CyberArk Conjur

nShield® HSM Integration Guide

2025-04-23

Table of Contents

Chapter 1. Introduction

CyberArk Conjur offers secrets management for applications and services. There are four different deployment models. The model tested in this Integration Guide is the Dynamic Access Provider (DAP). For more information, see Conjur Secrets Manager Enterprise features in the CyberArk Conjur online documentation.

The base product is provided as a containerized appliance and can be executed in Docker or Kubernetes. The testing in this Integration Guide uses a basic deployment of nCOP in Docker.

1.1. Container images

Two container images were created for the purpose of this integration: a hardserver container and a CyberArk Conjur application container. These images are stored in an external registry:

- `nshield-hwsp`

  A hardserver container image that controls communication between the HSM(s) and the application containers.

- `conjur-appliance`

  An Application Access Manager (AAM) container image from CyberArk that will host the Master DAP Server.

1.2. Product configurations

Entrust has successfully tested nShield HSM integration with CyberArk Conjur in the following configurations:

Software	Version
Security World	13.6.8
nCOP	1.1.3
Operating System	Ubuntu 24.04.2 LTS
CyberArk Conjur Appliance Image	13.5.0

1.3. Supported nShield hardware and software versions

Entrust has successfully tested with the following nShield hardware and software versions:

1.3.1. Connect XC

Security World Software	Firmware	Image	OCS	Softcard	Module
13.6.8	12.72.3 (FIPS 140-2 certified)	13.6.7	✓	✓	✓

1.3.2. nShield 5c

Security World Software	Firmware	Image	OCS	Softcard	Module
13.6.8	13.4.5 (FIPS 140-3 certified)	13.6.7	✓	✓	✓

1.4. Supported nShield HSM functionality

Feature	Support
Module-only key	Yes
OCS cards	Yes
Softcards	Yes
nSaaS	Yes
FIPS 140 Level 3 mode support	Yes

1.5. Requirements

Before installing these products, read the associated documentation:

- For the nShield HSM: *Installation Guide* and *User Guide*.
- If nShield Remote Administration is to be used: *nShield Remote Administration User Guide*.
- *nShield Container Option Pack User Guide*.
- DAP Deployment, refer to Conjur Secrets Manager Enterprise v13.5 in the CyberArk online documentation.
- HSM Master Key Encryption, refer to Encrypt the master key using an HSM in the CyberArk online documentation.

Furthermore, the following design decisions have an impact on how the HSM is installed and configured:

- Whether your Security World must comply with FIPS 140 Level 3 standards.

  If using FIPS 140 Level 3, it is advisable to create an OCS for FIPS authorization. For information about limitations on FIPS authorization, see the *Installation Guide* of the nShield HSM.

- Whether to instantiate the Security World as recoverable or not.

Entrust recommends that you allow only unprivileged connections unless you are performing administrative tasks.

1.6. More information

For more information about OS support, contact your CyberArk sales representative or Entrust nShield Support, https://nshieldsupport.entrust.com.

Access to the Entrust nShield Support Portal is available to customers under maintenance. To request an account, contact nshield.support@entrust.com.

Chapter 2. Procedures

2.1. Prerequisites

Before you can use nCOP and run the container images, complete the following steps:

1. Install Docker. For information, see Get Docker in the Docker online documentation.
2. Gain access to the Conjur appliance image.
3. Request the nCOP and Security World software from Entrust.
4. Set up the HSM. See the *Installation Guide* for your HSM.
5. Configure the HSM(s) to use the IP address of your container host machine as a client.
6. Load an existing Security World or create a new one on the HSM.
7. Copy the Security World and module files to your container host machine at a directory of your choice.
8. Create or edit the `cknfastrc` file in `/opt/nfast` and add one of the following config settings:
9. For OCS or Softcard protection:

```
CKNFAST_LOADSHARING=1
CKNFAST_NO_ACCELERATOR_SLOTS=1
```

10. For Module protection:

```
CKNFAST_FAKE_ACCELERATOR_LOGIN=1
```

11. Optionally, the following can be added to generate PKCS #11 debug logs at the example location:

```
CKNFAST_DEBUG=10
CKNFAST_DEBUGFILE=/opt/ncop/pkcs11.log
```

12. Create a `pkcs11.yml` file with the following content:

```
library: /opt/nfast/toolkits/pkcs11/libcknfast.so
wrapping_key: <wrapping_key name>
pin: <passphrase of ocs/softcard if required>
slot: <slot number for the intended key protection type>
```

By the default, the slot number for module protection is 0, for softcard protection 1, and for OCS protection 2. This can change depending on your HSM deployment. In our environment, slot 2 was used for softcard protection and slot 1 for OCS protection. The pin passphrase is not required if you are using module protection.

For more information on configuring and managing nShield HSMs, Security Worlds, and Remote File Systems, see the *User Guide* for your HSM(s).

Here is an example for **module** protection:

```
library: /opt/nfast/toolkits/pkcs11/libcknfast.so
wrapping_key: wrappingkey
slot: 0
```

Here is an example for **softcard** protection:

```
library: /opt/nfast/toolkits/pkcs11/libcknfast.so
wrapping_key: wrappingkey
pin: ncipher
slot: 2
```

Here is an example for **OCS** protection:

```
library: /opt/nfast/toolkits/pkcs11/libcknfast.so
wrapping_key: wrappingkey
pin: ncipher
slot: 1
```

2.2. Create and configure the nshield-hwsp container

The nShield hardserver container has to be configured to enable it to communicate with the CyberArk Conjur Master DAP Server in a later step, see Create and configure the Conjur application container and the Master DAP Server.

To deploy an nCOP container image for use with CyberArk Conjur:

1. Log in to the container host machine server with `root` privileges and launch a terminal window.
2. Set up the nCOP working directory:

```
% mkdir -p /opt/ncop
```

3. Transfer the nCOP tar file to the host machine and extract it into the `/opt/ncop` directory:

```
% tar xf ncop-1.1.3.tar -C /opt/ncop
```

4. Chand directory to `/opt/ncop`:

```
% cd /opt/ncop
```

5. Mount the Security World ISO file:

   Transfer the Security World ISO file to the host machine, then mount it so it can be used by the nCOP script that will create the hardserver image.

```
% mkdir SecWorld-13.6.8

% mount -o loop SecWorld_Lin64-13.6.8.iso SecWorld-13.6.8
```

6. Set up the hardserver image:

```
% ./make-nshield-hwsp SecWorld-13.6.8

Detecting nShield software version
Version is 13.6.8
Unpacking hwsp...
Removing redundant files...
Creating files...
.
.
.
 => exporting to image
2.4s
 => => exporting layers
2.4s
 => => writing image sha256:ecda66fxxxxxxxxxx5c06cfb335xxxxxxxxxxxbe16f27635f58576226a086917
0.0s
 => => naming to docker.io/library/nshield-hwsp:13.6.8
```

7. List the docker images to view the newly created hardserver image:

```
% docker images

REPOSITORY                    TAG       IMAGE ID       CREATED          SIZE
nshield-hwsp                  13.6.8    ecda66f40322   11 minutes ago   543MB
```

8. Configure `nshield-hwsp`:

   a. Set up the hardserver configuration file and directory:

```
% mkdir -p /opt/ncop/config1
```

```
% ./make-nshield-hwsp-config --output /opt/ncop/config1 config <hsm ip address>
```

b. Check that the configuration file information matches your HSM deployment:

```
% cat /opt/ncop/config1/config

syntax-version=1

[nethsm_imports]
local_module=1
remote_esn=5F08-02E0-D947
remote_ip=1X.1XX.1XX.XX
remote_port=9004
keyhash=732523000c324c8a674236d1ad783a4dae0179fe
privileged=0
```

c. Create a new socket so that application containers can use the hardserver:

```
% docker volume create socket1
```

d. Run the `nshield-hwsp` container:

```
% docker run -d -v /opt/ncop/config1:/opt/nfast/kmdata/config:ro -v socket1:/opt/nfast/sockets
nshield-hwsp:13.6.8
```

e. Check the status of `nshield-hwsp` using the `enquiry` command:

```
% NFAST_SERVER=/var/lib/docker/volumes/socket1/_data/nserver /opt/nfast/bin/enquiry
```

2.3. Create and configure the Conjur application container and the Master DAP Server

The assumption is that you are inside the `/opt/ncop` directory and that the Security World ISO file still mounted. Transfer the conjur-appliance tar file to the host machine.

1. Load the `conjur-appliance into the local Docker registry:

   The following command can be used to load the `conjur-appliance` .tar file into the local Docker repository:

```
% docker load -i <PATH-TO-TAR-FILE>/conjur-appliance-13.5.0.tar.gz
```

2. Extend the `conjur-appliance` image with the `nfast` utilities:

```
% ./extend-nshield-application --from registry.tld/conjur-appliance:13.5.0 --pkcs11 SecWorld-13.6.8

Detecting nShield software version
Version is 13.6.8
NOTICE: --pkcs11 included by default with 12.60 ISO. Flag ignored
Unpacking /opt/nfast/SecWorld-13.6.8/linux/amd64/hwsp.tar.gz ...
Unpacking /opt/nfast/SecWorld-13.6.8/linux/amd64/ctls.tar.gz ...
Adding files...
Building image...
[+] Building 18.2s (8/8) FINISHED
docker:default
 => [internal] load build definition from Dockerfile
0.0s
 => => transferring dockerfile: 257B
0.0s
 => [internal] load metadata for registry.tld/conjur-appliance:13.5.0
0.0s
 => [internal] load .dockerignore
0.0s
 => => transferring context: 2B
0.0s
 => [internal] load build context
2.5s
 => => transferring context: 264.45MB
2.3s
 => [1/3] FROM registry.tld/conjur-appliance:13.5.0
0.0s
 => [2/3] COPY opt /opt
11.8s
 => [3/3] RUN mkdir -p /opt/nfast/kmdata /opt/nfast/sockets && mkdir -m 1755 /opt/nfast/kmdata/tmp
0.2s
 => exporting to image
3.6s
 => => exporting layers
3.5s
 => => writing image sha256:cbf9fe5049afe72b3134d3c200baed0272020cc76cdd9b9af71f9a9e4c75ce28
0.0s
```

3. List the docker images

```
% docker images

REPOSITORY                     TAG       IMAGE ID       CREATED          SIZE
<none>                         <none>    cbf9fe5049af   45 seconds ago   1.99GB
nshield-hwsp                   13.6.8    ecda66f40322   14 minutes ago   543MB
registry.tld/conjur-appliance  13.5.0    ff712871eb0a   3 months ago     1.41GB
```

See the **IMAGE ID** for the new image. It will be labeled as **none**.

4. Tag the generated application image for convenience:

```
% docker tag <IMAGE ID> conjur-appliance-wnfast:13.6.8
```

5. List the images again to validate tagging.

```
% docker images

REPOSITORY                      TAG       IMAGE ID       CREATED          SIZE
conjur-appliance-wnfast         13.6.8    cbf9fe5049af   2 minutes ago    1.99GB
nshield-hwsp                    13.6.8    ecda66f40322   16 minutes ago   543MB
registry.tld/conjur-appliance   13.5.0    ff712871eb0a   3 months ago     1.41GB
```

6. Run the `conjur-appliance` container with the `nfast` container:

```
% docker run --name dap-wnfast -d \
  --restart=unless-stopped \
  --security-opt seccomp=/path/to/conjur-seccomp.json \
  -p "443:443" -p "5432:5432" -p "1999:1999" \
  -v /opt/nfast/kmdata:/opt/nfast/kmdata:rw \
  -v socket1:/opt/nfast/sockets \
  conjur-appliance-wnfast:13.6.8
```

7. Check the running containers

```
% docker ps -a

CONTAINER ID IMAGE                          COMMAND               CREATED            STATUS
PORTS                     NAMES
036009bab3fd conjur-appliance-wnfast:13.6.8 "/usr/local/bin/entr…" About a minute ago Up About a minute
0.0.0.0:443->443/tcp, ... dap-wnfast
4bafaddefabb nshield-hwsp:13.6.8            "/opt/nfast/sbin/nsh…" 12 minutes ago     Up 12 minutes
confident_vaughan
```

8. Perform the initial configuration of Conjur. The username is **admin**. For password requirements, see Configure the Conjur cluster in the CyberArk online documentation.

```
% docker exec dap-wnfast evoke configure master \
  --accept-eula --hostname dap-wnfast.example.com \
  --admin-password Mypassw0rD1! org1

.
.
.
You have accepted the CyberArk End User License Agreement.
Use `evoke eula show` in the future to display the EULA.
Configuration successful. Conjur Leader up and running.
```

9. Copy the `cknfastrc` and `pkcs11.yml` configuration files into the running container:

```
% docker cp /opt/nfast/cknfastrc dap-wnfast:/opt/nfast/cknfastrc

% docker cp pkcs11.yml dap-wnfast:/opt/conjur/etc/pkcs11.yml
```

10. Generate a new Key Encryption Key (KEK) for Conjur to be stored on the HSM:

```
% docker exec dap-wnfast evoke pkcs11 generate

...
2025-04-17 15:26:32 [2230]: pkcs11: 00000000 D    slot_destroy_hashmaps
2025-04-17 15:26:32 [2230]: pkcs11: 00000000 D    slot_destroy_hashmaps done
2025-04-17 15:26:32 [2230]: pkcs11: 00000000 <    rv 0x00000000
I, [2025-04-17T15:26:32.070847 #2230]  INFO -- : Using nCipher PKCS#11 13.6.8-209-a5bd9.
I, [2025-04-17T15:26:32.071106 #2230]  INFO -- : Using slot from config file
I, [2025-04-17T15:26:32.071177 #2230]  INFO -- : Using 5F08-02E0-D947 Rt1.
I, [2025-04-17T15:26:32.071837 #2230]  INFO -- : Generating a new wrapping key with ID "wrappingkey"...
I, [2025-04-17T15:26:32.094593 #2230]  INFO -- : All done.
```

11. Start the `conjur-appliance` container, which will act as the Master DAP Server, in Interactive mode:

```
% docker exec -i -t dap-wnfast /bin/bash
```

12. Check for the generated key. Make sure it was created with the intended key protection type, defined by the `pkcs11.yml` file.

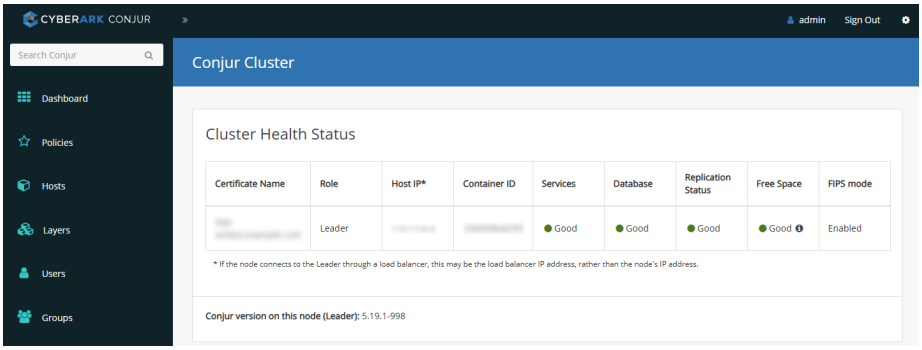
```
root@036009bab3fd:/# /opt/nfast/bin/nfkminfo -l

Keys with module protection:
 key_pkcs11_uaaexxxxxxxxxx60dbaxxxxxxxxxxx04bdde969a659 `Conjur master key wrapping key'
```

The KEK is now ready for use.

2.4. Web Interface

1. Log on to web interface: https://1X.1XX.1XX.XXX/
   a. User: admin
   b. Password: MypassW0rD1!
2. Select the **Settings** symbol in the top right.
3. Select **Conjur Cluster**.
4. The master node should now be displayed.



2.5. Example commands used with the KEK

Here are some examples of commands that can be used inside the docker container. For more examples, see Server Key Encryption Methods in the CyberArk online documentation.

1. Generate a random master key. Keep this file secure.

```
root@036009bab3fd:/# mkdir -p /secrets
root@036009bab3fd:/# openssl rand 32 > /secrets/master.key
```

2. Encrypt the server keys with the 32-byte master key

```
root@036009bab3fd:/# evoke keys encrypt /secrets/master.key

Encrypted 4 key files and adjusted 0 symlinks
NOTE: To allow services access to keys, the keys must be unlocked. For more information, run 'evoke keys
unlock --help'.
```

3. Unlock the server keys and restart the Conjur services

```
root@036009bab3fd:/# evoke keys unlock /secrets/master.key

Stopping service 'conjur'...
Stopping service 'nginx'...
Stopping service 'pg/main'...
Stopping service 'pg/audit'...
Stopping service 'seed'...
Stopping service 'syslog-ng'...
Starting service 'conjur'...
Starting service 'nginx'...
Starting service 'pg/main'...
Starting service 'pg/audit'...
Starting service 'seed'...
Starting service 'syslog-ng'...
Keys are unlocked. Services and scripts can now access encrypted data.
```

4. Wrap the master key

```
root@036009bab3fd:/#  evoke pkcs11 wrap /secrets/master.key
```

```
Using nCipher PKCS#11 13.6.8-209-a5bd9.
Using slot from config file
Using 5F08-02E0-D947 Rt1.
Using wrapping key "wrappingkey".
Wrapping the master key...
Wrapped key stored in /opt/conjur/etc/pkcs11-keys.
```

5. Lock the keys

```
root@036009bab3fd:/# evoke keys lock

Stopping service 'conjur'...
Stopping service 'nginx'...
Stopping service 'pg/main'...
Stopping service 'pg/audit'...
Stopping service 'seed'...
Keys are Locked. Services and scripts can no longer access encrypted data.
```

6. Unlock the keys

```
root@036009bab3fd:/#  evoke keys unlock

No master key found in the session keyring
Using PKCS#11 master key configuration...
Using nCipher PKCS#11 13.6.8-209-a5bd9.
Using slot from config file
Using 5F08-02E0-D947 Rt1.
Using wrapping key "wrappingkey"...
Service 'conjur' is not running. Skipping stop.
Service 'nginx' is not running. Skipping stop.
Service 'pg/main' is not running. Skipping stop.
Service 'pg/audit' is not running. Skipping stop.
Service 'seed' is not running. Skipping stop.
Stopping service 'syslog-ng'...
Starting service 'conjur'...
Starting service 'nginx'...
Starting service 'pg/main'...
Starting service 'pg/audit'...
Starting service 'seed'...
Starting service 'syslog-ng'...
Keys are unlocked. Services and scripts can now access encrypted data.
```

Chapter 3. Additional resources and related products

3.1. nShield Connect

3.2. nShield as a Service

3.3. nShield Container Option Pack

3.4. Entrust products

3.5. nShield product documentation