Red Hat Certificate System

nShield® HSM Integration Guide

2026-01-28

© Entrust Corporation. All rights reserved.

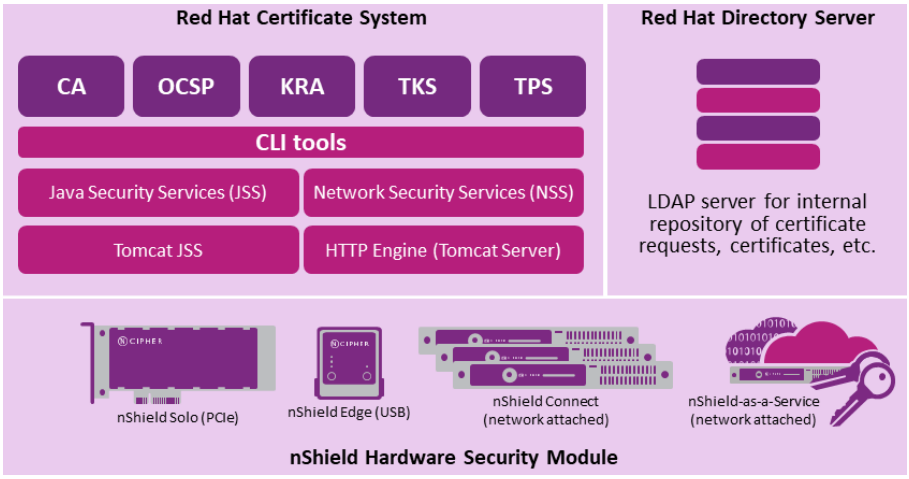
Table of Contents

1. Introduction . . . 1
    1.1. Requirements . . . 1
    1.2. Licensing . . . 2
    1.3. Product configurations . . . 2
    1.4. Supported nShield functionality . . . 3
    1.5. Policy requirements . . . 4
2. Procedures . . . 5
    2.1. Install the Operating System . . . 5
    2.2. Configure the HSM . . . 8
    2.3. Install RHDS . . . 11
    2.4. Install RHCS . . . 13
    2.5. Import the CA chain and user credentials into Firefox . . . 20
    2.6. Basic system tests . . . 21
3. Configure pkispawn . . . 23
    3.1. Modifying the sample pkispawn configuration file . . . 23
4. Additional resources and related products . . . 27
    4.1. nShield HSMs . . . 27
    4.2. nShield as a Service . . . 27
    4.3. Entrust products . . . 27
    4.4. nShield product documentation . . . 27

Chapter 1. Introduction

This guide explains how to configure a Red Hat Certificate System (RHCS) installation with an Entrust nShield Hardware Security Module (HSM). The integration between the HSM and Red Hat Certificate System uses the PKCS #11 cryptographic API.

The basic architecture of an RHCS deployment is shown in the diagram below:



This guide does not cover every step in the process of setting up all software. Some packages require that other packages are already configured, initialized, and running before they can be installed successfully.

1.1. Requirements

For an RHCS installation, you need to set up a Red Hat Enterprise Linux system. Conceptually, a CentOS platform will work in an identical manner, however the core RHCS packages may not be as up-to-date as those provided by Red Hat.

This guide does not cover the installation and configuration of the nShield Security World client software. For those instructions, see the *Installation Guide* for your HSM.

Requirements for the Red Hat Enterprise Linux server:

Component	Minimum Requirements	Recommended Requirements
Memory	2 GB	4 GB or more
Processor	1 CPU	1 CPU or more
Processor Cores	2	4 or more, AES-NI support

Component	Minimum Requirements	Recommended Requirements
Hard Disk	20 GB	80 GB or more
CD/DVD	Optional	Optional
Network Adapter	1	1
USB Controller	Only required for nShield Remote Administration	
Display	Standard configuration	

Components required for installation:

- Security World software v13.6.12.
- Red Hat Enterprise Linux v8.10.
- Linux firewall (`firewalld`).
- Static IP address.
- Mozilla Firefox.

Versions of Mozilla Firefox after v31.6.0 do not support client-side (web browser) initiated key generation functions.

- OpenJDK 64-bit.
- Red Hat Directory Server (RHDS) v11.

  Any LDAP-compliant database should be compatible with the integration.

- Red Hat Certificate System (RHCS) v10.

1.2. Licensing

There is no licensing that is imported into the product after installation. Contact Red Hat for appropriate licensing to purchase RHCS product support and RHN channel access.

1.3. Product configurations

RHCS v10.8 has been tested with the following nShield HSM configurations:

nShield HSM	Software	Firmware	Netimage	Security World	Ciphersuite
Connect XC	13.6.12	12.72.4 (FIPS 140-2 certified)	13.6.11	FIPS-140 Level 3	DLf3072s256mA EScSP800131Ar1
nShield 5C	13.6.12	13.4.5 (FIPS 140-3 certified)	13.6.12	FIPS-140 Level 3	DLf3072s256mA EScSP800131Ar1

The Common Criteria edition has been tested with the following settings:

nShield HSM	Software	Firmware	Netimage	Security World	Ciphersuite
Connect XC	13.6.12	12.60.15	13.3.2	Common Criteria	DLf3072s256mA EScSP800131Ar1
nShield 5C	13.6.12	13.5.1	13.6.12	Common Criteria	DLf3072s256mA EScSP800131Ar1

1.4. Supported nShield functionality

Red Hat Certificate System does not support module-protected keys. When you are enabling the use of an HSM, RHCS requires a token name, for which module protected keys have none. Using "accelerator" does not work.

Feature	Support	Feature	Support
Key Generation	Yes	Module-only keys	No
Key Management	Yes	FIPS 140 Level 3 mode support	Yes
1-of-N Operator Card Set	Yes	Common Criteria mode support	Yes
K-of-N Operator Card Set	Yes	Load Sharing	Yes
Softcards	Yes	Failover	Yes

1.5. Policy requirements

Entrust recommends that your organization operates its PKI using an approved organizational Certificate Policy, Certificate Practices Statement, and any other policy/procedure guidance necessary to govern the administration of the PKI and associated HSM(s). In particular, these documents should specify the following aspects of HSM administration:

- The number and quorum of Administrator Cards in the Administrator Card Set (ACS), and the policy for managing these cards.

  Common Criteria firmware requires a minimum quorum of two Administrator Cards.

- Whether application keys are to be protected by Operator Card Set (OCS) or Softcard protection mechanisms. Module key protection mechanism is not supported.
- The number and quorum of Operator Cards in the OCS (if OCS key protection is used), and the policy for managing these cards.
- Whether the security world should be compliant with FIPS 140 Level 3, FIPS 140 Level 2, or Common Criteria.

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

  See the *User Guide* for your HSM and the *nShield Security Manual* for more recommendations, or contact your local Entrust nShield Account Manager to arrange a technical discussion on architecture and best practices specific to your environment.

Compliance with US government security standards

It may be important for your organization to configure your RHEL OS in compliance with applicable standards such as FIPS.

- US Government Security Standards for Red Hat products, see https://access.redhat.com/articles/2918071.
- Information on enabling FIPS mode for RHEL v8, see https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/8/html/security_hardening/using-the-system-wide-cryptographic-policies_security-hardening

Chapter 2. Procedures

The instructions in this guide make reference to one server on which the Red Hat applications are installed, with two network interfaces and with unique IP addresses:

Application	Domain	IP Address
Red Hat Certificate System (RHCS)	pki.domain.com	10.0.0.2
Red Hat Directory Server (RHDS)	ldap.domain.com	10.0.0.3

In the instructions, use the domain names and IP addresses in your system.

2.1. Install the Operating System

1. Install Red Hat Enterprise Linux server on your target server platform or virtualized environment.

   No GUI is required for the installation of RHCS. However, to perform agent functions on the CA, you must use a system with a web browser (Mozilla Firefox is preferred).

2. If FIPS mode is required, ensure that the system is operating in FIPS mode.

   ```
   % sudo fips-mode-setup --enable
   ```

   Restart your system to allow the kernel to switch to FIPS mode:

   ```
   % sudo reboot
   ```

   After the restart, you can check the current state of FIPS mode:

   ```
   % fips-mode-setup --check

   FIPS mode is enabled.
   ```

3. Ensure that the system is operating in SELinux enforcing mode:

   ```
   % getenforce

   Enforcing
   ```

   If the result is Permissive or Disabled, you need to enable SELinux before continuing.

If the status was Permissive:

a. Make SELinux enforcing on reboot by editing `/etc/selinux/config`. Change `SELINUX=Permissive` to `SELINUX=enforcing`.

b. Set SELinux to enforcing for this session:

```
% setenforce 1
```

c. Check SELinux status again:

```
% getenforce
```

It should now show Enforcing.

If the status was Disabled:

a. Make SELinux enforcing on reboot by editing `/etc/selinux/config`. Change `SELINUX=Disabled` to `SELINUX=enforcing`.

b. Enable a `relabel` operation on reboot:

```
touch /.autorelabel
```

c. Reboot the server.

d. Check SELinux status again:

```
getenforce
```

It should now show Enforcing.

4. Ensure that the `firewalld` service is enabled and running:

```
systemctl enable firewalld
systemctl start firewalld
```

5. Configure the system time zone. With UTC:

```
timedatectl set-timezone UTC
timedatectl set-time "YYYY-MM-DD HH:MM:SS"
(HH:MM:SS is in 24-hour time)
```

If you are using NTP, see the Red Hat Enterprise Linux documentation.

6. Configure system hostname:

```
hostnamectl set-hostname <pki.domain.com>
```

Where *<pki.domain.com>* is your server FQDN.

7. If you are using the nShield Connect model, configure network interfaces as needed for external access to the RHCS subsystems and for the HSM. If you are installing the RHDS component on an external system, a network interface must be able to communicate with that system as well. See the Red Hat Enterprise Linux documentation for instructions on configuring the networking interfaces for your system, for example on using `nmcli` or other methods.

   Example:

   Modify `ens33` so that its IP address is 10.0.0.2:

   ```
   nmcli connection modify ens33 connection.autoconnect yes ipv4.method manual ipv6.method auto ipv4.addresses
   10.0.0.2/24 ipv4.gateway 10.0.0.1 ipv4.dns 10.0.0.1
   ```

   Add a second IP address, 10.0.0.3, to `ens33`:

   ```
   nmcli connection modify ens33 connection.autoconnect yes ipv4.method manual ipv6.method auto
   +ipv4.addresses 10.0.0.3/24 +ipv4.gateway 10.0.0.1 +ipv4.dns
   10.0.0.1
   nmcli connection up ens33
   ```

8. If you are not using DNS, configure `/etc/hosts` to include any aliases that might be used for the RHCS and RHDS subsystems.

   If DNS or `/etc/hosts` is not configured to resolve the applicable hostnames, services, especially RHDS, might hang for a long period of time when they are started.

   The following is an example configuration, assuming the CA and LDAP are on separate servers:

   ```
   127.0.0.1 localhost
   10.0.0.2 pki.domain.com pki
   10.0.0.3 ldap.domain.com ldap
   ```

9. Configure the yum repository to point to the Red Hat Network subscription channels for RHEL, RHCS, and RHDS.

   Optionally, point to a local yum repository if you have one configured. See the Red Hat Enterprise Linux documentation for instructions on configuring a yum repository.

10. Configure the `ulimits` for the OS by adding the following lines to the end of the

`/etc/security/limits.conf` file:

```
root soft nofile 65536
root hard nofile 65536
```

11. Configure any system security settings and lockdown procedures, such as screensaver settings, to ensure that the system is secure before generating the CA private key(s).

2.2. Configure the HSM

Install the HSM by following the instructions in the *Installation Guide* for the HSM.

Entrust recommends that you install the HSM before configuring the Security World software with your Apache HTTP Server.

2.2.1. Install the nShield Security World software

To install the Security World software and create the Security World:

1. Install the latest version of the Security World software as described in the *User Guide* for the HSM. Entrust recommends that you uninstall any existing Security World software before installing the new Security World software.

2.2.2. Enroll the new HSM client with the HSM

1. For a privileged client, such as the RFS, if enrolling as a client:

   ```
   nethsmenroll --privileged HSM_IP_ADDRESS
   ```

   For an unprivileged client, which includes most application servers such as a CA, OCSP responder, etc:

   ```
   nethsmenroll HSM_IP_ADDRESS
   ```

2. Verify enrollment with the enquiry command:

   ```
   enquiry -m 1
   ```

2.2.3. Configure cknfastrc for RHCS

1. Configure the `/opt/nfast/cknfastrc` file with the following settings:

Key Protection Mechanism	CKNFASTRC Configuration Parameters
Module protection	N/A: RHCS does not support this option

Softcard protection	`CKNFAST_OVERRIDE_SECURITY_ASSURANCES=wrapping_crypt`
	`CKNFAST_LOADSHARING=1`
	`CKNFAST_NO_ACCELERATOR_SLOTS=1`

OCS protection
(K-of-N with k=1)
	`CKNFAST_OVERRIDE_SECURITY_ASSURANCES=wrapping_crypt`
	`CKNFAST_LOADSHARING=1`
	`CKNFAST_NO_ACCELERATOR_SLOTS=1`

OCS protection
(K-of-N with k > 1)
	`CKNFAST_OVERRIDE_SECURITY_ASSURANCES=wrapping_crypt`
	`CKNFAST_LOADSHARING=1`
	`CKNFAST_NO_ACCELERATOR_SLOTS=1`
	`NFAST_NFKM_TOKENSFILE=/opt/nfast/nfast-nfkm-tokensfile`

2. To troubleshoot CA installation:
   - RHCS logs are in `/var/log/pki/<instance>`.
   - nShield hardserver logs are in `/opt/nfast/log/hardserver.log`.

2.2.4. Create or load a Security World

1. Initialize a Security World as described in the *User Guide* for the HSM.

   The decision to create a new Security World or to load an existing Security World is based on your organization's security practices.

2.2.5. Create an OCS or Softcard to protect the application keys

If you already have an OCS or a Softcard to use from an existing Security World:

1. Make sure the appropriate card(s) or the appropriate Softcard file(s) are in

`/opt/nfast/kmdata/local`.

2. Continue with the instructions in Install RHDS.

2.2.5.1. Method #1: Create an OCS

This `createocs` command is an example. Use parameters that satisfy your organization's security requirements.

1. Use the following parameters to create a new OCS:
   - K-of-N defined as 1/3.
   - No time-out.
   - Persistent card set.
   - Do not name cards individually.
   - Enable PIN recovery.
   - Pick a name for your OCS. The example below uses OCS1.

```
createocs -m 1 -N "OCS1" -Q 1/3 -T 0 --persist --pp-recovery
```

2. If you are using a FIPS 140 Level 3 Security World, you need to present an ACS or OCS card for FIPS authorization.
3. Enter in a password for each OCS card in the quorum. The passwords can be unique per card, but it is not recommended for most use cases.
4. Verify that the OCS card set is visible:

```
nfkminfo -c
```

2.2.5.2. Method #2: Create a Softcard

This `ppmk` command is an example. Use parameters that satisfy your organization's security requirements.

1. Create a new Softcard:
   - If you are using FIPS 140 Level 3 Security World, you need to present an ACS or OCS to provide FIPS authentication.
2. Pick a name for your Softcard. The example below uses `SOFTCARD1`.
3. Make the Softcard container recoverable.

```
ppmk --new --recoverable SOFTCARD1
```

4. Verify that the Softcard was created:

```
ppmk --list
```

2.2.5.3. Copy KMDATA files to other HSM client systems

The Security World files are located `/opt/nfast/kmdata/local` folder where they were generated.

To use the Security World on other HSM clients, copy the appropriate KMDATA files (`world`, `cards_*`, `card_*`, `softcard_*`, `module_*`) from `/opt/nfast/kmdata/local` to your other HSM clients.

This can be performed manually, using tools such as `rsync` or `scp`, or by configuring the `rfs-sync` capability using nShield tools.

For information on configuring and using `rfs-sync`, see *the User Guide* for your HSM.

2.3. Install RHDS

You need the Red Hat Directory Server packages from the appropriate Red Hat Network channel. Some packages are in the RHEL OS channel, and some are in the RHDS channel. For instructions to install these packages, see the Red Hat Directory Server documentation.

In these steps, the instance name ca-1 is used. Entrust recommends that you use a more descriptive identifier to suit the requirements of your organization. It is most convenient to name the RHDS instance the same as the RHCS instance, especially if you plan to create multiple instances of either on their respective servers.

For instructions to enable LDAPS on the directory server before installing RHCS, see the Red Hat Directory Server documentation. There are generally two methods for this:

- Create a TLS key/request and have the certificate signed by an already deployed CA, if one exists.
- Create a TLS key/request and self-sign it for temporary use in the LDAP server until the new RHCS CA is deployed, then re-sign the TLS certificate request on the new RHCS CA, remove the self-signed certificate from the LDAP server's NSS database, and install the new TLS certificate along with the new CA chain.

In the steps below, the RHDS instance is installed on the same system as RHCS. They can be installed on separate servers.

1. Open ports to the appropriate firewall zone:

```
firewall-cmd --permanent --add-port={389/tcp,636/tcp} --zone=<zone>
firewall-cmd --reload
```

2. Install the RHDS packages and dependencies:

```
yum install redhat-ds
```

3. Configure the LDAP service account user:

```
groupadd -r ldap
```

4. Configure the LDAP service account group:

```
useradd -g ldap ldap-ca-1
```

5. Configure RHDS:

   Directory Server v11:

```
dscreate interactive
```

The interactive setup is limited in scope. You can create an INF file with the options that you want. Additionally, the interactive installation options change with different major versions. See Red Hat Directory Server v11 documentation for an example installation on the newer versions.

   a. For **Would you like to continue with set up? [yes]:**, press **Enter**.
   b. When you are prompted about the number of file descriptors, enter **yes** and press **Enter**.

      See the Red Hat Enterprise Linux documentation about increasing the number of file descriptors, if necessary

   c. For **Choose a setup type [2]:**, accept the default, and press **Enter**.
   d. For **Computer name [pki.domain.com]:**, enter the FQDN for your LDAP server, for example *ldap.domain.com*, and press **Enter**.
   e. For **System User [dirsrv]:**, enter LDAP service account user that you created, **ldapca-1**, and press **Enter**.

f. For **System Group [dirsrv]:**, enter the LDAP service account group that you created, **ldap**, and press **Enter**.

g. For **Directory server network port [389]:**, accept the default, and press **Enter**.

h. For **Directory server identifier [???]:**, enter **ca-1**, then press **Enter**.

i. For **Suffix [dc=domain, dc=name]:** enter **o=ca-1-CA**, and press **Enter**. This must match the `pki_ds_base_dn` variable in `pkispawn.cfg`, see Pre-configure the RHCS instance.

j. For **Directory Manager DN [cn=Directory Manager]:**, accept the default, and press **Enter**.

k. Enter and confirm the password for the `cn=Directory Manager` account.

The LDAP instance is created.

You can control the RHDS instance with the following commands.

+

Function	Command
Enable on boot	`systemctl enable dirsrv@<RHCS-instance>`
Start manually	`systemctl start dirsrv@<RHCS-instance>`
Stop manually	`systemctl status dirsrv@<RHCS-instance>`
Restart manually	`systemctl restart dirsrv@<RHCS-instance>`

6. Test the RHDS connection:

```
ldapsearch -o ldif-wrap=no -x -LLL -h ds-ldap.domain.com -p 389 -D 'cn=Directory Manager' -W -b "cn=config"
-s base '(objectclass=*)' nsslapd-versionstring nsslapddefaultnamingcontext

Enter LDAP Password:
dn: cn=config
nsslapd-versionstring: 389-Directory/1.4.3.39
```

2.4. Install RHCS

You need the Red Hat Certificate System packages from the appropriate Red Hat Network channel. The CA and KRA subsystems are part of the RHEL OS channel, but the TKS and TPS subsystems require access to the Red Hat Certificate System channel. There are many dependency packages that need to be installed along with the RHCS packages, including `apache/httpd` and `apache/tomcat`. For package installation instructions, see the Red Hat Certificate System documentation.

The various subsystems for Red Hat Certificate System are installed and configured individually. The initial installation is performed using package management tools such as `yum`. Subsystem setup is accomplished using the command-line tool `pkispawn`.

2.4.1. Install the packages

The `redhat-pki` package installs packages to support all RHCS subsystems. This document assumes the use of the `pki-ca` subsystem.

To install just the CA subsystem:

```
% yum install pki-ca redhat-pki-server-theme
```

Restore SELinux context on `/opt/nfast` to account for the new `pki-selinux` policy

```
% restorecon -FRvv /opt/nfast
```

Ensure that your RHEL OS packages are updated, in particular `httpd`, `tomcat`, and `nss`. RHCS packages do not always have the correct dependency versions, and sometimes key generation might fail because `nss` needs to be updated to the latest version.

Restart the nfast hardserver.

```
% /opt/nfast/sbin/init.d-ncipher restart
```

2.4.2. Check the PKCS #11 connection to the HSM

2.4.2.1. Check with Mozilla NSS database tools

1. Create the temporary NSS database:

```
mkdir -p /opt/tempnssdb
cd /opt/tempnssdb
modutil -dbdir . -create -force
```

2. Add the nShield PKCS #11 library to the temporary NSS database:

```
modutil -dbdir . -add nfast -libfile /opt/nfast/toolkits/pkcs11/libcknfast.so -force
Module "nfast" added to database.
```

3. Once the NSS database is created and linked to the HSM, check PKCS#11 token info using RHCS tools:

```
% TokenInfo .

Database Path: .
Found external module 'NSS Internal PKCS #11 Module'
Found external module 'nfast'
Found external module 'p11-kit-proxy'
Found external token 'testSC'
```

4. Remove the temporary NSS database:

```
rm -rf /opt/tempnssdb
```

2.4.2.2. Check using HSM tools

Use the `ckcheckinst` command to test the PKCS #11 installation with nShield tools. This example uses an OCS, modify it if you are using Softcard protection.

In some cases, `ckcheckinst` might fail. However, this does not necessarily indicate a system configuration problem. Consult Entrust nShield Support if necessary.

```
% /opt/nfast/bin/ckcheckinst

PKCS#11 library interface version 2.40
                           flags 0
                  manufacturerID "nCipher Corp. Ltd               "
               libraryDescription "nCipher PKCS#11 13.6.12-317-3003"
          implementation version 13.06
        Loadsharing and Failover enabled

Slot  Status            Label
====  ======            =====
   0  No token present
   1  No token present
   2  No token present
   3  Operator card     "testOCS                        "
   4  Soft token        "testSC                         "

Select slot number to run library test or 'R'etry or to 'E'xit: 4
Using slot number 4.

Please enter the passphrase for this token (No echo set).
Passphrase:

Test                     Pass/Failed
----                     -----------

1 Generate RSA key pair   Pass
2 Generate DSA key pair   Pass
3 Encryption/Decryption   Pass
4 Signing/Verification    Pass
```

```
Deleting test keys        ok

PKCS#11 library test successful.
```

2.4.3. Configure the firewall

1. Open ports to support RHCS functions. These ports are the default for RHCS. If you plan to use non-default ports, add those ports to the firewall instead of the ones listed in this table.

Service	Port
CA HTTP Proxy	8080/tcp
CA HTTPS Proxy	8443/tcp
CA Security Domain	8443/tcp
CA Tomcat Server	8005/tcp
CA AJP	8009/tcp

2. Configure the firewall:

```
firewall-cmd --permanent --add-port={8080/tcp,8443/tcp,8005/tcp,8009/tcp}
```

2.4.4. Configure service account users and groups

Create the appropriate user and group accounts before staging the system.

Members of the `pkiadmin` system group have full access to tasks in the agent service interface:

```
groupadd -r pkiadmin
```

Members of the `pkiaudit` system group can read the signed audit logs.

```
groupadd -r pkiaudit
```

To create a new service account user, and assign the account to the `pkiadmin` group:

```
useradd -g pkiuser -G nfast,pkiadmin,pkiaudit -d /usr/share/pki -s /sbin/nologin -c "RHCS ca-1" -r pkiuser-ca-1
```

2.4.5. Pre-configure the RHCS instance

Run `pkispawn` for an initial creation phase so you can pre-configure parameters, such as certificate distinguished names and validity periods. For information on the two-step installation, see the *RHCS Installation Guide*.

See Configure pkispawn.

Do not modify `/etc/pki/default.cfg` directly.

1. Create a copy of `/etc/pki/default.cfg`, for example copy it to `/opt/pkispawn.cfg`, and modify it to match your system.

   Information on how to modify the default configuration file:

   - Configure pkispawn.
   - `man 5 pki_default.cfg`.
   - *Red Hat Certificate System Planning, Installation, and Deployment Guide*.

If you are using a FIPS Level 3 world file, have an OCS card connected to provide FIPS Authorization.

1. Generate an INF for subsystem setup.

   If you are using OCS-protected keys and K>1, use the preload command with `pkispawn`:

   ```
   preload -m <module number> -f "<preload FilePath>" --cardset-name=<OCS Cardset-Name> pkispawn -f
   /path/to/pkispawn.cfg -s CA --skip-configuration
   ```

   Insert the OCS cards and, if prompted, enter the OCS passphrase.

   For other OCS scenarios and other protection methods, run `pkispawn` on its own:

   ```
   pkispawn -f /path/to/pkispawn.cfg -s CA --skip-configuration
   ```

2. At this point, you can modify any of the pre-configuration files as necessary. See the information on two-step installation in the *RHCS Installation Guide*.
3. The output of the `pkispawn` commands above should be something like this:

   ```
   Loading deployment configuration from /opt/pkispawn.cfg.
   Installation log: /var/log/pki/pki-ca-spawn.20251007210906.log
   Installing CA into /var/lib/pki/ca-1.

       ======================================================================
                                INSTALLATION SUMMARY
       ======================================================================
   ```

```
      The CA subsystem of the 'ca-1' instance
      must still be configured!

      To check the status of the subsystem:
            systemctl status pki-tomcatd@ca-1.service

      To restart the subsystem:
            systemctl restart pki-tomcatd@ca-1.service

      The URL for the subsystem is:
            https://pki.interop.local:8443/ca

      PKI instances will be enabled upon system boot

    ==========================================================================
```

2.4.6. Configure the CA instance

Determine whether you are creating a root CA or a subordinate (issuing) CA, then define an appropriate configuration file for `pkispawn` using either:

2.4.6.1. Create a root CA

Set `pki_external=False` for the `pkispawn.cfg` file so that `pkispawn` self-signs the `caSigningCert` object.

If you are using OCS-protected keys and K>1, use the preload command with `pkispawn`:

```
preload -m <module number> -f "<preload FilePath>" --cardset-name=<OCS Cardset-Name> pkispawn -s CA -vv -f
/path/to/pkispawn.cfg --skip-installation
```

Insert the OCS cards and, if prompted, enter the OCS passphrase.

For other OCS scenarios and other protection methods, run the single-phase `pkispawn` on its own:

```
pkispawn -s CA -vv -f /path/to/pkispawn.cfg --skip-installation
```

Output should be similar to this:

```
...
...
    ==========================================================================
                              INSTALLATION SUMMARY
    ==========================================================================

      Administrator's username:             caadmin
      Administrator's PKCS #12 file:
            /root/.dogtag/ca-1/ca_admin_cert.p12

      This CA subsystem of the 'ca-1' instance
```

```
    has FIPS mode enabled on this operating system.

    REMINDER:  Don't forget to update the appropriate FIPS
               algorithms in server.xml in the 'ca-1' instance.

    To check the status of the subsystem:
          systemctl status pki-tomcatd@ca-1.service

    To restart the subsystem:
          systemctl restart pki-tomcatd@ca-1.service

    The URL for the subsystem is:
          https://pki.interop.local:8443/ca

    PKI instances will be enabled upon system boot

  ================================================================================
```

2.4.6.2. Create a subordinate or issuing CA

Set `pki_external=True` for the `pkispawn.cfg` file so that `pkispawn` does not self-sign the `caSigningCert object`, but rather creates a PKCS#10 certificate request file to be signed by an external root CA.

1. Set the `pki_external_step_two` parameter to false in the `pkispawn.cfg` file.
2. Run `pkispawn` phase 1:

   ```
   pkispawn -s CA -vv -f /path/to/pkispawn.cfg --skip-installation
   ```

3. Sign the certificate request generated at the end of phase 1, and put it in `/etc/pki/<instance>/alias/<instance>_caSigningCert.cer`.
4. Put the CA chain PKCS #7 in `/etc/pki/<instance>/alias/caChain.p7c`.

   This file should not include the new subordinate CA's certificate, just the signing CA hierarchy.

5. Change the `pki_external_step_two` parameter to true in `pkispawn.cfg`.
6. Run `pkispawn` phase 2.

   If you are using OCS-protected keys and K>1, use the preload command with `pkispawn`:

   ```
   preload -m <module number> -f "<preload FilePath>" --cardset-name=<OCS Cardset-Name> pkispawn -s CA -vv -f
   /path/to/pkispawn.cfg
   ```

   Insert the OCS cards and, if prompted, enter the OCS passphrase.

   For other OCS scenarios and other protection methods, run `pkispawn` on its own:

```
pkispawn -s CA -vv -f /path/to/pkispawn.cfg
```

2.4.6.3. Configure the CA and CRL profiles

See Configure pkispawn.

Restart the CA service after modifying profiles or modifying `/etc/pki/<instance>/ca/CS.cfg`.

2.4.6.4. RHCS instance control commands

If you are using OCS-protected keys and K>1

1. Open another terminal.
2. Run the following command:

```
preload -m <module number> -f "<preload FilePath>" --cardset-name=<OCS Cardset-Name> pause
```

3. Insert the OCS cards and, if prompted, enter the OCS passphrase.
4. Leave the preload command paused until you finished to run the control commands, for example, systemctl start or systemctl restart, that you intend to run. This may take a minute or two. It is best to check the CA web page to ensure the system is up and running before you cancel the preload command.

Function	Command
Enable on boot	Do not use. The service would fail because it needs the OCS passphrase to unlock the CA private signing key.
Start manually	`systemctl start pki-tomcatd@<instance>`
Stop manually	`systemctl stop pki-tomcatd@<instance>`

Check status	`systemctl status pki-tomcatd@<instance>`
	
	`pkidaemon status <instance>`

2.5. Import the CA chain and user credentials into Firefox

Launch Firefox.

If this is a sub CA, manually import external certificates in the CA chain:

1. Navigate to the drop-down menu in the upper-right corner (three horizontal bars).
2. Navigate to **Preferences >Privacy and Security**.
3. Scroll down to **Certificates**, and select **View Certificates**.
4. On the **Authorities** tab, select **Import to import CA chain certificates into the store**.

   Repeat for each certificate in the external trust chain.

5. Configure certificate trust:
   a. If prompted during import, check all the options to trust the certificate for various activities.
   b. On the **Authorities** tab, find your CA certificate in the list organized by the organization name specified during installation. Select **Edit Trust**, select all of the options to trust the root certificate, and select **OK**.
   c. Repeat for each certificate that was imported.
6. Select **OK**.

To import the CA's own chain, for both root and sub CA types:

1. Navigate to the **Retrieval** tab on the end-entity page.
2. Select **Import CA Certificate Chain** link.
3. Select **Import the CA certificate chain into your browser**, and select **Submit**.
4. Select all of the trust options, and select **OK**.

To import the default agent's credential for both root and sub CA types:

1. Navigate to the drop-down menu in the upper-right corner (three horizontal bars).
2. Navigate to **Preferences > Privacy and Security**.
3. Scroll down to **Certificates**, and select **View Certificates**.
4. On the **Your Certificates** tab, select **Personal**, then select **Import**.
5. Navigate to the folder in which your default agent credential was created, for example `/etc/pki/<instance>/agent_alias`, and select the admin PKCS #12 file that is located there.
6. Enter the PKCS #12 passphrase defined in `pkispawn` configuration and select **OK**.
7. Select **OK** to close View Certificates.

2.6. Basic system tests

2.6.1. Verify CA keys

The application URLs can be found as follows, using *pki.domain.com* as an example:

Web Site	URL
Unsecure URL	http://<pki.domain.com>:8080/ca/ee/ca
Secure Agent URL	https://<pki.domain.com>:8443/ca/agent/ca
Secure EE URL	https://<pki.domain.com>:8443/ca/ee/ca
Secure Admin URL	https://<pki.domain.com>:8443/ca/services

2.6.2. Test the CA functionality

1. Open a web browser from the CA. Do not use the browser icon.

   ```
   pkidaemon status <instance>
   ```

2. Ctrl-click on **Secure Agent URL**.
3. Choose the appropriate agent certificate, then select **OK**.
4. Enter the Firefox security database password if it is configured.
5. Navigate to **Update Revocation List**.
6. Ensure that **Issuing Point** is set to **MasterCRL**.
7. Select the appropriate **Signature Algorithm**. The default is likely to change when you configure `/etc/pki/<instance>/ca/CS.cfg`, but for testing pick whatever you would pick for the final configuration.
8. Create a new CRL:
   a. Select **Update**.
   b. When the **Certification Revocation List Update** has been scheduled, check the CS logs to see results.
9. Select the link on the left for **Display Revocation List**.
10. Ensure that **Issuing Point** is set to **MasterCRL**.
11. Ensure that **Display Type** is set to **Entire CRL**.
12. Select **Display** to view the new CRL.
13. To verify the CRL, verify that its timestamp is from within the last minute.

Chapter 3. Configure pkispawn

3.1. Modifying the sample pkispawn configuration file

3.1.1. Hostnames

Set them as appropriate in your system. If you run multiple instances on the same server, it is not recommended to use your system FQDN as the RHCS server.

- RHCS server: `pki_hostname=pki.domain.com`
- RHDS server: `pki_ds_hostname=ldap.domain.com`

3.1.2. Differences for a root CA and subordinate CA

For a root CA, the `caSigningCert` is self-signed in `pkispawn` phase 1.

- `pki_external=False`

For a subordinate CA, phase 1 completes basic setup, and creates the `caSigningCert` key and certificate request. Phase 2 imports the signed CA certificate chain and finishes the setup.

- `pki_external=True`
- Phase 1: `pki_external_step_two=False`
- Phase 2: `pki_external_step_two=True`

The `pki_subordinate*` parameters are used if you want your root CA to be part of the same RHCS security domain. This is an unlikely scenario because the root will be offline. Do not change these parameters.

3.1.3. Ports

The defaults work for a single instance on the host. If you are using multiple hosts, it is recommended to use high ports, for example in the 63000 range.

- `pki_security_domain_https_port=8443`
- `pki_http_port=8080`
- `pki_https_port=8443`
- `pki_ajp_port=8009`

- `pki_tomcat_server_port=8005`

3.1.4. Certificate Distinguished Names

The six certificate DNs, especially the CA's own certificate, are important to an enterprise-class PKI.

Change `*_subject_dn=` to your DN, based on your policy.

- `pki_ca_signing_subject_dn=cn=<CA Common Name>,OU=Group,OU=Division,O=Company,C=US`
- `pki_sslserver_subject_dn=cn=<server FQDN>,OU=Group,OU=Division,O=Company,C=US`
- `pki_subsystem_subject_dn=cn=<CA Common Name> Subsystem Certificate,OU=Group,OU=Division,O=Company,C=US`
- `pki_admin_subject_dn=cn=<CA Common Name> Agent Certificate,OU=Group,OU=Division,O=Company,C=US`
- `pki_audit_signing_subject_dn=cn=<CA Common Name> Audit Certificate,OU=Group,OU=Division,O=Company,C=US`
- `pki_ocsp_signing_subject_dn=cn=<CA Common Name> OCSP Certificate,OU=Group,OU=Division,O=Company,C=US`

3.1.5. Algorithms and key size

Six key pairs are created during installation. For policy reasons, keys should match.

For all keys, change all `*_key_algorithm=`, `*_key_size=`, and `*_key_type=` parameters to match your key configuration.

- CA signing key (`pki_ca_signing_key_*`)
- Instance subsystem key (`pki_ subsystem_key_*`)
- SSL/TLS web server key (`pki_sslserver_key_*`)
- Internal OCSP (`pki_ocsp_signing_key_*`)
- Instance audit signing key (`pki_audit_signing_key_*`)
- Default administrator key (`pki_admin_key_*`)

Example for the CA's signing key, using RSA2048 and SHA-256:

- `pki_ca_signing_key_algorithm=SHA256withRSA`
- `pki_ca_signing_key_size=2048`
- `pki_ca_signing_key_type=rsa`

Example for the CA's signing key using ECC (nistp256) and SHA-256:

- `pki_ca_signing_key_algorithm=SHA256withEC`
- `pki_ca_signing_key_size=nistp256`
- `pki_ca_signing_key_type=ecc`

3.1.6. nShield HSM

Change all `*_token=` variables to match the name of your OCS or Softcard token.

- `pki_audit_signing_token=OCS1`
- `pki_sslserver_token=OCS1`
- `pki_subsystem_token=OCS1`
- `pki_token_name=OCS1`
- `pki_ca_signing_token=OCS1`
- `pki_ocsp_signing_token=OCS1`

3.1.7. Account passwords

Change all applicable `*_password=` variables from the defaults.

[configure-pkispawn:::pkispawn-config-example] has a default value of `password` for all the passwords.

`pki_replication_password` is only for cloning CAs.

- For the RHCS security domain, to join additional subsystems to the CA's security domain (`pki_security_domain_password`)
- For the RHCS instance's NSS database (`pki_server_database_password`)
- For the LDAP directory server (`pki_ds_password`)
- For the HSM token (`pki_token_password`)

Default admin user passwords. They should match:

- Administrator credential, for example for logging in to pkiconsole (`pki_admin_password`)
- Default administrator credential in an NSS database (`pki_client_database_password`)
- Default administrator credential in a PKCS #12 file (`pki_client_pkcs12_password`)

3.1.8. Default agent credential

Load this PKCS #12 file into Firefox, or another web browser on any system, to be able to access the agent web page and issue certificates or CRLs.

After the installation completes, this file is located in `/etc/pki/<instance>/agent_alias`.

<instance> is the `pki_instance_name variable` in the `pkispawn` configuration file.

Chapter 4. Additional resources and related products

4.1. nShield HSMs

4.2. nShield as a Service

4.3. Entrust products

4.4. nShield product documentation