BIG-IP® Systems and Thales® HSM: Implementations

Version 11.5

Table of Contents

Legal Notices

Publication Date

This document was published on December 9, 2015.

Publication Number

MAN-0495-00

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

AAM, Access Policy Manager, Advanced Client Authentication, Advanced Firewall Manager, Advanced Routing, AFM, APM, Application Acceleration Manager, Application Security Manager, ARX, AskF5, ASM, BIG-IP, BIG-IQ, Cloud Extender, CloudFucious, Cloud Manager, Clustered Multiprocessing, CMP, COHESION, Data Manager, DevCentral, DevCentral [DESIGN], DNS Express, DSC, DSI, Edge Client, Edge Gateway, Edge Portal, ELEVATE, EM, Enterprise Manager, ENGAGE, F5, F5 [DESIGN], F5 Certified [DESIGN], F5 Networks, F5 SalesXchange [DESIGN], F5 Synthesis, f5 Synthesis, F5 Synthesis [DESIGN], F5 TechXchange [DESIGN], Fast Application Proxy, Fast Cache, FirePass, Global Traffic Manager, GTM, GUARDIAN, iApps, IBR, Intelligent Browser Referencing, Intelligent Compression, IPv6 Gateway, iControl, iHealth, iQuery, iRules, iRules OnDemand, iSession, L7 Rate Shaping, LC, Link Controller, Local Traffic Manager, LTM, LineRate, LineRate Systems [DESIGN], LROS, LTM, Message Security Manager, MSM, OneConnect, Packet Velocity, PEM, Policy Enforcement Manager, Protocol Security Manager, PSM, Real Traffic Policy Builder, SalesXchange, ScaleN, Signalling Delivery Controller, SDC, SSL Acceleration, software designed applications services, SDAC (except in Japan), StrongBox, SuperVIP, SYN Check, TCP Express, TDR, TechXchange, TMOS, TotALL, Traffic Management Operating System, Traffix Systems, Traffix Systems (DESIGN), Transparent Data Reduction, UNITY, VAULT, vCMP, VE F5 [DESIGN], Versafe, Versafe [DESIGN], VIPRION, Virtual Clustered Multiprocessing, WebSafe, and ZoneRunner, are trademarks or service marks of F5 Networks, Inc., in the U.S. and other countries, and may not be used without F5's express written consent.

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: *http://www.f5.com/about/guidelines-policies/patents*

Acknowledgments

This product includes software developed by Bill Paul.

This product includes software developed by Jonathan Stone.

This product includes software developed by Manuel Bouyer.

This product includes software developed by Paul Richards.

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes software developed by the Politecnico di Torino, and its contributors.

This product includes software developed by the Swedish Institute of Computer Science and its contributors.

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by the Computer Systems Engineering Group at the Lawrence Berkeley Laboratory.

This product includes software developed by Christopher G. Demetriou for the NetBSD Project.

This product includes software developed by Adam Glass.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Dean Huxley.

This product includes software developed by John Kohl.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Herb Peyerl.

This product includes software developed by Jochen Pohl for the NetBSD Project.

This product includes software developed by Chris Provenzano.

This product includes software developed by Theo de Raadt.

This product includes software developed by David Muir Sharnoff.

This product includes software developed by SigmaSoft, Th. Lockert.

This product includes software developed for the NetBSD Project by Jason R. Thorpe.

This product includes software developed by Jason R. Thorpe for And Communications, http://www.and.com.

This product includes software developed for the NetBSD Project by Frank Van der Linden.

This product includes software developed for the NetBSD Project by John M. Vinopal.

This product includes software developed by Christos Zoulas.

This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.

This product includes software developed by Balazs Scheidler (bazsi@balabit.hu), which is protected under the GNU Public License.

This product includes software developed by Niels Mueller (nisse@lysator.liu.se), which is protected under the GNU Public License.

In the following statement, *This software* refers to the Mitsumi CD-ROM driver: This software was developed by Holger Veit and Brian Moore for use with 386BSD and similar operating systems. *Similar operating systems* includes mainly non-profit oriented systems for research and education, including but not restricted to NetBSD, FreeBSD, Mach (by CMU).

This product includes software developed by the Apache Group for use in the Apache HTTP server project (http://www.apache.org/).

This product includes software licensed from Richard H. Porter under the GNU Library General Public License (© 1998, Red Hat Software), www.gnu.org/copyleft/lgpl.html.

This product includes the standard version of Perl software licensed under the Perl Artistic License (© 1997, 1998 Tom Christiansen and Nathan Torkington). All rights reserved. You may find the most current standard version of Perl at http://www.perl.com.

This product includes software developed by Jared Minch.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product contains software based on oprofile, which is protected under the GNU Public License.

This product includes RRDtool software developed by Tobi Oetiker (http://www.rrdtool.com/index.html) and licensed under the GNU General Public License.

This product contains software licensed from Dr. Brian Gladman under the GNU General Public License (GPL).

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

This product includes Hypersonic SQL.

This product contains software developed by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and others.

This product includes software developed by the Internet Software Consortium.

This product includes software developed by Nominum, Inc. (http://www.nominum.com).

This product contains software developed by Broadcom Corporation, which is protected under the GNU Public License.

This product contains software developed by MaxMind LLC, and is protected under the GNU Lesser General Public License, as published by the Free Software Foundation.

This product includes Intel QuickAssist kernel module, library, and headers software licensed under the GNU General Public License (GPL).

This product includes software licensed from Gerald Combs (gerald@wireshark.org) under the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or any later version. Copyright ©1998 Gerald Combs.

This product includes software developed by Thomas Williams and Colin Kelley. Copyright ©1986 - 1993, 1998, 2004, 2007

Permission to use, copy, and distribute this software and its documentation for any purpose with or without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Permission to modify the software is granted, but not the right to distribute the complete modified source code. Modifications are to be distributed as patches to the released version. Permission to distribute binaries produced by compiling modified sources is granted, provided you

1. distribute the corresponding source modifications from the released version in the form of a patch file along with the binaries,

2. add special version identification to distinguish your version in addition to the base release version number,
3. provide your name and address as the primary contact for the support of your modified version, and
4. retain our contact information in regard to use of the base software.

Permission to distribute the released version of the source code along with corresponding source modifications in the form of a patch file is granted with same provisions 2 through 4 for binary distributions. This software is provided "as is" without express or implied warranty to the extent permitted by applicable law.

This product contains software developed by Google, Inc. Copyright ©2011 Google, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product includes software developed by Jeremy Ashkenas and DocumentCloud, and distributed under the MIT license. Copyright © 2010-2013 Jeremy Ashkenas, DocumentCloud.

This product includes gson software, distributed under the Apache License version 2.0. Copyright © 2008-2011 Google Inc.

Chapter 1

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. The RFS helps automate the key distribution process, but you are not required to use RFS with this solution.

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/products-and-services/products-and-services
/hardware-security-modules/general-purpose-hsms/nshield-connect
```

Note: *The BIG-IP system does not support Thales nShield Connect in 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 address and the cluster member IP address for each blade installed in the chassis to the allowed list.
- 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`.
- You have created the Thales Security World (security architecture).
- The BIG-IP system is licensed for "External Interface and Network HSM."

Important: *If the BIG-IP system contains an internal HSM (FIPS Cavium card), you cannot use internal and external HSMs at the same time. In this case, the external HSM takes precedence. To use the internal HSM, you must uninstall the Thales HSM client.*

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 (Release 11.40 or higher)

- The nShield_Connect_and_netHSM_User_Guide.pdf

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
Setting up the RFS on the BIG-IP system (optional)
Setting up the Thales nShield Connect client on the BIG-IP system
Configuring the Thales nShield Connect client for multiple HSMs in an HA group

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 v11.40. We expect later versions to be compatible. 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 BIG-IP system using an account with administrator privileges.
2. Create a directory under `/shared` named `thales_install/amd64/nfast`.
   ```
   mkdir /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=192.27.13.59 --rfs_interface=eth0
   ```

After you set up the RFS on the BIG-IP system, you must set up the Thales nShield Connect client on each BIG-IP system that you want to use with the Thales nShield Connect device.

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. 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 ut the Thales client on a VIPRION® system, you must install and configure the client software on each blade installed in the chassis. The Thales software and configuration files do not sync between blades.

1. Log in to the command-line interface of the BIG-IP system using an account with administrator privileges.
2. Set the HSM vendor, including the password used to log in to the HSM:

   ```
   tmsh create sys crypto fips external-hsm vendor thales password <password>
   ```

   The password is required only if you are using softcard protection for keys.
3. 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.
4. 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_IP_address>
--rfs_username=<remote_RFS_server_username_for_SSH_login>
--interface=<Interface_name_of_Thales_client_on_BIG-IP>
```

The following example sets up the client where the Thales nShield Connect device has an IP address of `192.27.13.59`, the remote RFS has an IP address of `192.27.13.58`, the user name for an SSH login to the RFS is `root`, and the Thales client interface is the management (`eth0`) interface:

```
nethsm-thales-install.sh --hsm_ip_addr=192.27.13.59
--rfs_ip_addr=192.27.12.58 --rfs_username=root --interface=eth0
```

- 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 `172.27.13.59` and the RFS is installed on the BIG-IP system using the `eth0` interface:

```
nethsm-thales-install.sh --hsm_ip_addr=172.27.13.59 --rfs_interface=eth0
```

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 BIG-IP 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 the 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. 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
```

Chapter 2

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/products-and-services/products-and-services
/hardware-security-modules/general-purpose-hsms/nshield-connect
```

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 softcard 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 a token-, module-, or softcard-protected key and certificate, and creating a client SSL profile to use the key and certificate.

Task list

Configuring the key protection type

Generating a token-, module-, or softcard-protected key/certificate using Thales nShield Connect

Configuring hardware-protected HSM keys using tmsh

Adding certificates using tmsh

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

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

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 module key protection by opening the configuration file `/opt/nfast/cknfastrc` and changing the value of the following parameter to `1`, as shown.

      ```
      CKNFAST_NO_ACCELERATOR_SLOTS=1
      ```

   2. Disable softcard key protection by moving any previously created `softcard*` files from the `/opt/nfast/kmdata/local` directory to the `/opt/nfast/kmdata/` directory.
   3. Enable OCS key protection by creating the OCS cardset using the Thales-provided `createocs` utility.

   softcard

   1. Disable module protection by opening the configuration file `/opt/nfast/cknfastrc` and changing the value of the following parameter to `1`, as shown.

      ```
      CKNFAST_NO_ACCELERATOR_SLOTS=1
      ```

   2. Disable OCS key protection by moving any previously created `cards*` files from the `/opt/nfast/kmdata/` directory to the `/opt/nfast/kmdata/local` directory.
   3. 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.*

   ---

3. After you make any configuration changes, you must restart the `pkcsd11` and `tmm` services.

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

Generating a token-, module-, or softcard-protected key/certificate using Thales nShield Connect

Before you generate a token-, module-, or softcard-protected key/certificate, make sure that the Thales nShield Connect client is running on the BIG-IP® system.

You can use the `fipskey.nethsm` utility to generate token-, module-, or softcard-protected private keys and self-signed certificates on the BIG-IP system. You can use the generated `.csr` file to request a signed certificate from a certificate authority (CA).

1. Log in to the command-line interface of the system using an account with administrator privileges.
2. Generate a key, using one of these options:
   - Generate a token-protected key (the default method):

     ```
     fipskey.nethsm --genkey -o <output_file> -c token
     ```
   - Generate a module-protected key:

     ```
     fipskey.nethsm --genkey -o <output_file> -c module
     ```
   - Generate a softcard-protected key:

     ```
     fipskey.nethsm --genkey -o <output_file> -c softcard
     ```

   This example generates the four files that follow, using the default protection type (token):

   ```
   fipskey.nethsm --genkey -o my_key -c token
   ```

   - `/config/ssl/ssl.key/my_key.key` (local key)
   - `/config/ssl/ssl.csr/my_key.csr` (CSR file)
   - `/config/ssl/ssl.crt/my_key.crt` (self-signed certificate)
   - `/opt/nfast/kmdata/local/<filename>` (protected key)

   The local key points to the protected key, which is encrypted.

After you generate a key and certificates, you need to load the local key into the BIG-IP configuration using `tmsh`.

Configuring hardware-protected HSM keys using tmsh

You can use the Traffic Management Shell (`tmsh`) to load the corresponding local HSM (FIPS) keys into the BIG-IP® system.

Note: *This procedure loads the local key, not the actual hardware key, which never leaves the HSM.*

1. Log in to the command-line interface of the BIG-IP system using an account with administrator privileges.
2. Open the Traffic Management Shell (`tmsh`).

   ```
   tmsh
   ```
3. Configure the local key.

   ```
   install sys crypto key <key_object_name> from-local-file <keyname>
   ```

   This example loads the external HSM key named `my_key.key` from a local key file stored in the `/config/ssl/ssl.key/` directory:

   ```
   install sys crypto key my_key.key from-local-file
   /config/ssl/ssl.key/my_key.key
   ```

   The Thales client software maps the local key to the appropriate protected key.

Adding certificates using tmsh

You can use the Traffic Management Shell (`tmsh`) to add existing certificates to the BIG-IP® system configuration.

1. Log in to the command-line interface of the BIG-IP system using an account with administrator privileges.
2. Open the Traffic Management Shell (`tmsh`).
   ```
   tmsh
   ```
3. Add the certificate.
   ```
   install sys crypto cert <cert_object_name> from-local-file <path_to_cert_file>
   ```
   This example loads the certificate named `my_key.crt` from a local certificate file stored in the `/config/ssl/ssl.crt/` directory:
   ```
   install sys crypto cert my_key.crt from-local-file
   /config/ssl/ssl.crt/my_key.crt
   ```

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_uced028e5251b7b
6891e7e59dec5428d871f92241b-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.*

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

After you have installed the external HSM key and certificate to the BIG-IP® system, you can use the key and certificate as part of a client SSL profile.

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. Select clientssl in the Parent Profile list.
5. From the **Configuration** list, select **Advanced**.

   This selection makes it possible for you to modify additional default settings.
6. Select the **Custom** check box for **Configuration**.
   The settings in the Configuration area become available for configuring.
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.

Chapter 3

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 Global Traffic Manager™ (GTM™), 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/products-and-services/products-and-services
/hardware-security-modules/general-purpose-hsms/nshield-connect
```

Task list

Generating an external key for creating manually managed DNSSEC keys
Configuring hardware-protected HSM keys using tmsh
Adding certificates using tmsh
Creating a DNSSEC key using an external HSM key and certificate

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® GTM™ devices in the configuration synchronization group.

You can use the `fipskey.nethsm` utility to generate keys and self-signed certificates to be used to create manually managed DNSSEC private keys. You can use the generated `.csr` file to request a signed certificate from a certificate authority (CA).

Tip: *For information about creating automatically managed DNSSEC private keys, see Configuring DNSSEC with an external HSM in BIG-IP® DNS Services: Implementations at* `http://support.f5.com`.

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

   ```
   fipskey.nethsm --genkey -o <output_file>
   ```

   Note: *The output file name cannot include periods.*

   This example generates four files, using the default protection type (token):

   ```
   fipskey.nethsm --genkey -o my_key
   ```

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

   The local key points to the protected key, which is encrypted.

After you generate a key and certificates, you need to load the local key into the BIG-IP configuration using `tmsh`.

Configuring hardware-protected HSM keys using tmsh

You can use the Traffic Management Shell (`tmsh`) to load the corresponding local HSM (FIPS) keys into the BIG-IP® system.

Note: *This procedure loads the local key, not the actual hardware key, which never leaves the HSM.*

1. Log in to the command-line interface of the BIG-IP system using an account with administrator privileges.
2. Open the Traffic Management Shell (`tmsh`).
   ```
   tmsh
   ```
3. Configure the local key.
   ```
   install sys crypto key <key_object_name> from-local-file <keyname>
   ```
   This example loads the external HSM key named `my_key.key` from a local key file stored in the `/config/ssl/ssl.key/` directory:
   ```
   install sys crypto key my_key.key from-local-file
   /config/ssl/ssl.key/my_key.key
   ```
   The Thales client software maps the local key to the appropriate protected key.

Adding certificates using tmsh

You can use the Traffic Management Shell (`tmsh`) to add existing certificates to the BIG-IP® system configuration.

1. Log in to the command-line interface of the BIG-IP system using an account with administrator privileges.
2. Open the Traffic Management Shell (`tmsh`).
   ```
   tmsh
   ```
3. Add the certificate.
   ```
   install sys crypto cert <cert_object_name> from-local-file <path_to_cert_file>
   ```
   This example loads the certificate named `my_key.crt` from a local certificate file stored in the `/config/ssl/ssl.crt/` directory:
   ```
   install sys crypto cert my_key.crt from-local-file
   /config/ssl/ssl.crt/my_key.crt
   ```

Creating a DNSSEC key using an external HSM key and certificate

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

Appendix A

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.

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 must run the configuration script on each blade.

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.

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`)

Option	Description
`-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
`-n [integer]`	Slot number to read cards from ***Note:*** *Not designed to work with the Thales HSM devices.*
`-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
`-u`	Uninstalls Thales software and cleans up.
`-v`	Prints verbose output about operations
`--hsm_ip_addr=<ip_addr>`	Thales HSM IP address
`--rfs_ip_addr=<ip_addr>`	Remote RFS server IP address
`--rfs_username=<ssh username>`	Remote RFS server user name for SSH login

Option	Description
`--rfs_interface=<interface_name>`	Interface identifier for the Remote File System (RFS) server. Default is the management interface (eth0).
`--interface=<interface_name>`	Interface identifier for the BIG-IP® system to be used as the Thales HSM client. Default is the management interface (eth0).
`--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
`-v`	Prints verbose output about operations
`--hsm_ip_addr=<ip_addr>`	Thales HSM IP address
`--rfs_ip_addr=<ip_addr>`	Remote File System (RFS) server IP address
`--rfs_username=<ssh_user_name>`	RFS server username for SSH login
`--rfs_interface=<interface_name>`	Interface identifier for the BIG-IP® system used as the Thales HSM client. Default is the management interface (eth0).
`--verbose=<level>`	Indicates message verbosity level. The default value is zero, and all levels greater than zero indicate verbose output.

Index