Apache HTTP Server: Integration Guide

THALES LUNA HSM AND DPOD LUNA CLOUD HSM

Document Information

Document Part Number	007-011228-001
Revision	L
Release Date	10 March 2023

Trademarks, Copyrights, and Third-Party Software

CONTENTS

Overview

This guide describes how to integrate Luna HSM or Luna Cloud HSM service with Apache HTTP Server to securely store SSL cryptographic keys. By integrating with Luna HSMs, Apache HTTP Server can utilize the Gem Engine with OpenSSL to access HSM resources. The advantages of generating cryptographic keys for Apache HTTP Server using Luna HSMs include:

- Secure generation, storage and protection of the cryptographic keys on FIPS 140-2 level 3 validated hardware.
- Full life cycle management of the keys.
- Significant performance improvements by off-loading cryptographic operations from application servers.
- HSM audit trail*
- Using cloud services with confidence.

* Luna Cloud HSM services do not have access to the secure audit trail.

Certified Platforms

This integration is certified on the following platforms:

Certified platforms on Luna HSM

Certified platforms on Luna Cloud HSM

Certified platforms on Luna HSM

HSM Type	Apache HTTP Server	Platforms
Luna HSM	2.4.x 2.2.x 2.0.x	Windows Server Red Hat Enterprise Linux

NOTE: This integration is tested with Luna HSMs in HA and FIPS Mode.

Luna HSM: Luna HSM appliances are purposefully designed to provide a balance of security, high performance, and usability that makes them an ideal choice for enterprise, financial, and government organizations. Luna HSMs secure the cryptographic keys physically and logically and accelerate cryptographic processing. Luna HSM on premise offerings include the Luna Network HSM, Luna PCIe HSM, and Luna USB HSMs. Luna HSMs are also available for access as an offering from cloud service providers such as IBM cloud HSM and AWS cloud HSM classic.

Certified platforms on Luna Cloud HSM

HSM Type	OpenSSL Toolkit	Platforms
Luna Cloud HSM	2.4.x	Windows Server Red Hat Enterprise Linux

Luna Cloud HSM: Luna Cloud HSM provides on-demand HSM and Key Management services through a simple graphical user interface. With Luna Cloud HSM, security is simple, cost effective and easy to manage because there is no hardware to buy, deploy and maintain. As an Application Owner, you click and deploy services, generate usage reports and maintain just the services you need.

Prerequisites

Before you proceed with the integration, complete the following tasks:

Configure Luna HSM

Configure Luna Cloud HSM Service

Downloading GemEngine Toolkit

Configure Luna HSM

If you are using Luna HSM:

1. Verify that the HSM is set up, initialized, provisioned, and ready for deployment. Refer to the Luna HSM documentation for more information.
2. Create a partition on the HSM that will be later used by Apache HTTP Server.
3. If you are using a Luna Network HSM, register a client for the system and assign the client to the partition to create an NTLS connection. Initialize the Crypto Officer and Crypto User roles for the registered partition.
4. Ensure that each partition is successfully registered and configured. The command to see the registered partitions is:

```
# <LunaClient Installation Directory>/bin/lunacm
```

```
lunacm (64-bit) v10.5.0-470. Copyright (c) 2021 SafeNet. All rights reserved.
Available HSMs:


Slot Id ->                   1
Label ->                     apache1
Serial Number ->             1312109862208
Model ->                     LunaSA 7.7.1
Firmware Version ->          7.7.1
Bootloader Version ->        1.1.2
Configuration ->             Luna User Partition With SO (PW) Key Export With
Cloning Mode
Slot Description ->          Net Token Slot
FM HW Status ->              Non-FM
```

```
Slot Id ->                2
Label ->                  apache2
Serial Number ->          1280780175894
Model ->                  LunaSA 7.7.1
Firmware Version ->       7.7.2
Bootloader Version ->     1.1.2
Configuration ->          Luna User Partition With SO (PW) Key Export With
Cloning Mode
Slot Description ->       Net Token Slot
FM HW Status ->           Non-FM


Slot Id ->                9
HSM Label ->              HA
HSM Serial Number ->      11312109862208
HSM Model ->              LunaVirtual
HSM Firmware Version -> 7.7.1
HSM Configuration ->      Luna Virtual HSM (PW) Key Export With Cloning Mode
HSM Status ->             N/A - HA Group
```

5. For PED-authenticated HSM, enable partition policies 22 and 23 to allow activation and auto-activation.

> **NOTE:** Refer to Luna HSM documentation for detailed steps about creating NTLS connection, initializing the partitions, and assigning various user roles.

Set up Luna HSM High-Availability

Refer to the Luna HSM documentation for HA steps and details regarding configuring and setting up two or more HSM boxes on host systems. You must enable the HAOnly setting in HA for failover to work so that if the primary goes down due to any reason all calls automatically route to the secondary until the primary recovers and starts up.

Setting Luna Network HSM Configuration

When Luna Client is installed, a configuration file is loaded at the following location:

/etc/Chrystoki.conf

This file is automatically configured and does not require any changes to communicate with the HSM. However for Luna Client 6.x onwards, this configuration needs to be edited for slot id because the default slot id is 0, but LunaCA3 engine is configured to use slot id as 1. To set the slot id to 1, you need to make the following changes in the configuration file:

```
Presentation = {
   OneBaseSlotId = 1;
}
```

Using Luna HSM in FIPS mode

Under FIPS 186-3/4, the RSA methods permitted for generating keys are 186-3 with primes and 186-3 with aux primes. This means that RSA PKCS and X9.31 key generation is no longer approved for operation in a FIPS-compliant HSM. If you are using the Luna HSM in FIPS mode, you have to make the following change in configuration file:

For Linux

```
Misc = {
RSAKeyGenMechRemap = 1;
}
```

For Windows

```
[Misc]
RSAKeyGenMechRemap=1
```

The above setting redirects the older calling mechanism to a new approved mechanism when Luna HSM is in FIPS mode.

> **NOTE:** The above setting is not required for Universal Client. This setting is applicable only for Luna Client 7.x.

Configure Luna Cloud HSM service

You can configure Luna Cloud HSM Service in the following ways:

> Standalone Cloud HSM service using minimum client package

> Standalone Cloud HSM service using full Luna client package

> Luna HSM and Luna Cloud HSM service in hybrid mode

> **NOTE:** Luna Client v10.x or higher is required for configuring Luna HSM device and Luna Cloud HSM service in hybrid mode.

Standalone Cloud HSM service using minimum client package

To configure Luna Cloud HSM service using minimum client package:

1. Transfer the downloaded .zip file to your Client workstation using pscp, scp, or other secure means.
2. Extract the .zip file into a directory on your client workstation.
3. Extract or untar the appropriate client package for your operating system. Do not extract to a new subdirectory; place the files in the client install directory.

   ```
   [Windows]

   cvclient-min.zip

   [Linux]

   cvclient-min.tar

   # tar -xvf cvclient-min.tar
   ```

4. Run the `setenv` script to create a new configuration file having information required by the Luna Cloud HSM service.

```
[Windows]
Right-click setenv.cmd and select Run as Administrator.
[Linux]
Source the setenv script.
# source ./setenv
```

5. Run the `LunaCM` utility and verify the Cloud HSM service is listed.

Standalone Cloud HSM service using full Luna client package

To configure Luna Cloud HSM service using full Luna client package:

1. Transfer the downloaded .zip file to your Client workstation using pscp, scp, or other secure means.
2. Extract the .zip file into a directory on your client workstation.
3. Extract or untar the appropriate client package for your operating system. Do not extract to a new subdirectory; place the files in the client install directory.

   ```
   [Windows]
   cvclient-min.zip
   [Linux]
   cvclient-min.tar
   # tar -xvf cvclient-min.tar
   ```

4. Run the `setenv` script to create a new configuration file containing information required by the Luna Cloud HSM service.

   ```
   [Windows]
   ```

   Right-click setenv.cmd and select Run as Administrator.

   ```
   [Linux]
   Source the setenv script.
   # source ./setenv
   ```

5. Copy the server and partition certificates from the Cloud HSM service client directory to Luna client certificates directory:

   > **NOTE:** Skip this step for Luna Client v10.2 or higher.

   Cloud HSM Certificates:

   ```
   server-certificate.pem
   partition-ca-certificate.pem
   partition-certificate.pem
   ```

   LunaClient Certificate Directory:

   ```
   [Windows default location for Luna Client]
   C:\Program Files\Safenet\Lunaclient\cert\
   [Linux default location for Luna Client]
   /usr/safenet/lunaclient/cert/
   ```

6. Open the configuration file from the Cloud HSM service client directory and copy the `XTC` and `REST` section.

   ```
   [Windows]
   ```

   ```
   crystoki.ini
   ```

   ```
   [Linux]
   ```

   ```
   Chrystoki.conf
   ```

7. Edit the Luna Client configuration file and add the `XTC` and `REST` sections copied from Cloud HSM service client configuration file.

8. Change server and partition certificates path from step 5 in `XTC` and `REST` sections. Do not change any other entries provided in these sections.

   > **NOTE:** Skip this step for Luna Client v10.2 or higher.

   [XTC]

   ```
   . . .
   PartitionCAPath=<LunaClient_cert_directory>\partition-ca-certificate.pem
   PartitionCertPath00=<LunaClient_cert_directory>\partition-certificate.pem
   . . .

   [REST]
   . . .
   SSLClientSideVerifyFile=<LunaClient_cert_directory>\server-certificate.pem
   . . .
   ```

9. Edit the following entry from the `Misc` section and update the correct path for the plugins directory:

   ```
   Misc]
   PluginModuleDir=<LunaClient_plugins_directory>
   ```

   ```
   [Windows Default]
   ```

   ```
   C:\Program Files\Safenet\Lunaclient\plugins\
   ```

   ```
   [Linux Default]
   ```

   ```
   /usr/safenet/lunaclient/plugins/
   ```

   Save the configuration file. If you wish, you can now safely delete the extracted Cloud HSM service client directory.

10. Reset the `ChrystokiConfigurationPath` environment variable and point back to the location of the Luna Client configuration file.

    ```
    [Windows]
    ```

    Click Environment Variables. In both list boxes for the current user and system variables, edit ChrystokiConfigurationPath and point to the crystoki.ini file in the Luna client install directory.In the Control Panel, search for environment and select Edit the system environment variables.

```
[Linux]
```

Either open a new shell session, or export the environment variable for the current session pointing to the location of the Chrystoki.conf file:

```
# export ChrystokiConfigurationPath=/etc/
```

11. Run the `LunaCM` utility and verify that the Cloud HSM service is listed. In hybrid mode, both Luna and Cloud HSM service will be listed.

> **NOTE:** Follow the Luna Cloud HSM documentation for detailed steps for creating service, client, and initializing various user roles.

Luna HSM and Luna Cloud HSM service in hybrid mode

To configure Luna HSM and Luna Cloud HSM service in hybrid mode, follow the steps mentioned under the Standalone Cloud HSM service using full Luna client package section above.

> **NOTE:** Luna Client v10.x or higher is required for configuring Luna HSM device and Luna Cloud HSM service in hybrid mode.

Use Luna Cloud HSM Service in FIPS mode

Cloud HSM service operates in both FIPS and non-FIPS mode. If your organization requires non-FIPS algorithms for your operations, enable the Allow non-FIPS approved algorithms check box when configuring your Cloud HSM service. The FIPS mode is enabled by default. Refer to the Mechanism List in the SDK Reference Guide for more information about available FIPS and non-FIPS algorithms.

Download GemEngine toolkit

Download the GemEngine toolkit with Gem Engine support from Thales Customer Support portal.

> **NOTE:**
>
> The Doc ID for downloading the GemEngine v1.6 from support portal is KB0026742.
>
> The Doc ID for downloading the GemEngine v1.5 from support portal is KB0024584.
>
> The Doc ID for downloading the GemEngine v1.3 from support portal is KB0017806.
>
> The Doc ID for downloading the GemEngine v1.2 from support portal is KB0016309.

It is recommended that you should familiarize yourself with Apache HTTP Server. Refer to Apache HTTP Server Documentation for more information about Apache HTTP Server.

Integrating Luna HSM with Apache HTTP Server using Gem Engine

To integrate Luna HSM with Apache HTTP Server using Gem Engine, follow the steps mentioned below in accordance with your environment configuration:

> Integrate Apache HTTP Server with Luna HSM on UNIX

> Integrate Apache HTTP Server with Luna HSM on Windows

Integrate Apache HTTP Server with Luna HSM on UNIX

Integration of Apache HTTP Server with Luna HSM using GemEngine on UNIX involves the following use cases:

> Integrating Luna HSM with Apache HTTP Server by generating new SSL keys

> Integrating Luna HSM with Apache HTTP Server by migrating existing SSL keys

Integrating Luna HSM with Apache HTTP Server by generating new SSL keys

The integration involves the following steps:

> Downloading and extracting the required software packages

> Compiling and Installing the Gem Engine and OpenSSL using the GemEngine toolkit and Apache

> Configuring Gem Engine for OpenSSL on UNIX

> Configuring Apache HTTP Server for SSL

Downloading and extracting the required software packages

1. Download and extract the OpenSSL source tarball.

   Example:

   Download openssl-3.0.7.tar.gz from https://www.openssl.org/source/

   ```
   # tar xvfz openssl-3.0.7.tar.gz
   ```

2. Download and extract an Apache (httpd) source tarball from https://httpd.apache.org/download.cgi and place the .tar.gz file inside the gemengine directory.

   Example:

   Download and extract the httpd-2.4.55.tar.gz file:

   ```
   # tar xzvf httpd-2.4.55.tar.gz
   ```

3. Download and extract an apr source tarball from https://apr.apache.org/download.cgi and place the .tar.gz file inside the gemengine directory.

   Example:

   Download apr-1.7.0.tar.gz and extract:

   ```
   # tar xzvf apr-1.7.0.tar.gz
   ```

4. Download and extract an apr util source tarball from https://apr.apache.org/download.cgi and place the .tar.gz file inside the gemengine directory.

   Example:

   Download apr-util-1.6.1.tar.gz and extract:

   ```
   # tar xzvf apr-util-1.6.1.tar.gz
   ```

5. Download and extract an apr iconv source tarball from https://apr.apache.org/download.cgi and place the .tar.gz file inside the gemengine directory.

   Example:

   Download apr-iconv-1.2.2.tar.gz and extract:

   ```
   # tar xzvf apr-iconv-1.2.2.tar.gz
   ```

Compiling and installing the Gem Engine and OpenSSL using GemEngine toolkit and Apache

1. From gemengine directory, run the gembuild config command using the -prefix option.

   ```
   # ./gembuild config --openssl-source=<openssl-source path> --apache-
   source=<httpd-src path> --apr-source=<apr-src path> --apr-iconv-
   source=<iconvsrc path> --apr-util-source=<utilsrc path>  --prefix=/usr/local --
   config-bits=64 --fips-module=no
   ```

2. Compile and install OpenSSL.

   ```
   # ./gembuild openssl-build
   
   # ./gembuild openssl-install
   ```

3. Compile and install gem dynamic engine and verify engine.

   ```
   # ./gembuild engine-build
   
   # ./gembuild engine-install
   
   # /usr/local/ssl/bin/openssl engine gem -v
   
   (gem) Gem engine support
   
   enginearg, openSession, closeSession, login, logout, engineinit,
   
   CONF_PATH, ENGINE_INIT, ENGINE2_INIT,engine2init,DisableCheckFinalize,
   
   SO_PATH, GET_HA_STATE, SET_FINALIZE_PENDING, SKIP_C_INITIALIZE,
   
   IntermediateProcesses
   ```

4. Compile and install the sautil command.

   ```
   # ./gembuild sautil-build
   
   # ./gembuild sautil-install
   ```

   By default, this installs the sautil command to <prefix>/sautil/bin/sautil where <prefix> is the directory specified with --prefix option in the step 7.

   If a different location is desired, use the --sautil-prefix option to specify the desired directory either by redoing the step 4 with the option or by specifying the option as part of the ./gembuild sautil-install command.

5. Add openssl and sautil to PATH. Example:

   ```
   # export PATH=/usr/local/ssl/bin:/usr/local/sautil/bin:$PATH
   ```

6. Compile and install Apache:

   ```
   # ./gembuild apache-build
   ```

Configuring Gem Engine for OpenSSL on UNIX

> Configure Gem Engine for Luna HSM

> Configure Gem Engine for Luna Cloud HSM service

Configure Gem Engine for Luna HSM

To configure Gem Engine for Luna HSM:

1. Run the Optimize.sh command in the gemengine directory to configure the Luna Network HSM/Luna PCI-E HSM configuration file (/etc/Chrystoki.conf) for Apache:

   ```
   # ./Optimize.sh fork
   ```

   The Luna Network HSM/Luna PCI-E HSM configuration file (/etc/Chrystoki.conf) is now configured for Apache HTTP Server.

   Luna Network HSM

   ```
   Misc = {
      PE1746Enabled = 0;
      Apache = 0;
   }
   GemEngine = {
   LibPath = /usr/safenet/lunaclient/lib/libCryptoki2.so;
   LibPath64 = /usr/safenet/lunaclient/lib/libCryptoki2_64.so;
   EnableDsaGenKeyPair = 1;
   EnableRsaGenKeyPair = 1;
   DisablePublicCrypto = 1;
   EnableRsaSignVerify = 1;
   EnableLoadPubKey = 1;
   EnableLoadPrivKey = 1;
   DisableCheckFinalize = 0;
   DisableEcdsa = 1;
   DisableDsa = 0;
   DisableRand = 0;
   EngineInit = 1:10:11;
   }
   ```

2. Run the sautil utility to open the session on the Luna HSM slot:

   ```
   # /usr/local/sautil/bin/sautil -v -s 1 -i 10:11 -o -q
   ```

Configure Gem Engine for Luna Cloud HSM service

To configure Gem Engine for Luna Cloud HSM:

1. Create a text file to store the partition crypto officer password in that file.

   ```
   # echo <partition_password> > <path_to_my_passfile>/passfile
   ```

2. Open the /<ChrystokiConfigurationFile_Directory>/Chrystoki.conf file and add the following text to the GemEngine section:

   ```
   GemEngine = {
   LibPath = <path to LibCryptoki2.so>;
   LibPath64 = <path to LibCryptoki2_64.so>;
   EnableDsaGenKeyPair = 1;
   EnableRsaGenKeyPair = 1;
   DisablePublicCrypto = 1;
   EnableRsaSignVerify = 1;
   EnableLoadPubKey = 1;
   EnableLoadPrivKey = 1;
   DisableCheckFinalize = 1;
   DisableEcdsa = 1;
   DisableDsa = 0;
   DisableRand = 0;
   EngineInit = <Partition Label>:0:0:passfile=<path_to_my_passfile>/passfile;
   EnableLoginInit = 1;
   ```

   > **NOTE:** `<Partition Label>` must be replaced with the actual label of your partition and `passfile` must point to the actual path of the file containing CO password.

3. In the Misc section of Chrystoki.conf file, add the following flag and save the changes. Do not delete the default values already present in the Misc section.

   ```
   Misc = {
   FinalizeOnClose = 1;
   }
   ```

Configuring Apache HTTP server for SSL

> [Generate certificates](#)

> [Update Apache HTTP Server to start SSL](#)

Generate certificates

Depending on the HSM you are using, generate a certificate:

> **NOTE:** It is recommended to use the CA signed certificate in the production environment.

> [Generate certificates for Luna HSM](#)

> [Generate certificates for Luna Cloud HSM Service](#)

Generate certificates for Luna HSM

To generate certificates for Luna HSM:

1. Go to your gemengine directory.

2. Run gembuild to generate RSA keys for apache:

   ```
   # ./gembuild apache-genrsa
   ```

   > **NOTE:** The above command will generate a self-signed certificate. It is recommended to use the CA signed certificate in production environment. Self-Signed certificate is suitable for test environment only.

Generate certificates for Luna Cloud HSM Service

[Generate CA-signed certificates](#)

[Generate self-signed certificates](#)

Generate CA-signed Certificates

The steps to generate CA signed certificate are as follows:

1. Execute the command below to generate the keys on Luna HSM and save the certificate request and key reference:

   ```
   # openssl req -engine gem -new -newkey rsa:2048 -nodes -sha256 -keyout
   server.key -out server.csr –keyform engine
   ```

   The public key and private key will be generated on the HSM and the private key reference generated on the HSM will be saved in the server.key file. You'll be requiring this later. The Certificate Signing Request (CSR) will be saved in the server.csr file that needs to be submitted to the CA for obtaining a CA-signed certificate.

2. Run the cmu list to verify the generated key pair on Luna HSM.

   ```
   # <DPoD Client Directory>/bin/64/cmu list
   ```

   Provide partition password when prompted.

3. Submit the CSR file (server.csr) to a CA, such as VeriSign or Entrust. The CA authenticates the request and returns a signed certificate or a certificate chain. Save the CA-signed certificate in the system directory and provide your key reference (server.key) and CA-signed certificate (server.pem) in the Apache HTTP Server configuration.

Generate self-signed certificates

The steps to generate self-signed certificates are as follows:

1. Execute the command below to generate the keys on Luna HSM and save the key reference.

   ```
   # openssl genrsa -engine gem -out server.key 2048
   ```

   The server.key is the key reference to Private Key Generated on HSM. You will require it later.

2. Generate a self-signed certificate that can be used for test purpose by executing the following command:

   ```
   # openssl req -new -engine gem -x509 -key server.key -sha256 -out server.pem
   ```

   Here, server.pem is the self-signed certificate in PEM format.

Update Apache HTTP Server to start SSL

1. Go to Apache installation directory, update Apache configuration file (httpd.conf) and edit the ServerName field with the hostname or IP address of the server with the value specified for the CN in the certificate created above.

2. Go to Apache installation directory for conf/extra for SSL configuration (such as: /usr/local/apache2/conf/extra), update httpd-ssl.conf and edit the Virtual Host section as below:

   ```
   <VirtualHost Hostname or IP Address: 443>
   ```

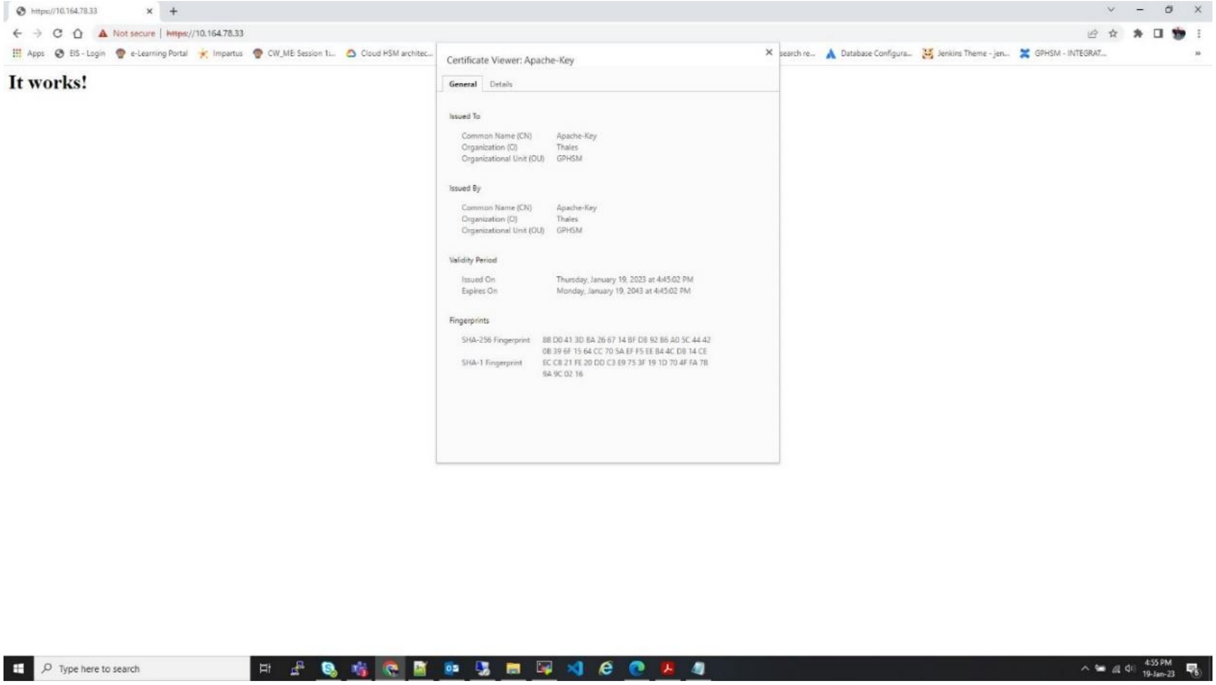
3. Start the Apache server:

   ```
   # /usr/local/apache2/bin/apachectl -k start
   ```

4. Open IE or Firefox browser and access the following HTTP server:

   https://<HostName or IP Address>:443

5. Accept the certificate.



It works!

Certificate Viewer: Apache-Key

General | Details

Issued To

Common Name (CN) Apache-Key
Organization (O) Thales
Organizational Unit (OU) GPHSM

Issued By

Common Name (CN) Apache-Key
Organization (O) Thales
Organizational Unit (OU) GPHSM

Validity Period

Issued On Thursday, January 19, 2023 at 4:45:02 PM
Expires On Monday, January 19, 2043 at 4:45:02 PM

Fingerprints

SHA-256 Fingerprint 88 D0 41 3D BA 26 67 14 BF D8 92 B6 A0 5C 44 42 0B 39 6F 15 64 CC 70 5A EF F5 EE B4 4C D8 14 CE
SHA-1 Fingerprint EC C8 21 FE 20 DD C3 E9 75 3F 19 1D 70 4F FA 7B 9A 9C 02 16

Integrating Luna HSM with Apache HTTP Server by migrating existing SSL keys

It is assumed that Apache HTTP server is already configured and running on SSL where SSL certificate and keys are generated by OpenSSL and saved somewhere in the directory. Before proceeding, ensure that you have completed the Prerequisites.

To migrate the SSL keys:

1. Configure OpenSSL to use GemEngine by executing the steps mentioned in the Configuring GemEngine for OpenSSL on UNIX section.
2. Locate the directory where the SSL private key and certificate are stored.
3. Extract the certificate public key using the command below:

```
# openssl rsa -in /usr/local/apache2/conf/ssl.key/server.key -pubout -out
/usr/local/apache2/conf/ssl.crt/pubkey.pem
```

4. Extract the private key in PKCS#8 format using the below command:

```
# openssl pkcs8 -in /usr/local/apache2/conf/ssl.key/server.key -topk8 -nocrypt
-out /usr/local/apache2/conf/ssl.key/privatekey.pem
```

5. Using the CMU utility provided with Luna Client, import the public key and private key to the HSM.

   For Public Key:

```
# <LunaClient Installation Directory>/bin/cmu import -inputFile
/usr/local/apache2/conf/ ssl.crt/pubkey.pem -label apache2_public_key -
pubkey=rsa
```

   For Private Key:

```
# <LunaClient Installation Directory>/bin/cmu importkey -PKCS8 -in
/usr/local/apache2/conf/ ssl.key/privatekey.pem -keyalg RSA
```

6. Verify that the keys are generated on Luna HSM partition and note the private key handle that will be used later.

```
# <LunaClient Installation Directory>/bin/cmu list

Certificate Management Utility (64-bit) v10.5.0-470. Copyright (c) 2022
SafeNet. All rights reserved.


handle=2000001  label= CMU Unwrapped RSA Private Key

handle=2000002  label= apache2_public_key
```

7. Optionally, in case you have multiple keys, you can recognize those by providing them labels. To do so, execute the following command:

```
# <LunaClient Installation Directory>/bin/cmu setattribute -handle=2000001 -
label=apache2_priv_key
```

8. Verify that the private key label corresponds to the label of public key.

```
# <LunaClient Installation Directory>/bin/cmu list -password userpin1

Certificate Management Utility (64-bit) v10.5.0-470. Copyright (c) 2021
SafeNet. All rights reserved.


handle=2000001  label= apache2_priv_key

handle=2000002  label= apache2_public_key
```

9. Copy the sautil utility to /usr/local/bin according to your OpenSSL version.

   Example:

```
# cp <gem-engine directory>/builds/linux/rhel/64/3.0/sautil /usr/local/bin
```

10. Run the sautil utility to create Private Key Reference to actual private key imported in Luna HSM.

```
sautil -v -s 1 -i 10:11 -a 0:RSA -f /usr/local/apache2/conf/
ssl.key/HSMKey_ref_new.pem -o -p userpin1 -c
```

After successful completion, HSMKey_ref.pem will be generated. You need to specify HSMKey_ref.pem in SSL setting in extras/httpd-ssl.conf file.

11. Remove the Private Key generated by OpenSSL that you were using before importing the key in to HSM along with the PKCS#8 format key generated in step 4.

12. Add/modify the following lines in /usr/local/apache2/conf/extras/httpd-ssl.conf file:

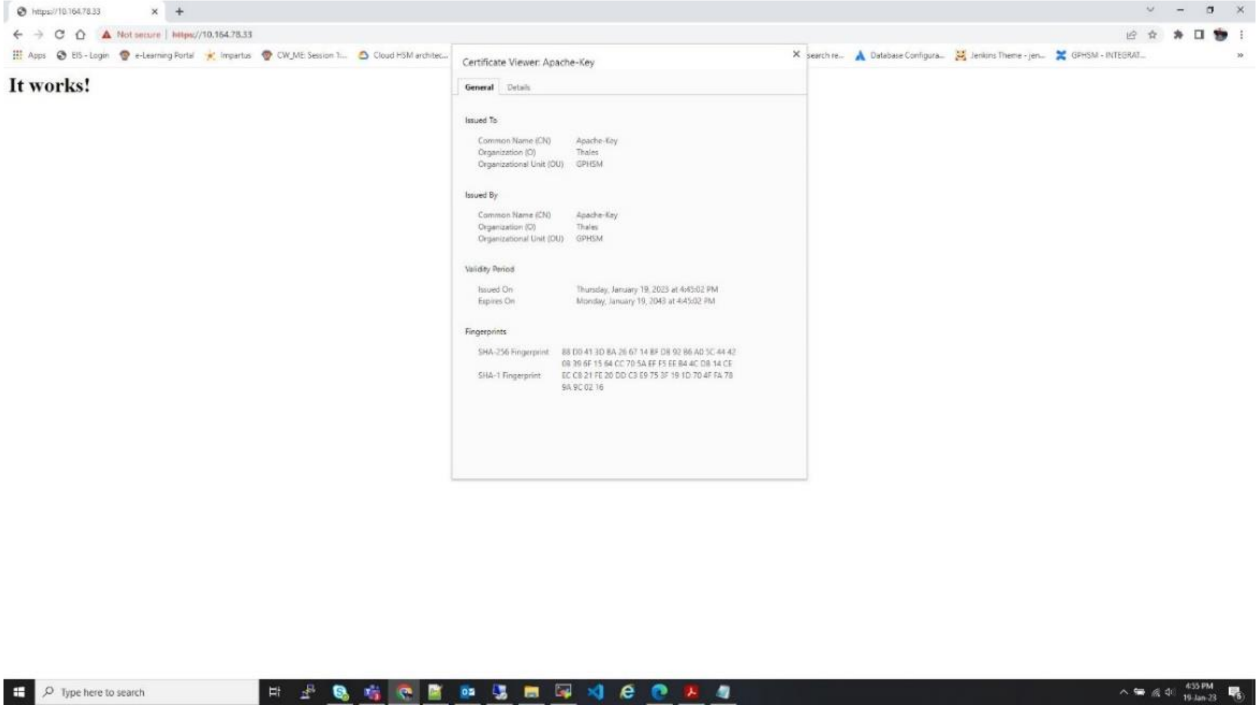
```
SSLCryptoDevice gem

<VirtualHost _default_:443>

SSLCertificateFile /usr/local/apache2/conf/ssl.crt/server.pem

SSLCertificateKeyFile /usr/local/apache2/conf/ssl.key/HSMKey_ref.pem

</VirtualHost>
```

13. Restart the Apache HTTP Server:

```
# /usr/local/apache2/bin/apachectl -k restart
```

14. Access the Apache HTTP Server over port 443 in web browser and accept the certificate:

```
https://<HostName or IP Address>:443
```



It works!

Certificate Viewer: Apache-Key

General | Details

Issued To

Common Name (CN) Apache-Key
Organization (O) Thales
Organizational Unit (OU) GPHSM

Issued By

Common Name (CN) Apache-Key
Organization (O) Thales
Organizational Unit (OU) GPHSM

Validity Period

Issued On Thursday, January 19, 2023 at 4:45:02 PM
Expires On Monday, January 19, 2043 at 4:45:02 PM

Fingerprints

SHA-256 Fingerprint 88 D0 41 3D 8A 26 67 14 BF D8 92 B6 A0 5C 44 42 08 39 6F 15 64 CC 70 5A EF F5 EE B4 4C D8 14 CE
SHA-1 Fingerprint EC C8 21 FE 20 DD C3 E9 75 3F 19 1D 70 4F FA 78 9A 9C 02 16

Integrate Apache HTTP Server with Luna HSM on Windows

Integration of Apache HTTP Server with Luna HSM using GemEngine on Windows involves the following use cases:

> Integrating Luna HSM with Apache HTTP Server by generating new SSL keys

> Integrating Luna HSM with Apache HTTP Server by migrating existing SSL keys

Integrating Luna HSM with Apache HTTP Server by generating new SSL keys

The integration involves the following steps:

> Install and configure GemEngine toolkit

> Configuring GemEngine for OpenSSL on Windows

> Generating keys and certificates

> Configuring Apache HTTP Server for SSL

Install and configure GemEngine toolkit

1. Initially, install and configure the GemEngine toolkit on Windows, Navigate to the GemEngine toolkit <engine-directory>\builds\win and extract the file sautil-win64-openssl-x.x.xx.tar.gz and ssl-win64-openssl-x.x.xx.tar.gz in C:\ directory.
2. Add C:\cygwin\usr\local\sautil\bin and C:\cygwin\usr\local\ssl\bin to your system path (Control Panel -> System -> Change Settings -> Advanced -> Environment Variables -> System Variables).

   NOTE: We recommend adding these files to the Path. This step is not mandatory.

3. Rename the openssl executable file in C:\Apache24\bin to openssl.exe.old, so that new openssl executable which comes with the GemEngine toolkit will be used.
4. Check the OpenSSL and engine directory of the OpenSSL bundled with GemEngine:

   ```
   # openssl version -d
   ```

   The output will appear like this:

   OPENSSLDIR: C:\cygwin\usr\local\ssl

5. Copy C:\Apache24\conf\openssl.cnf file to the folder specified above in OPENSSLDIR.
6. Copy engines directory from C:\cygwin\usr\local\ssl\lib directory to C:\Apache24\lib location.

Configuring GemEngine for OpenSSL on Windows

> Configure Gem Engine for Luna HSM

> Configure Gem Engine for Luna Cloud HSM service

Configure Gem Engine for Luna HSM

To configure Gem Engine for Luna HSM:

1. Add the following text to the C:\Program Files\SafeNet\LunaClient\crystoki.ini file:

   ```
   [GemEngine]
   LibPath = <LunaClient Installation Directory>\win32\cryptoki.dll
   LibPath64 = <LunaClient Installation Directory>\cryptoki.dll
   EnableDsaGenKeyPair = 1
   EnableRsaGenKeyPair = 1
   DisablePublicCrypto = 1
   EnableRsaSignVerify = 1
   EnableLoadPubKey = 1
   EnableLoadPrivKey = 1
   DisableCheckFinalize = 1
   DisableEcdsa = 1
   DisableDsa = 0
   DisableRand = 0
   EngineInit = 0:10:11
   ```

   where 0 is slot id and 10:11 is the application ID in EngineInit.

2. Open the persistent session with the HSM:

   ```
   # sautil.exe -v -s 0 -i 10:11 -o -q
   ```

   where 0 is slot id and 10:11 is the application ID.

3. Provide the partition password when prompted.

> **NOTE:** For more login methods for session, refer to the *README-GEM-CONFIG* file present in the `<GemEngine_Directory>\docs` folder.

Configure Gem Engine for Luna Cloud HSM service

1. Create a text file `passfile` in `<path_to_my_passfile>` and store the partition password in it.

2. Open the `crystoki.ini` file and add the following text to the `GemEngine` section:

   ```
   [GemEngine]
   LibPath = <path to LibCryptoki2.so>
   LibPath64 = <path to LibCryptoki2_64.so>
   EnableDsaGenKeyPair = 1
   EnableRsaGenKeyPair = 1
   DisablePublicCrypto = 1
   ```

```
EnableRsaSignVerify = 1
EnableLoadPubKey = 1
EnableLoadPrivKey = 1
DisableCheckFinalize = 1
DisableEcdsa = 1
DisableDsa = 0
DisableRand = 0
EngineInit = <Partition Label>:0:0:passfile=<path_to_my_passfile>/passfile;
EnableLoginInit = 1
```

> **NOTE:** `<Partition Label>` must be replaced with the actual label of your partition and `passfile` must point to the actual path of the file containing CO password.

Generating Keys and Certificates

1. Use the OpenSSL command to generate keys and certificates on the HSM:

   ```
   # cd C:\Apache24\conf
   # openssl.exe genrsa -engine gem 2048
   ```

   This will output the private key reference. Save the private key reference in a file called C:\Apache24\conf\server.key.

2. Create a self-signed certificate using the above private key:

   ```
   # openssl.exe req -engine gem -new -x509 -days 365 -key server.key -out
   server.crt –keyform engine
   ```

Configuring Apache HTTP Server for SSL

To configure Apache HTTP server for SSL:

1. Uncomment the following lines in the C:\Apache24\conf\httpd.conf file:

   ```
   #LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
   #LoadModule ssl_module modules/mod_ssl.so
   #Include conf/extra/httpd-ssl.conf
   ```

2. Add/modify the following lines in C:\Apache24\conf\extras\httpd-ssl.conf file:

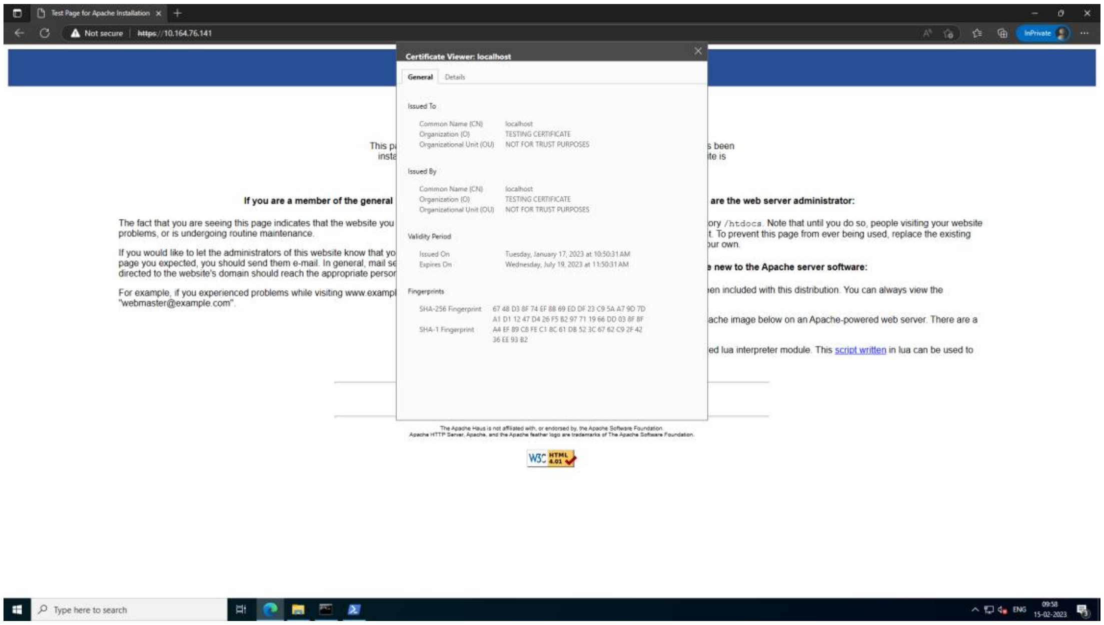
   ```
   SSLCryptoDevice gem
   <VirtualHost _default_:443>
   SSLCertificateKeyFile ${SRVROOT}/conf/server.key
   SSLCertificateFile ${SRVROOT}/conf/server.crt
   </VirtualHost>
   ```

3. Start the Apache HTTP Server:

   ```
   # C:\Apache24\bin\httpd.exe -k start
   ```

4. Access the HTTP Server over port 443 in web browser and accept the certificate:

```
https://<HostName or IP Address>:443
```



Certificate Viewer: localhost

General | Details

Issued To

Common Name (CN)	localhost
Organization (O)	TESTING CERTIFICATE
Organizational Unit (OU)	NOT FOR TRUST PURPOSES

Issued By

Common Name (CN)	localhost
Organization (O)	TESTING CERTIFICATE
Organizational Unit (OU)	NOT FOR TRUST PURPOSES

Validity Period

Issued On	Tuesday, January 17, 2023 at 10:50:31 AM
Expires On	Wednesday, July 19, 2023 at 11:50:31 AM

Fingerprints

SHA-256 Fingerprint	67 48 D3 8F 74 EF 88 69 ED DF 23 C9 5A A7 9D 7D A1 D1 12 47 D4 26 F5 B2 97 71 19 66 DD 03 8F 8F
SHA-1 Fingerprint	A4 EF 89 CB FE C1 8C 61 DB 52 3C 67 62 C9 2F 42 36 EE 93 B2

The Apache Haus is not affiliated with, or endorsed by, the Apache Software Foundation.
Apache HTTP Server, Apache, and the Apache feather logo are trademarks of The Apache Software Foundation.

Integrating Luna HSM with Apache HTTP Server by migrating existing SSL keys

It is assumed that Apache HTTP server is already configured and running on SSL where SSL certificate and keys are generated by OpenSSL and saved somewhere in directory. Before proceeding ensure that you have completed the Prerequisites.

1. Configure OpenSSL to use GemEngine by executing the steps mentioned in the Configuring GemEngine for OpenSSL on Windows section.
2. Locate the directory where the SSL private key and certificate are stored.
3. Extract the certificate public key by using the command below:

   ```
   # openssl rsa -in server.key -pubout -out pubkey.pem
   ```

4. Extract the private key in PKCS#8 format by using the below command:

   ```
   # openssl pkcs8 -in server.key -topk8 -nocrypt -out privatekey.pem
   ```

5. Using the CMU utility provided with the Luna Client, import the public key and private key to the HSM:

   For Public Key:

   ```
   # <LunaClient Installation Directory>\Cmu.exe import -inputFile pubkey.pem -label apache_pub_key -pubkey=rsa
   ```

   For Private Key:

   ```
   # <LunaClient Installation Directory>\Cmu.exe importkey -PKCS8 -in privatekey.pem -keyalg RSA
   ```

6. Verify that the keys are generated on Luna HSM partition and note the private key handle which will be used later:

```
# <LunaClient Installation Directory>\Cmu.exe list -password userpin1

Certificate Management Utility (64-bit) v10.5.0-470. Copyright (c) 2022
SafeNet. All rights reserved.

handle=2000001  label=apache_pub_key

handle=2000002  label=CMU Unwrapped RSA Private Key
```

7. Run the sautil utility to create Private Key Reference to actual private key imported in Luna HSM:

```
# sautil.exe -v -s 0 -i 0:0 -a 0:RSA -f HSMKey_ref.pem -o -q –c
```

8. Provide the HSM partition CO password and key handle when prompted. After successful completion, HSMKey_ref.pem will be generated that you need to specify in SSL setting in extras/httpd-ssl.conf file.

9. Remove the private key generated by OpenSSL that you were using before importing the key in to Luna HSM along with the PKCS#8 format key generated in step 4.

10. Add/modify the following lines in the C:\Apache24\conf\extras\httpd-ssl.conf file:

```
SSLCryptoDevice gem

<VirtualHost _default_:443>

SSLCertificateKeyFile ${SRVROOT}/conf/HSMKey_ref.pem

SSLCertificateFile ${SRVROOT}/conf/server.crt

</VirtualHost>
```

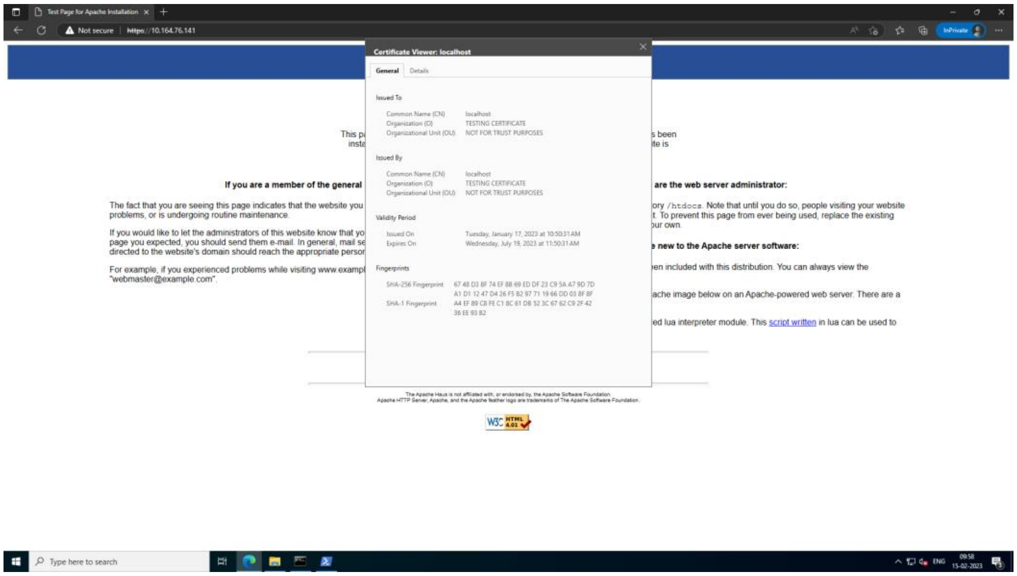
> **NOTE:** If the certificate is signed by root CA, then add SSLCertificateChainFile ${SRVROOT}/conf/server-ca.crt, where server-ca.cert is signing/root CA certificate inside VirtualHost section.

11. Restart the Apache HTTP Server:

```
# C:\Apache24\bin\httpd.exe -k restart
```

12. Access the Apache HTTP Server over port 443 in web browser and accept the certificate:

```
https://<HostName or IP Address>:443
```



Certificate Viewer: localhost

General | Details

Issued To

Common Name (CN) localhost
Organization (O) TESTING CERTIFICATE
Organizational Unit (OU) NOT FOR TRUST PURPOSES

Issued By

Common Name (CN) localhost
Organization (O) TESTING CERTIFICATE
Organizational Unit (OU) NOT FOR TRUST PURPOSES

Validity Period

Issued On Tuesday, January 17, 2023 at 10:50:31 AM
Expires On Wednesday, July 19, 2023 at 11:50:31 AM

Fingerprints

SHA-256 Fingerprint 67 48 D3 8F 74 EF 88 69 ED DF 23 C9 5A A7 9D 7D A1 D1 12 47 D4 26 F5 B2 97 71 19 66 DD 03 8F 8F
SHA-1 Fingerprint A4 EF 89 CB FE C1 BC 61 DB 52 3C 67 62 C9 2F 42 36 EE 93 B2

The Apache Haus is not affiliated with, or endorsed by, the Apache Software Foundation.
Apache HTTP Server, Apache, and the Apache feather logo are trademarks of The Apache Software Foundation.

Contacting Customer Support

If you encounter a problem during this integration, contact your supplier or Thales Customer Support. Thales Customer Support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between Thales and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

Customer Support Portal

The Customer Support Portal, at https://supportportal.thalesgroup.com, is a database where you can find solutions for most common problems. The Customer Support Portal is a comprehensive, fully searchable repository of support resources, including software and firmware downloads, release notes listing known problems and workarounds, a knowledge base, FAQs, product documentation, technical notes, and more. You can also use the portal to create and manage support cases.

> **NOTE:** You require an account to access the Customer Support Portal. To create a new account, go to the portal and click on the **REGISTER** link.

Telephone Support

If you have an urgent problem, or cannot access the Customer Support Portal, you can contact Thales Customer Support by telephone at +1 410-931-7520. Additional local telephone support numbers are listed on the support portal.