THALES

Oracle WebLogic Server: Integration Guide

THALES LUNA HSM

Document Information

Document Part Number	007-012420-001
Revision	N
Release Date	28 September 2022

Trademarks, Copyrights, and Third-Party Software

CONTENTS

Overview

This guide provides you step by step instructions for integrating Oracle WebLogic Server with Luna HSM. You can use the instructions contained in this document to set up a small test lab that has Oracle WebLogic Server running with Luna HSM to secure SSL private keys. This document also includes steps for Oracle WebLogic Server installation, which is a prerequisite for Luna HSM integration.

This guide is intended for experienced administrators who are responsible for planning, implementation, and deployment of WebLogic Server. These administrators are expected to be familiar with the WebLogic Server concepts as well as the WebLogic Server Administrative Console.

Oracle WebLogic Server is an enterprise-ready Java EE application server that supports the deployment of many types of distributed applications and is an ideal foundation for building applications based on Service Oriented Architectures (SOA). SOA is a design methodology aimed at maximizing the reuse of application services. The benefits that you can obtain by securing Oracle Web Logic Server private keys with Luna HSM include:

- Secure generation, storage, and protection of the encryption keys on FIPS 140-2 level 3 validated hardware
- Full life cycle management of the keys
- Access to secure audit trail
- Using cloud services with confidence

Certified Platforms

This integration is certified on the following platforms:

- Certified platforms on Luna HSM v7.x
- Certified platforms on Luna HSM v5.x and v6.x

Certified platforms on Luna HSM v7.x

Oracle WebLogic Server	Platforms tested	Luna HSM appliance software and firmware	Luna HSM client software version	JDK
WLS 14.1.1.0.0	RHEL 7(64 bit)	Software: 7.7.1 Firmware: 7.7.2	10.4.0	Oracle JDK 11.0.6
WLS 14.1.1.0.0	RHEL 7(64 bit)	Software: 7.7.1 Firmware: 7.7.2	10.4.0	Oracle JDK 1.8.0_331
WLS 12.2.1.4.0	RHEL 7(64 bit)	Software: 7.7.1 Firmware: 7.7.2	10.4.0	Oracle JDK 1.8.0_331

Oracle WebLogic Server	Platforms tested	Luna HSM appliance software and firmware	Luna HSM client software version	JDK
WLS 12.2.1.3	RHEL 7(64-bit)	Software: 7.2.0 Firmware: 7.2.0	7.2.0	Oracle JDK 1.8_131
WLS 12.2.1.1	RHEL 7(64-bit)	Software: 7.1.0 Firmware: 7.1.0	7.1.0	Oracle JDK 1.8_91
WLS 12.2.1.1	RHEL 7(64-bit)	Software: 7.0.0 Firmware: 7.0.1	7.0.0	Oracle JDK 1.8_91

Certified platforms on Luna HSM v5.x and v6.x

Oracle WebLogic Server	Platforms tested	Luna HSM appliance software and firmware	Luna HSM client software version	JDK
WLS 12.2.1	RHEL 7.0(64-bit)	Software: 6.3.0 Firmware: 6.27.0 and 6.10.9	6.x (v6.3.0)	Oracle JDK 1.8.0_91
WLS 12.2.1	RHEL 6.5(64-bit)	Software: 6.2.2 Firmware: 6.24.3 and 6.10.9	6.x (v6.2.2)	Oracle JDK 1.8.0_91
WLS 12.2.1	RHEL 6.5(64-bit)	Software: 6.2.1 Firmware:6.24.2 and 6.10.9	6.x (v6.2.1)	Oracle JDK 1.8.0_91
WLS 12.1.2	RHEL 6.5(64-bit)	Software: 6.2.0 f/w 6.24.0 and 6.10.9	6.x (v6.2)	Oracle JDK 1.7.0_79
WLS 12.1.2	RHEL 7.0(64 bit)	Software: 6.0.0 Firmware: 6.22.0	6.x (v6.1)	Oracle JDK 1.7.0_79
WLS 12.1.2	RHEL 6.0(64 bit) RHEL 6.2(64 bit)	Software: 5.4.1 Firmware: 6.21.0	5.x (v5.0.x, 5.1.x, 5.2.x, 5.3.x, 5.4.x)	Oracle JDK 1.7.0_51

Oracle WebLogic Server	Platforms tested	Luna HSM appliance software and firmware	Luna HSM client software version	JDK
WLS 10.3.6	RHEL 5.5 (64 bit) RHEL 6.2 (64 bit) RHEL 6.0 (64 bit) Solaris 10 SPARC v9 (32 bit)	Software: 5.4.0 Firmware: 6.21.0	5.x (v5.0.x, 5.1.x, 5.2.x, 5.3.x, 5.4.x)	Oracle JRockit 1.6.0_45 Oracle JDK 1.6.0_45 Oracle JDK 1.7.0_25 Oracle JDK 1.7.0_72
WLS 10.3.5	RHEL 5.8(64 bit) RHEL 6.2(64 bit)	Software: 5.2.1 Firmware: 6.10.1	5.x (v5.0.x, 5.1.x, 5.2.x, 5.3.x, 5.4.x)	Oracle JDK 1.6.0_45

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 HSM. Luna HSMs are also available for access as an offering from cloud service providers such as IBM Cloud HSM and AWS CloudHSM Classic.

Prerequisites

You need to complete the following tasks before proceeding with the integration:

> Configure Luna HSM

> Install JDK

> Install Oracle WebLogic Server

Configure Luna HSM

To configure a Luna HSM device with Oracle WebLogic Server:

1. Verify that the HSM is set up, initialized, provisioned, and ready for deployment.
2. Create a partition on the HSM that will be later used by Oracle WebLogic 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 the partition is successfully registered and configured. The command to see the registered partition is:

```
lunacm (64-bit) v10.4.0-417. Copyright (c) 2021 SafeNet. All rights
reserved.
```

```
   Available HSMs:

   Slot Id ->              0
   Label ->                Oracle_WebLogic
   Serial Number ->        1280780175882
   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

   Current Slot Id: 0
```

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

Controlling user access to HSM

NOTE: This section is applicable only for Linux users.

By default, only the root user has access to the HSM. You can specify a set of non-root users that are permitted to access the HSM by adding them to the **hsmusers** group. The client software installation automatically creates the **hsmusers** group. The **hsmusers** group is retained when you uninstall the client software, allowing you to upgrade the software while retaining your **hsmusers** group configuration.

Adding a user to hsmusers group

To allow non-root users or applications access to the HSM, assign the users to the **hsmusers** group. The users you assign to the **hsmusers** group must exist on the client workstation.

1. Ensure that you have **sudo** privileges on the client workstation.
2. Add a user to the hsmusers group.

   ```
   # sudo gpasswd --add <username> hsmusers
   ```

   Where `<username>` is the name of the user you want to add to the hsmusers group.

Removing a user from hsmusers group

1. Ensure that you have sudo privileges on the client workstation.
2. Remove a user from the hsmusers group.

   ```
   # sudo gpasswd -d <username> hsmusers
   ```

   Where `<username>` is the name of the user you want to remove from the **hsmusers** group. You must log in again to see the change.

   NOTE: The user you delete will continue to have access to the HSM until you reboot the client workstation.

Set up Luna HSM High-Availability Group

Refer to the Luna HSM documentation for 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 get automatically routed to the secondary until the primary recovers and starts up.

Set up Luna HSM in FIPS Mode

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

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 Luna HSM in FIPS mode, you have to make the following change in the configuration file:

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

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

Install JDK

Before installing WebLogic Server, you need to install JDK. Download the JDK software from the Oracle support site. You can download and install one of the following JDK variants:

- Java Development Kit 11
- Java Development Kit 8
- Java Development Kit 7
- Java Development Kit 6
- JRockit

After installing JDK, the root user needs to perform the following actions:

1. Create a new group and user.

```
# groupadd -g 1000 oinstall
# useradd -u 1100 -g oinstall oracle
# passwd oracle
```

2. Create the directories in which the Oracle software will be installed.

```
# mkdir -p /u01/app/oracle/middleware
# chown -R oracle:oinstall /u01
# chmod -R 775 /u01/
```

After creating the user/group and directories for WebLogic Server, log in as “oracle” user and run the installer. Create the WebLogic domain and apply the appropriate patch to support Luna HSM.

Install Oracle WebLogic Server

1. Install Oracle WebLogic server, as described in Oracle documentation.
2. Apply a patch to support Luna HSM after completing Oracle WebLogic installation. Patch information for the WebLogic Server is provided below:
   - Oracle WebLogic Server 12.1.2: p17436068_121200_Generic.zip
   - Oracle WebLogic Server 10.3.6: p17436068_1036_Generic.zip
   - Oracle WebLogic Server 10.3.5: p17436068_1035_Generic.zip

   Note: Refer to the ReadMe.txt for installing the patch or Oracle documentation to use smart update utility for applying the patch.

Integrating Luna HSM with Oracle WebLogic Server

To integrate Luna HSM for Oracle WebLogic Server:

1. Create the HSM keystore and copy the **libLunaAPI.so** and **LunaProvider.jar** file from the **<Luna_installation_directory>/jsp/lib** folder to the **<JDK_installation_directory>/jre/lib/ext**.

   NOTE: Skip this step if you are using JDK 11.

2. Edit the java.security file:

   To configure java.security file for JDK 8

   i. Edit the Java Security Configuration file java.security located in the directory <JDK_installation_directory>/jre/lib/security.

   ii. Add the Luna Provider to the java.security file, as shown below:

```
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=sun.security.jgss.SunProvider
security.provider.6=com.sun.security.sasl.Provider
security.provider.7=org.jcp.xml.dsig.internal.dom.XMLDSigRI
security.provider.8=sun.security.smartcardio.SunPCSC
security.provider.9=com.safenetinc.luna.provider.LunaProvider
security.provider.10=sun.security.rsa.SunRsaSign
```

   iii. Save the changes to the **java.security** file.

   To configure java.security file for JDK 11

   i. Edit the Java Security Configuration file **java.security** located in the directory **<JDK_installation_directory>/conf/security**.

   ii. Add the Luna Provider to the **java.security** file, as shown below:

```
security.provider.1=SUN
security.provider.2=SunEC
security.provider.3=SunJSSE
security.provider.4=SunJCE
security.provider.5=SunJGSS
security.provider.6=SunSASL
security.provider.7=XMLDSig
security.provider.8=SunPCSC
security.provider.9=JdkLDAP
security.provider.10=JdkSASL
security.provider.11=SunPKCS11
security.provider.12=com.safenetinc.luna.provider.LunaProvider
security.provider.13=SunRsaSign
```

iii. Save the changes to the **java.security** file.

> **NOTE:** If using Luna HSM 5.2.1 and above, skip the steps 3 and 4 and proceed with step 5.

iv. Edit the Luna Configuration file make the following changes in the Misc section:

```
Misc = {
   AppIdMajor=1;
   AppIdMinor=1;
}
```

3. Use SALOGIN utility to open the session with Luna using Application ID defined in the Luna Configuration file. For example:

```
# /usr/safenet/lunaclient/bin/salogin -o -s 1 -i 1:1 -v -p <Partition Password>
```

4. Export the environment variables, as indicated in the examples below.

JDK 8:

```
# export JAVA_HOME=<JAVA_HOME>
# export PATH=$JAVA_HOME/bin:$PATH
```

JDK 11:

```
# export JAVA_HOME=<JAVA_HOME>
# export PATH=$JAVA_HOME/bin:$PATH
# export CLASSPATH=/usr/safenet/lunaclient/jsp/lib/LunaProvider.jar:$CLASSPATH
# export LD_LIBRARY_PATH=/usr/safenet/lunaclient/jsp/lib/:$LD_LIBRARY_PATH
```

5. Go to the WebLogic Server Domain directory, as indicated in the example below.

```
# cd /u01/app/oracle/middleware/user_projects/domains/base_domain/bin/
```

6. Set the domain environment variables by executing the `setDomainEnv.sh`, as indicated in the example below.

```
# . ./setDomainEnv.sh
```

7. Create a **lunastore** file and make following entry:

```
tokenlabel:<Partition Name>
```

8. Place the **lunastore** file in the "/home/oracle/Oracle/Middleware/Oracle_Home/user_projects/domains/base_domain" directory.

9. Generate a new keystore and key pair using Java Keytool utility, as indicated below.

For JDK 7/8

```
# keytool -genkeypair -alias lunakey -keyalg RSA -sigalg SHA256withRSA -keypass
temp123# -keysize 2048 -keystore lunastore -storepass temp123# -storetype luna
```

For JDK 11

```
# keytool -genkeypair -alias lunakey -keyalg RSA -keysize 2048 -sigalg
SHA256withRSA -keypass temp123# -keystore lunastore -storetype Luna -
storepass temp123# -providerpath
"/usr/safenet/lunaclient/jsp/lib/LunaProvider.jar" -providerclass
com.safenetinc.luna.provider.LunaProvider -J-
Djava.library.path=/usr/safenet/lunaclient/jsp/lib/ -J-cp -
J/usr/safenet/lunaclient/jsp/lib/LunaProvider.jar
```

Example:

```
# keytool -genkeypair -alias lunakey -keyalg RSA -sigalg SHA256withRSA -
keypass temp123# -keysize 2048 -keystore lunastore -storepass temp123# -
storetype luna

What is your first and last name?
[Unknown]: Hostname
What is the name of your organizational unit?
[Unknown]: Testing Only
What is the name of your organization?
[Unknown]: My Org
What is the name of your City or Locality?
[Unknown]: My City
What is the name of your State or Province?
[Unknown]: My State
What is the two-letter country code for this unit?
[Unknown]: UK
Is CN= Hostname, OU= Testing Only, O= My Org, L= My City, ST= My State, C=
UK correct?
[no]: yes
```

A new key pair will be generated on Luna HSM.

10. Generate a certificate request from a key in the keystore, as indicated below.

For JDK 7/8

```
# keytool -certreq -alias lunakey -sigalg SHA256withRSA -file certreq_file -
storetype luna -keystore lunastore
```

For JDK 11

```
# keytool -certreq -alias lunakey -sigalg SHA256withRSA -file certreq_file
-keystore lunastore -storetype Luna -providerpath
```

```
"/usr/safenet/lunaclient/jsp/lib/LunaProvider.jar" -providerclass
com.safenetinc.luna.provider.LunaProvider -J-
Djava.library.path=/usr/safenet/lunaclient/jsp/lib/ -J-cp -
J/usr/safenet/lunaclient/jsp/lib/LunaProvider.jar
```

11. Submit the CSR file to a CA such as VeriSign and Entrust. The CA authenticates the request and returns a signed certificate or a certificate chain. Save the reply in the current working directory.
12. Import the CA root certificate and signed certificate or a certificate chain into the keystore.
    - To import the CA root certificate, execute the following command:

      For JDK 7/8

      ```
      # keytool -trustcacerts -importcert -alias rootca -file root.cer -
      keystore lunastore -storetype luna
      ```

      For JDK 11

      ```
      # keytool -trustcacerts -importcert -alias rootca -file root.cer -
      keystore lunastore -storetype Luna -providerpath
      "/usr/safenet/lunaclient/jsp/lib/LunaProvider.jar" -providerclass
      com.safenetinc.luna.provider.LunaProvider -J-
      Djava.library.path=/usr/safenet/lunaclient/jsp/lib/ -J-cp -
      J/usr/safenet/lunaclient/jsp/lib/LunaProvider.jar
      ```

    - To import the signed certificate reply or certificate chain execute the following command:

      For JDK 7/8

      ```
      # keytool -trustcacerts -importcert -alias lunakey -file mycert.cer -
      keystore lunastore -storetype luna
      ```

      For JDK 11

      ```
      # keytool -trustcacerts -importcert -alias lunakey -file mycert.p7b -
      keystore lunastore -storetype Luna -providerpath
      "/usr/safenet/lunaclient/jsp/lib/LunaProvider.jar" -providerclass
      com.safenetinc.luna.provider.LunaProvider -J-
      Djava.library.path=/usr/safenet/lunaclient/jsp/lib/ -J-cp -
      J/usr/safenet/lunaclient/jsp/lib/LunaProvider.jar
      ```

      `root.cer` and `mycert.p7b` are the CA Root Certificate and Signed Certificate request respectively.
13. Start the WebLogic Server using the following command:

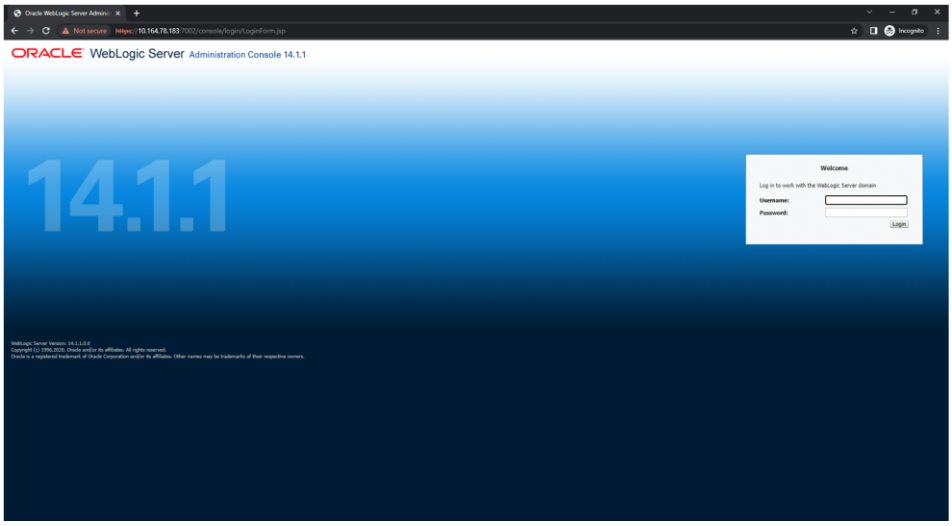
    ```
    # ./startWebLogic.sh
    ```

14. Open the Administration Console using the link **http://hostname:7001/console**.
15. Navigate to **Domain Structure** > **Environment** and complete the following steps:
    a. Click **Servers**.
    b. Click **AdminServer**.
    c. Click **Lock & Edit.**
    d. Select the **SSL Listen Port Enabled** checkbox.
    e. Click **Save**.

16. Open the **Keystores** tab and then complete the following steps:
    a. Click **Change**.
    b. From the drop-down menu, select **Custom Identity and Custom Trust**.
    c. Click **Save**.
    d. In the **Custom Identity Keystore** field, enter **lunastore**.
    e. In the **Custom Identity Keystore Type** field, enter **luna.**
    f. In the **Custom Identity Keystore Passphrase** field, enter **<part_password>**. Confirm the passphrase.
    g. In the **Custom Trust Keystore** field, enter **lunastore.**
    h. In the **Custom Trust Keystore Type** field, enter **luna**.
    i. In the **Custom Trust Keystore Passphrase** field, enter **<part_password>**. Confirm the passphrase.
    j. Click **Save**.
17. Open the **SSL** tab.
    a. In the **Private Key Alias** field, enter lunakey.
    b. In the **Private Key Passphrase** field, enter **<part_password>**. Confirm the passphrase.
    c. Click **Save**.
18. Click **Advanced**.
    a. Select the **Use JSSE SSL** check box.

    > **Note:** In Oracle Weblogic 12c, **USE JSSE SSL** option would not be available.

    b. Click **Save**.
19. Click **Activate Changes**.
20. Log out from the **Administration** console.
21. Restart the WebLogic Server. Open the **Administration** console using the link **https://hostname:7002/console**.



Troubleshooting

Problem

SSL_BAD_MAC_ALERT message received when accessing the Administration Console using **https://hostname:7002/console**.

Solution

Ensure that you have applied the appropriate patch for Luna HSM support and selected the **Use JSSE SSL** check box under the **SSL > Advanced tab** in the Administration Console.

Contacting Customer Support

If you encounter a problem while installing, registering, or operating this product, refer to the documentation. If you cannot resolve the issue, 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.

Email Support

You can also contact technical support by email at technical.support.DIS@thalesgroup.com.