THALES

JBoss Application Server: Integration Guide

THALES LUNA HSM AND LUNA CLOUD HSM

Document Information

Document Part Number	007-012123-001
Revision	P
Release Date	12 April 2023

Trademarks, Copyrights, and Third-Party Software

CONTENTS

Overview

The aim of this document is to assist security administrators in installing, configuring, and integrating Thales Luna Hardware Security Modules (HSMs) with JBoss Application Server. Transferring cryptographic operations from the JBoss server to the Luna HSMs can bring about significant performance improvements. Additionally, the high-value SSL private key and certificate of the server can be secured within a FIPS 140-2 certified hardware security module.

JBoss is a collection of offerings for enterprise customers seeking preconfigured profiles of JBoss Enterprise Middleware components that have been tested and certified to provide an integrated experience. It is the open source implementation of the Java EE suite of services. JBoss's user-friendly server architecture and high flexibility make it an ideal choice for both beginners and senior architects who require a customizable middleware platform. Since JBoss Application Server is Java-based, it is cross-platform, simple to install, and can be used on any operating system that supports Java. The readily available source code is a powerful tool for learning how to debug and understand the server. It also provides the flexibility to create custom versions for personal or business use.

The advantages of utilizing Thales Luna HSMs for SSL key generation in JBoss include:

- Ensuring secure key generation, storage, and protection through FIPS 140-2 level 3 validated hardware.
- Providing full life cycle management of the keys.
- Maintaining an audit trail through HSM.
- Achieving significant performance enhancements by offloading cryptographic operations from application servers.

> **Note:** The Luna Cloud HSM service does not have access to the secure audit trail.

Certified platforms

> Certified platforms for Luna HSM

> Certified platforms for Luna Cloud HSM

Certified platforms for Luna HSM

HSM Type	Platforms
Luna HSM	Red Hat Enterprise Linux

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 physically and logically secure cryptographic keys and accelerate cryptographic processing. The 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 for Luna Cloud HSM

HSM Type	Platforms
Luna Cloud HSM	Red Hat Enterprise Linux

Luna Cloud HSM: Luna Cloud HSM platform provides on-demand, cloud-based 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 beginning the integration, ensure that you have completed the following tasks:

> Configure Luna HSM

> Configure Luna Cloud HSM service

> Java Setup

> JBoss Setup

Configure Luna HSM

To configure Luna HSM:

1. Ensure the HSM is set up, initialized, provisioned, and ready for deployment.
2. Create a partition on the HSM for use by JBoss.
3. Establish an NTLS connection by creating a certificate and exchanging it between the Luna Network HSM and the client, followed by registering the client and assigning a partition to it, and then initializing the Crypto Officer and Crypto User roles for the registered partition.
4. Verify that the partition is successfully registered and configured by running the following command:

```
# /usr/safenet/lunaclient/bin/lunacm
lunacm (64-bit) v10.5.0-470. Copyright (c) 2022 SafeNet. All rights reserved.

Available HSMs:

Slot Id ->              0
Label ->                jboss1
Serial Number ->        1312109862213
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 ->              1
Label ->                jboss2
```

```
Serial Number ->         1280780175898
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 ->               8
HSM Label ->             HA
HSM Serial Number ->     11312109862213
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

Current Slot Id: 0
```

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

> **NOTE:** For detailed instructions on how to establish an NTLS connection, initialize partitions, and manage different user roles, please refer to the Luna HSM documentation.

Set up Luna HSM High-Availability

To obtain information on configuring and setting up two or more HSM boxes on host systems for high availability, please consult the Luna HSM documentation. In order to enable failover functionality, it is necessary to activate the HAOnly option in the HA setup. This will enable automatic routing of all calls to the secondary system in the event of a primary system failure, until the primary system is restored and operational again.

Configure Luna Cloud HSM service

Follow these steps to set up your Luna Cloud HSM:

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 using the following command:

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

> **NOTE:** Do not extract to a new subdirectory. Place the files in the client install directory.

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

   ```
   source ./setenv
   ```

   NOTE: To add the configuration to an already installed UC client, use the `–addcloudhsm` option when running the setenv script.

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

   NOTE: If your organization requires non-FIPS algorithms for your operations, ensure that the Allow non-FIPS approved algorithms check box is checked. For more information, refer to Supported Mechanisms.

Java setup

Set the following variables of Java to use JBoss & Java KEYTOOL utility:

```
export JAVA_HOME=<Path to Java installation Directory>
export PATH=$JAVA_HOME/bin:$PATH
```

JBoss Setup

To proceed with the integration process, it is necessary to have JBoss server installed on the target machine. You can refer to the JBoss documentation for guidance on how to install the server.

1. Set the JBOSS_HOME variable and provide the path of JBoss installation directory.

   ```
   export JBOSS_HOME=<Path to JBoss installation directory>
   ```

2. Run the following command to start the JBoss Server.

   ```
   # sh $JBOSS_HOME/bin/standalone.sh
   ```

   After installing and starting JBoss, check whether the server is functioning correctly by browsing http://localhost:8080/.Integrating JBoss with Thales Luna HSM or Luna Cloud HSM

Configure JBoss for SSL acceleration

1. Copy `libLunaAPI.so` and `LunaProvider.jar` from `/usr/safenet/lunaclient/jsp/lib` location to `$JAVA_HOME/jre/lib/ext/.`

   Example:

   ```
   cp /usr/safenet/lunaclient/jsp/lib/libLunaAPI.so $JAVA_HOME/jre/lib/ext/
   cp /usr/safenet/lunaclient/jsp/lib/LunaProvider.jar $JAVA_HOME/jre/lib/ext/
   ```

2. Modify the java.security file to include the Luna Provider. It is available at $JAVA_HOME/jre/lib/security location.

   ```
   security.provider.1=com.safenetinc.luna.provider.LunaProvider
   security.provider.2=sun.security.provider.Sun
   security.provider.3=sun.security.rsa.SunRsaSign
   security.provider.4=sun.security.ec.SunEC
   ```

```
security.provider.5=com.sun.net.ssl.internal.ssl.Provider
security.provider.6=com.sun.crypto.provider.SunJCE
security.provider.7=sun.security.jgss.SunProvider
security.provider.8=com.sun.security.sasl.Provider
security.provider.9=org.jcp.xml.dsig.internal.dom.XMLDSigRI
security.provider.10=sun.security.smartcardio.SunPCSC
```

3. Generate a keystore by going to the configuration directory located at `$JBOSS_HOME/standalone/configuration/`.
4. Create a keystore file and include the entry `tokenlabel:<Partition_Name>` in it, where `<Partition_Name>` is the name of the partition on the Luna HSM.
5. Use the Java keytool utility to create a new keystore that uses the luna provider to generate a key and certificate on the Luna HSM partition.

   ```
   keytool -genkey -keystore <keystore name> -storepass <partition password> -
   alias <key label> -keypass <partition password> -keyalg <key algorithm> -
   keysize <size of key> -sigalg <signing algorithm> -validity <no. of days> -
   storetype <name of keystore>
   ```

   For Example:

   RSA Keys:

   ```
   keytool -genkey -keystore mykeystore -storepass userpin1 -alias jboss -
   keypass userpin1 -keyalg RSA -keysize 2048 -sigalg SHA1withRSA -validity
   365 -storetype luna
   ```

   ECDSA Keys:

   ```
   keytool -genkey -keystore mykeystore -storepass userpin1 -alias jboss -
   keypass userpin1 -keyalg EC -keysize 256 -sigalg SHA1withECDSA -validity
   365 -storetype luna
   ```

6. Generate a key and certificate in the Luna Network HSM and a keystore in the current directory by entering the required details, as prompted.

   > **NOTE:** If the HSM is set to FIPS mode, then SHA1withRSA and SHA1withECDSA signature algorithms should be substituted with SHA256withRSA and SHA256withECDSA, respectively.

7. Create a Certificate Signing Request (CSR) using the key that has been generated.

   For RSA Keys:

   ```
   keytool -certreq -alias <key label> -file <request file> -keypass
   <partition password> -keystore <keystore name> -storepass <partition
   password> -sigalg <signing algorithm> -storetype <name of keystore>
   ```

   For Example:

   RSA Keys:

   ```
   keytool -certreq -alias jboss -file cert.csr -keypass userpin1 -keystore
   mykeystore -storepass userpin1 -sigalg SHA1withRSA -storetype luna
   ```

ECDSA Keys:

```
keytool -certreq -alias jboss -file cert.csr -keypass userpin1 -keystore
mykeystore -storepass userpin1 -sigalg SHA1withECDSA -storetype luna
```

> **NOTE:** If the HSM is set to FIPS mode, then SHA1withRSA and SHA1withECDSA signature algorithms should be substituted with SHA256withRSA and SHA256withECDSA, respectively.

8. Modify the `java.security` file available at `$JAVA_HOME/jre/lib/security`.

For JAVA version 1.8.0_261 and above:

```
security.provider.1=sun.security.provider.Sun
security.provider.2=sun.security.ec.SunEC
security.provider.3=com.sun.net.ssl.internal.ssl.Provider
security.provider.4=com.sun.crypto.provider.SunJCE
security.provider.5=com.safenetinc.luna.provider.LunaProvider
security.provider.6=sun.security.rsa.SunRsaSign
security.provider.7=sun.security.jgss.SunProvider
security.provider.8=com.sun.security.sasl.Provider
security.provider.9=org.jcp.xml.dsig.internal.dom.XMLDSigRI
security.provider.10=sun.security.smartcardio.SunPCSC
```

For JAVA versions below 1.8.0_261:

```
security.provider.1=sun.security.provider.Sun
security.provider.2=sun.security.rsa.SunRsaSign
security.provider.3=sun.security.ec.SunEC
security.provider.4=com.sun.net.ssl.internal.ssl.Provider
security.provider.5=com.sun.crypto.provider.SunJCE
security.provider.6=com.safenetinc.luna.provider.LunaProvider
security.provider.7=sun.security.jgss.SunProvider
security.provider.8=com.sun.security.sasl.Provider
security.provider.9=org.jcp.xml.dsig.internal.dom.XMLDSigRI
security.provider.10=sun.security.smartcardio.SunPCSC
```

9. Submit the contents of the generated CSR to the CA for signing the certificate request.
10. Save the root certificate and signed certificate obtained from the CA in the current directory with the filenames `RootCA.cer` and `jboss.cer`, respectively.

11. Import the CA root certificate in the key store:

```
keytool -import -trustcacerts -alias rootCA -file RootCA.cer -keystore
mykeystore -storepass <password> -storetype luna
```

12. Import the signed certificate in the key store:

```
keytool -import -trustcacerts -alias jboss -keypass <partition password> -
file jboss.cer -keystore mykeystore -storepass <keystore password> -
storetype luna
```

> **NOTE:** Before importing the certificate in key store you must import the CA Root certificate and Intermediate certificate also if any.

13. Make changes to the standalone.xml configuration file located in the `$JBOSS_HOME/standalone/configuration` directory, as described in the next step.

14. Edit the `<ssl>` configuration under the `<management> <security-realms>` section, as indicated below:

```
<security-realm name="ApplicationRealm">
<server-identities>
<ssl>
<keystore path="mykeystore" relative-to="jboss.server.config.dir"
provider="luna"                                keystore-password="userpin1"
alias="jboss" key-password="userpin1"/>
</ssl>
</server-identities>
```

> **NOTE:** Here, “userpin1” refers to the Luna HSM partition pin.

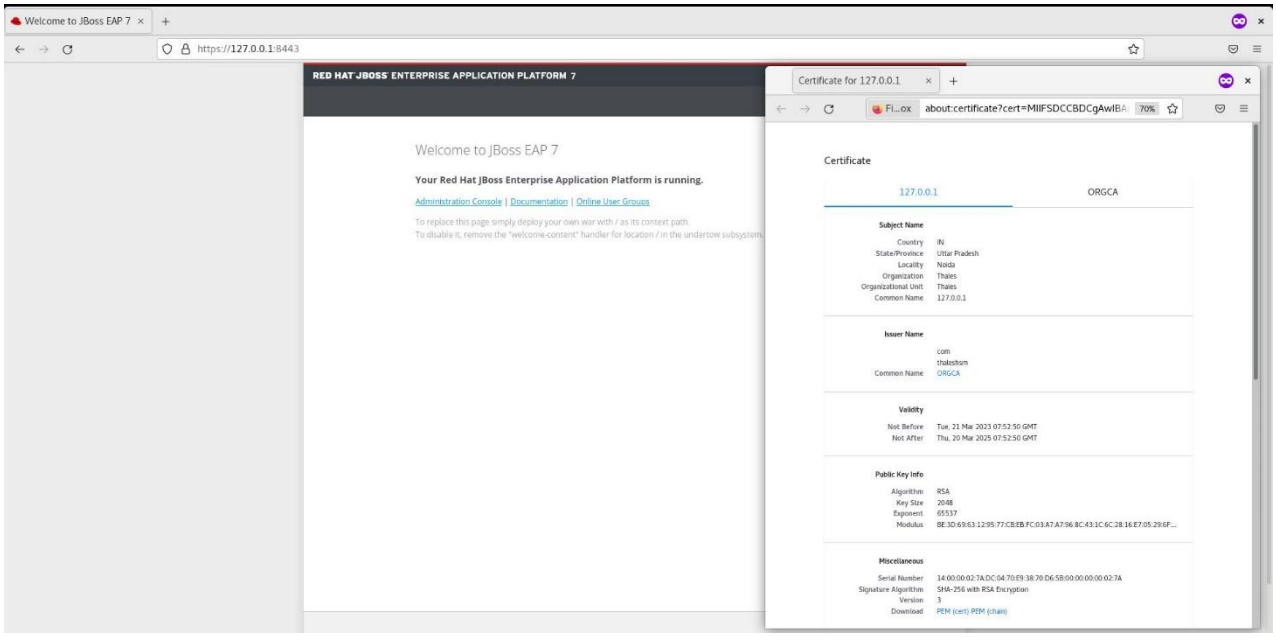
15. Initiate the server using the following command:

```
sh $JBOSS_HOME/bin/standalone.sh
```

16. Access the SSL-enabled server by entering the following URL in a web browser:

https://<hostname or ip address>:8443

18. Open the web console by accepting the certificate, thus confirming the successful configuration of SSL on JBoss, while ensuring the security of the Private Key and SSL Certificate on the Thales Luna HSM.



The integration between JBoss and Thales Luna HSM or Luna Cloud HSM has now been successfully accomplished.

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.