Encryption Guide

Hammerspace version 4.3

About This Guide

This guide will assist in configuring Hammerspace data encryption for Object storage.

Overview

Hammerspace has the capability of encrypting data that is stored in object and cloud storage. This also includes data encryption as data is transferred between sites in a multi-site Global File System configuration.

The architecture approach is using a plug-in architecture for Hammerspace to expand into additional vendors over time. This also enables customers to easily switch from one vendor to another without having to re-encrypt all their data.

Supported Key Management solutions

The only supported external Key Management solution in this release is nCipher HSM.

nCipher HSM

Hammerspace Key Management System

Overview

The nCipher hardware security module (HSM) is the key management service currently supported by Hammerspace. Specifically, Hammerspace, requires the nCipher Web Services Option Pack, which provides access to nCipher HSM platforms through a standard REST-based web interface.

Hammerspace cloud upload and download services use AES secret-key encryption to encrypt file chunks before uploading them to the cloud. The key is encrypted (wrapped) by a master AES key obtained from the HSM; that wrapped key is then stored with each chunk in the cloud.

When the file chunks are downloaded, the encryption context containing the wrapped key is also downloaded, decrypted with the configured HSM master key and then used to decrypt the file chunks.

The entire communications channel between the cloud mover and the HSM is also encrypted and secured using standard TLS encryption and security protocols, including mutual authentication between client and server components. The PKI materials for the REST TLS channel are themselves secured by the HSM.

Requirements

The following components are required to add an nCipher HSM as a key management service to Hammerspace:

- One (or more) nCipher HSM hardware device(s)
- One 64-bit Linux server that conforms with nCipher Web Services Option Pack (WSOP) requirements. CentOS 7.6 or newer works well. A VM is sufficient, but a bare metal box will also work. Installation of a GUI desktop is not required. 32 GB of disk space is recommended. At least 4 GB of RAM is recommended. At least 2 CPUs are required; 4 is recommended for better performance. The system should be configured with a static IP address. A DNS hostname is optional, but highly recommended to make certificate management simpler in the face of future network configuration changes.
- The required nCipher Security World software components should be installed on the WSOP system, along with the WSOP components. This document describes the process for installing version 12.50.4 of nCipher Security World and version 1.0.0 of the Web Services Option Pack.
- The optional Java JCE Security World components are also required. These come bundled with Security World and using them does not incur an additional cost.
- The Security World installed on the WSOP server should be configured with the nCipher HSM devices.
- The WSOP server should be configured with appropriate PKI materials.

The following sections provide specific instructions for setting up the requirements outlined above.

Linux WSOP Server Setup

The installation instructions are included here as a reference only.

1. Download a supported version of CentOS DVD image, CentOS 7.6 is later is preferred.
2. If using a virtual machine, configure the virtual machine as per the requirements in the overview section. Power on the VM or bare metal system and begin the CentOS installation process. Move through the initial screens to the Installation Summary screen.
3. Select Network & Host Name first. Click the Configure button and configure the ethernet adapter with a static IP address on your network by selecting the IPv4 Settings tab. Click Save and enter the desired host and domain name at the bottom of the window. Enable the ethernet adapter in the upper right corner. Click Done to return to the Installation Summary screen.

A static IP is required so that the WSOP PKI materials (certificates) can be configure with proper stable identities. A DNS hostname is highly recommended to allow the PKI materials to be configured with an abstract name for the WSOP server's address, which could change later.

4. Select Time & Date and select the proper Timezone. If you have your own Network Time Protocol (NTP) server, select the gear icon in the upper right corner and add your NTP server address to the list. Disable defaults if desired. Click Done to return to the Installation Summary screen.
5. Select Installation Destination and click Done to return to the Installation Summary screen.

   You may modify the disk configuration if you wish, but the default configuration is sufficient. Entering this screen and immediately leaving it removes the warning on the Installation Summary screen.
6. Select Software Selection and then select the Server radio button (a GUI is not required). Select any desired add-ons on the left. None of these add-ons are required, but some additional packages will be needed after installation has completed. Click Done to return to the Installation Summary screen.
7. Click Begin Installation at the bottom. While installing, click Root Password and set a secure root password for the system. Also, while installing, click User Creation and create a non-root administrative user. The passwords for these accounts should be kept secure to avoid compromising the contents of the Security World that will be configured on this system at a later stage.
8. When installation has completed, click Reboot at the bottom of the screen to boot up the newly installed system.

Java 8 Installation

Install Oracle Java 8 Java Runtime Environment (JRE). Navigate in a web browser to the Oracle Java SE Downloads page at https://www.oracle.com/technetwork/java/javase/downloads/index.html, scroll down until you find the entry for the latest Java SE 8 update, and click the JRE Download button. Accept the license agreement and download the file named **jre-8uXXX-linux-x64.rpm**. The **XXX** in the filename should be the latest update number. Copy this file into the /root directory on your Linux server. Run the following command from the file location to install the package:

```
# rpm -Uvh jre-8u*-linux-x64.rpm
warning: jre-8u221-linux-x64.rpm: Header V3 RSA/SHA256 Signature, key ID ec551f03: NOKEY
Verifying...                ########################################## [100%]
Preparing...                ########################################## [100%]
```

```
Updating / installing...
   1:jre1.8-1.8.0_221-fcs      ################################## [100%]
Unpacking JAR files...
        plugin.jar...
        javaws.jar...
        deploy.jar...
        rt.jar...
        jsse.jar...
        charsets.jar...
        localedata.jar...
# java -version
java version "1.8.0_221"
Java(TM) SE Runtime Environment (build 1.8.0_221-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.221-b11, mixed mode)
```

Legacy NIS Installation [CentOS 8 ONLY]

If you are installing on a CentOS 8 system, you will need to install the legacy NIS package:

```
# yum install libnsl.x86_64
…
Total download size: 87 k
Installed size: 147 k
Is this ok [y/N]: y
…
libnsl-2.28-42.el8.1.x86_64.rpm
…
Importing GPG key 0x8483C65D:
 Userid     : "CentOS (CentOS Official Signing Key) <security@centos.org>"
 Fingerprint: 99DB 70FA E1D7 CE22 7FB6 4882 05B5 55B3 8483 C65D
 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial
Is this ok [y/N]: y
Key imported successfully
…
Installed:
  libnsl-2.28-42.el8.1.x86_64

Complete!
#
```

Security World Software Installation

1. Use an SSH Client (PuTTY on Windows or ssh on another Linux system) to open a remote console to the new Linux server. Login as the administrative user you created during installation.
2. Use the **su** utility to change into the root account (use a dash after the **su** command to enable the root environment and move you to the root user's home directory):

   $ **su -**
   Password: ********
   # **pwd**
   /root

3. Copy the Security World software packages into a directory within the root user's home directory (e.g., **/root/nCipher**). Specifically, the **SecWorld-linux64-user-12.50.4.iso.zip** and **wsop-user-1.00.00.zip** files should be copied into this directory.
4. Unzip these files using the **unzip** utility. You should be left with three new files beside the zip files, **SecWorld-linux64-user.12.50.4.iso**, **wsop-user-1.00.00.iso** and **rnotes_1_1.pdf** (release notes that ship with the wsop ISO image).
5. Create directories in which to mount the iso images, then mount these images within the new directories:

   # **mkdir -p /mnt/sw**
   # **mount -t iso9660 -o loop /root/nCipher/SecWorld-linux64-user.12.50.4.iso /mnt/sw**
   mount: /mnt/sw: WARNING: device write-protected, mounted read-only.
   # **mkdir -p /mnt/wsop**
   # **mount -t iso9660 -o loop /root/nCipher/wsop-user-1.00.00.iso /mnt/wsop**
   mount: /mnt/wsop: WARNING: device write-protected, mounted read-only.

6. Within the **/mnt/sw/document** directory, you should now find documentation for installing Security World software. Locate files named **nShield_Connect_Installation_Guide.pdf** and **nShield Connect User Guide for Unix**; copy these files to a Windows system and open them for reference. Also do this with the file, **/mnt/wsop/document/WSOP_User_Guide.pdf**. References to these documents will be made from here on.
7. Instructions for installing Security World software begin in Chapter 4 of the nShield Connect Installation Guide. Instructions for installing on Linux systems begins at the bottom of page 26. You are already logged in as root so change to the root directory (cd /) and begin with step 4 of the instructions:

Note The instructions in the nShield Connect Installation Guide are general purpose and based on the idea that Security World software might be installed for many purposes. This guide is more explicit, indicating exactly the packages needed for using Security World software with the WSOP.

tar xf /mnt/sw/linux/libc6_11/amd64/nfast/hwsp/agg.tar
tar xf /mnt/sw/linux/libc6_11/amd64/nfast/ctls/agg.tar
tar xf /mnt/sw/linux/libc6_11/amd64/nfast/nhfw/agg.tar
tar xf /mnt/sw/linux/libc6_11/amd64/nfast/hwcrhk/user.tar
tar xf /mnt/sw/linux/libc6_11/amd64/nfast/hwcrhk/gnupg.tar
tar xf /mnt/sw/linux/libc6_11/amd64/nfast/javasp/agg.tar
tar xf /mnt/sw/linux/libc6_11/amd64/nfast/jcecsp/user.tar
tar xf /mnt/wsop/linux/libc6_11/amd64/wsop/wsopinst/user.tar
/opt/nfast/sbin/install
---- Stopping any nCipher servers ----

No nCipher init scripts installed.

---- Cleaning up any old install ----

No nCipher components requiring cleanup found.

---- Installing ----
-- Running install fragment 10nfastug
Checking for user 'nfast' in group 'nfast'
Creating nfast group.
Creating nfast user.
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
Checking user 'nfast' is in correct group 'nfast'
users created correctly

-- Running install fragment 11systemd
Register the SELinux policy for nFast, will take some time.

-- Running install fragment 15makefiles
Setting up directories.
Making default config file.
Making default cardlist file

-- Running install fragment 45drivers
Unloading old nCipher PCI nfp driver.
Checking for PCI nfp hardware.

```
Warning: No suitable pre-built PCI driver available.

No nCipher PCI nfp devices found.
Installing startup scripts for 'drivers'.
Not linking in init scripts or loading drivers.

-- Running install fragment 46exard
Remove old nCipher PCI miniHSM devices.
Checking for nCipher PCI miniHSM hardware.
No nCipher PCI miniHSM devices found.
Installing startup scripts for 'exard'.
Not linking in init scripts or loading drivers.

-- Running install fragment 50hardserver
Configuring hardserver privileges.
ls: cannot access '/dev/nfastpci*': No such file or directory
Installing startup scripts for 'hardserver'.
Linking in init scripts
Adding and enabling a systemd unit
Synchronizing state of nc_hardserver.service with SysV service script with
/usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nc_hardserver
Created symlink /etc/systemd/system/multi-user.target.wants/nc_hardserver.service →
/etc/systemd/system/nc_hardserver.service.
Note: Forwarding request to 'systemctl enable nc_hardserver.service'.
Synchronizing state of nc_hardserver.service with SysV service script with
/usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nc_hardserver
Warning: Installed, but no directly attached hardware was found.  If
you have an nCipher PCI card , re-run 'install' script with hardware
attached, or with '-d' option, or consult nCipher support.
Starting nCipher 'hardserver' server process.
waiting for nCipher server to become operational ...
nCipher server now running

-- Running install fragment 60cmdadp

-- Running install fragment 70edgecfg

 ---- Installation complete ----
#
```

8. If you wish, edit your **/root/.bash_profile** script, prepending **‘/opt/nfast/bin:’** to the **PATH** variable defined therein. After editing, source the script to make the

changes take effect in the current shell (**source /root/.bash_profile**). Test the installation now to ensure all is working properly by running the **nfkminfo** utility. The output should be as shown in the top portion of the example output at the bottom of the Linux section of Chapter 4 in the nShield Connect Installation Guide.

Note: The bottom portion of this output will show up when an HSM has been added to the Security World.

9. Copy the **nCipherKm.jar** file into the Oracle JRE Extensions directory:

```
# cp /opt/nfast/java/classes/nCipherKM.jar /usr/java/latest/lib/ext/
```

NOTE: This step is documented in the nShield Connect User Guide for Unix, Chapter 9: Application Interfaces, in the section entitled "nCipherKM JCA/JCE CSP", step 2.

10. The WSOP server is a standard embedded Tomcat service running the WSOP web application. The web application communicates with the Hardserver via the Hardserver's non-privileged listen port. This port is not enabled by default. Enable it now as follows:

```
# /opt/nfast/bin/config-serverstartup -s --port  9000
[server_settings] change successful; you must restart the hardserver for this to take effect
# /opt/nfast/sbin/init.d-ncipher restart
 -- Running shutdown script 50hardserver

 -- Running shutdown script 46exard|

 -- Running shutdown script 45drivers

 -- Running startup script 45drivers

 -- Running startup script 46exard

 -- Running startup script 50hardserver
waiting for nCipher server to become operational ...
nCipher server now running
```

Configuring the Linux Server to Use an Existing Security World

The steps to configure the server are very specific to the customer environment. The instructions on how to do this are provided in the nShield Connect User Guide for Unix.

Generating PKI Key Materials and Starting the WSOP Server

At this point the Security World software has been installed and configured to use the existing security world and HSM devices, and the Hardserver is running with the non-privileged port open, listening for requests from the WSOP server. The WSOP server components have been copied into the **/opt/nfast/wsop** directory, but the WSOP Tomcat application server has not yet been installed and started.

The next step is to generate public key infrastructure (PKI) key materials for the WSOP application server, as it cannot be started without these materials in place. To be clear, the only requirement here is that a properly configured Java key store be configured and installed into the **/opt/nfast/wsop/tls** directory.

The following steps will guide you through this process. These instructions assume you are still logged in as root and sitting in the root directory of the Linux WSOP server:

1. Rename the example files:

   # **pushd /opt/nfast/wsop/conf**
   /opt/nfast/wsop/conf /
   # **mv application.properties.example application.properties**
   # **mv wsop_java.security.example wsop_java.security**
   # **popd**
   /

2. Generate a Java key store containing a self-signed server certificate, using the Java **keytool** utility:

   # **export JAVA_TOOL_OPTIONS='-Dprotect=module -DignorePassphrase=true **
   -Djava.security.properties=/opt/nfast/wsop/conf/wsop_java.security'
   # **keytool -genkeypair -storetype nCipher.sworld -keyalg RSA **
   -keysize 2048 -validity 3650 -alias mykey
   -keystore /opt/nfast/wsop/tls/keystore.ks
   Picked up JAVA_TOOL_OPTIONS: -Dprotect=module -DignorePassphrase=true
   -Djava.security.properties=/opt/nfast/wsop/conf/wsop_java.security
   Enter keystore password: **<contrive a reasonable password>**
   Re-enter new password: **<repeat>**
   What is your first and last name?
   [Unknown]: **<enter Linux WSOP server host name here>**
   What is the name of your organizational unit?
   [Unknown]: **<enter your organization within your company>**
   What is the name of your organization?
   [Unknown]: **<enter your company name>**
   What is the name of your City or Locality?
   [Unknown]: **<enter your city name>**
   What is the name of your State or Province?

```
 [Unknown]:  <enter your two-letter state>
What is the two-letter country code for this unit?
 [Unknown]:  US
Is CN=???, OU=???, O=???, L=???, ST=??, C=US correct?
 [no]:  yes

Enter key password for <mykey>
      (RETURN if same as keystore password): <press ENTER>
```

Note: You may also use a CA-signed certificate for your WSOP application server's identity. In this case, follow the instructions on page 8 of the WSOP User Guide, the section entitled, "Using a Certificate Signed by a CA".

3. Edit the **/opt/nfast/wsop/conf/application.properties** file and modify the key store password to match the one you selected when you created the key store in the previous step. (The key store filename should already be **keystore.ks**.)

4. Change into the root home directory (/root) and create a self-signed client certificate and private key pair using the **openssl** tool:

```
# cd
# openssl req -x509 -newkey rsa:2048 -keyout client-key.pem \
-out client-cert.pem -days 3650 -nodes
Generating a 2048 bit RSA private key
.....................................................................+++
.........+++
writing new private key to 'client-key.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]: US
State or Province Name (full name) [Some-State]: <two-letter state>
Locality Name (eg, city) []: <city>
Organization Name (eg, company) [Internet Widgits Pty Ltd]: <company>
Organizational Unit Name (eg, section) []: <organization>
Common Name (e.g. server FQDN or YOUR name) []: <client host name>
Email Address []: <press ENTER>
```

Note: The Common Name used above should uniquely identity the Hammerspace client that will converse with the WSOP server. If possible, it

should be the host name of the Hammerspace metadata server (MDS). If unknown, any unique host-like name will work.

5. Still in the root user's home directory, export the WSOP server certificate to a DER certificate file, then convert that DER file to PEM format so it can be used with Hammerspace:

```
# keytool -export -alias mykey -file server-cert.der \
-storetype nCipher.sworld -keystore /opt/nfast/wsop/tls/keystore.ks
Picked up JAVA_TOOL_OPTIONS: -Dprotect=module -DignorePassphrase=true
-Djava.security.properties=/opt/nfast/wsop/conf/wsop_java.security

Enter keystore password: <ENTER>

**********  WARNING WARNING WARNING  **********
* The integrity of the information stored in your keystore  *
* has NOT been verified!  In order to verify its integrity, *
* you must provide your keystore password.                  *
**********  WARNING WARNING WARNING  **********

Certificate stored in file <server-cert.der>
# openssl x509 -inform der -in server-cert.der -out server-cert.pem
```

Note: The WARNING above is not important. It's merely telling us that since we didn't provide the proper password, the integrity of the key store was not verified by keytool when it opened it to read the server's certificate.

6. Back in the root directory, import the client certificate (but not the private key) into a new Java trust store for the WSOP server:

```
# cd /
# keytool -import -trustcacerts -file /root/client-cert.pem \
-alias client2 -keystore /opt/nfast/wsop/tls/truststore.ts
Picked up JAVA_TOOL_OPTIONS: -Dprotect=module -DignorePassphrase=true
-Djava.security.properties=/opt/nfast/wsop/conf/wsop_java.security

WARNING: nCipher provider was called for keystore mechanism 'JKS'.
A new keystore was loaded, so a software keystore is being used.

Enter keystore password: changeit
Re-enter new password: changeit
…
Trust this certificate? [no]:  yes
Certificate was added to keystore
```

Note: The WARNING above is fine – it just means you didn't specify ***-storetype nCipher.sworld*** *on the* ***keytool*** *command line, which is ok because this is merely a trust store and will not hold any private key material.*

Note: The trust store password was set to 'changeit' because all Java key stores require a password but trust stores hold public information and it's industry convention to set the trust store password to this value. The application properties file is already configured with this trust store password.

7. Install the WSOP application server by running the installation script:

```
# /opt/nfast/wsop/sbin/install
-- Running install fragment wsopd
Checking for user 'wsopd' in group 'wsopd'
Creating wsopd group.
Creating wsopd user.
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
Checking user 'wsopd' is in correct group 'wsopd'
users created correctly
Installing startup scripts for 'wsopd'.
Created symlink /etc/systemd/system/multi-user.target.wants/wsopd.service →
/etc/systemd/system/wsopd.service.
Starting nCipher 'wsopd' server process.
 ---- Installation complete ----
```

8. The WSOP REST service listens on port 18001 for remote requests. If you enabled the CentOS firewall during server installation it must be configured to allow inbound traffic on this port:

```
# firewall-cmd --zone=public --add-port=18001/tcp --permanent
success
# firewall-cmd --reload
success
```

Adding an AES Key to the HSM for Hammerspace

The WSOP only supports "simple" keys of two types, an AES secret key and an RSA public/private key. For Hammerspace, generate an AES secret key using the following command from the WSOP server command line:

```
# generatekey simple protect=module type=AES size=256 ident=key1 \
```

plainname=hs-key1 nvram=no

key generation parameters:

operation	Operation to perform	generate
application	Application	simple
verify	Verify security of key	yes
type	Key type	AES
size	Key size	256
ident	Key identifier	key1
plainname	Key name	hs-key1
nvram	Blob in NVRAM (needs ACS)	no

Key successfully generated.

Path to key: /opt/nfast/kmdata/local/key_simple_key1

Testing the WSOP Server

From the root user's home directory, use the **curl** utility to verify the WSOP application server is working correctly and is properly accessing the Hardserver for key information:

curl --cacert server-cert.pem --cert ./client-cert.pem --key client-key.pem --header 'Accept: application/json' "https://<wsop-host>:18001/km/v1/keys"

{"keys":[{"created":"2019-10-02T03:09:54Z","kid":"urn:uuid:a22071d9-8ace-3a67-ac59-61129fb8f46b","sworldAppname":"simple","sworldIdent":"key1"}]}

The key returned was the simple 256-bit AES key added in the previous step. If the command was successful, we know the WSOP service is correctly configured and properly responding to requests.

*Note the **kid** field – we'll use this in the next step when we add the HSM to Hammerspace.*

Configuring the HSM in Hammerspace

The final step is to configure Hammerspace with the access information for the HSMs exposed through the WSOP REST service. This must be done through the Hammerspace Admin command line.

At the Hammerspace Admin prompt, enter the following command (note that anything between quotes on this command line may contain line wraps – for example, the PEM certs and keys may contain line wraps as long as the text is in quotes):

admin@host.domain> **kms-add --endpoint <wsop-host>:18001 --key-id urn:uuid:<kid-from-above> --name nCipher-hsm --type NCIPHER_WSOP --client-certificate '-----BEGIN CERTIFICATE-----MIIDq** ... **dY0E=-----END CERTIFICATE-----' --client-private-key '-----BEGIN PRIVATE KEY-----MIIEv** ... **Hsw==-----END PRIVATE KEY-----' --server-certificate-chain '-----BEGIN CERTIFICATE-----MIIDb** ... **bLaMg==-----END CERTIFICATE-----'**

Name: nCipher-hsm
Type: KmsType.NCIPHER_HSM
Internal ID: 1
ID: fb1e46c5-d493-428c-b755-7de8ac2b613d
Endpoint: https://isis:18001
Key identifier: urn:uuid:<kid-from-above>
Client id/cert: -----BEGIN CERTIFICATE-----MIIDq ...
Server id/cert: -----BEGIN CERTIFICATE-----MIIDb ...

The command line will accept line-wrap characters (CR, LF, CRLF, etc) within quoted parameter arguments. For instance, in the above sample command line the **--client-certificate** argument is a string of text between single quotes. Any text within these quotes may have line-wrap characters embedded.

You have now successfully completed the configuration.