BIG-IP® System and Thales HSM: Implementation

Version 13.1

Table of Contents

Setting Up the Thales HSM

Overview: Setting up the Thales HSM

The Thales nShield Connect is an external HSM that is available for use with BIG-IP® systems. Because it is network-based, you can use the Thales nShield Connect solution with all BIG-IP platforms, including VIPRION® Series chassis and BIG-IP Virtual Edition (VE).

The Thales nShield Connect architecture includes a component called the Remote File System (RFS) that stores and manages the encrypted key files. The RFS can be installed on the BIG-IP system or on another server on your network.

The BIG-IP system is a client of the RFS, and all BIG-IP systems that are enrolled with the RFS can access the encrypted keys from this central location.

Only RSA-based cipher suites use the network HSM.

After you install the Thales nShield Connect client on the BIG-IP system, the keys stored in the Thales HSM and the corresponding certificates are available for use with Access Policy Manager® and Application Security Manager™.

For additional information about using Thales nShield Connect, refer to the Thales website: (`https://www.thales-esecurity.com`).

***Note:** If you are installing Thales nShield Connect on a BIG-IP system that will be licensed for Appliance mode, you must install the Thales nShield Connect software prior to licensing the BIG-IP system for Appliance mode.*

Prerequisites for setting up Thales nShield Connect with BIG-IP systems

Before you can use Thales nShield Connect with the BIG-IP® system, you must make sure that these requirements are in place:

- The Thales nShield Connect device is installed on your network.
- The IP address of the BIG-IP client that is visible to the Thales HSM is on the allowed list of clients on the Thales nShield Connect device. If you are implementing Thales nShield Connect with a VIPRION® system, you need to add the cluster management IP addresses and the cluster member IP address for each blade installed in the chassis to the allowed list. This applies to using the management network. If you use a TMM interface with a floating IP address, only that IP address is required.
- The RFS server is installed. This could be an external server on your network or on the local BIG-IP system.
- The Thales nShield Connect device, the RFS, and the BIG-IP system can initiate connections with each other through port `9004` (default).
- You have created the Thales Security World (security architecture).
- The BIG-IP system is licensed for "External Interface and Network HSM."

***Important:** You cannot run the BIG-IP system with both internal and external HSMs at the same time.*

***Note:** BIG-IP TMOS with Thales HSM only supports IPv4.*

Additionally, before you begin the installation process, make sure that you can locate these items on the installation DVD that ships with the Thales hardware unit:

- The Thales Security World Software for Linux 64bit
- The nShield_Connect_and_netHSM_User_Guide.pdf

Note: *For supported Thales client and HSM versions with BIG-IP TMOS versions information, see the Interoperability Matrix for BIG-IP TMOS with Thales and HSM supplemental document available on AskF5.*

Task summary

The implementation process involves preparation of the Thales nShield Connect device.

Task list

Installing Thales nShield Connect components on the BIG-IP system

Before you can set up the Thales nShield Connect components on a BIG-IP® system, you must obtain the Thales 64-bit Linux ISO CD and copy files from the CD to specific locations on the BIG-IP system using secure copy (SCP). F5 Networks has tested these integration steps with Thales security World Software for Linux 64bit. For questions about Thales components, consult your Thales representative.

You can install files from the Thales 64 bit Linux ISO CD to the BIG-IP system.

1. Log in to the command-line interface of the system using an account with administrator privileges.
2. Create a directory under `/shared` named `thales_install/amd64/nfast`.
   ```
   mkdir -p /shared/thales_install/amd64/nfast
   ```
3. In the new directory, create subdirectories named `ctls`, `hwcrhk`, `hwsp`, and `pkcs11`.
4. Copy files from the CD and place them in the specified directories:

File to copy from the CD	Location to place file on BIG-IP
/linux/libc6_3/amd64/nfast/ctls/agg.tar	/shared/thales_install/amd64/nfast/ctls/agg.tar
/linux/libc6_3/amd64/nfast/hwcrhk/user.tar	/shared/thales_install/amd64/nfast/hwcrhk/user.tar
/linux/libc6_3/amd64/nfast/hwsp/agg.tar	/shared/thales_install/amd64/nfast/hwsp/agg.tar
/linux/libc6_3/amd64/nfast/pkcs11/user.tar	/shared/thales_install/amd64/nfast/pkcs11/user.tar

Setting up the RFS on the BIG-IP system (optional)

Before you set up the Remote File System (RFS) on the BIG-IP® system, make sure that the Thales nShield Connect device is installed on your network.

Note: *Setting up the RFS on the BIG-IP system is optional. If the RFS is running on another server on your network, you do not need to perform this task.*

If the RFS is not running on another server in your network, you need to set up the RFS on the BIG-IP® system.

1. Log in to the command-line interface of the BIG-IP system using an account with administrator privileges.
2. Run the script to set up the RFS.

```
nethsm-thales-rfs-install.sh --hsm_ip_addr=<Thales_nShield Connect device
IP address> --rfs_interface=<local interface name>
```

This example sets up the RFS to run on the BIG-IP system, where the IP address of the Thales nShield Connect device has an IP address of `192.27.13.59`:

```
nethsm-thales-rfs-install.sh --hsm_ip_addr=192.168.13.59 --
rfs_interface=eth0
```

The RFS interface option is the interface the BIG-IP uses to connect to the HSM.

After you have set up the RFS, you must setup a Security World before attempting to connect to the BIG-IP as a client.

Setting up the Thales nShield Connect client on the BIG-IP system

Before you set up the Thales client, make sure that the Thales nShield Connect client is installed on the BIG-IP® system and that the Security World has been set up. Additionally, make sure that the RFS is installed and set up on either a remote server or on the BIG-IP system on your network.

***Note:** If the Thales nShield Connect client was installed on a BIG-IP system before the RFS was installed on the network, then you must reinstall the client on the BIG-IP system.*

***Note:** The BIG-IP system IP address might not be the same as the IP address of the outgoing packet, such as when a firewall modifies the IP address.*

To use the Thales nShield Connect device with the BIG-IP system, you must first set up the Thales client on the BIG-IP system. For the enrollment to work properly, the IP address of the BIG-IP system must be a client of the networked HSM. In the case of the VIPRION® system and connecting over the admin interfaces, each blade and the chassis IP address need to be added as a client. You set up the IP address using the front panel of the nShield Connect device, or by pushing the client configuration. For details about how to add, edit, and view clients, refer to the Thales documentation.

If you are setting up the Thales client on a VIPRION® system, you run the configuration script only on the primary blade, and then the system propagates the configuration to the additional active blades.

1. Log in to the command-line interface of the BIG-IP system using an account with administrator privileges.
2. Verify that the F5 interface you will use to communicate with the nShield Connect has been entered on the front panel of the HSM; that is, the Thales nShield Connect must permit connections from the F5 source IP address.
3. Set up the Thales nShield Connect client, using one of these options:
   - Option 1: Set up the client when the RFS is remote.

```
nethsm-thales-install.sh
--hsm_ip_addr=<nShield_Connect_device_IP_address>
--rfs_ip_addr=<remote_RFS_server_IP_address>
--rfs_username=<remote_RFS_server_username_for_SSH_login>
--protection=<protection_type>
```

The following example sets up the client where the Thales nShield Connect device has an IP address of `192.168.13.59`, the remote RFS has an IP address of `192.168.13.58`, the user name for an SSH login to the RFS is `root`, and the Thales client interface is the management interface:

```
nethsm-thales-install.sh --hsm_ip_addr=192.168.13.59 --
rfs_ip_addr=192.168.12.58 --rfs_username=root
```

   - Option 2: Set up the client when the RFS is set up on the local BIG-IP system:

```
nethsm-thales-install.sh
--hsm_ip_addr=<nShield_Connect_device_IP_address>
--rfs_interface=<local_RFS_server_interface>
```

The following example sets up the client where the Thales nShield Connect device has an IP address of `172.168.13.59` and the RFS is installed on the BIG-IP system using the `eth0` interface:

```
nethsm-thales-install.sh --hsm_ip_addr=172.168.13.59 --rfs_interface=eth0
```

In addition, the RFS installed on the BIG-IP system may use the TMM interface (namely a VLAN):

```
nethsm-thales-install.sh --hsm_ip_addr=10.20.20.1 --
rfs_interface=<VLAN_name>
```

4. Reload the PATH environment variable.

   If you are installing the Thales nShield Connect on a VIPRION system, you need to reload the PATH environment variable on any blades with already-open sessions: `source ~/.bash_profile`.
5. You can use the default number of threads provided, or you can specify the number of threads using the num-threads option. This can also be adjusted later using `tmsh`.

Setting up the Thales nShield Connect client on a newly added or activated blade (optional)

After you set up the Thales nShield Connect client on the primary blade of a VIPRION® system, the system propagates the configuration to the additional active blades. If you subsequently add a secondary blade, activate a disabled blade, or power-on a powered-off blade, you need to run a script on the new secondary blade.

1. Log in to the command-line interface of the system using an account with administrator privileges.
2. Run this script on any new or re-activated secondary blade:
   `thales-sync.sh`
3. If you make the new blade a primary blade before running the synchronization script, you need to run the regular client setup procedure on the new primary blade only.
   `nethsm-thales-install.sh`

Configuring the Thales nShield Connect client for multiple HSMs in an HA group

Before starting this task, you need to set up the Thales nShield Connect client on the BIG-IP® system.

You can perform these additional steps to configure the Thales nShield Connect client for multiple HSMs.

1. Log in to the command-line interface of the system using an account with administrator privileges.
2. Enroll each additional HSM in the HA group.
   `/opt/nfast/bin/nethsmenroll --force <HSM_ip_address> $(anonkneti <HSM_ip_address>)`

   Perform this step for each of the additional HSMs in the HA group. For the enrollment to work properly, the IP address of the BIG-IP system must be a client of each networked HSM. You set up the IP address using the front panel of the nShield Connect device, or by pushing the client configuration. For details about how to add, edit, and view clients, refer to the Thales documentation.
3. Update the permissions.

```
          chmod 755 -R /opt/nfast/bin
          chown -R nfast:nfast /opt/nfast/kmdata/
```

```
    chmod 700 -R /opt/nfast/kmdata/tmp/nfpriv_root
    chown -R root:root /opt/nfast/kmdata/tmp/nfpriv_root
```

4. Verify installation.

   `/opt/nfast/bin/enquiry`

   This command displays all the installed modules that have the status `Operational`. Note that three HSMs are operational in this example.

```
        Server:
        :
        serial number        CB9E-745E-F901 A1D0-2DBE-AD98 5286-D07F-7601
        mode                 operational
```

5. Restart the `pksc11` service.

   `tmsh restart sys service pkcs11d`

6. Restart the `TMM` service.

   `tmsh restart sys service tmm`

7. Wait until the TMM is active.

8. Verify installation.

   `/opt/nfast/bin/enquiry`

Managing External HSM Keys for LTM

Overview: Managing external HSM keys for LTM

You can use the Thales nShield Connect to store and manage token-, module-, and softcard-protected keys.

For additional information about using Thales nShield Connect, refer to the Thales website: (`https://www.thales-esecurity.com`).

About key protection

There are three types of key protection available for use with the BIG-IP® system and Thales Connect:

- *Module-protected keys* are directly protected by the external HSM through the security world and can be used at any time without further authorization.
- *Softcard-protected keys* are protected by a softcard and can be used by only an operator who possesses the assigned passphrases.
- *Token-protected keys* are protected by a cardset and can be used by only an operator who possesses the Operator Card Set (OCS) token and any assigned passphrases.

All options are equally secure, and the main difference is the authorization requirement. As a general rule, if you have no particular security or regulatory requirement, you can default to module protection. Thales prefers the use of physical tokens for authorization. In the case of Operator Cards, Thales recommends making a 1/N card set, where N is greater than the total number of nShield Connects. For more information about card sets, refer to the Thales user guides.

Task summary

The implementation process involves configuring a key protection type, and then creating and loading token-, module-, or softcard- protected keys and certificates, and creating a client SSL profile to use the key and certificate.

Task list

Configuring the key protection type

On the BIG-IP® system, you can choose among the Thales-supported types of key protection: module, softcard, and OCS. By default, the installation script sets up the appliance to create and use module-protected keys. F5 recommends that you keep only one set of cardset files (cards* or softcard*) in the `$NFAST_KMDATA/local` directory.

In this release, only one type of key protection (PKCS#11 slot) can be configured for active use. You need to configure the key protection type for a slot by enabling the type you want, and disabling the others.

1. Log in to the command-line interface of the BIG-IP system using an account with administrator privileges.
2. Complete one of these steps, depending on your preferred key protection option:

Option	Description
module	The module-protected key option is enabled by default. To enable this protection type, no further action is required, and you can proceed to the next section.
OCS	**1.** Disable softcard key protection by moving any previously created `softcard*` files from the `/opt/nfast/kmdata/local` directory to the `/opt/nfast/kmdata/` directory. **2.** Enable OCS key protection by creating the OCS cardset using the Thales-provided `createocs` utility.
softcard	**1.** Disable OCS key protection by moving any previously created `cards*` files from the `/opt/nfast/kmdata/local` directory to the `/opt/nfast/kmdata` directory. **2.** Enable softcard key protection by creating the softcard cardset using the Thales-provided `ppmk` utility.

Note: *The softcard passphrase used in the* `ppmk` *command must match the passphrase used for setting up the Thales nShield Connect client on the BIG-IP system (used in the command* `tmsh create/modify sys crypto fips external-hsm password <password>`*).*

Note: *If OCS is configured with a passphrase for Thales HSM, the user must enter it when prompted for* `Thales HSM slot password`*, even if the user only wants to use module keys.*

Note: *To revert back to module protection, change* `CKNFAST_NO_ACCELERATOR_SLOTS=1` *to* `CKNFAST_NO_ACCELERATOR_SLOTS=0` *and remove any softcard or OCS files out of* `/opt/nfast/kmdata/local`*.*

3. After you make any configuration changes, you must restart the `pkcs11` and `tmm` services.

```
tmsh restart sys service pkcs11d
tmsh restart sys service tmm
```

Generating a key/certificate using tmsh

You can use the Traffic Management Shell (`tmsh`) to generate a key and certificate.

1. Log in to the command-line interface of the system using an account with administrator privileges.
2. Open the TMOS Shell (`tmsh`).
   `tmsh`
3. Generate the key.

   ```
   create sys crypto key <key_name> gen-certificate common-name <cert_name>
   security-type nethsm
   ```

   This example generates an external HSM key named `test_key` and a certificate named `test_thales.com` with the security type of `nethsm`:

   ```
   create sys crypto key test_key gen-certificate common-name test_thales.com
   security-type nethsm
   ```
4. Verify that the key was created.
   `list sys crypto key test_key.key`
   Information about the key displays:

```
sys crypto key test_key.key {
key-id <32-digit string>
key-size 2048
key-type rsa-private
security-type nethsm
}
```

When you generate a key/certificate using `tmsh`, the system creates a HSM private key. It also creates a local key, which points to the HSM key, residing in the HSM.

Creating a self-signed digital certificate

If you are configuring the BIG-IP® system to manage client-side HTTP traffic, you perform this task to create a self-signed certificate to authenticate and secure the client-side HTTP traffic. If you are also configuring the system to manage server-side HTTP traffic, you must repeat this task to create a second self-signed certificate to authenticate and secure the server-side HTTP traffic.

1. On the Main tab, click **System** > **Certificate Management** > **Traffic Certificate Management**.
   The Traffic Certificate Management screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Self**.
5. In the **Common Name** field, type a name.

   This is typically the name of a web site, such as `www.siterequest.com`.
6. In the **Division** field, type your department name.
7. In the **Organization** field, type your company name.
8. In the **Locality** field, type your city name.
9. In the or **State or Province** field, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.
13. In the **Subject Alternative Name** field, type a name.

    This name is embedded in the certificate for X509 extension purposes.

    By assigning this name, you can protect multiple host names with a single SSL certificate.
14. From the **Security Type** list, select **NetHSM**.
15. From the **Key Type** list, **RSA** is selected as the default key type.
16. From the **Size** list, select a size, in bits.
17. Click **Finished**.

Requesting a certificate from a certificate authority

You perform this task to generate a certificate signing request (CSR) that can then be submitted to a third-party trusted certificate authority (CA).

Note: *F5 Networks recommends that you consult the CA to determine the specific information required for each step in this task.*

1. On the Main tab, click **System** > **Certificate Management** > **Traffic Certificate Management**.
   The Traffic Certificate Management screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Certificate Authority**.

5. In the **Common Name** field, type a name.
   This is typically the name of a web site, such as `www.siterequest.com`.
6. In the **Division** field, type your department name.
7. In the **Organization** field, type your company name.
8. In the **Locality** field, type your city name.
9. In the or **State or Province** field, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.
13. In the **Subject Alternative Name** field, type a name.
    This name is embedded in the certificate for X509 extension purposes.
    By assigning this name, you can protect multiple host names with a single SSL certificate.
14. In the **Challenge Password** field, type a password.
15. In the **Confirm Password** field, re-type the password you typed in the **Challenge Password** field.
16. From the **Security Type** list, select **NetHSM**.
17. From the **Key Type** list, **RSA** is selected as the default key type.
18. From the **Size** list, select a size, in bits.
19. Click **Finished**.
    The Certificate Signing Request screen displays.
20. Do one of the following to download the request into a file on your system.
    - In the **Request Text** field, copy the certificate.
    - For **Request File**, click the button.
21. Follow the instructions on the relevant certificate authority web site for either pasting the copied request or attaching the generated request file.
22. Click **Finished**.
    The Certificate Signing Request screen displays.

The generated certificate signing request is submitted to a trusted certificate authority for signature.

Deleting a key from the BIG-IP

You perform this task to delete an existing key from the BIG-IP.

1. On the Main tab, click **System** > **Certificate Management** > **Traffic Certificate Management**.
   The Traffic Certificate Management screen opens.
2. From the **SSL Certificate List**, select the check box next to the key you wish to delete.
3. Click **Delete**.

The key you selected is deleted from BIG-IP.

Note: *The key stored in NetHSM is not deleted.*

Creating a client SSL profile to use an external HSM key and certificate

After you have added the external HSM key and certificate to the BIG-IP® system configuration, you can use the key and certificate as part of a client SSL profile. This task describes using the browser interface. Alternatively, you can use the Traffic Management Shell (`tmsh`) command-line utility.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client screen opens.

2. Click **Create**.
   The New Client SSL Profile screen opens.
3. In the **Name** field, type a name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. From the **Configuration** list, select **Advanced**.

   This selection makes it possible for you to modify additional default settings.
6. For the Configuration area, select the **Custom** check box.
   The settings in the Configuration area become available for modification.
7. Using the **Certificate Key Chain** setting, specify one or more certificate key chains:
   a) From the **Certificate** list, select the name of a certificate that you imported.
   b) From the **Key** list, select the name of the key that you imported.
   c) From the **Chain** list, select the chain that you want to include in the certificate key chain.
   d) Click **Add**.
8. Click **Finished**.

After you have created the client SSL profile, you must assign the profile to a virtual server, so that the virtual server can process SSL traffic according to the specified profile settings.

Migrating existing software-protected or unprotected keys to the Thales HSM

Before you begin this task, make sure that the Thales nShield Connect client is installed and configured on the BIG-IP® system.

If you already have regular RSA keys, you can migrate them to the Thales HSM.

***Note:** A Thales HSM device that is configured with the `Strict FIPS 140-2 Level 3` compliance flag prevents importation of extraneous private keys.*

---

1. Log in to the command-line interface of the system using an account with administrator privileges.
2. Migrate the key.

   ```
   fipskey.nethsm --export -i <input_key_full_path_filename> -o
   output_key_filename
   ```

   This example generates the four files that follow:

   ```
   fipskey.nethsm --export -i regular_key -o hsm_key
   ```

   - `/config/ssl/ssl.key/hsm_key.key` (local key)
   - `/config/ssl/ssl.csr/hsm_key.csr` (CSR file)
   - `/config/ssl/ssl.crt/hsm_key.crt` (self-signed certificate)
   - `/opt/nfast/kmdata/local/protected_key_filename` (protected key)

If you migrated a key that has a certificate that is already issued by a reputable issuing CA, you should migrate the key, but continue using the old certificate. After you migrate the existing key to the Thales HSM, you must load the key into the BIG-IP system using `tmsh`, and then modify the client SSL profile, or create a new client SSL profile that uses the new key and the existing certificate.

Importing existing SSL keys into Thales nShield device for use by the BIG-IP system

You import existing SSL keys when you have pre-existing keys you want the BIG-IP® system to use. You need to perform these steps for each key you want to import into the Thales system.

1. Log in to the command-line interface of the system using an account with administrator privileges.
2. Copy certificate(s) and key(s) you want to import onto the BIG-IP system and place them in the `/var/tmp` directory on the BIG-IP system.

```
/var/tmp/user.key
/var/tmp/user.crt
```

3. Ensure adequate permissions are set so that other users on the system are not able to view the `.key` files copied.

   `chmod 600 /var/tmp/user.key`
4. Import the key into Thales nShield Connect external HSM using the **generatekey** utility.

   `/opt/nfast/bin/generatekey --import pkcs11 certreq=yes`

   The system interactively prompts you for information.
5. When prompted to enter the name of the PEM file that contains the RSA key, enter the full path to the key copied to the BIG-IP system (pemreadfile).

   For example, `/var/tmp/user.key`.
6. When prompted to enter the file name where the key will be written, enter the full path to the pseudo key (embedsavefile).

   This is the pseudo key required by BIG-IP system.

   For example, `/var/tmp/imported_user.key`.
7. When prompted to enter the key name, type a name for the key (`plainname`).

   This is the name with which the key is associated in the nShield RFS. No path is required, as `plainname` is not written to a file on disk.

   For example, `userkey`.

   When the key import is complete, the `generatekey` utility will generate two files.

   - `imported_user.key`
   - `imported_user_req`
8. Modify the ownership and permissions of the key you created. After successful import, take note of the path to key to modify ownership.

```
chown nfast:nfast /opt/nfast/kmdata/local/key_pkcs11_uced028e5251b7b
6891e7e59dec5428d871f92241b-c70e6451e8d793ca80a497267ccb9bc73bd55edb
chmod 755 /opt/nfast/kmdata/local/key_pkcs11_uced028e5251b7b6891e7e59dec5428d871f92241b-
c70e6451e8d793ca80a497267ccb9bc73bd55edb
```

Important: *If this step is omitted, you might see permission errors when running* `rfs-sync`*.*

9. Sync the nShield generated pseudo-key (embedsavefile) to the RFS.

```
[root@LBHAS64:Active:Standalone] tmp # rfs-sync --update
[root@LBHAS64:Active:Standalone] tmp # rfs-sync --commit
```

   If the BIG-IP system this procedure is performed on is also the RFS, the `rfs-sync` commands above will report `0 committed`. This is expected behavior, as the keys imported are automatically stored in the RFS directory.

10. Import the pseudo key and SSL certificate using `tmsh` for use by BIG-IP client SSL profile using this syntax:

```
tmsh install sys crypto key [name] from-local-file [/path/to/pseudo_key.key]
tmsh install sys crypto cert [name] from-local-file [/path/to/real_certificate.crt]
```

   For example:

```
tmsh install sys crypto key import.key from-local-file /var/tmp/imported_user.key
tmsh install sys crypto cert import.crt from-local-file /var/tmp/user.crt
```

11. Save the configuration.

   `tmsh save sys config`

   If you need to import more SSL certificates and keys, repeat all preceding steps for each certificate and key pair.

12. Create an SSL profile that references the above key and certificate.
13. Create a virtual server that uses the above SSL profile (or assign to an existing virtual server).
14. Verify that the virtual server passes traffic correctly.
15. You can safely remove the certificates and keys from `/var/tmp` directory used in this procedure as they are no longer required by the BIG-IP system.

Note: *Once the pseudo key has been installed with* `tmsh`*, the copy in* `/var/tmp` *is no longer used.*

Note: *Unless the SSL key file is deleted in a secure manner, it might be possible for someone to recover the file from the disk. Consider using the* `shred` *utility (type* `man shred` *at the command line for details) to delete any key files copied to the BIG-IP system once they have been successfully imported into the Thales nShield device.*

Note: *When you create a new Thales key for BIG-IP HA, you must run the command* `rfs-sync --update` *on all standby BIG-IP devices to update the local Thales encrypted file object cache. Without this action, SSL traffic using this key will fail when BIG-IP fails over to one of the unsynced standby devices.*

Generating External HSM Key-Cert Pairs for DNSSEC

Overview: Generating external HSM key and certificate pairs for manually managed DNSSEC keys

When the BIG-IP® system is a BIG-IP DNS (previously Global Traffic Manager), you can use the Thales nShield Connect to store and manage DNSSEC keys.

For additional information about using Thales nShield Connect, refer to the Thales website: (`https://www.thales-esecurity.com`).

Task list

Generating an external key for creating manually managed DNSSEC keys

Before you generate the key, make sure that the Thales nShield Connect client is running on all BIG-IP® DNS devices in the configuration synchronization group.

You can use the Traffic Management Shell (`tmsh`) to generate a key and certificate.

1. Log in to the command-line interface of the system using an account with administrator privileges.
2. Open the TMOS Shell (`tmsh`).
   ```
   tmsh
   ```
3. Generate the key.
   ```
   create sys crypto key <key_name> gen-certificate common-name <cert_name>
   security-type nethsm
   ```
   This example generates an external HSM key named `test_key` and a certificate named `test_thales.com` with the security type of `nethsm`:
   ```
   create sys crypto key test_key gen-certificate common-name test_thales.com
   security-type nethsm
   ```
4. Verify that the key was created.
   ```
   list sys crypto key test_key.key
   ```
   Information about the key displays:

```
sys crypto key test_key.key {
key-id <32-digit string>
key-size 2048
key-type rsa-private
security-type nethsm
}
```

When you generate a key/certificate using `tmsh`, the system creates a HSM private key. It also creates a local key, which points to the HSM key, residing in the HSM.

Creating a DNSSEC key using an external HSM key and certificate

Before you create a DNSSEC key using an external key and certificate, make sure that you have generated a key and certificate using Thales nShield Connect, and that you have loaded the key and certificate.

You can create manually managed DNSSEC zone-signing and key-signing keys for use with an external HSM. For more information, see *Configuring DNSSEC with an external HSM* in *BIG-IP® DNS Services: Implementations* at `http://support.f5.com`.

Additional Information

Creating a backup of the Thales RFS

Before you back up the RFS, make sure that the Thales nShield Connect Remote File System (RFS) server is installed on your network.

You back up the `/shared/nfast/kmdata/local/` directory of the RFS to a secure location, so that you can recover the RFS state, if needed. The RFS contains all of the Thales nShield Connect keys.

1. If the RFS is not installed on the BIG-IP system, rename the `/shared/nfast` directory to `/shared/nfast.org`.
   This directory can be used to recover old data, if necessary.
2. Follow the Thales best practices for backing up the RFS server.

Upgrading the BIG-IP software when using the Thales HSM

After a BIG-IP® system software or hotfix upgrade, you must run the Thales client setup script to restore your default Thales configuration. Any local keys and certificates you loaded into the BIG-IP system before upgrading (using the command `tmsh install sys crypto`) appear in the upgrade partition, but they are usable only after you run the Thales client setup script. If you are restoring the Thales client on a VIPRION® system, you run the configuration script only on the primary blade, and then the system propagates the configuration to the additional active blades.

***Note:** If you will need CSRs, keys, or certs that were not loaded into the BIG-IP system, before you upgrade, copy the files into the `/shared` directory. After the upgrade, copy them back to their appropriate directories in the new partition: `/config/ssl/ssl.key/`, `/config/ssl/ssl.crt`, or `/config/ssl/ssl.csr`.*

1. Log in to the command-line interface of the BIG-IP system using an account with administrator privileges.
2. Run one of these scripts, using the arguments that are appropriate for your configuration:
   - If the BIG-IP is an RFS server in addition to being a Thales client, use: `nethsm-thales-rfs-install.sh` and `nethsm-thales-install.sh`
   - If the BIG-IP is only a Thales client use: `nethsm-thales-install.sh`

The protected keys, which are stored in `/opt/nfast/kmdata/`, are available in the new partition, regardless of whether the keys and certs were loaded into the BIG-IP system.

Uninstalling Thales nShield Connect components from the BIG-IP system

If you no longer need to use the Thales nShield Connect on a BIG-IP® system, you should uninstall the files.

1. Log in to the command-line interface of the system using an account with administrator privileges.
2. Uninstall the client and clean up.

```
nethsm-thales-install.sh -u [-v]
```

Replacing a broken Thales HSM without breaking existing keys

You can replace a broken Thales HSM without destroying existing keys.

1. Turn on the new HSM, and give it an IP address, netmask, and default route.

   ```
   nethsm-thales-install.sh -u [-v]
   ```

2. Reconfigure the RFS, so that you can point the HSM to it. If the RFS is a BIG-IP® system, use `nethsm-thales-rfs-install.sh --hsm_ip_addr=<new HSM IP>`. When the script asks if you want to uninstall, type `yes`. When the script asks if you want to overwrite the config, type `yes`. When the script asks if you want to use the existing Security World, type `yes`.
3. Using the front panel on the HSM, configure the HSM to know its RFS. Once this is done, using the front panel again, add at least the BIG-IP system and the RFS to the HSM as clients.
4. Using the front panel of the HSM and a quorum of the ACS, load the Security World onto the HSM.
5. On the RFS, unenroll the old HSM, because otherwise it will still show up in enquiry results.

   ```
   nethsmenroll --force -r <old HSM IP> <old HSM ESM> <old HSM hash>
   ```

   Note that this requires the ESM and HSM hash from the broken machine. The command `anonkneti` will not work because the old HSM is broken or already removed from the network. This needs to be done even if the RFS is a BIG-IP system and the `nethsm-thales-rfs-install.sh` command was used.
6. Rerun the install script on any BIG-IP systems that use the HSM. When that script asks if you want to uninstall, type `yes`. When that script asks if you want to backup, type whichever you prefer.
7. On each BIG-IP system, unenroll the old HSM, because it will still show up in enquiry results.

   ```
   nethsmenroll --force -r <old HSM IP> <old HSM ESM> <old HSM hash>
   ```

   Note that this requires the ESM and HSM hash from the broken machine. The command `anonkneti` will not work, since the old HSM is broken or already removed from the network.

fipskey.nethsm utility options

The `fipskey.nethsm` utility includes these options:

Option	Description
`-o`	Name applied to `.key`, `.csr`, and `.crt` output files ***Important:*** *This parameter is required.*
`-c [token | module | softcard]`	Type of protection (default value is `token`)
`-e [hex]`	Public exponent to use when generating RSA keys only. ***Note:*** *Do not provide a value for this option, unless advised to do so by F5® Technical Support.*
`-g [sha1]`	Digest used to sign key and certificate
`-k [name]`	Key name
`-m [yes | no]`	Store key in non-volatile RAM

Option	Description
`-n [integer]`	Slot number to read cards from
`-r [yes | no]`	Key recovery available
`-s [integer]`	Size of key/certificate pair (in bits)
`-t [RSA]`	Key type
`-v [yes | no]`	Verification available
`-C`	Country identifier
`-D`	Domain name
`-E`	Email address for key contact
`-L`	Locality identifier
`-N`	Substitute alternative name ***Note:*** *Applies only to SafeNet Luna HSM.*
`-O`	Organization identifier
`-P`	Province identifier
`-U`	Organization unit identifier

nethsm-thales-install.sh utility options

The `nethsm-thales-install.sh` utility includes these options:

Option	Description
`-h`	Displays help.
`-v`	Prints verbose output about operations.
`--hsm_ip_addr=<ip_addr>`	Thales HSM IP address(es). For multiple HSMs, use a double-quoted value with space-separated IP addresses (such as `--hsm_ip_addr="10.10.10.100.10.10.10.101"`).
`--rfs_interface=<interface_name>`	Interface identifier for the local Remote File System (RFS) server. Default is the management interface (eth0).
`--protection=<protection_type>`	Indicates which type of key protection to use. Valid options are [m]odule, [o]cs, or [s]oftcard. When this option is not used, the protection defaults to module protection.
`--verbose=<level>`	Indicates message verbosity level. The default value is zero, and all levels greater than zero indicate verbose output.

nethsm-thales-rfs-install.sh utility options

The `nethsm-thales-rfs-install.sh` utility includes these options:

Option	Description
`-h`	Displays help.
`--u`	Uninstalls Thales software and cleans up Thales directories.
`-v`	Prints verbose output about the executing operations.
`--hsm_ip_addr=<ip_addr>`	Thales HSM IP address(es). For multiple HSMs, use a double-quoted value with space-separated IP addresses (such as `--hsm_ip_addr="10.10.10.100.10.10.10.101 "`).
`--interface=<interface_name>`	Interface identifier of BIG-IP® to be used as Thales HSM Client (eth0). The default is the management interface.
`--num_threads=<threads>`	Indicates the number of threads pkcs11d will use. The default is 20.
`--rfs_interface=<interface_name>`	Local Remote File System (RFS) server interface name (eth0).
`--rfs_ip_addr=<ip_addr>`	Remote RFS server IP address.
`--rfs_username=<ssh_username>`	Remote RFS server username for SSH login.
`--protection=<protection_type>`	Indicates which type of key protection to use. Valid options are [m]odule, [o]cs, or [s]oftcard. When this option is not used, the protection defaults to module protection.
`--verbose=<level>`	Indicates message verbosity level. The default value is zero, and all levels greater than zero indicate verbose output.

Legal Notices

Legal notices

Publication Date

This document was published on November 13, 2017.

Publication Number

MAN-0495-05

Copyright

F5 Networks, Inc. (F5) believes the information it furnishes to be accurate and reliable. However, F5 assumes no responsibility for the use of this information, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, copyright, or other intellectual property right of F5 except as specifically described by applicable user licenses. F5 reserves the right to change specifications at any time without notice.

Trademarks

For a current list of F5 trademarks and service marks, see *http://www.f5.com/about/guidelines-policies/trademarks*.

All other product and company names herein may be trademarks of their respective owners.

Patents

This product may be protected by one or more patents indicated at: *https://f5.com/about-us/policies/patents*.

Link Controller Availability

This product is not currently available in the U.S.

Export Regulation Notice

This product may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this product from the United States.

RF Interference Warning

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

FCC Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This unit generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

Canadian Regulatory Compliance

This Class A digital apparatus complies with Canadian ICES-003.

Standards Compliance

This product conforms to the IEC, European Union, ANSI/UL and Canadian CSA standards applicable to Information Technology products at the time of manufacture.

Index