THALES

OpenSSH: Integration Guide

THALES LUNA HSM

Document Information

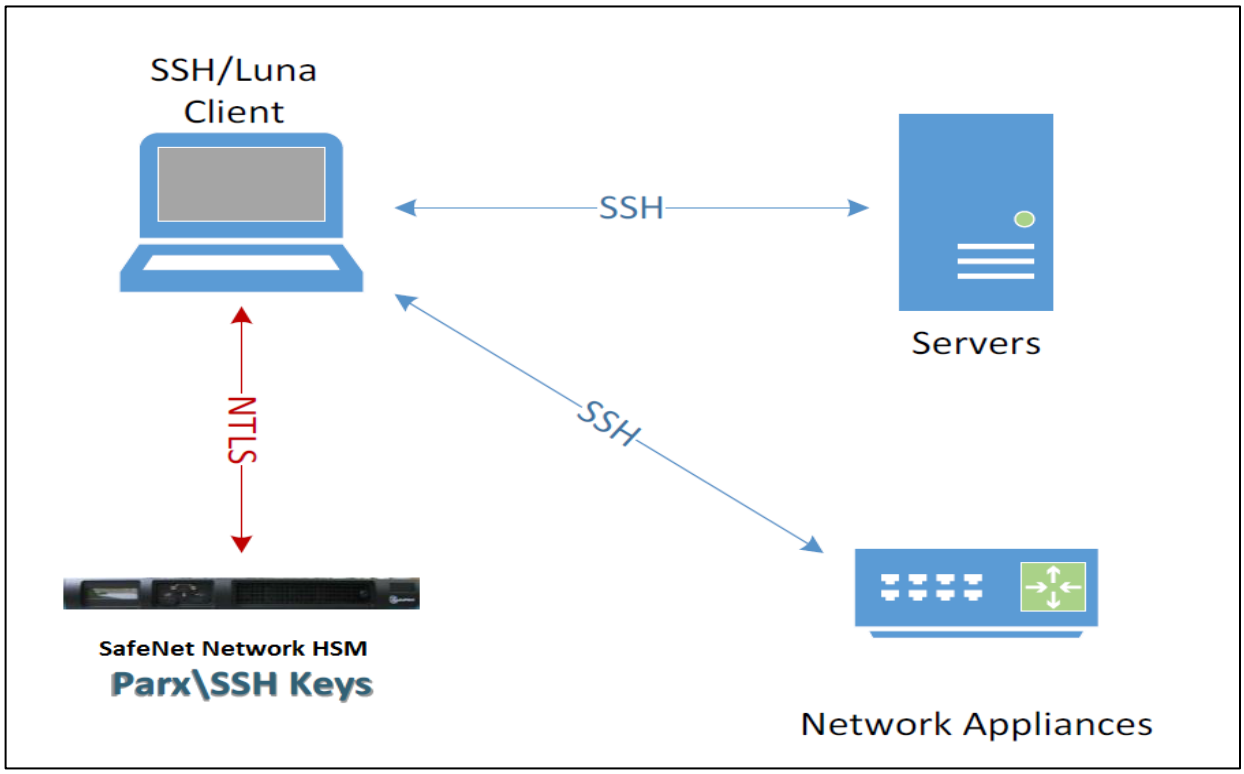
Document Part Number	007-013723-001
Revision	C
Release Date	7 December 2020

Trademarks, Copyrights, and Third-Party Software

CONTENTS

Overview

OpenSSH (also known as Open BSD Secure Shell) is a suite of security-related network-level utilities based on the Secure Shell (SSH) protocol that help secure network communications via the encryption of network traffic over multiple authentication methods and by providing secure tunneling capabilities.



This document guides through the steps for integrating OpenSSH with a Thales Luna HSM. You can use a Thales Luna HSM to secure the SSH Keys. The benefits of securing the cryptographic keys with a Thales Luna HSM include:

- Secure generation, storage and protection of the Identity signing private key on FIPS 140-2 level 3 validated hardware.
- Full life cycle management of the keys.
- HSM audit trail.
- Significant performance improvements by off-loading cryptographic operations from application servers.

Certified Platforms

This integration is certified with Luna HSM on the following operating systems:

Platform Certified	OpenSSH Version
Ubuntu 20	OpenSSH-8.2p1
Windows Server 2019	OpenSSH-8.1p1-Beta
Cent OS 6.6 (64 bit)	OpenSSH-5.3p1
Solaris SPARC 11	OpenSSH-6.6p1

NOTE: Any Luna Client version will support this integration, provided the supported Luna HSM and OpenSSH are used.

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

Prerequisites

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

Configure Thales Luna HSM

To configure Luna HSM:

1. Ensure that the HSM is set up, initialized, provisioned and ready for deployment.
2. Create a partition that will be later used by OpenSSH.
3. Create and exchange certificate between the Luna Network HSM and Client system. Register client and assign partition to create an NTLS connection. Initialize 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 partitions is:

```
# /usr/safenet/lunaclient/bin/lunacm (For Linux)

# /opt/safenet/lunaclient/bin/lunacm (For Solaris)

# C:\Program Files\SafeNet\LunaClient>lunacm.exe (For Windows)

lunacm (64-bit) v10.2.0-111.Copyright(c) 2020 SafeNet. All rights reserved.
Available HSMs:
Slot Id ->              0
Label ->                partition_ssh
```

```
Serial Number ->          1238696044903
Model ->                  LunaSA 7.4.0
Firmware Version ->       7.4.0
Configuration ->          Luna User Partition With SO (PW) Signing 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 for creating NTLS connection, initializing the partitions, and assigning various user roles.

To use 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. RSA PKCS and X9.31 key generation are no longer approved for operation in a FIPS-compliant HSM. If you are using Luna HSM in FIPS mode, make the following change in configuration file:

```
[Misc]

RSAKeyGenMechRemap=1
```

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

NOTE: For Luna Client 10.x onwards, this setting is not needed. It is applicable for Luna Client 7.x only.

To configure Luna HSM HA (High-Availability)

Refer to Luna HSM documentation for HA steps and details regarding configuring and setting up two or more HSM appliances on Windows and UNIX systems. You must enable the HAOnly setting in HA for failover to work so that if primary stop functioning for some reason, all calls automatically routed to secondary till primary starts functioning again.

Integrating Luna HSM with OpenSSH on Linux

To set up OpenSSH with SafeNet HSM on Linux distribution, OpenSC is required for PKCS11 supports that provides the tools to support key generation on tokens. To integrate Luna HSM with OpenSSH on Linux:

1. Install the OpenSC.

   ```
   # yum install opensc
   ```

   Alternatively you can install from source file from https://github.com/OpenSC/OpenSC/wiki

2. Generate a key pair on the HSM partition using the pkcs11-tool specifying the libCryptoki2_64.so crypto module and provide the partition password when prompted.

   ```
   # pkcs11-tool --module=/usr/safenet/lunaclient/lib/libCryptoki2_64.so --
      login --keypairgen --key-type rsa:2048 --id "numerical unique id" --label
      "friendly-key-name"
   ```

For Example:

```
# pkcs11-tool --module=/usr/safenet/lunaclient/lib/libCryptoki2_64.so --
   login --keypairgen --key-type rsa:2048 --id "10001001" --label "myssh-
   key"
```

```
root@localhostranjan:~# pkcs11-tool --module=/usr/safenet/lunaclient/lib/libCryptoki2_64.so --login --keypa
irgen --key-type rsa:2048 --id "10001001" --label "myssh-key"
Using slot 0 with a present token (0x0)
Logging in to "partition_ssh".
Please enter User PIN:
Key pair generated:
Private Key Object; RSA
  label:      myssh-key
  ID:         10001001
  Usage:      decrypt, sign, unwrap
warning: PKCS11 function C_GetAttributeValue(ALWAYS_AUTHENTICATE) failed: rv = CKR_ATTRIBUTE_TYPE_INVALID (
0x12)

  Access:     sensitive, always sensitive, never extractable, local
Public Key Object; RSA 2048 bits
  label:      myssh-key
  ID:         10001001
  Usage:      encrypt, verify, wrap
  Access:     local
```

3. Start the SSH agent.

```
# eval `ssh-agent -P "/usr/safenet/lunaclient/lib/*" -s`
```

4. Register the HSM partition with the SSH authentication agent using the Thales Luna client crypto provider using the below command.

```
# ssh-add -s /usr/safenet/lunaclient/lib/libCryptoki2_64.so
```

Provide partition password when prompted.

```
root@localhostranjan:~# ssh-add -s /usr/safenet/lunaclient/lib/libCryptoki2_64.so
Enter passphrase for PKCS#11:
Card added: /usr/safenet/lunaclient/lib/libCryptoki2_64.so
```

5. Display the keys available to the SSH agent using the below command:

```
# ssh-add -l
```

```
root@localhostranjan:~# ssh-add -l
2048 SHA256:6WR8+B8c1t7siFzWqnZR6eV1qRHc3YXuRw1p006sJ8M myssh-key (RSA)
```

6. Install the public key in the ~/.ssh/authorized_keys file of the remote server. Remote server is any server that you want to connect with SSH client.Provide remote server login password when prompted.

```
# ssh-copy-id 10.164.76.144
```

```
root@localhostranjan:~# ssh-copy-id 10.164.76.144
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already in
stalled
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the
 new keys
root@10.164.76.144's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh '10.164.76.144'"
and check to make sure that only the key(s) you wanted were added.
```

Now if you login to the remote server and check ~/.ssh/authorized_keys file, you will find details such as follows:

```
# cat .ssh/authorized_keys

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDwnklleZyQa1gPaEyZC4EF+BAXRc4uf2F6RxUg0x1YDQF
mkQAg+9UYPTQXtzFugA/MCqVUeQ70U5Idg6jRopAlIt8vwsZFaJnbmA6JwWXcTA3ISX6mFfoNGv
do7K/rtuvhaASIDMmNk4XTKxv2ScOL980eNw26KwMWDAFDvL4nExdQgarnvw/giUzoQn4vTYwIk
n5hj5uUvI3DXRaFk45LA9//8drqPuP555QYn7lFobZ/2pDa1lKePGbVWBCV6GKOEjVdcFAW83hI
JuMn5uAMwd+9IDp76hhMj+JHpwgLBMJIbXI9xgKWD5HphvuFEg6zasUnhiwapq62vZ8MC9sB
myssh-key
```

7. Connect to the remote SSH server specifying the Luna crypto provider utilizing the –I ssh option.

```
# ssh -v -I/usr/safenet/lunaclient/lib/libCryptoki2_64.so 10.164.76.144
```

This completes the integration of Luna HSM with OpenSSH on Linux.

Additional Notes

1. The SSH client must have an assigned and connected partition on the HSM when attempting to connect to the remote SSH server.
2. Use the ./vtl verify command on the SSH client to confirm.
3. Create symbolic link to the crypto provider libCryptoki2_64.so. In this example, the link is “/cp” and that is all that needs to be specified in the ssh connection after adding the symbolic link.

```
# ln -s /usr/safenet/lunaclient/lib/libCryptoki2_64.so /cp

# ssh -v -I/cp 10.164.76.5
```

4. The ssh –v option is for debug and demonstration purposes, add more v’s to get more detail, ssh –vvvv.
5. The sshd_conf file can be configured to not use passwords and force RSA authentication.
6. To stop using your private key for whatever reason, disconnect the HSM partition “card” from the authentication agent with the following command:

```
# ssh-add –e /usr/safenet/lunaclient/lib/libCryptoki2_64.so
```

Followed with the ssh-add –l command to confirm the agent has no identities.

7. Currently the SSH registration is not maintained between logins & reboots. The ssh-agent must be started and the SSH registration to the HSM must be run with the following commands:

```
# eval `ssh-agent -s`

# ssh-add -s /usr/safenet/lunaclient/lib/libCryptoki2_64.so
```

Integrating Luna HSM with OpenSSH on Solaris

Open SSH is not installed by default on Solaris system but you can download the Open SSH source code and built it on Solaris. To integrate Luna HSM with OpenSSH HSM on Solaris:

1. Download the OpenSSH source files form the below URL:

```
https://www.openssh.com/
```

2. Extract the downloaded file and then go to the extracted folder.

```
# tar zxvf openssh-6.6p1.tar.gz
# cd openssh-6.6p1
```

3. Set the CFLAGS and LDFLAGS variable to build the source for 64 bit platform.

```
# export CFLAGS=-m64
# export LDFLAGS=-m64
```

4. Build and install the Open SSH using below commands. It will by default get installed in "/usr/local". If you want to specify any other directory you can use --prefix=<directory location> with configure command.

```
# mkdir /var/empty
# chown root:sys /var/empty
# chmod 755 /var/empty
# groupadd sshd
# useradd -g sshd -c 'sshd privsep' -d /var/empty -s /bin/false sshd
# ./configure
# make package
# pkgadd -d OpenSSH-OpenSSH_6.6p1-Solaris-sparc.pkg
```

5. To use the OpenSSH set the PATH variable, else Sun SSH will be used by default.

```
# export PATH=/usr/local/bin:/usr/local/sbin:$PATH
# ssh -V
OpenSSH_6.6p1, OpenSSL 1.0.0e 6 Sep 2011
```

6. Generate a key pair on the HSM partition using the CMU utility provided with the Luna Client using below command and provide the partition password when prompted. After password it will prompt for RSA Mechanism, select PKCS.

```
# ./cmu generatekeypair -modulusBits=2048 -publicExponent=65537 -sign=T -
verify=T -encrypt=T -decrypt=T -wrap=T -unwrap=T –id="numerical unique id"
–label="friendly-key-name"
```

For Example:

```
# ./cmu generatekeypair -modulusBits=2048 -publicExponent=65537 -sign=T -
verify=T -encrypt=T -decrypt=T -wrap=T -unwrap=T -id=10000001 -label=myssh-
key
Please enter password for token in slot 0 : ********
Select RSA Mechanism Type -
[1] PKCS [2] FIPS 186-3 Only Primes [3] FIPS 186-3 Auxiliary Primes : 1
```

NOTE: Where label and id is any user defined string.

7. After generating the key, start the SSH agent by using the below command:

```
# eval `ssh-agent -s`
```

8. Register the HSM partition with the SSH authentication agent using the SafeNet HSM client crypto provider using the below command and provide partition password when prompt displays.

```
# ssh-add -s /opt/safenet/lunaclient/lib/libCryptoki2_64.so
```

Provide partition password when prompted.

On success following will be displayed:

```
Card added: /usr/safenet/lunaclient/lib/libCryptoki2_64.so
```

9. Display the keys available to the SSH agent using the below command:

```
# ssh-add -L
```

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCsL+XpWzQy/9bUTPJaGi49xuwGTkm+KbFmpMDwS65c7Se
Z4+LpEJV3dqeaHnABuGGaCek6hl9O+P3sR6iz4Vs6yupc4mjagG1I1VUSphUAAvcHwDo+C4Aoxj
36A8+EWFzWpXdgJQQ6SUzYr7V/l6LsSoXcGogDdkFAzKXqsN6BzACYZYU8camVI8H7hFaQ0ptPW
AC1EAGthL+9/t0wCDWMekc32+/js/n2jXlxxAl+T5WRHJ3BzaxHOQ6QCCDGv1Omfv4W+ipkOOHs
gVslTVROR1gl4Ho9UGuSedDmNvn2GelM9UJ6blddSeCSbidQmamPUNxFxmRvKu/I+exuyfgf
/opt/safenet/lunaclient/lib/libCryptoki2_64.so
```

10. Login to the remote server and copy the public key in the ~/.ssh/authorized_keys of the remote server. Remote server is any UNIX based server that you may want to connect with SSH client.

11. Check ~/.ssh/authorized_keys on remote server to ensure that Public Key is copied correctly:

```
# cat ~/.ssh/authorized_keys
```

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCsL+XpWzQy/9bUTPJaGi49xuwGTkm+KbFmpMDwS65c7Se
Z4+LpEJV3dqeaHnABuGGaCek6hl9O+P3sR6iz4Vs6yupc4mjagG1I1VUSphUAAvcHwDo+C4Aoxj
36A8+EWFzWpXdgJQQ6SUzYr7V/l6LsSoXcGogDdkFAzKXqsN6BzACYZYU8camVI8H7hFaQ0ptPW
AC1EAGthL+9/t0wCDWMekc32+/js/n2jXlxxAl+T5WRHJ3BzaxHOQ6QCCDGv1Omfv4W+ipkOOHs
gVslTVROR1gl4Ho9UGuSedDmNvn2GelM9UJ6blddSeCSbidQmamPUNxFxmRvKu/I+exuyfgf
/opt/safenet/lunaclient/lib/libCryptoki2_64.so
```

> **NOTE:** Luna crypto library should be available in the ~/.ssh/authorized_keys.

12. Edit the /usr/local/etc/sshd_config and make the following changes:

```
#RSAAuthentication yes
```

```
#PasswordAuthentication no
```

13. Connect to the remote SSH server specifying the Luna crypto provider utilizing the –I ssh option.

```
# ssh -v -I/opt/safenet/lunaclient/lib/libCryptoki2_64.so 10.164.76.21
```

This completes the integration of Luna HSM with OpenSSH on Solaris.

Integrating Luna HSM with OpenSSH on Windows

To integrate Luna HSM with OpenSSH on Windows:

1. Go to https://github.com/PowerShell/Win32-OpenSSH/wiki/Install-Win32-OpenSSH and follow the instruction to download and install the latest OpenSSH package for Windows.
2. Generate a key pair on the HSM partition using the CMU utility provided with the Luna Client using below command and provide the partition password when prompted. After password it will prompt for RSA Mechanism, select PKCS.

   ```
   # .\cmu.exe generatekeypair -labelpublic "any_label" -labelprivate "any_label" -keytype "rsa" -sign T -verify T -wrap T -unwrap T -encrypt T -decrypt T -publicExponent 65537 -modulusBits 2048
   ```

   For Example:

   ```
   # .\cmu.exe generatekeypair -labelpublic "sshpub" -labelprivate "sshpriv" -keytype "rsa" -sign T -verify T -wrap T -unwrap T -encrypt T -decrypt T -publicExponent 65537 -modulusBits 2048

   Please enter password for token in slot 0 : ********

   Select RSA Mechanism Type -

   [1] PKCS [2] FIPS 186-3 Only Primes [3] FIPS 186-3 Auxiliary Primes : 1

   ...The key pair was successfully generated -> public handle(661), private handle(396)
   ```

3. Start the ssh-agent if it is not running using Powershell.

   ```
   Start-Service ssh-agent

   Get-Service ssh-agent
   ```

   ```
   PS C:\Users\Administrator> Start-Service ssh-agent
   PS C:\Users\Administrator> Get-Service ssh-agent

   Status   Name               DisplayName
   ------   ----               -----------
   Running  ssh-agent          OpenSSH Authentication Agent
   ```

4. List the public key and provide parititon password when prompted:

   ```
   ssh-keygen -D "C:\Program Files\SafeNet\LunaClient\cryptoki.dll"
   ```

   ```
   PS C:\Program Files\SafeNet\LunaClient> ssh-keygen -D "C:\Program Files\SafeNet\LunaClient\cryptoki.dll"
   Enter PIN for 'partition_ssh':
   ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCh0s0dIO61B8GPrzTziGrLaFe530xJk9Co831vNuIX6GZDuPrI8MK9V4UG08xIiBzpgoOwMTeameOQdHYV
   uHh58/SORnBoWFLgXwZa78DBXJIC04+xKaT9MdM9yF0nlqbKoOTkW26zyJ3ZK/78jGWlbLcgFcopzl0E44YyMiugRmQK3HLw4Suxml81LjLFOyyDnP1zYMtZ
   3pcL4MCW6DoroKMmbMrfsKUvfNN/AX73fiDlFhQhLKcB8JL/UifMZEBf4CT5QvMawo3BNG2tYa63QiehRYOIvOKpRQyixF/ieNAqkLne0BZV9p9exfCr2Qaw
   lZZBPQz4WZ1O9dMT6sNr
   ```

5. Copy the public key that is listed above and paste it to the .ssh\authorized_keys file on remote server.
6. Connect to the remote server with ssh and provide the partition password when prompted.

   ```
   ssh.exe -v -I "C:\Program Files\SafeNet\LunaClient\cryptoki.dl" username@remote_server_IP
   ```

This completes the integration of Luna HSM with OpenSSH on Windows.

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.