Red Hat Certificate System

nShield® HSM Integration Guide

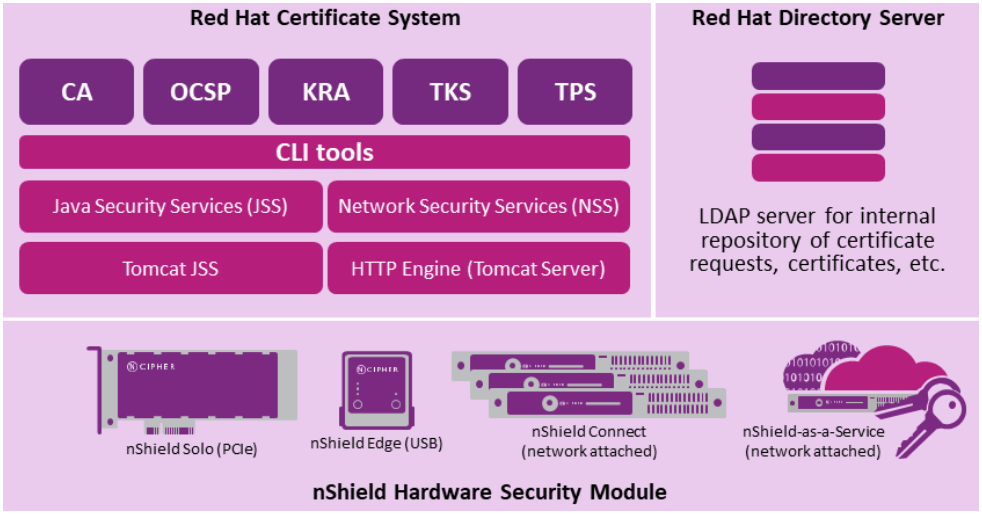
11 Apr 2023

Contents

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 v12.40.2, v12.60.11, v12.80.4, or v13.2.2.
- Red Hat Enterprise Linux v7.8 or later.
- 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) v10 or later.

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

- Red Hat Certificate System (RHCS) v9 or later.

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 v9.6 has been field tested with the following nShield HSM configurations:

Software	Firmware	Netimage	Security World	Ciphersuite
12.40.2	3.4.2 vsn37 (FIPS)	12.45.2 vsn30	FIPS-140-2 Level 3	DLf3072s256mRijndael

Software	Firmware	Netimage	Security World	Ciphersuite
12.60.11 [1]	3.4.2 vsn37 (FIPS)	12.60.10 vsn31	FIPS-140-2 Level 3	DLf3072s256mRijndael
12.60.11	12.50.11 (FIPS Certified)	12.60.10 vsn31	FIPS-140-2 Level 3 [2]	DLf3072s256mAEScSP800131Ar1

RHCS v10.2 has been field tested with the following nShield HSM configurations:

Software	Firmware	Netimage	Security World	Ciphersuite
12.80.4	12.50.11 (FIPS Certified)	12.60.10 vsn30	FIPS-140-2 Level 3 [2]	DLf3072s256mAEScSP800131Ar1
12.80.4	12.72.1 (FIPS Certified)	12.80.5	FIPS-140-2 Level 3 [2]	DLf3072s256mAEScSP800131Ar1

RHCS v10.4 has been field tested with the following nShield 5c HSM configurations:

Software	Firmware	Netimage	Security World	Ciphersuite
13.2.2	13.2.2 (FIPS Pending)	13.2.2	FIPS-140-2 Level 3 [2] [3]	DLf3072s256mAEScSP800131Ar1

[1] The nShield 12.40 Compatibility Package (version 1.0.0) is required. To obtain the package, contact Entrust nShield Support, https://nshieldsupport.entrust.com.

[2] This RHCS configuration only works with the CA and OCSP subsystem components. KRA, TKS, or TPS components are not supported with this configuration. If KRA, TKS, or TPS components are required, use either an unrestricted world instead of FIPS 140-2 Level 3, or use the configurations with 3.4.2 firmware.

[3] Hotfix TAC_955 is required for this configuration. An unrestricted world may be used without the need for a hotfix.

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	Strict FIPS mode support	Yes
1-of-N Operator Card Set	Yes	Common Criteria mode support	N/A
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.
- Whether application keys are to be protected by Operator Card Set (OCS), Softcard, or module key protection mechanisms.
- 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 140-2 Level 3, Common Criteria, or unrestricted mode.

  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 v7, see https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/security_guide/chap-federal_standards_and_regulations#sect-Federal_Information_Processing_Standard.
- Information on enabling FIPS mode for RHEL v8, see https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/security_hardening/using-the-system-wide-cryptographic-policies_security-hardening.

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.
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

This section is a general HSM setup procedure to support RHCS. For detailed instructions on configuration, see the *Installation Guide* and the *User Guide* for your HSM.

2.2.1. Install the nShield Security World software

1. Mount the Security World ISO:

```
mkdir -p /mnt/iso
mount /path/to/<filename>.iso /mnt/iso
```

2. Open a terminal window, and change to the root directory.

```
cd /
```

3. Extract the required .tar.gz files to install all the software bundles.

   For v12.50 and earlier firmware:

```
tar xf /mnt/iso/linux/libc6_11/amd64/nfast/hwsp/agg.tar
tar xf /mnt/iso/linux/libc6_11/amd64/nfast/ctls/agg.tar
tar xf /mnt/iso/linux/libc6_11/amd64/nfast/javasp/agg.tar
tar xf /mnt/iso/linux/libc6_11/amd64/nfast/jcecsp/user.tar
tar xf /mnt/iso/linux/libc6_11/amd64/nfast/pkcs11/user.tar
tar xf /mnt/iso/linux/libc6_11/amd64/nfast/hwcrhk/user.tar
```

   For v12.60 and later firmware:

```
tar zxf /mnt/iso/linux/amd64/ctls.tar.gz
tar zxf /mnt/iso/linux/amd64/hwsp.tar.gz
tar zxf /mnt/iso/linux/amd64/javasp.tar.gz
tar zxf /mnt/iso/linux/amd64/redist.tar.gz
```

4. Create the `nfast` user and group:

```
groupadd -r nfast
useradd -r -g nfast -d /opt/nfast -s /bin/bash nfast
```

5. Install the nShield software components:

```
/opt/nfast/sbin/install
```

6. Stop services:

```
systemctl stop nc_hardserver
```

7. Configure the `nfast` user or group to read everything:

```
chown -R nfast:nfast /opt/nfast
```

8. Add the `nfast/bin` folder to the system path:

```
echo "export PATH=\$PATH:/opt/nfast/bin" > /etc/profile.d/nfast_path.sh
chown root:root /etc/profile.d/nfast_path.sh
chmod 0644 /etc/profile.d/nfast_path.sh
```

9. Start and enable all Security World system processes:

```
systemctl start nc_hardserver
systemctl enable nc_hardserver
```

10. Configure the firewall:

```
firewall-cmd --permanent --add-port=9004/tcp
firewall-cmd --reload
firewall-cmd --list-all
```

2.2.2. Connect only: Configure basic network settings for the HSM

If you are using an nShield Solo or nShield Edge, continue with the instructions in Method #3, Solo or Edge: Using the fet command.

1. If this is a new HSM, perform a factory reset:

   System > Factory state

2. Enable IPv4 for interface #1 (repeat for interface #2 if necessary):

   System > System configuration > Network config > Set up interface #1 > configure #1 IPv4 > IPv4 Enable/Disable > Enable > Finish

3. Configure the IPv4 static address for interface #1 (repeat for interface #2 if necessary):

   System > System configuration > Network config > Set up interface #1 > configure #1 IPv4 > Static IPv4 address > HSM_IP_ADDRESS > HSM_SUBNET_MASK > Finish

4. Configure the default gateway (if one exists):

   System > System configuration > Network config > Set Default gateway > IPv4 Gateway > HSM_GATEWAY_IP_ADDRESS > Next > Finish

2.2.3. Connect only: Configure the RFS

1. Ensure that the RFS can ping the HSM. Run the following command on the RFS:

```
ping <hsm-ip-address>
```

2. Open a command prompt as administrator, obtain the ESN and KNETI hash of the HSM:

```
anonkneti <hsm-ip-address> EEEE-SSSS-NNNN abcdef0123456789abcdef0123456789abcdef01
```

The command returns the `HSM_ESN` (`EEEE-SSSS-NNNN`) and `HSM_KNETI_HASH` (`abcdef0123456789abcdef0123456789abcdef01`) that will be used in the next step. These should be compared the front panel of the HSM (scroll down the front panel for status).

3. Use the output of `anonkneti` to set up the RFS for this HSM:

```
rfs-setup --force HSM_IP_ADDRESS HSM_ESN HSM_KNETI_HASH
```

4. Configure the RFS IP address on the HSM:

   System > System configuration > Remote File System > Define IPv4 RFS > Set IP address of RFS> RFS_IP_ADDRESS > Port 9004 (default) > Finish

2.2.4. Connect only: Configure the keyboard layout

Select the keyboard layout as appropriate. The default is UK.

System > System configuration >Keyboard layout > <region-option> >Finish

2.2.5. Connect only: Create a new HSM client connection

Most HSM clients do not need to be, and should not be, privileged. It is simpler to create a Security World, a Softcard or an OCS, and the instructions in this section use a privileged client. It is recommended to change the client to unprivileged after installation.

1. Navigate to the client configuration screen:

   System > System configuration > Client Config

2. Select **Unprivileged for normal use**, or **Priv. on any port for a privileged client connection.**
3. For v12.50 and earlier nShield Connect netimage:

   New IPv4 Client > Enter your IP address (CLIENT_IP_ADDRESS) > Unprivileged > NO (nToken) > Finish

4. For v12.60 and later nShield Connect netimage:
   a. Obtain the KNETI hash of your v12.60 or later Security World client:

```
anonkneti -m 0 127.0.0.1
```

b. **New IPv4 Client > Enter your IP address (CLIENT_IP_ADDRESS) > Unprivileged > YES (secure authentication) >Confirm KNETI hash of HSM client from previous step > Finish**

5. Repeat steps 2-4 for each client.

2.2.6. Connect only: Enroll the new HSM client with the HSM

1. For a privileged client (e.g. the RFS, if enrolling as a client):

```
nethsmenroll --privileged HSM_IP_ADDRESS
```

For an unprivileged client (e.g. most application servers such as a CA, OCSP responder, etc):

```
nethsmenroll HSM_IP_ADDRESS
```

2. Verify enrollment with the enquiry command:

```
enquiry -m 1
```

2.2.7. Load feature licenses

Usually customers only obtain a FEM card with their HSM purchase. Additional licenses purchased thereafter are provided by Entrust nShield Support as a text file.

2.2.7.1. Method #1, Connect only: Using a FEM card

To load all of the features at once using the FEM card:

HSM > HSM feature enable > Read FEM from card

2.2.7.2. Method #2, Connect only: Using text files from Entrust nShield support

1. Copy your feature files to the following folder on the RFS:

```
/opt/nfast/kmdata/hsm-EEEE-SSSS-NNNN/features/
```

2. Update ownership/permissions:

```
cd /opt/nfast/kmdata/hsm-EEEE-SSSS-NNNN/features/
chown -R nfast:nfast *
chmod 0660 *
```

3. Attach a USB keyboard to the front of the HSM. It is helpful to rename the feature files stored on the RFS to a shorter name (such as `ecc.txt`) so they are easier to type.
4. Navigate to the feature enablement section:

   HSM > HSM feature enable

5. Repeat the following step for each feature file to load on the HSM:

   Read FEM from a file > type in the full filename of feature file > Confirm

2.2.7.3. Method #3, Solo or Edge: Using the fet command

1. Run the `fet` command.
2. Follow the prompts to load licenses.

2.2.7.4. All HSM models: Restart the hardserver process on the HSM client(s)

1. If you are loading additional client licenses into the HSM, restart your hardserver process on the HSM client. You only need to do this on new systems that need to be enrolled, already enrolled systems are not affected:

   ```
   systemctl restart nc_hardserver
   ```

2. Verify that the additional licenses are available:

   ```
   enquiry
   ```

   Check the max exported modules line, which should reflect your new client license count.

2.2.8. 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=none`
	`CKNFAST_LOADSHARING=1`
	`CKNFAST_NO_ACCELERATOR_SLOTS=1`

OCS protection
(K-of-N with k=1)
	`CKNFAST_OVERRIDE_SECURITY_ASSURANCES=none`
	`CKNFAST_LOADSHARING=1`
	`CKNFAST_NO_ACCELERATOR_SLOTS=1`

OCS protection
(K-of-N with k > 1)
	`CKNFAST_OVERRIDE_SECURITY_ASSURANCES=none`
	`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`.

2.2.9. Create or load a Security World

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

Do not create a new Security World and load an existing Security World on the same HSM.

2.2.9.1. Create a new Security World

If you create a new Security World, first decide if your organization needs to enforce FIPS 140-2 Level 3 mode.

For information on available parameters and settings for a new Security World, see the *User Guide* for your HSM or use the `new-world --help` command.

For information on which cipher suites are supported in FIPS 140-2 Level 3 mode, see the *User Guide* for your HSM.

Perform the steps in this section from a privileged client, from example from the RFS if the RFS is enrolled as a privileged client. All steps use `-m` 1 for module #1.

1. Switch to initialization mode.

   Connect or Solo:

   ```
   nopclearfail -m 1 --initialization
   ```

   Edge:

   a. Select and hold the **mode** button until the LED changes to **i** for initialization.

      The **i** LED starts to flash.

   b. Select and briefly hold the **clear** button to change the mode.

      The **i** LED stops flashing. The **status** LED starts to blink in a regular on-off pattern about every half second.

2. Create a Security World.

   The `new-world` commands in this step are examples. Use parameters that satisfy your organization's security requirements.

   12.40.2 and earlier Security World client, using FIPS 140-2 Level 3:

   ```
   new-world -m 1 --initialize -Q 2/5 -c DLf3072s256mRijndael --strict-fips-140-2-level-3 p
   ```

   12.50 and later Security World client, using the Compatibility Pack, and using FIPS 140-2 Level 3:

   Use the `new-world` binary from the 12.40 Compatibility Pack because the default version no longer allows the creation of a DLf3072s256mRijndael world, which is required for RHCS to work.

   ```
   new-world-1240 -m 1 --initialize -Q 2/5 -c DLf3072s256mRijndael --strict-fips-140-2-level-3 p
   ```

   12.50 and later Security World client, using an unrestricted world:

   ```
   new-world -m 1 --initialize -Q 2/5 -c DLf3072s256mAEScSP800131Ar1 p
   ```

12.50. and later Security World client, using FIPS 140-2 Level 3:

```
new-world -m 1 --initialize -Q 2/5 -c DLf3072s256mAEScSP800131Ar1 --mode=fips-140-2-level-3 p
```

3. Follow the on-screen prompts to define passphrases for the ACS. The passphrases can be unique to each card, or you can use the same password for all cards.
4. Switch to operational mode.

Connect or Solo:

```
nopclearfail -m 1 --operational
```

Edge:

a. Select and hold the **mode** button until the LED changes to **o** for operational.

The **o** LED starts to flash.

b. Press and briefly hold the **clear** button to change the mode.

The **o** LED stops flashing. The **status** LED starts to blink in a regular on-off pattern about every half second.

2.2.9.2. Load an existing Security World

1. Switch to initialization mode.

Connect or Solo:

```
nopclearfail -m 1 --initialization
```

Edge:

a. Select and hold the **mode** button until the LED changes to **i** for initialization.

The **i** LED starts to flash.

b. Select and briefly hold the **clear** button to change the mode.

The **i** LED stops flashing. The **status** LED starts to blink in a regular on-off pattern about every half second.

2. Load the existing Security World:

```
new-world -m 1 --program
```

3. When prompted, present the existing ACS quorum and associated passphrases.
4. Switch to operational mode.

   Connect or Solo:

   ```
   nopclearfail -m 1 --operational
   ```

   Edge:

   a. Select and hold the **mode** button until the LED changes to **o** for operational.

      The **o** LED starts to flash.

   b. Press and briefly hold the **clear** button to change the mode.

      The **o** LED stops flashing. The **status** LED starts to blink in a regular on-off pattern about every half second.

2.2.9.3. Connect only: Update the state of the Security World on the HSM

If you are using an nShield Solo or nShield Edge, continue with the instructions in Create an OCS or Softcard to protect the application keys.

If your RFS is servicing multiple HSMs without storing Security World data, this update fails. Continue with the integration procedures and contact Entrust nShield Support if necessary.

1. Update the state:

   ```
   nethsmadmin -m 1 -w
   ```

2. Check the state:

   ```
   nethsmadmin -m 1 -c
   Initiating world check...
   World state: 0x37b7000c Initialized Usable StrictFIPS140
   ```

2.2.10. 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.

2.2.10.1. 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 140-2 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
   ```

2.2.10.2. 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 140-2 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
```

2.2.10.3. Copy KMDATA files to other HSM client systems

The Security World files are located on the RFS where they were generated.

To use the Security World on other HSM clients, assuming your RFS is not also your CA server, 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 v10:

```
setup-ds.pl
```

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. The *<instance>* is defined during `setup-ds.pl`:

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

dn: cn=config
nsslapd-versionstring: 389-Directory/1.3.9.1
nsslapd-defaultnamingcontext: o=ca-1-CA
```

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. You can install just the individual subsystems. Choose from:

- `pki-ca`
- `pki-kra`
- `pki-tks`
- `pki-tps`
- `pki-ocsp`
- `pki-console`

For example, to install just the CA subsystem:

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

Alternatively, to install all RHCS packages for all subsystems:

```
yum install redhat-pki 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.

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 token 'accelerator'
Found external token 'OCS1'
```

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.

```
ckcheckinst
PKCS#11 library interface version       2.01
                flags                   0
                manufacturerID          "nCipher Corp. Ltd"
                libraryDescription      "nCipher PKCS#11 12.40+"
                implementation version  12.40

Slot    Status          Label
====    ======          =====
0       Fixed token     "accelerator"
1       Operator card   "OCS1"

Select slot number to run library test or 'R'etry or to 'E'xit: 1
Using slot number 1.

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

2. 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
   ```

3. 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*.

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
```

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

Web Site	URL
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.

Appendix A: Configure pkispawn

A.1. Modifying the sample pkispawn configuration file

A.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`

A.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.

A.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`

A.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`

A.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 (nistp384) and SHA-384:

- `pki_ca_signing_key_algorithm=SHA384withEC`
- `pki_ca_signing_key_size=nistp384`
- `pki_ca_signing_key_type=ecc`

A.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`

A.1.7. Account passwords

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

Sample pkispawn configuration file 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`)

A.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.

A.2. Sample pkispawn configuration file

The following is a sample pkispawn configuration file for Red Hat Certificate System v9. For Red Hat Certificate System v10 and later versions, see Red Hat documentation or the `default.cfg` file on your RHCS system for an updated example. The following is a sample `pkispawn` configuration file for Red Hat Certificate System v9. For Red Hat Certificate System v10 and later versions, see the Red Hat documentation or the `default.cfg` file on your RHCS system for an updated example.

```
[DEFAULT]
JAVA_HOME=%(java_home)s
NSS_DEFAULT_DB_TYPE=%(nss_default_db_type)s
pki_admin_cert_file=%(pki_client_dir)s/admin.cer
pki_admin_cert_request_type=pkcs10
pki_admin_dualkey=False
pki_admin_key_algorithm=SHA256withRSA
pki_admin_key_size=2048
pki_admin_key_type=rsa
pki_admin_password=password
pki_audit_group=pkiaudit
pki_audit_signing_key_algorithm=SHA256withRSA
pki_audit_signing_key_size=2048
pki_audit_signing_key_type=rsa
pki_audit_signing_signing_algorithm=SHA256withRSA
pki_audit_signing_token=OCS1
pki_ca_hostname=%(pki_security_domain_hostname)s
pki_ca_port=%(pki_security_domain_https_port)s
pki_ca_signing_cert_path=/etc/pki/ca-1/alias/ca-1_caSigningCert.cer
pki_ca_signing_nickname=caSigningCert cert-%(pki_instance_name)s CA
pki_cert_chain_nickname=caSigningCert External CA
pki_cert_chain_path=/etc/pki/ca-1/alias/caChain.p7c
pki_client_admin_cert=%(pki_client_dir)s/%(pki_subsystem_type)s_admin.cer
pki_client_admin_cert_p12=%(pki_client_dir)s/%(pki_subsystem_type)s_admin.p12
pki_client_cert_database=%(pki_client_database_dir)s/cert8.db
pki_client_database_dir=%(pki_client_subsystem_dir)s
pki_client_database_password=password
pki_client_database_purge=False
pki_client_dir=/etc/pki/ca-1/agent_alias
pki_client_key_database=%(pki_client_database_dir)s/key3.db
pki_client_password_conf=%(pki_client_subsystem_dir)s/password.conf
pki_client_pkcs12_password=password
pki_client_pkcs12_password_conf=%(pki_client_subsystem_dir)s/pkcs12_password.conf
pki_client_secmod_database=%(pki_client_database_dir)s/secmod.db
pki_client_subsystem_dir=%(pki_client_dir)s
pki_configuration_path=%(pki_root_prefix)s/etc/pki
pki_ds_bind_dn=cn=Directory Manager
pki_ds_create_new_db=True
pki_ds_ldap_port=389
pki_ds_ldaps_port=636
pki_ds_password=password
pki_ds_remove_data=True
pki_ds_secure_connection=False
pki_ds_secure_connection_ca_nickname=Directory Server CA certificate
pki_ds_secure_connection_ca_pem_file=
pki_existing=False
pki_external_ca_cert_chain_path=%(pki_cert_chain_path)s
pki_group=pkiuser
pki_hostname=pki.domain.com
pki_hsm_enable=True
pki_hsm_libfile=/opt/nfast/toolkits/pkcs11/libcknfast.so
pki_hsm_modulename=nfast
pki_http_port=8080
pki_https_port=8443
pki_instance_conf_link=%(pki_instance_path)s/conf
pki_instance_configuration_path=%(pki_configuration_path)s/%(pki_instance_name)s
```

```
pki_instance_database_link=%(pki_instance_path)s/alias
pki_instance_log_path=%(pki_log_path)s/%(pki_instance_name)s
pki_instance_logs_link=%(pki_instance_path)s/logs
pki_instance_name=ca-1
pki_instance_path=%(pki_path)s/%(pki_instance_name)s
pki_issuing_ca=%(pki_issuing_ca_uri)s
pki_issuing_ca_hostname=%(pki_security_domain_hostname)s
pki_issuing_ca_https_port=%(pki_security_domain_https_port)s
pki_issuing_ca_uri=https://%(pki_issuing_ca_hostname)s:%(pki_issuing_ca_https_port)s
pki_log_path=%(pki_root_prefix)s/var/log/pki
pki_path=%(pki_root_prefix)s/var/lib/pki
pki_pkcs12_password=password
pki_pkcs12_path=
pki_registry_path=%(pki_root_prefix)s/etc/sysconfig/pki
pki_replication_password=password
pki_restart_configured_instance=True
pki_san_for_server_cert=
pki_san_inject=False
pki_security_domain_hostname=%(pki_hostname)s
pki_security_domain_https_port=8443
pki_security_domain_name=Security Domain
pki_security_domain_password=password
pki_security_domain_user=admin
pki_self_signed_token=internal
pki_server_database_password=password
pki_server_database_path=%(pki_instance_configuration_path)s/alias
pki_skip_configuration=False
pki_skip_ds_verify=False
pki_skip_installation=False
pki_skip_sd_verify=False
pki_source_conf_path=/usr/share/pki/%(pki_subsystem_type)s/conf
pki_source_cs_cfg=/usr/share/pki/%(pki_subsystem_type)s/conf/CS.cfg
pki_source_registry=/usr/share/pki/setup/pkidaemon_registry
pki_source_server_path=/usr/share/pki/server/conf
pki_source_setup_path=/usr/share/pki/setup
pki_source_subsystem_path=/usr/share/pki/%(pki_subsystem_type)s
pki_sslserver_key_algorithm=SHA256withRSA
pki_sslserver_key_size=2048
pki_sslserver_key_type=rsa
pki_sslserver_nickname=Server-Cert cert-%(pki_instance_name)s CA
pki_sslserver_subject_dn=cn=ca-1.domain.com,OU=Group,OU=Division,O=nCipher Security,C=US
pki_sslserver_token=OCS1
pki_subsystem_archive_log_path=%(pki_subsystem_log_path)s/archive
pki_subsystem_conf_link=%(pki_subsystem_path)s/conf
pki_subsystem_configuration_path=%(pki_instance_configuration_path)s/%(pki_subsystem_
type)s
pki_subsystem_database_link=%(pki_subsystem_path)s/alias
pki_subsystem_key_algorithm=SHA256withRSA
pki_subsystem_key_size=2048
pki_subsystem_key_type=rsa
pki_subsystem_log_path=%(pki_instance_log_path)s/%(pki_subsystem_type)s
pki_subsystem_logs_link=%(pki_subsystem_path)s/logs
pki_subsystem_nickname=subsystemCert cert-%(pki_instance_name)s CA
pki_subsystem_path=%(pki_instance_path)s/%(pki_subsystem_type)s
pki_subsystem_registry_link=%(pki_subsystem_path)s/registry
pki_subsystem_subject_dn=cn=CA-1 Subsystem Certificate,OU=Group,OU=Division,O=nCipher
Security,C=US
pki_subsystem_token=OCS1
pki_theme_enable=True
pki_theme_server_dir=/usr/share/pki/common-ui
pki_token_name=OCS1
pki_token_password=password
pki_user=pkiuser-ca-1
[Tomcat]
pki_ajp_host=localhost
pki_ajp_port=8009
pki_cgroup_cpu_systemd_service=%(pki_cgroup_cpu_systemd_service_path)s/%(pki_systemd_service)
s
pki_cgroup_cpu_systemd_service_path=/sys/fs/cgroup/cpu\,cpuacct/system/%(pki_systemd_
service)s
pki_cgroup_systemd_service=%(pki_cgroup_systemd_service_path)s/%(pki_instance_name)s
```

```
pki_cgroup_systemd_service_path=/sys/fs/cgroup/systemd/system/%(pki_systemd_service)s
pki_clone=False
pki_clone_pkcs12_password=password
pki_clone_pkcs12_path=
pki_clone_reindex_data=False
pki_clone_replicate_schema=True
pki_clone_replication_clone_port=
pki_clone_replication_master_port=
pki_clone_replication_security=None
pki_clone_setup_replication=True
pki_clone_uri=https://%(pki_master_hostname)s:%(pki_master_https_port)s
pki_enable_access_log=True
pki_enable_java_debugger=False
pki_enable_on_system_boot=True
pki_enable_proxy=False
pki_instance_conf_log4j_properties=%(pki_instance_configuration_path)s/log4j.properties
pki_instance_lib=%(pki_instance_path)s/lib
pki_instance_lib_log4j_properties=%(pki_instance_lib)s/log4j.properties
pki_instance_registry_path=%(pki_instance_type_registry_path)s/%(pki_instance_name)s
pki_instance_systemd_link=%(pki_instance_path)s/%(pki_instance_name)s
pki_instance_type=Tomcat
pki_instance_type_registry_path=%(pki_registry_path)s/tomcat
pki_master_hostname=%(pki_security_domain_hostname)s
pki_master_https_port=%(pki_security_domain_https_port)s
pki_proxy_http_port=80
pki_proxy_https_port=443
pki_security_manager=true
pki_server_external_certs_path=
pki_server_pkcs12_password=password
pki_server_pkcs12_path=
pki_source_catalina_properties=%(pki_source_server_path)s/catalina.properties
pki_source_context_xml=%(pki_source_server_path)s/context.xml
pki_source_server_xml=%(pki_source_server_path)s/server.xml
pki_source_servercertnick_conf=%(pki_source_server_path)s/serverCertNick.conf
pki_source_tomcat_conf=%(pki_source_server_path)s/tomcat.conf
pki_subsystem_registry_path=%(pki_instance_registry_path)s/%(pki_subsystem_type)s
pki_subsystem_signed_audit_log_path=%(pki_subsystem_log_path)s/signedAudit
pki_systemd_service=/lib/systemd/system/pki-tomcatd@.service
pki_systemd_service_link=%(pki_systemd_target_wants)s/pki-tomcatd@%(pki_instance_
name)s.service
pki_systemd_target=/lib/systemd/system/pki-tomcatd.target
pki_systemd_target_wants=/etc/systemd/system/pki-tomcatd.target.wants
pki_tomcat_bin_link=%(pki_instance_path)s/bin
pki_tomcat_bin_path=/usr/share/tomcat/bin
pki_tomcat_common_lib_path=%(pki_tomcat_common_path)s/lib
pki_tomcat_common_path=%(pki_instance_path)s/common
pki_tomcat_common_webapps_path=%(pki_instance_path)s/common/webapps
pki_tomcat_lib_path=/usr/share/tomcat/lib
pki_tomcat_server_port=8005
pki_tomcat_subsystem_webapps_path=%(pki_subsystem_path)s/webapps
pki_tomcat_systemd=/usr/sbin/tomcat
pki_tomcat_tmpdir_path=%(pki_instance_path)s/temp
pki_tomcat_webapps_path=%(pki_instance_path)s/webapps
pki_tomcat_webapps_subsystem_path=%(pki_tomcat_subsystem_webapps_path)s/%(pki_subsystem_
type)s
pki_tomcat_webapps_subsystem_webinf_classes_path=%(pki_tomcat_webapps_subsystem_
path)s/WEB-INF/classes
pki_tomcat_webapps_subsystem_webinf_lib_path=%(pki_tomcat_webapps_subsystem_path)s/WEBINF/
lib
pki_tomcat_work_catalina_host_path=%(pki_tomcat_work_catalina_path)s/localhost
pki_tomcat_work_catalina_host_run_path=%(pki_tomcat_work_catalina_host_path)s/_
pki_tomcat_work_catalina_host_subsystem_path=%(pki_tomcat_work_catalina_host_path)s/%
(pki_subsystem_type)s
pki_tomcat_work_catalina_path=%(pki_tomcat_work_path)s/Catalina
pki_tomcat_work_path=%(pki_instance_path)s/work
[CA]
pki_admin_email=%(pki_admin_name)s@localhost
pki_admin_name=%(pki_admin_uid)s
pki_admin_nickname=CA-1 Agent Certificate
pki_admin_subject_dn=cn=CA-1 Agent Certificate,OU=Group,OU=Division,O=nCipher Security,
C=US
```

```
pki_admin_uid=admin
pki_audit_signing_cert_path=
pki_audit_signing_csr_path=
pki_audit_signing_nickname=auditSigningCert cert-%(pki_instance_name)s CA
pki_audit_signing_subject_dn=cn=CA-1 Audit Certificate,OU=Group,OU=Division,O=nCipher
Security,C=US
pki_ca_signing_csr_path=/etc/pki/ca-1/alias/ca-1_caSigningCert.req
pki_ca_signing_key_algorithm=SHA256withRSA
pki_ca_signing_key_size=2048
pki_ca_signing_key_type=rsa
pki_ca_signing_record_create=True
pki_ca_signing_serial_number=1
pki_ca_signing_signing_algorithm=SHA256withRSA
pki_ca_signing_subject_dn=cn=CA-1,OU=Group,OU=Division,O=nCipher Security,C=US
pki_ca_signing_token=OCS1
pki_ca_starting_crl_number=0
pki_default_ocsp_uri=
pki_ds_base_dn=o=%(pki_instance_name)s-CA
pki_ds_database=%(pki_instance_name)s-CA
pki_ds_hostname=ldap.domain.com
pki_external=False
pki_external_pkcs12_password=password
pki_external_pkcs12_path=%(pki_pkcs12_path)s
pki_external_step_two=False
pki_import_admin_cert=False
pki_master_crl_enable=True
pki_ocsp_signing_cert_path=
pki_ocsp_signing_csr_path=
pki_ocsp_signing_key_algorithm=SHA256withRSA
pki_ocsp_signing_key_size=2048
pki_ocsp_signing_key_type=rsa
pki_ocsp_signing_nickname=ocspSigningCert cert-%(pki_instance_name)s CA
pki_ocsp_signing_signing_algorithm=SHA256withRSA
pki_ocsp_signing_subject_dn=cn=CA-1 OCSP Certificate,OU=Group,OU=Division,O=nCipher Security,
C=US
pki_ocsp_signing_token=OCS1
pki_profiles_in_ldap=False
pki_random_serial_numbers_enable=False
pki_replica_number_range_end=100
pki_replica_number_range_start=1
pki_req_ext_add=False
pki_req_ext_critical=False
pki_req_ext_data=1E0A00530075006200430041
pki_req_ext_oid=1.3.6.1.4.1.311.20.2
pki_request_number_range_end=10000000
pki_request_number_range_start=1
pki_serial_number_range_end=10000000
pki_serial_number_range_start=1
pki_share_db=False
pki_source_admincert_profile=%(pki_source_conf_path)s/%(pki_admin_key_type)sAdminCert.
profile
pki_source_caauditsigningcert_profile=%(pki_source_conf_path)s/caAuditSigningCert.profile
pki_source_cacert_profile=%(pki_source_conf_path)s/caCert.profile
pki_source_caocspcert_profile=%(pki_source_conf_path)s/caOCSPCert.profile
pki_source_emails=/usr/share/pki/ca/emails
pki_source_flatfile_txt=%(pki_source_conf_path)s/flatfile.txt
pki_source_profiles=/usr/share/pki/ca/profiles
pki_source_proxy_conf=%(pki_source_conf_path)s/proxy.conf
pki_source_registry_cfg=%(pki_source_conf_path)s/registry.cfg
pki_source_servercert_profile=%(pki_source_conf_path)s/%(pki_sslserver_key_type)sServer-
Cert.profile
pki_source_subsystemcert_profile=%(pki_source_conf_path)s/%(pki_subsystem_key_
type)sSubsystemCert.profile
pki_sslserver_cert_path=
pki_sslserver_csr_path=
pki_subordinate=False
pki_subordinate_create_new_security_domain=False
pki_subordinate_security_domain_name=%(pki_dns_domainname)s Subordinate Security Domain
pki_subsystem_cert_path=
pki_subsystem_csr_path=
pki_subsystem_emails_path=%(pki_subsystem_path)s/emails
```

```
pki_subsystem_name=CA %(pki_hostname)s %(pki_https_port)s
pki_subsystem_profiles_path=%(pki_subsystem_path)s/profiles
```