Hyperledger Fabric

nShield® HSM Integration Guide

2024-02-12

Table of Contents

Chapter 1. Introduction

This document describes how to integrate the Hyperledger Fabric with the Entrust nShield Container Option Pack (nCOP). This utilizes Entrust nShield hardware security module (HSM) as a Root of Trust for storage encryption, to protect the private keys and meet FIPS 140 Level 2 and 3 criteria.

1.1. Product configurations

Entrust has successfully tested nShield HSM integration with Hyperledger Fabric in the following configurations:

Product	Version
Fabric CA	1.5.2
Docker CE	20.10.12
Go	1.16.12
Host OS	Red Hat Enterprise Linux 8
Container OS	Ubuntu Bionic

1.2. Supported nShield hardware and software versions

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

1.2.1. Connect XC

Security World Software	Firmware	Image	OCS	Softcard	Module
12.80.4	12.50.11	12.80.4	✓	✓	✓

1.2.2. Connect +

Security World Software	Firmware	Image	OCS	Softcard	Module
12.80.4	12.50.8	12.80.4	✓	✓	✓

1.3. Supported nShield HSM functionality

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

1.4. Requirements

Familiarize yourself with:

- Hyperledger Fabric documentation: Hyperledger Fabric CA User's Guide.
- The nShield HSM: *Installation Guide* and *User Guide*.
- Your organizational Certificate Policy and Certificate Practice Statement, and a Security Policy or Procedure in place covering administration of the PKI and HSM:
  - The number and quorum of administrator cards in the Administrator Card Set (ACS), and the policy for managing these cards.
  - The number and quorum of operator cards in the Operator Card Set (OCS), and the policy for managing these cards.
  - The keys protection method: Module, Softcard, or OCS.
  - The level of compliance for the Security World, FIPS 140 Level 3.
  - Key attributes such as key size, time-out, or need for auditing key usage.

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

Chapter 2. Procedures

Prerequisites:

1. Install the Entrust nShield HSM using the instructions in the *Installation Guide* for the HSM.
2. Configure the Entrust nShield HSM to have the IP address of your container host machine as a client.
3. Install the Entrust nShield Security World Software, and configure the Security World as described in the *User Guide* for the HSM. The Security World and module files will be copied to the container. Instructions for this are detailed later in this guide.
4. Edit or create the `cknfastrc` file located in `/opt/nfast/`:
   - If using Module protection:

     ```
     CKNFAST_FAKE_ACCELERATOR_LOGIN=1
     ```

   - If using OCS or Softcard protection:

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

5. Install Git, cURL, Go, and Docker. See the Hyperledger Fabric documentation for more information on the prerequisites.
6. Get the Hyperledger Fabric repositories. The following installs both the fabric-ca-server and fabric-ca-client binaries in `$GOPATH/bin`:

   ```
   go get -u github.com/hyperledger/fabric-ca/cmd/...
   ```

2.1. Install nCOP on the host machine

The following is an example installation of nCOP. See *nShield Container Option Pack User Guide* for more details.

1. Log in to the container host machine server as **root**, and launch a terminal window.
2. Set up the nCOP working directory:

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

```
% tar xf ncop-1.1.1.tar -C /opt/ncop
```

3. Mount the Security World:

```
% mkdir SecWorld-12.80.4

% mount -o loop SecWorld_Lin64-12.80.4.iso SecWorld-12.80.4
```

4. Set up the hardserver image:

```
% cd /opt/ncop

% ./make-nshield-hwsp SecWorld-12.80.4
```

5. 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>

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

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

```
% docker volume create socket1
```

   c. Run the `nshield-hwsp` container:

```
% docker run -d -v /opt/ncop/config1:/opt/nfast/kmdata/config:ro -v socket1:/opt/nfast/sockets
nshield-hwsp:12.80.4
```

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

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

2.2. Build Hyperledger with PKCS #11 enabled

The default `Dockerfile` that comes with the Hyperledger Fabric CA software makes an image containing Alpine Linux. nCOP does not currently support Alpine Linux. The `Dockerfile` will need to be changed to make an image with a supported Operating System.

1. Edit the `Dockerfile`. For example (Ubuntu):

```
% vi /root/go/pkg/mod/github.com/hyperledger/fabric-ca@v1.5.2/images/fabric-ca/Dockerfile

ARG GO_VER
ARG ALPINE_VER
FROM ubuntu:bionic
ARG GO_LDFLAGS
ARG GO_TAGS

RUN apt-get update
RUN apt-get install -y curl gcc git musl-dev
RUN rm -rf /var/lib/apt/lists/*

ENV GOLANG_VERSION 1.13.2
RUN curl -sSL https://storage.googleapis.com/golang/go$GOLANG_VERSION.linux-amd64.tar.gz | tar -v -C
/usr/local -xz
ENV PATH /usr/local/go/bin:$PATH
RUN mkdir -p /go/src /go/bin && chmod -R 777 /go
ENV GOROOT /usr/local/go
ENV GOPATH /go
ENV PATH /go/bin:$PATH
ADD . $GOPATH/src/github.com/hyperledger/fabric-ca
RUN go install -tags "${GO_TAGS}" -ldflags "${GO_LDFLAGS}" \
        github.com/hyperledger/fabric-ca/cmd/fabric-ca-server \
        && go install -tags "${GO_TAGS}" -ldflags "${GO_LDFLAGS}" \
        github.com/hyperledger/fabric-ca/cmd/fabric-ca-client

ENV FABRIC_CA_HOME /etc/hyperledger/fabric-ca-server
EXPOSE 7054
CMD fabric-ca-server start -b admin:adminpw
```

2. Build the container image:

```
% cd /root/go/pkg/mod/github.com/hyperledger/fabric-ca@v1.5.2

% GO_TAGS=pkcs11 make docker
```

   The following images will be generated:

```
% docker images

REPOSITORY              TAG             IMAGE ID       CREATED        SIZE
hyperledger/fabric-ca   1.5.2           42c3a53f293f   2 hours ago   832MB
hyperledger/fabric-ca   amd64-1.5.2     42c3a53f293f   2 hours ago   832MB
hyperledger/fabric-ca   latest          42c3a53f293f   2 hours ago   832MB
```

3. Run the `make-nshield-application` (nCOP) script to build the final image. For example:

```
% cd /opt/ncop

% ./make-nshield-application --from hyperledger/fabric-ca:amd64-1.5.2 --tag hyperpkcs11nshield:amd64 --java
/opt/nfast/SecWorld-12.80.4
```

   In this example:

--from This is the Docker images container that you want a source image.

--tag This is the new image name that will appear in the list of Docker images.

--java This is an option to install Java, it is optional for this integration.

/mnt This is where your Security World ISO is mounted.

2.3. Configure and start the Hyperledger Fabric CA server

The volumes that will be used in the `docker run` command are: In the following steps, `fabric-ca-server-config.yaml` will be generated and then edited within the container. This file can also be created beforehand to simplify these steps.

The volumes that will be used in the `docker run` command are:

-v /opt/nfast/sockets.hwsp:/opt/nfast/sockets

Required for hardserver communication.

-v /opt/nfast/kmdata/local:/opt/nfast/kmdata/local:rw

Used to share `kmdata\local` between Security World and the app container.

-v /opt/nfast/cknfastrc:/opt/nfast/cknfastrc

File used when selecting PKCS #11 key protection type.

1. Run the container interactively with a bash prompt:

```
% docker run -it \
       -v socket1:/opt/nfast/sockets \
       -v /opt/nfast/kmdata/local:/opt/nfast/kmdata/local:rw \
       -v /opt/nfast/cknfastrc:/opt/nfast/cknfastrc \
       hyperpkcs11nshield:amd64-java
```

2. Run `enquiry` and `nfkminfo` within the container. The output should show a usable module and Security World.

```
% /opt/nfast/bin/enquiry

% /opt/nfast/bin/nfkminfo
```

3. Run `fabric-ca-server` to generate a new config file to edit. Then, type

Control-C or **X** multiple times to exit.

```
% fabric-ca-server start -b root:root
```

4. Edit the `fabric-ca-server-config.yaml` file:

```
% cd /etc/hyperledger/fabric-ca-server

% vi fabric-ca-server-config.yaml
```

5. Find the "bccsp" section and add the PKCS #11 settings. For example:

```
bccsp:
    default: PKCS11
    pkcs11:
        Library: /opt/nfast/toolkits/pkcs11/libcknfast.so
        Pin: 123456
        label: fabric
        hash: SHA2
        security: 256
```

   In this example:

   - The name of the Softcard or OCS is "fabric" and the pin is "123456".
   - If using module protection, the label will be "accelerator".
   - If using module protection with `loadsharing=1` in the `cknfastrc` file, the label will be "loadshared accelerator".
   - The pin can be left empty if using module protection.

6. Delete any keystore in `/etc/hyperledger/fabric-ca-server` such as the `msp` directory and the old `.pem` file so that new ones are generated with the HSM when the server is started.
7. Start the server:

```
% fabric-ca-server start
```

The new HSM protected PKCS #11 key can be found at `/opt/nfast/kmdata/local`. The cert is in the Hyperledger directory as `ca-cert.pem`.

2.4. Enroll and register a Fabric CA client

1. Edit the `fabric-ca-server-config.yaml` file and change the identity section as needed.
2. Start the server if it is not currently running. It needs to be running for the `enroll` command to work.

3. Create a directory environment variable:

```
% export FABRIC_CA_CLIENT_HOME=$HOME/fabric-ca/clients/admin
```

4. Enroll under the identity in the server YAML file:

```
% fabric-ca-client enroll -u http://root:root@localhost:7054
```

   This was done to generate a client YAML file to edit. The client is not yet enrolled through the HSM.

5. Edit the client YAML file in your home directory environment path.
6. Delete the `msp` directory and edit the YAML file in the home directory you exported.
7. Edit the "bccsp" section and mirror the server YAML BCCSP for the HSM.
8. Run the `enroll` command again with the server identity:

```
% fabric-ca-client enroll -u http://root:root@localhost:7054
```

9. Register the client. For example:

```
% fabric-ca-client register --id.name ica.example --id.type client --id.secret root --csr.names
C=es,ST=madrid,L=Madrid,O=example.com --csr.cn ica.example -m ica.example --id.attrs
'"hf.IntermediateCA=true"' -u http://localhost:7054 --loglevel debug
```

2.5. Peers and ordering nodes

To set up peers and ordering nodes with the Entrust nShield HSM:

- Edit one more YAML file for each node. Use the same PKCS #11 BCCSP template as shown above. This will be the `core.yaml` file for a peer node and the `orderer.yaml` file for a ordering node.
- Run the enrollment lines from the peer or ordering nodes to the main fabric CA server to enroll it.

See the Hyperledger Fabric documentation for more information.

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 digital security solutions

3.5. nShield product documentation