Fortanix®

Integration Guide

USING FORTANIX DATA SECURITY MANAGER FOR NGINX AND APACHE TLS PRIVATE KEYS

Fortanix Inc. | 800 West El Camino Real | Suite 180 | Mountain View, CA 94040 | United States of America |
+1 (650) 943-2484 | info@fortanix.com | www.fortanix.com

1.0 INTRODUCTION 3
1.1 Overview 3
2.0 PREREQUISITES 3
2.1 Installing Pre-Requisites 4
3.0 SETTING UP FORTANIX DATA SECURITY MANAGER 5
4.0 SETTING UP OPENSSL TO USE PKCS#11 5
5.0 TESTING SETUP 6
5.1 Testing the Engine Operation 6
5.2 Testing the PKCS#11 Library 6
5.2.1 Verify PKCS#11 Library 7
5.2.2 List Keys from Fortanix Data Security Manager 7
5.2.3 Generate a Key in Fortanix Data Security Manager 7
5.3 Testing the OpenSSL with PKCS#11 Library 8
5.3.1 Encrypting Data With the Key in Fortanix Data Security Manager 8
6.0 IMPORTING TLS KEY AND CERTIFICATE INTO FORTANIX DATA SECURITY MANAGER 9
6.1 Import Using WebUI 9
6.2 Import Using pkcs11-tool 10
7.0 CONFIGURING NGINX TO USE KEY FROM FORTANIX DATA SECURITY MANAGER 11
7.1 Update NGINX Configuration File 11
7.2 Update NGINX Startup File 12
7.3 Restarting NGINX 13
7.4 Verification 13
8.0 CONFIGURING APACHE TO USE KEY FROM FORTANIX DATA SECURITY MANAGER 13
8.1 Create pkcs11 config file 13
8.2 Red hat 7 – set up mod nss and update nss configuration file 14
8.3 Red hat 8 - Update APACHE SSL Configuration File 15

8.4 Update apache Startup File 16
8.5 Restarting apache 17
8.6 Verification 18
9.0 TROUBLESHOOTING 18
10.0 DOCUMENT INFORMATION 19
10.1 Document Location 19
10.2 Document Updates 19
10.3 Revision History ..... Error! Bookmark not defined.

1.0 INTRODUCTION

This article describes how to use **Fortanix Data Security Manager (DSM)** for protecting **NGINX and Apache TLS Keys on Red Hat.** It also contains the information that a user requires for:

- Setting up Fortanix DSM
- Setting up Open SSL to use PKCS11
- Testing the setup
- Configuring NGINX to use key from Fortanix DSM
- Configuring Apache to use key from Fortanix DSM

1.1 OVERVIEW

Fortanix DSM can be used to protect the TLS private key for your NGINX and Apache server, keeping the private key secure even if the host running these servers is compromised.

This document describes how to set up your NGINX and Apache server running on Red Hat to use a TLS private key stored in Fortanix DSM.

2.0 PREREQUISITES

The following software components are required. The procedure described here was tested with the versions mentioned below:

- NGINX
  - Red Hat Enterprise Linux Server (7.6)
  - NGINX (1.18.0)
  - OpenSSL (1.0.2-k)
  - OpenSC (0.19.0)
  - OpenSSL PKCS11 engine (0.4.10)
- Apache
  - Red Hat Enterprise Linux Server (8.2)
    - Apache (2.4.37)
    - mod_ssl (2.4.37)
    - OpenSSL (1.1.1-c)
    - OpenSC (0.19.0)
    - OpenSSL PKCS11 engine (0.4.10)
  - Red Hat Enterprise Linux Server (7.8)

  - Apache (2.4.6)
  - mod_nss (1.0.14)
  - nss-tools (3.44.0)
  - OpenSSL (1.0.2k)
  - OpenSC (0.19.0)
  - OpenSSL PKCS11 engine (0.4.10)
- Fortanix PKCS11 library (3.19.1348)

2.1 INSTALLING PRE-REQUISITES

- NGINX

  If you need to install NGINX, please refer to the instructions in the URL:

  https://www.nginx.com/resources/wiki/start/topics/tutorials/install/
- Apache and mod_ssl

  If you need to install Apache, install it by running the following commands:

  On Red Hat 8 you will need to install “mod_ssl” and on Red Hat 7 you will need to install “mod_nss”.

```
sudo yum install httpd
Red Hat 8
sudo yum install mod_ssl

Red Hat 7
sudo yum install mod_nss
sudo yum install nss-tools
```

- OpenSC

  Run the following command to run OpenSC. This package provides PKCS#11 tools and PKCS#11 engine plugin for the OpenSSL library which allows accessing PKCS#11 modules in a semi-transparent way.

```
sudo yum install opensc
```

- Install OpenSSL PKCS#11 engine using the following command:

```
wget https://download-ib01.fedoraproject.org/pub/epel/7/x86_64/Packages/o/openssl-pkcs11-0.4.10-1.el7.x86_64.rpm
sudo rpm -i openssl-pkcs11-0.4.10-1.el7.x86_64.rpm
```

- Fortanix PKCS#11 library

  Download Fortanix PKCS#11 library RPM package from the following URL:

  https://support.fortanix.com/hc/en-us/articles/360018312391-PKCS-11

  Install it by running the following command:

```
sudo rpm -i <package name>, for example
sudo rpm -i fortanix-pkcs11-3.19.1348-0.x86_64.rpm
```

3.0 SETTING UP FORTANIX DATA SECURITY MANAGER

For this integration, create a group and app in Fortanix DSM. For instructions on creating a group and app in Fortanix DSM please check the Getting Started Guide.

Note the API key of app you added. We will need this in the next steps where `API_KEY` is mentioned.

4.0 SETTING UP OPENSSL TO USE PKCS#11

1. Save the current OpenSSL configuration file (`openssl.cnf`) before making changes:

```
sudo cp /etc/pki/tls/openssl.cnf /etc/pki/tls/openssl.cnf.bak
```

2. Now set up OpenSSL configuration file as follows to have the PKCS#11 engine section.

```
# PKCS11 engine config
openssl_conf = openssl_def
[openssl_def]
engines = engine_section
[req]
distinguished_name = req_distinguished_name
[req_distinguished_name]
# empty.
```

```
[engine_section]
pkcs11 = pkcs11_section
[pkcs11_section]
engine_id = pkcs11
dynamic_path = /usr/lib64/openssl/engines/libpkcs11.so
MODULE_PATH = /opt/fortanix/pkcs11/fortanix_pkcs11.so
PIN = API_KEY
init = 1
```

NOTE: Replace `API_KEY` in the highlighted line with the API key of the app you created earlier.

5.0 TESTING SETUP

Use the following commands to verify the setup is correct:

5.1 TESTING THE ENGINE OPERATION

1. To verify that the engine is properly operating run the following command.

```
openssl engine pkcs11 -t
```

This should return:

```
(pkcs11) pkcs11 engine
     [ available ]
```

5.2 TESTING THE PKCS#11 LIBRARY

1. Export the following environment variables (replace values as appropriate for your environment) and use pkcs11-tool as shown below:

```
export OPENSSL_CONF=/etc/pki/tls/openssl.cnf
export FORTANIX_API_ENDPOINT=https://sdkms.fortanix.com
export FORTANIX_API_KEY=API_KEY
```

NOTE: Replace `API_KEY` in the with the API key of the app you created earlier.

5.2.1 VERIFY PKCS#11 LIBRARY

Run the following command to verify if the PKCS#11 library works correctly:

```
pkcs11-tool --module /opt/fortanix/pkcs11/fortanix_pkcs11.so -I
```

This should return output as follows:

```
*** Cryptoki library has already been initialized ***
Cryptoki version 2.40
Manufacturer     Fortanix
Library          Fortanix PKCS#11 3.19.1348 (ver 3.19)
Using slot 0 with a present token (0x0)
```

5.2.2 LIST KEYS FROM FORTANIX DATA SECURITY MANAGER

Run the following command to list keys from Fortanix DSM.

This should return keys that are already there in the group:

```
pkcs11-tool --module /opt/fortanix/pkcs11/fortanix_pkcs11.so --
login --pin $FORTANIX_API_KEY -O
```

5.2.3 GENERATE A KEY IN FORTANIX DATA SECURITY MANAGER

Run the following command to generate an RSA key in Fortanix DSM:

```
pkcs11-tool --module /opt/fortanix/pkcs11/fortanix_pkcs11.so --
login --pin $FORTANIX_API_KEY -k --id `uuidgen | tr -d -` --label
"Test RSA key" --key-type rsa:2048
```

On success the output will be as follows:

```
*** Cryptoki library has already been initialized ***
Using slot 0 with a present token (0x0)
Key pair generated:
Private Key Object; RSA
```

```
  label:      Test RSA key
  ID:         bc017c71c51447aeaa3af6c38a42f810
  Usage:      decrypt, sign, unwrap
Public Key Object; RSA 2048 bits
  label:      Test RSA key
  ID:         bc017c71c51447aeaa3af6c38a42f810
  Usage:      encrypt, verify, wrap
```

Verify that this command succeeds, and the RSA key is generated in Fortanix DSM. You can verify by looking at the Security Objects page in the WebUI or running the pkcs11-tool command mentioned earlier.

Note the Key ID returned from this command. We will need it in the next steps for verification.

5.3 TESTING THE OPENSSL WITH PKCS#11 LIBRARY

Now we will verify if OpenSSL with Fortanix PKCS#11 library is working correcting by performing some cryptographic operations using the key we generated earlier in Fortanix DSM.

5.3.1 ENCRYPTING DATA WITH THE KEY IN FORTANIX DATA SECURITY MANAGER

Run the following command to encrypt some data with the key we created in the previous step. For ID here, use the Key ID you got in the previous step (in this example, it is `bc017c71c51447aeaa3af6c38a42f810`).

1. Create an input text file that will be used for encryption. For example, you can do the following:

```
cat > input.txt
```

2. Test the encryption.
3. Press CTRL+C to complete the creation of the file.
4. Run the following command to encrypt the file:

```
openssl rsautl -engine pkcs11 -in input.txt -out cipher.data -
encrypt -keyform engine -inkey 1:bc017c71c51447aeaa3af6c38a42f810
```

This should succeed and create a file "`cipher.data`" which contains the encrypted data. You can also verify that the key in Fortanix DSM was used for this operation by looking at the event log for the key.

5. You can also use the following command to decrypt the data and verify that decryption also works:

```
openssl rsautl -engine pkcs11 -in cipher.data -out output.txt -
decrypt -keyform engine -inkey 1:bc017c71c51447aeaa3af6c38a42f810
```

6.0 IMPORTING TLS KEY AND CERTIFICATE INTO FORTANIX DATA SECURITY MANAGER

You can import your existing TLS key and certificate into Fortanix DSM either using WebUI or using the pkcs11-tool.

6.1 IMPORT USING WEBUI

To import using the WebUI, do the following:

1. Log in to the Fortanix DSM UI and select the **Security Objects**  icon from the left panel and then the  icon to open a new security object form.
2. Provide a name for the key and select **IMPORT** option to import a key. (***Note this name, we will need to specify this in the NGINX configuration file***).
3. Select **RSA** in the **Choose a type** section for the security object type.
4. Select **Base64** for the value format and upload your key.
5. Associate the key to the previously created group.
6. Click **IMPORT** to create the security object. (See example screenshot below).

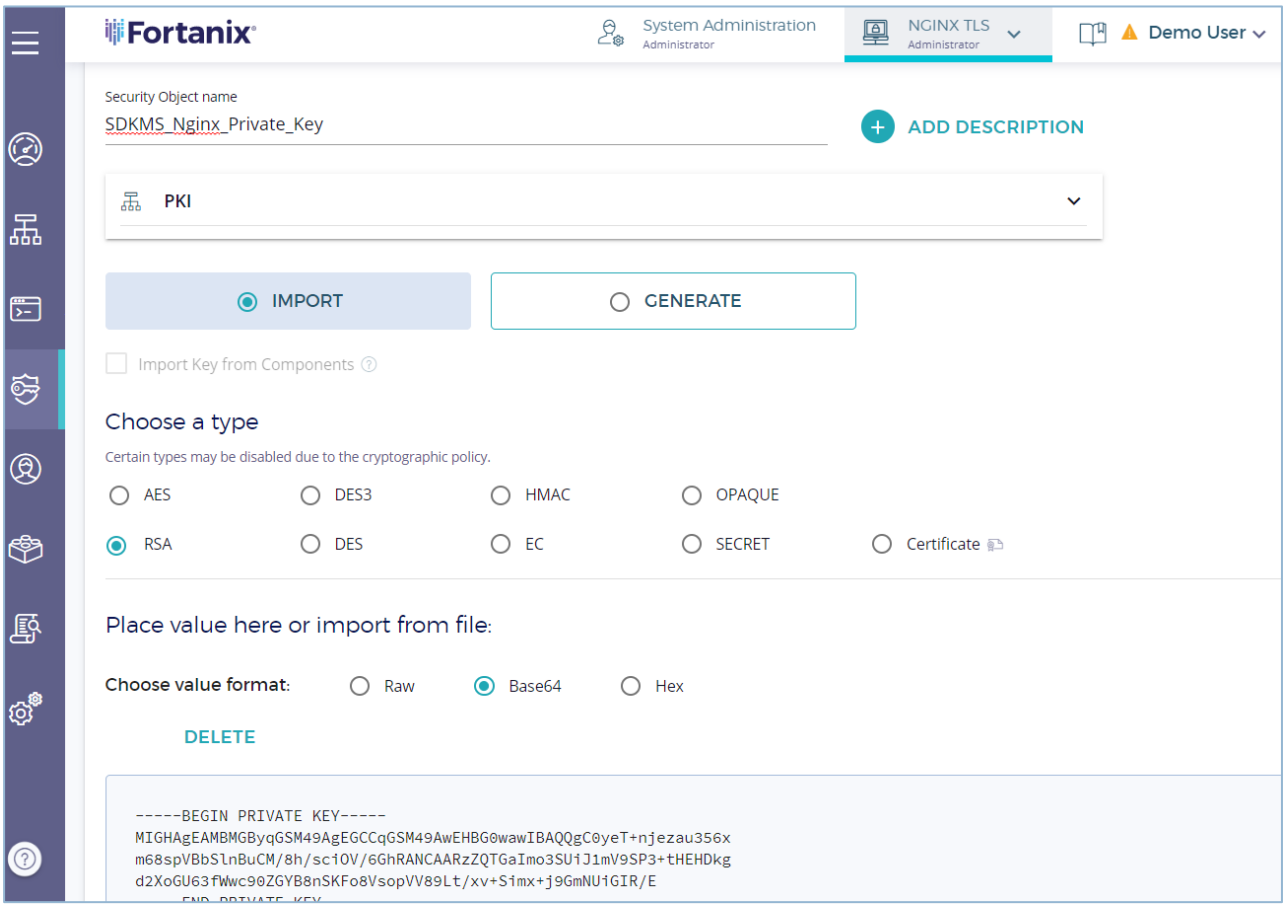

FIGURE 1: IMPORT SECURITY OBJECT

6.2 IMPORT USING PKCS11-TOOL

To import using the pkcs11-tool, do the following:

Assuming that you have the private key in a file `key.pem`, you can use the following commands to import the key into Fortanix DSM.

```
openssl rsa -inform pem -outform der -in key.pem -out key.der

pkcs11-tool --module $PKCS11_LIBRARY --login --pin <API key> --
write-object key.der --type privkey --id `uuidgen | tr -d -` --
label "YOUR_PRIVATE_KEY_LABEL"
```

Note the Key ID value from the output of this command. You will need it when importing a TLS certificate. For the examples mentioned in this document, the key label value will be "`SDKMS_Nginx_Private_Key`".

For Apache running on Red Hat 7, you will also need to import your TLS certificate and associate this certificate to the private key you imported. Assuming that you have the certificate in a file `cert.pem`, you can use the following commands to import the certificate into Fortanix DSM. Please note the `ID_VALUE` in this command is the Key ID that was displayed when you imported the private key in the previous command.

```
openssl rsa -inform pem -outform der -in cert.pem -out cert.der

pkcs11-tool --module /opt/fortanix/pkcs11/fortanix_pkcs11.so --
login --pin <API Key> --write-object cert.der --type cert --id
ID_VALUE --label "SDKMS_Apache_Certificate"
```

For the examples mentioned in this document, the key label value will be "`SDKMS_Nginx_Private_Key`".

7.0 CONFIGURING NGINX TO USE KEY FROM FORTANIX DATA SECURITY MANAGER

Assuming you already have NGINX server configured to use TLS keys, the following steps will guide to you to make changes so that while the certificate file is stored locally, (in this example - `/etc/nginx/certificate.crt`) the corresponding key is securely stored in Fortanix DSM and all crypto operations with this private key are offloaded to Fortanix DSM.

If you do not already have TLS keys and would like to generate TLS keys in Fortanix DSM and generate CSR, you can use OpenSSL and pkcs11-tool to accomplish that. Please see the details at Generating a TLS key and importing a CA-issued certificate.

7.1 UPDATE NGINX CONFIGURATION FILE

Update the NGINX configuration file (`/etc/nginx/nginx.conf`) file to specify the SSL private key which needs to be accessed using the PKCS#11 engine. The highlighted key name after the word '`label`' corresponds to the name of the key in Fortanix DSM when you imported it (in this example : **`SDKMS_Nginx_Private_Key`**). The lines to be edited are shown in the following snippet:

```
##
# SSL Settings
```

```
##
ssl_certificate /etc/nginx/certificate.crt;
ssl_certificate_key engine:pkcs11:label_key_name;
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_prefer_server_ciphers on;
```

 **NOTE**: Only the line "`ssl_certificate_key`" needs to be changed as the certificate file can stay on the server.

7.2 UPDATE NGINX STARTUP FILE

Update the NGINX service start-up file to pass the necessary environment variables for integration with Fortanix DSM. These environment variables allow the PKCS#11 engine and PKCS#11 library to work and point to your Fortanix DSM URL.

Edit the file `/usr/lib/systemd/system/nginx.service` and add the environment variables under the "`Service`" section as follows (see highlighted text). Please update the value of `FORTANIX_API_ENDPOINT` to point to your instance of Fortanix DSM.

```
[Unit]
Description=nginx - high performance web server
Documentation=http://nginx.org/en/docs/
After=network-online.target remote-fs.target nss-lookup.target
Wants=network-online.target
[Service]
Environment="OPENSSL_CONF=/etc/pki/tls/openssl.cnf"
Environment="FORTANIX_API_ENDPOINT=YOUR SDKMS URL"
Type=forking
PIDFile=/var/run/nginx.pid
ExecStart=/usr/sbin/nginx -c /etc/nginx/nginx.conf
ExecReload=/bin/kill -s HUP $MAINPID
ExecStop=/bin/kill -s TERM $MAINPID
[Install]
WantedBy=multi-user.target
```

7.3 RESTARTING NGINX

Now, to restart NGINX and make it use the key stored in Fortanix DSM, do the following:

```
sudo systemctl daemon-reload
sudo systemctl restart nginx
```

7.4 VERIFICATION

To verify that the configuration is working correctly, connect to your NGINX server using a browser. You should be able to connect successfully. You can also check the key usage event logs in Fortanix DSM to verify that the key is being used as you connect to your NGINX server.

8.0 CONFIGURING APACHE TO USE KEY FROM FORTANIX DATA SECURITY MANAGER

This section describes how to configure Apache on Red Hat 8 using mod_ssl and on Red Hat 7 using mod_nss.

Assuming that you already have an Apache server configured to use TLS keys, the following steps will guide to you to make changes so that the private key is securely stored in Fortanix DSM and all crypto operations with this private key are offloaded to Fortanix DSM.

- In case of Red Hat 8, in this example certificate file is stored locally, (`/etc/pki/tls/certs/localhost.crt`)
- In case of Red Hat 7, in this example we store certificate as well in Fortanix DSM.

If you do not already have TLS keys and would like to generate TLS keys in Fortanix DSM and generate CSR, you can use OpenSSL and pkcs11-tool to accomplish that. *Please see the details at Generating a TLS key and importing a CA-issued certificate*.

8.1 CREATE PKCS11 CONFIG FILE

Create PKCS11 config file `/etc/pkcs11/pkcs11.conf` and add the following configuration:

```
api_key = "Your App API Key"
api_endpoint = "Your SDKMS URL"
```

8.2 RED HAT 7 – SET UP MOD NSS AND UPDATE NSS CONFIGURATION FILE

On Red Hat 7 mod_nss is used. You need to set up NSS and update Apache mod_nss configuration file (`/etc/httpd/conf.d/nss.conf`) to specify SSL certificate and the key needs to be fetched using the PKCS#11 engine.

Run the following commands to set up NSS:

```
sudo mkdir /etc/nss
sudo certutil -N -d /etc/nss --empty-password
sudo modutil -add pkcs11 -libfile
/opt/fortanix/pkcs11/fortanix_pkcs11.so -dbdir /etc/nss -force
sudo modutil -enable pkcs11 -dbdir /etc/nss -force
sudo chown -R apache:apache /etc/nss
```

Create a PIN file that contains the Fortanix Token and its Pin by creating a file "`etc/nss/pin.txt`" with the content as follows:

```
Fortanix Token:API_KEY
```

For example,

```
sudo su
echo Fortanix Token: NmM4YjZhOTQtYmEzZS00N....h3 > /etc/nss/pin.txt
```

Edit the NSS configuration file (`/etc/httpd/conf.d/nss.conf`) and make the following changes:

- Update the pass phrase gathering process by changing the value of "`NSSPassPhraseDialog`" as follows:

```
NSSPassPhraseDialog  file:/etc/nss/pin.txt
```

- Update the server certificate database by changing the value of "`NSSCertificateDatabase`" as follows:

```
NSSCertificateDatabase /etc/nss
```

- Update the SSL certificate Nickname by changing the value of "`NSSNickname`" as follows:

```
NSSNickname "Fortanix Token:Your Certificate Label"
```

For example:

```
NSSNickname "Fortanix Token:SDKMS_Apache_Certificate"
```

8.3 RED HAT 8 - UPDATE APACHE SSL CONFIGURATION FILE

On Red Hat 8 mod_ssl is used. We need to update the Apache mod_ssl configuration file (`/etc/httpd/conf.d/ssl.conf`) file to specify the SSL private key which needs to be accessed using PKCS#11 engine.

The highlighted Key ID after the word '`id=`' corresponds to the `CKA_ID` of the key in Fortanix DSM when you imported or created it. As per the PKCS#11 URI format (RFC 7512), the value of '`id`' must be percent-encoded. You can find the `CKA_ID` of the key from the WebUI, by going to the detailed view of the security object then click the "**Attributes**" tab as shown in the following screenshot.

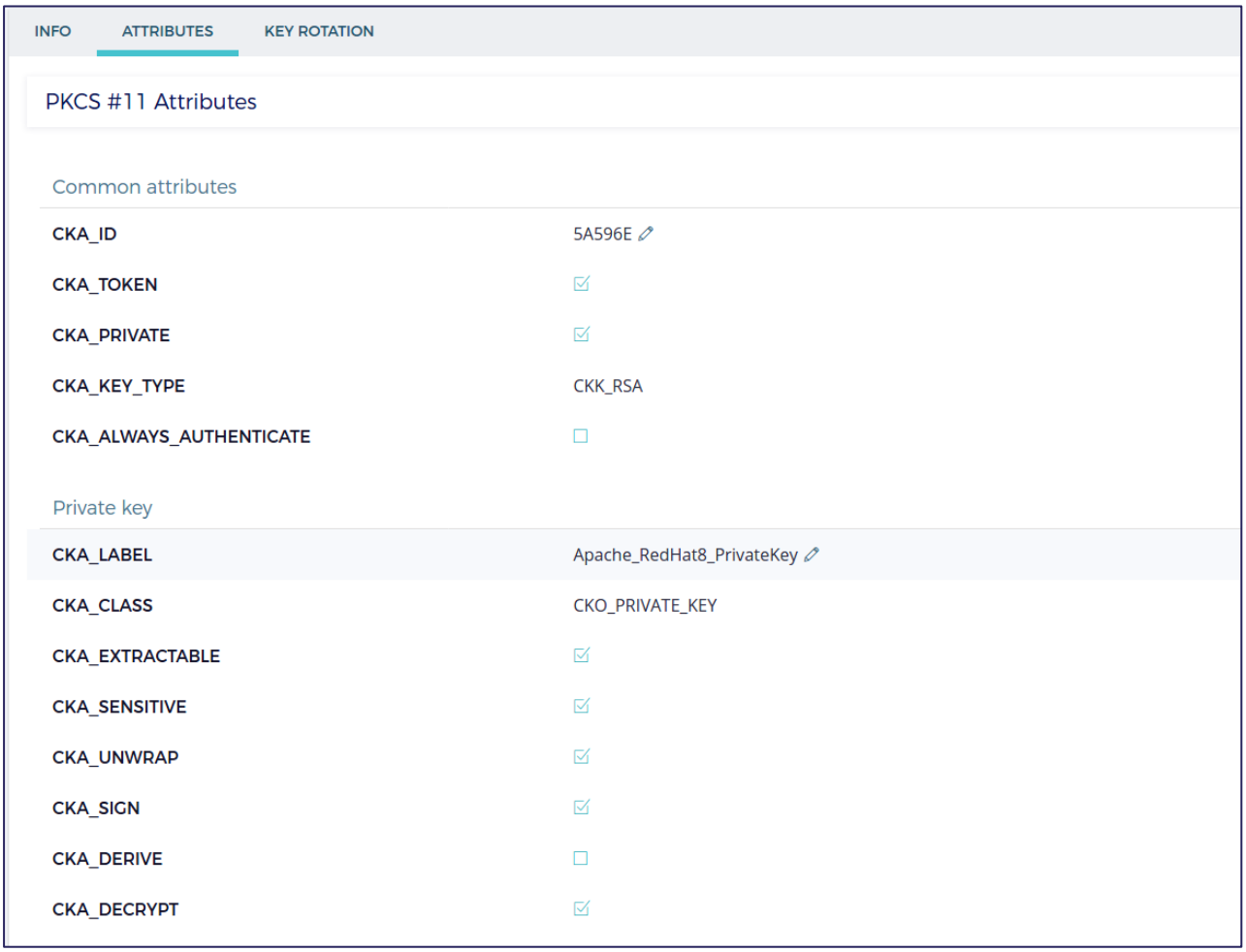

FIGURE 2: ATTRIBUTES TAB OF SECURITY OBJECT

The lines to be edited are shown in the following snippet:

```
SSLCertificateFile /etc/pki/tls/certs/localhost.crt

SSLCertificateKeyFile pkcs11:id=%5A%59%6E;type=private
```

NOTE:

- Only the line "**SSLCertificateKeyFile**" needs to be changed as the certificate file can stay on the server.
- Note the value specified in percent encoded format.

8.4 UPDATE APACHE STARTUP FILE

NOTE: This section is same for Red Hat 7 and 8.

Update the Apache service start-up file to pass the necessary environment variables for integration with Fortanix DSM. These environment variables allow PKCS#11 engine and PKCS#11 library to work and point to your Fortanix DSM URL.

Edit the file `/usr/lib/systemd/system/httpd.service` and add the environment variables under the "`Service`" section as follows (see highlighted text). Please update the value of `FORTANIX_API_ENDPOINT` to point to your instance of Fortanix DSM.

```
[Unit]
Description=The Apache HTTP Server
Wants=httpd-init.service
After=network.target remote-fs.target nss-lookup.target httpd-
init.service
Documentation=man:httpd.service(8)

[Service]
Type=notify
Environment=LANG=C
Environment="OPENSSL_CONF=/etc/pki/tls/openssl.cnf"
Environment="FORTANIX_API_ENDPOINT=YOUR SDKMS URL"
Environment="FORTANIX_PKCS11_NUM_SLOTS=1"

ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND
ExecReload=/usr/sbin/httpd $OPTIONS -k graceful
# Send SIGWINCH for graceful stop
KillSignal=SIGWINCH
KillMode=mixed
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

8.5 RESTARTING APACHE

Now, to restart Apache and make it use the key stored in Fortanix DSM, do the following:

```
sudo systemctl daemon-reload
sudo systemctl restart httpd
```

8.6 VERIFICATION

To verify that the configuration is working correctly, connect to your Apache server using a browser. You should be able to connect successfully. You can also check the key usage event logs in Fortanix DSM to verify that the key is being used as you connect to your Apache server.

9.0 TROUBLESHOOTING

If you see any issues, please use the steps mentioned in the Section 5.0 - *Testing Setup* to verify if each piece of integration is working correctly. If you still see issues, please look at the PKCS#11 logs in `/var/log/messages` and share with the Fortanix support team: (support@fortanix.com).

10.0 DOCUMENT INFORMATION

10.1 DOCUMENT LOCATION

The latest published version of this document is located at the URL:

https://support.fortanix.com/hc/en-us/articles/360047562571-Using-Fortanix-Data-Security-Manager-for-NGINX-and-Apache-TLS-Keys

10.2 DOCUMENT UPDATES

This document will typically be updated on a periodic review and update cycle.

For any urgent document updates, please send an email to: support@fortanix.com