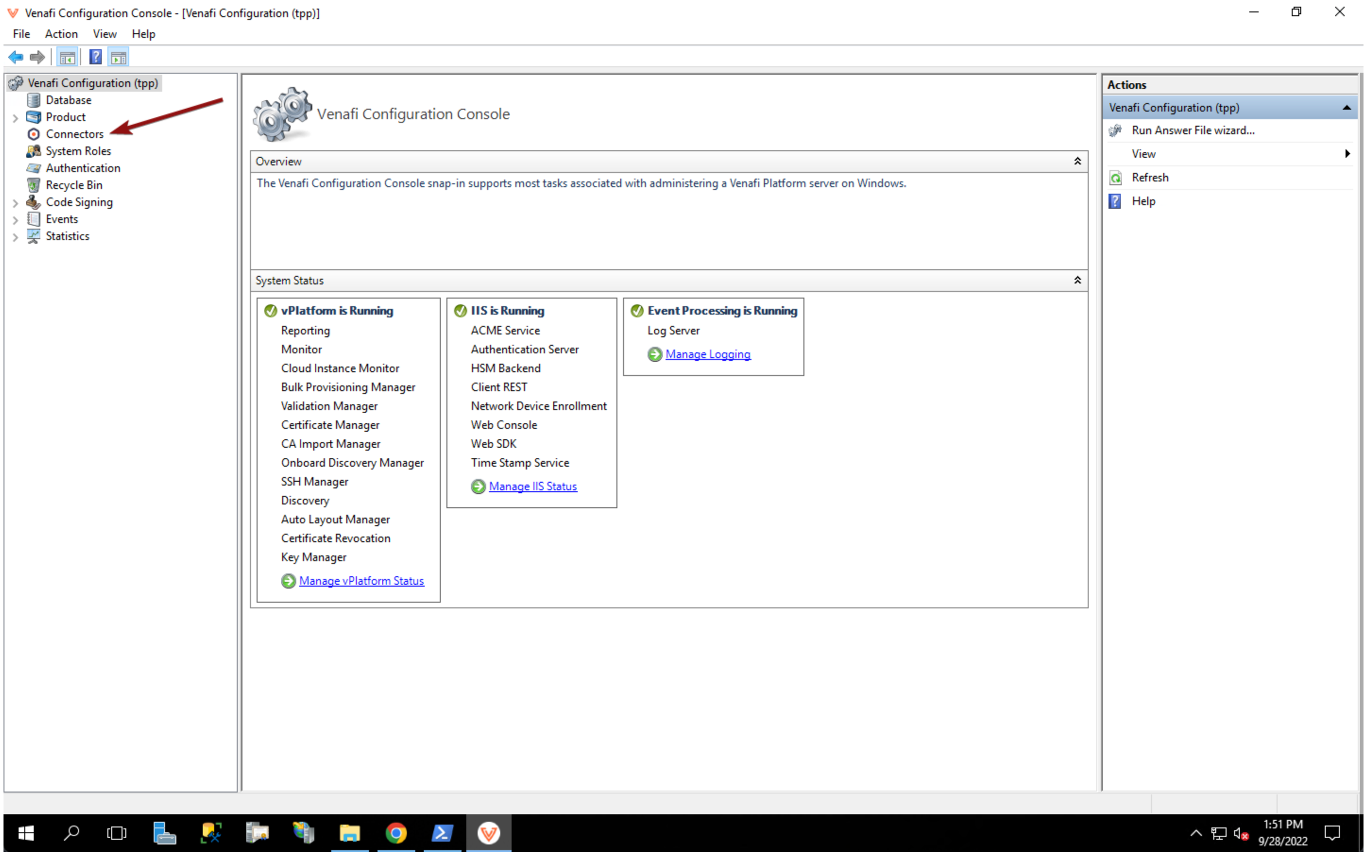
In addition, the **<WRAP-PRESERVE-USAGE>** define must be changed to **NO** in the **<CONFIG>** section.

[8] CONFIGURING VENAFI TPP TO USE THE VECTERA PLUS

This section describes how to integrate Venafi Trust Protection Platform (TPP) with the Futurex KMES Series 3 for data encryption, key generation, and key storage.

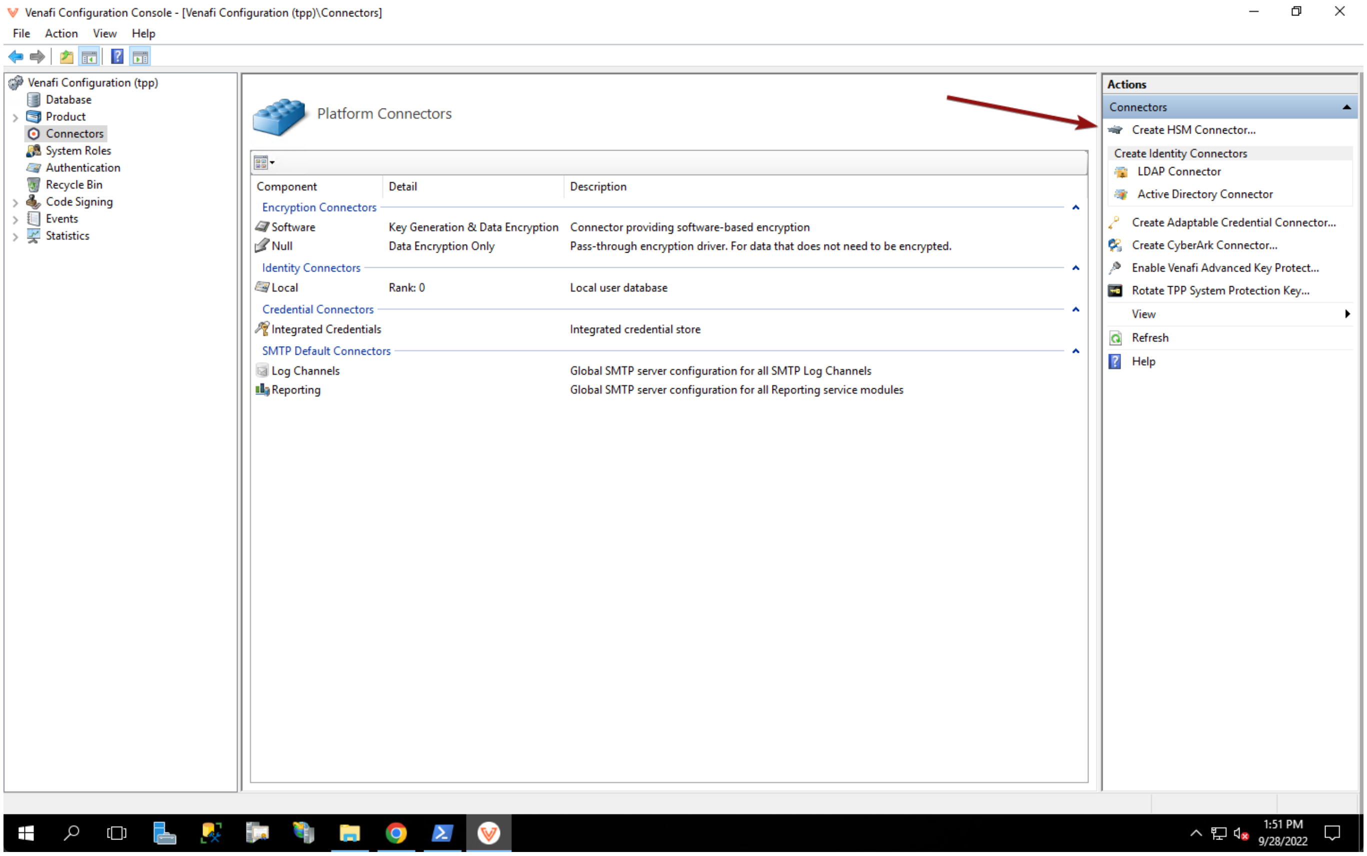
[8.1] CREATE AN HSM CONNECTOR AND GENERATE AN HSM-PROTECTED ENCRYPTION KEY

1. Open the **Venafi Configuration Console** application.
2. Select the **Connectors** node.



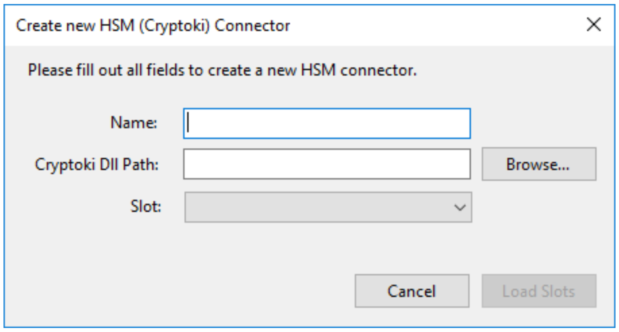
The Venafi Configuration Console snap-in supports most tasks associated with administering a Venafi Platform server on Windows.

3. Select **[Create HSM Connector...]** in the **Actions** panel.


Venafi Configuration Console - [Venafi Configuration (tpp)\Connectors]
File
Action
View
Help
Venafi Configuration (tpp)
Database
Product
Connectors
System Roles
Authentication
Recycle Bin
Code Signing
Events
Statistics
Platform Connectors
Actions
Connectors
Create HSM Connector...
Create Identity Connectors
LDAP Connector
Active Directory Connector
Create Adaptable Credential Connector...
Create CyberArk Connector...
Enable Venafi Advanced Key Protect...
Rotate TPP System Protection Key...
View
Refresh
Help
1:51 PM
9/28/2022

Component	Detail	Description
Encryption Connectors		
Software	Key Generation & Data Encryption	Connector providing software-based encryption
Null	Data Encryption Only	Pass-through encryption driver. For data that does not need to be encrypted.
Identity Connectors		
Local	Rank: 0	Local user database
Credential Connectors		
Integrated Credentials		Integrated credential store
SMTP Default Connectors		
Log Channels		Global SMTP server configuration for all SMTP Log Channels
Reporting		Global SMTP server configuration for all Reporting service modules

4. Enter the local master admin username and password and click **[OK]**. This will pull up the **Create new HSM (Cryptoki) Connector** window:



Please fill out all fields to create a new HSM connector.

5. For **Name**, enter any name for the HSM connector.

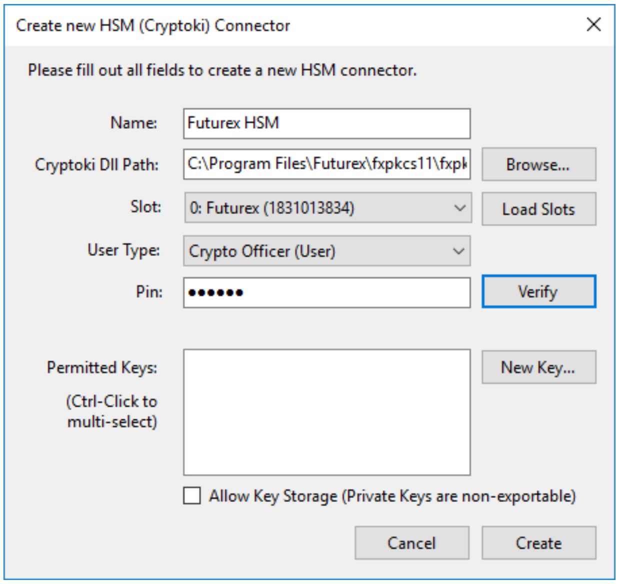
6. For **Cryptoki Dll Path**, select **[Browse...]** and locate the following path to the Futurex PKCS #11 DLL file:

```
C:\Program Files\Futurex\fxpkcs11\fxpkcs11.dll
```

7. Click **[Load Slots]**.

8. Select the slot number configured in your Futurex PKCS #11 configuration file (the default is slot **0**). This is where TPP will access the encryption keys.

9. For **User Type**, leave the default option selected, **Crypto Officer (User)**. Venafi will use the identity configured in the Futurex PKCS #11 file to connect to the Vectera Plus.
10. For **Pin**, enter the password for the identity configured in the Futurex PKCS #11 file.
11. Select **[Verify]**.
12. If the connection to the KMES Series 3 is successful, a new **Permitted Keys** section will populate in the window:



Create new HSM (Cryptoki) Connector

Please fill out all fields to create a new HSM connector.

Name: Futurex HSM

Cryptoki Dll Path: C:\Program Files\Futurex\fxpkcs11\fxpk

Slot: 0: Futurex (1831013834)

User Type: Crypto Officer (User)

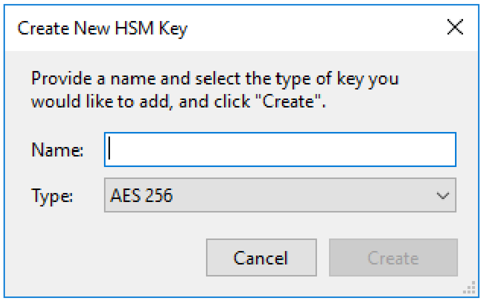
Pin: ••••••

Permitted Keys:

(Ctrl-Click to multi-select)

☐ Allow Key Storage (Private Keys are non-exportable)

13. Select **[New Key...]**. This will open the **Create New HSM Key** window.



Create New HSM Key

Provide a name and select the type of key you would like to add, and click "Create".

Name:

Type: AES 256

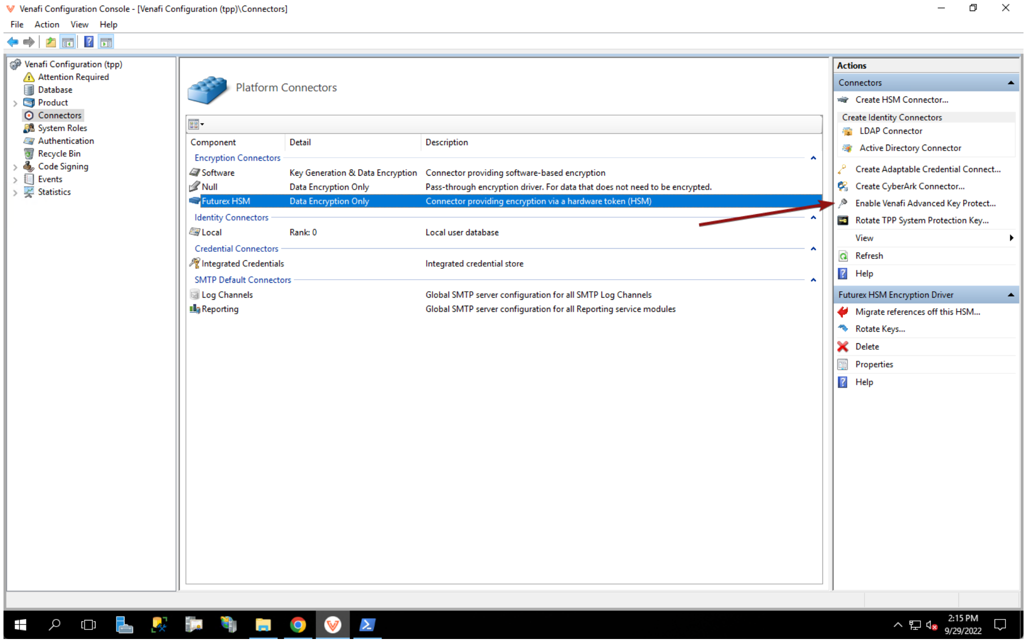
14. Enter a **Name** and select the **Type** for the key, then click **[Create]**.
15. If key creation is successful, the key is now viewable in the **Keys** menu in the KMES Series 3 application interface. The name of the key is shown in the list of **Permitted Keys** in the **Create New HSM Key** window.

    Note: If you plan to use Venafi CodeSign Protect to store private code signing keys in the KMES Series 3, select the **Allow Key Storage** checkbox here.
16. Select **[Create]** to save and close the window.

[8.2] ENABLE VENAFI ADVANCED KEY PROTECT

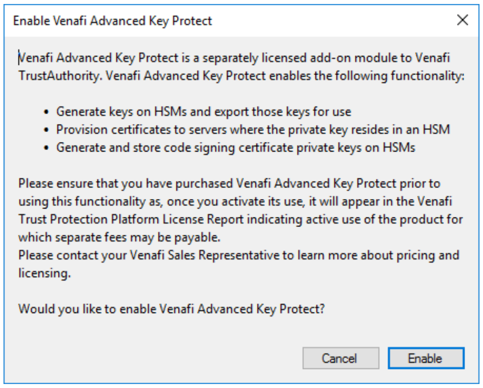
Venafi Advanced Key Protect is required for HSM Private Key Generation. In addition, Venafi Code Signing Certificate Private Key Storage requires this feature to be enabled. To enable Venafi Advanced Key Protect:

1. Open the **Venafi Configuration Console** application.
2. Select **[Enable Venafi Advanced Key Protect...]** in the **Actions** panel.


Venafi Configuration Console - [Venafi Configuration (tpp)\Connectors]
File
Action
View
Help
Venafi Configuration (tpp)
Attention Required
Database
Product
Connectors
System Roles
Authentication
Recycle Bin
Code Signing
Events
Statistics
Platform Connectors
Actions
Connectors
Create HSM Connector...
Create Identity Connectors
LDAP Connector
Active Directory Connector
Create Adaptable Credential Connect...
Create CyberArk Connector...
Enable Venafi Advanced Key Protect...
Rotate TPP System Protection Key...
View
Refresh
Help
Futurex HSM Encryption Driver
Migrate references off this HSM...
Rotate Keys...
Delete
Properties
Help
2:15 PM
9/29/2022

Component	Detail	Description
Encryption Connectors		
Software	Key Generation & Data Encryption	Connector providing software-based encryption
Null	Data Encryption Only	Pass-through encryption driver. For data that does not need to be encrypted.
Futurex HSM	Data Encryption Only	Connector providing encryption via a hardware token (HSM)
Identity Connectors		
Local	Rank: 0	Local user database
Credential Connectors		
Integrated Credentials		Integrated credential store
SMTP Default Connectors		
Log Channels		Global SMTP server configuration for all SMTP Log Channels
Reporting		Global SMTP server configuration for all Reporting service modules

3. Enter the local master admin username and password and click **[OK]**.
4. Review the information in the following dialog, and select **[Enable]** if you wish to proceed.



Venafi Advanced Key Protect is a separately licensed add-on module to Venafi TrustAuthority. Venafi Advanced Key Protect enables the following functionality:

- Generate keys on HSMs and export those keys for use
- Provision certificates to servers where the private key resides in an HSM
- Generate and store code signing certificate private keys on HSMs

Please ensure that you have purchased Venafi Advanced Key Protect prior to using this functionality as, once you activate its use, it will appear in the Venafi Trust Protection Platform License Report indicating active use of the product for which separate fees may be payable.
Please contact your Venafi Sales Representative to learn more about pricing and licensing.

Would you like to enable Venafi Advanced Key Protect?

5. Restart the IIS, Venafi Platform, and Logging services:
   a. Select the **Product** node.
   b. Select **Website** and then click **[Restart]** in the **Actions** panel.
   c. Select **Venafi Platform** and then click **[Restart]** in the **Actions** panel.
   d. Select **Logging** and then click **[Restart]** in the **Actions** panel.

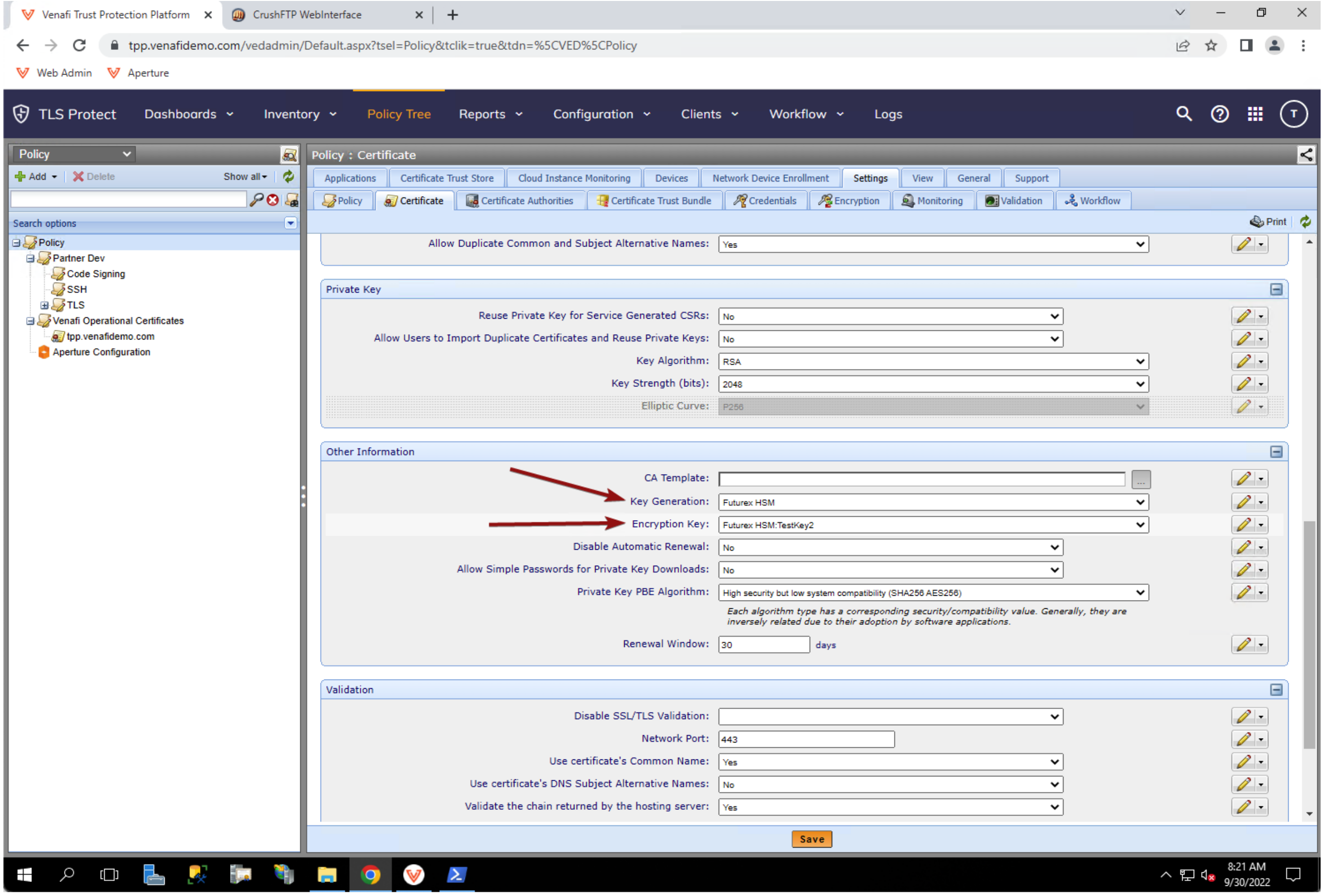
[8.3] HSM PRIVATE KEY GENERATION

Venafi Trust Protection Platform uses the Futurex KMES Series 3 for private key generation for SSH keys and certificates.

Note: Certificate Authority (CA) template objects are used in Venafi TPP to manage the certificate lifecycle. Creating one is a prerequisite to HSM Key Generation. See Venafi documentation for more information.

Configure the Venafi platform policy to enable the KMES Series 3 for HSM key generation

1. Log in to the admin console: **https://[IP_address_of_Venafi_TPP]/vedadmin**.
2. Select **Policy Tree** in the main menu at the top of the page.
3. In the **Policy : Certificate** window, select the **Certificate** tab.
4. Under **Other Information**:
   a. Select the name of the **HSM Connector** you created for the KMES Series 3 in the **Key Generation** drop-down menu.
   b. Select the name of the **HSM-Protected Encryption Key** you created on the KMES Series 3.



Policy : Certificate

Allow Duplicate Common and Subject Alternative Names: Yes

Private Key

Reuse Private Key for Service Generated CSRs: No
Allow Users to Import Duplicate Certificates and Reuse Private Keys: No
Key Algorithm: RSA
Key Strength (bits): 2048
Elliptic Curve: P256

Other Information

CA Template:
Key Generation: Futurex HSM
Encryption Key: Futurex HSM:TestKey2
Disable Automatic Renewal: No
Allow Simple Passwords for Private Key Downloads: No
Private Key PBE Algorithm: High security but low system compatibility (SHA256 AES256)

Each algorithm type has a corresponding security/compatibility value. Generally, they are inversely related due to their adoption by software applications.

Renewal Window: 30 days

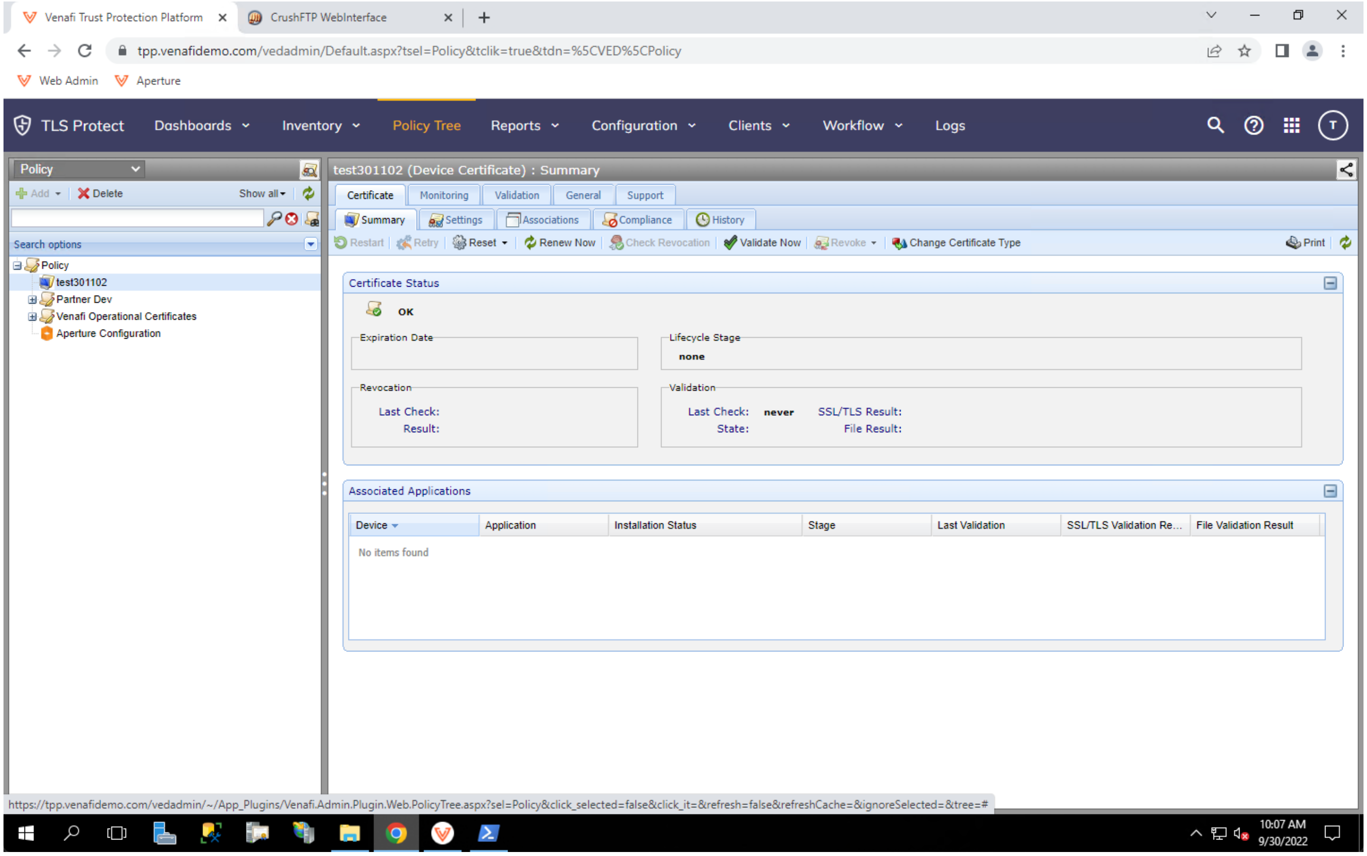
Validation

Disable SSL/TLS Validation:
Network Port: 443
Use certificate's Common Name: Yes
Use certificate's DNS Subject Alternative Names: No
Validate the chain returned by the hosting server: Yes

5. Click the **[Save]** button at the bottom of the page.

Generate the certificate

1. Select **Policy Tree** in the main menu at the top of the page.
2. On the left hand side of the page, click the **[Add]** button under the **Policy** dropdown and select **Certificates** > **Certificate**.
3. Under **General Information**, enter the required information, and for **Management Type**, select **Provisioning** or **Enrollment**.
4. Under **CSR Handling**, leave **Service Generated CSR** selected for **CSR Generation** and leave **Generate Key/CSR on Application** set to **No**.
5. Under **Subject DN**, enter the required information.
6. Under **Private Key**, select the **Key Algorithm** to use and the desired **Key Strength** in bits.
7. Under **Other Information**, search for and select the previously configured **CA Template**.
8. Click the **[Save]** button.
9. Select the newly generated certificate from the policy tree. The **Certificate Status** should be **OK**.



Certificate Status

OK

Expiration Date

Lifecycle Stage: none

Revocation
Last Check:
Result:

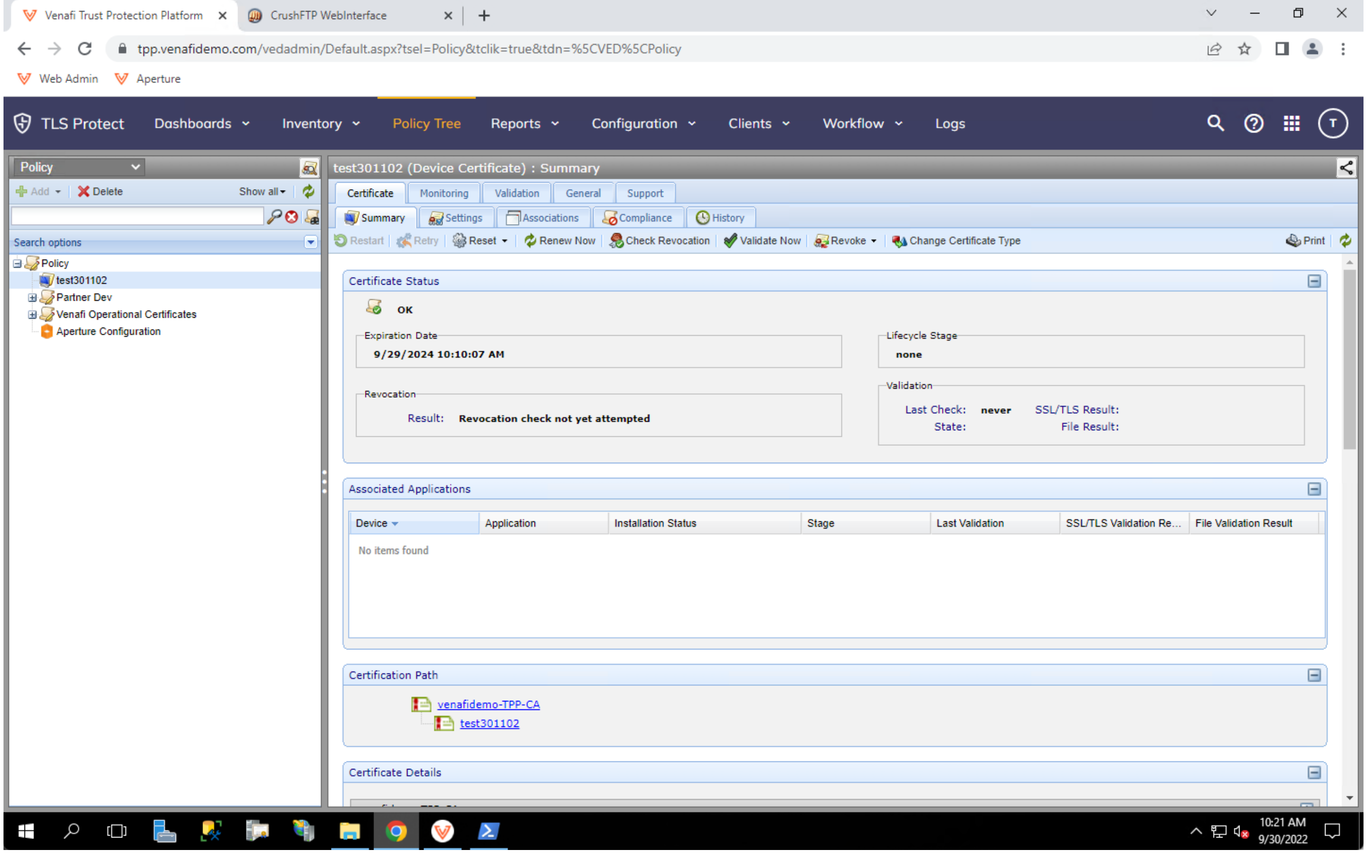
Validation
Last Check: never
State:
SSL/TLS Result:
File Result:

Associated Applications

Device	Application	Installation Status	Stage	Last Validation	SSL/TLS Validation Re...	File Validation Result
No items found						

10. Select **Renew Now**. The **Certificate Status** will change to **Queued for renewal**.

11. After about a minute, click **Refresh**. The certificate details will appear in the window.



tpp.venafidemo.com/vedadmin/Default.aspx?tsel=Policy&tclik=true&tdn=%5CVED%5CPolicy

test301102 (Device Certificate) : Summary

Certificate Status

OK

Expiration Date: 9/29/2024 10:10:07 AM

Lifecycle Stage: none

Revocation — Result: Revocation check not yet attempted

Validation — Last Check: never — State: — SSL/TLS Result: — File Result:

Associated Applications

Device	Application	Installation Status	Stage	Last Validation	SSL/TLS Validation Re...	File Validation Result

No items found

Certification Path

venafidemo-TPP-CA
test301102

Certificate Details

12. If you selected Provisioning for Management Type, associate the certificate to the intended application object.

[8.4] CODE SIGNING

Venafi CodeSign Protect can store private code signing keys in the KMES Series 3. This section describes the basic steps to configure this functionality for the integration. See Venafi documentation for more details.

Note: Certificate Authority (CA) template objects are used in Venafi TPP to manage the certificate lifecycle. Creating one is a prerequisite to CodeSign. See Venafi documentation for more information.

Note: To use the KMES Series 3 for key storage, **Key Storage** must be enabled on the **HSM Connector**, as noted in step 15 of section 8.1.

Assign permissions to a Code Signing Administrator

1. Open the **Venafi Configuration Console** application.
2. Select the **System Roles** node.
3. Select **[Add CodeSign Protect Administrator]** in the **Actions** panel.
4. Select a user to grant CodeSign Protect Administrator permissions.

Create a Code Signing Flow

1. Open the **Venafi Configuration Console** application.
2. Under the **Code Signing** node, select **Custom Flows**.
3. Select **Add new Code Signing Flow** in the **Actions** panel.
4. Enter a name for the Code Signing Flow.
5. Select the newly created Code Signing Flow and add an approver through the **Actions** panel.

Create an Environment Template for the Code Signing Project

1. Open the **Venafi Configuration Console** application.
2. Under the **Code Signing** node, select **Environment Templates**.
3. Select **Certificate** in the **Actions** panel under **Add Single Template**.
4. Enter a name for the Code Signing Environment Template and select **[Create]**.
5. In the **Properties** window, within the **Settings** tab, enter a **Description** and select a **Certificate Container** and **Signing Flow**.
6. Open the **Certificate Authority** tab and select a **CA Template**, then **click [Add]**.
7. Open the **Keys** tab and select which key sizes to allow for **RSA** and **Elliptic Curve** keys.
8. Open the **Key Storage** tab and select the **KMES Series 3** Connector, then click **[Add]**.
9. Enter any optional information in the remaining tabs, then click **[OK]**.

Create a new Code Signing Project

1. Log in to Aperture: **https://[IP_address_of_Venafi_TPP]/aperture/codesign**.
2. Select **Projects** in the main menu at the top of the page.
3. Select **[Create Project]**.
4. Enter a **Project Name** and **Description**.
5. Select **[Create]**.

Create an Environment for the project with a new HSM private key and certificate

1. Inside the newly created Code Signing Project, navigate to the **Environments** tab and select **Add Environment** > **Certificate & Key**.
2. Enter the **Environment Name**.
3. Select the **Environment Template** that you created for this Code Signing project.

4. For **Creation Type**, select **Create New**. The **Key Storage Location** should now list the KMES Series 3 Connector.
5. Enter any other necessary information for the certificate.
6. Select **[Save]**.
7. Select **[Submit For Approval]** to generate a new certificate and private key once it is approved.

Approving the Project

1. Log in to Aperture: **https://[IP_address_of_Venafi_TPP]/aperture/codesign**.
2. Select **Approvals** in the main menu at the top of the page.
3. Under **Pending Approvals**, select the Project Creation request you just submitted.
4. Select **[Approve]**.Approval processed successfully
5. Navigate back to the project, and under the **Environments** tab you should see that a **Certificate & Key** were created in **Hardware** (i.e., the KMES Series 3).

[9] VIEW THE KEYS AND CERTIFICATES THAT VENAFI TPP CREATED ON THE HSM

To view the keys and certificates that Venafi TPP created on the HSM, we will use the **PKCS11Manager** utility packaged with the Futurex PKCS #11 module.

1. In File Explorer, navigate to the directory where the FXPKCS11 module is installed and run the **PKCS11Manager.exe** file by double-clicking on it.

   This will present the following main menu:

```
Main Menu
    1. Print Library/Token Info

    2. Generate Key

    3. Find Objects
    4. Modify Objects
    5. Delete Objects

    6. Generate Random Data

    7. Sign Data

    8. Login
    9. Logout

    0. Exit
```

2. Type **8** to login, then press **Enter**.
3. Type **1**, then press **Enter**.
4. Type the password of the identity that is defined in the FXPKCS11 configuration file, then press **Enter**.

   If successful, you will receive confirmation that you are logged in.

5. Type **3** to find objects, then press **Enter**.
6. Type **1** to find all objects, then press **Enter**.

   PKCS11Manager will print info for all keys and certificates the connecting identity has permission to view. The output will look similar to the below:

```
Total number of found objects: 5
  Object ID: 1
    Internal ID: 1
    Excrypt Board Slot: 0
    Class: CKO_SECRET_KEY
    Key Type: AES
    Token: Yes
    Private: No
    Sensitive: Yes
    Modifiable: Yes
    Value Len: 32
    Value Bits: 256
    Label: TestKey855
    KCV: E645
    Usage: ED
```

```
Object ID: 2
  Internal ID: 2
  Excrypt Board Slot: 1
  Class: CKO_PRIVATE_KEY
  Token: Yes
  Private: No
  Sensitive: Yes
  Modifiable: Yes
  Modulus Bits: 2048
  Label: RSA 2048 1bb264b04b5f495f8a7c1ed43b6551cd
  ID: 10976
  KCV: CC7D
  Usage: SV
Object ID: 3
  Internal ID: 3
  Excrypt Board Slot: 2
  Class: CKO_PUBLIC_KEY
  Token: Yes
  Private: No
  Sensitive: No
  Modifiable: Yes
  Modulus Bits: 2048
  Label: RSA 2048 1bb264b04b5f495f8a7c1ed43b6551cd
  ID: 10976
  KCV: 8F48
  Usage: V
Object ID: 4
  Internal ID: 4
  Excrypt Board Slot: 3
  Class: CKO_PRIVATE_KEY
  Token: Yes
  Private: No
  Sensitive: Yes
  Modifiable: Yes
  Label: EC_P384 66573c87189c47d7ad2d82e4f072e603
  ID: 10979
  KCV: 7ECC
  Usage: SV
Object ID: 5
  Internal ID: 5
  Excrypt Board Slot: 4
  Class: CKO_PUBLIC_KEY
  Token: Yes
  Private: No
  Sensitive: No
  Modifiable: Yes
  Label: EC_P384 66573c87189c47d7ad2d82e4f072e603
  ID: 10979
  KCV: 7DA5
  Usage: V
```

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