CIPHERMAIL EMAIL ENCRYPTION

Ciphermail HSM Configuration Guide

April 12, 2017, Rev: 6852

Contents

1 Introduction

This guide explains how to configure the CipherMail Email Encryption Gateway for HSM support. It is assumed that a CipherMail gateway is already installed in the directory `/usr/share/djigzo`[1] and that the gateway is fully functional.

Requirements

- A functional CipherMail Enterprise Gateway
- HSM with support for PKCS#11
- CipherMail HSM module

Note: By default the HSM module only handles private key operations, generation of private/public keys and random generation directly. All other operations, like symmetric key encryption, hashing etc. are handled by a software based provider. The main reason for this is that the software based provider is faster for most operations and generally supports more algorithms.

2 PKCS#11 Configuration

This section explains how to configure a PKCS#11 provider for CipherMail. Java uses the PKCS#11 interface to communicate with an HSM.

2.1 Configure CipherMail PKCS#11 Module

A soft-link to the PKCS#11 configuration files should be added to the CipherMail configuration directory:

```
$ cd /usr/share/djigzo/conf/
$ sudo ln -s /usr/share/djigzo-hsm/conf/hsm/
```

2.2 Configure IAIK PKCS#11 provider

This section explains how to configure the HSM module to be used with the IAIK PKCS#11 provider. To load the IAIK PKCS#11 jar file at startup, a soft-link to the IAIK lib directory and a soft-link to the HSM module should be added to the lib.d directory:

```
$ cd /usr/share/djigzo/lib/lib.d
$ sudo ln -s /usr/share/djigzo-hsm/lib/ hsm
```

Replace the evaluation IAIK jars (iaik_jce.jar and iaikPkcs11Provider.jar) with the licensed IAIK jars.

```
$ sudo cp iaik_jce.jar /usr/share/djigzo-hsm/lib/
$ sudo cp iaikPkcs11Provider.jar /usr/share/djigzo-hsm/lib/
```

[1] If a different directory is used, update the installation instructions accordingly.

CipherMail should be configured to store the private keys on the HSM:

```
$ cd /usr/share/djigzo/conf/spring/spring.d
$ sudo ln -s /usr/share/djigzo-hsm/conf/spring/hsm.xml
$ sudo ln -s /usr/share/djigzo-hsm/conf/spring/hsm-iaik.xml
$ sudo ln -s /usr/share/djigzo-hsm/conf/spring/hsm-pgp.xml
$ sudo ln -s /usr/share/djigzo-hsm/conf/spring/hsm-iaik-watchdog.xml
```

Note: hsm-iaik-watchdog will create a test key the first time it is activated and will periodically check (every 30 sec) whether the HSM can still be accessed. If there is problem with the HSM connection, the back-end will be restarted automatically.

3 HSM configuration

This section explains how to configure a particular HSM for CipherMail.

3.1 SafeNet ProtectServer

This section explains how to configure HSM support for SafeNet ProtectServer. The software for the SafeNet ProtectServer should be installed according to the SafeNet instructions. It is assumed that SafeNet is installed to `/opt/PTK/` and that slot 0 is used.

3.1.1 Configure ProtectServer

Initialize Admin Token SO and Administrator PINs[2] and initialize slot 0 and set the SO and USER PIN for slot 0. The slot must be initialized as the user that runs CipherMail:

```
$ sudo su djigzo -s /bin/bash
$ export LD_LIBRARY_PATH=/opt/PTK/lib
$ /opt/PTK/bin/ctconf -fc
$ /opt/PTK/bin/ctkmu t -s0 -ldjigzo
$ exit
```

Note: if the software only ProtectServer is used, the `ctconf` command should be run by the user that runs the CipherMail server (by default this should be user djigzo).

3.1.2 IAIK PKCS#11 configuration

A soft-link to the SafeNet specific properties file should be created:

```
$ cd /usr/share/djigzo/conf/hsm/
$ sudo ln -s iaik-pkcs11-config.properties.safenet iaik-pkcs11-config.properties
```

[2]The -fc option sets the "No Public Crypto" flag to make sure that for each token the CKF_LOGIN_REQUIRED flag is set and that either the USER or SO must be logged in.

Change PIN The PIN for authenticating to the HSM is stored in the file `/usr/share/djigzo/conf/hsm/iaik-pkcs11-config.properties` (the default PIN is set to "123") and should be set to the USER PIN which was selected in the previous steps when the slot was initialized.

Select the 32bit or 64bit library The IAIK provider uses a shared module (libpkcs11wrapper.so) for accessing the HSM. On a 32bit system a different module should be used than on a 64bit system. The correct library version should be configured in `/usr/share/djigzo/conf/hsm/iaik-pkcs11-config.properties`

For a 32bit system use:

```
PKCS11_WRAPPER_PATH=/usr/share/djigzo/lib/lib.d/hsm/linux-x86/release/libpkcs11wrapper.so
```

For a 64bit system replace `linux-x86` by `linux-x86_64`

Note: make sure that libcryptoki.so is readable by user djigzo!

3.1.3 Finish

CipherMail should be restarted for the changes to take effect:

```
$ sudo service djigzo restart
```

CipherMail should now be ready to use the HSM. Check the log file (`/var/log/djigzo.log`) for any problems.

3.2 Utimaco CryptoServer

This section explains how to configure HSM support for Utimaco CryptoServer. The software for the Utimaco CryptoServer should be installed according to the Utimaco instructions. It is assumed that the Utimaco software is installed in `/opt/utimaco/` and that slot 0 is used.

3.2.1 Configure CryptoServer

The default configuration file for CryptoServer should be stored in `/etc/utimaco/cs2_pkcs11.ini`. The `cs2_pkcs11.ini` file should at least contain the device to use (in the following example the HSM is a CryptoServer Lan running on 192.168.1.34 port 3001).

```
[Global]
Timeout = 5000
Logging = 0
Logpath = /tmp

[CryptoServer]
Device     = TCP:3001@192.168.1.34
Timeout    = 600000
AppTimeout = 864000
SlotCount  = 100
```

Note: The AppTimeout setting in `/etc/utimaco/cs2_pkcs11.ini` should be set to `AppTimeout = 864000` or some other high value.

3.2.2 IAIK PKCS#11 configuration

A soft-link to the Utimaco specific properties file should be created:

```
$ cd /usr/share/djigzo/conf/hsm/
$ sudo ln -s iaik-pkcs11-config.properties.utimaco iaik-pkcs11-config.properties
```

Change PIN The PIN for authenticating to the HSM is stored in the file `/usr/share/djigzo/conf/hsm//hsm/iaik-pkcs11-config.properties` (the default PIN is set to "123") and should be set to the user PIN which was selected for the slot.

Select the 32bit or 64bit library Set the correct version (32bit or 64bit) of the CryptoServer PKCS#11 library in the file `/usr/share/djigzo/conf/hsm/iaik-pkcs11-config.properties`

Note: make sure that libcs2_pkcs11.so is readable by user djigzo!

3.2.3 Finish

CipherMail should be restarted for the changes to take effect:

```
$ sudo service djigzo restart
```

3.3 nCipher

This section explains how to configure HSM support for nCipher HSM. The nCipher support software should already be installed according to the nCipher installation instructions. It is assumed that the nCipher support software (including the PKCS#11 library) is installed in `/opt/nfast/` and that slot 0 is used.

3.4 Configure nCipher tools

To allow the user *djigzo* to write to the kmdata directory, the user *djigzo* should be added to the *nfast* group.

3.4.1 Configure PKCS#11

The back-end should be configured to use the nCipher provided PKCS#11 library.

IAIK PKCS#11 configuration A soft-link to the nCipher specific properties file should be created:

```
$ cd /usr/share/djigzo/conf/hsm/
$ sudo ln -s iaik-pkcs11-config.properties.ncipher iaik-pkcs11-config.properties
```

Select the 32bit or 64bit library By default the 64bit PKCS#11 libraries are selected. If a 32bit OS is used, change the paths in the following file to use the 32bit version of the libraries: `/usr/share/djigzo/lib/lib.d/hsm/iaik-pkcs11-config.properties`.

Allow module protected keys To allow module protected keys, the following line should be added to the nCipher configuration file `/opt/nfast/cknfastrc`[3].

```
$ sudo vi /opt/nfast/cknfastrc
```

add:

```
CKNFAST_FAKE_ACCELERATOR_LOGIN=1
```

3.4.2 Finish

nCipher and CipherMail should be restarted for the changes to take effect:

```
$ sudo /etc/init.d/nc_hardserver restart
$ sudo service djigzo restart
```

Check CipherMail back-end logs to see if the back-end starts correctly:

```
$ less /var/log/djigzo.log
```

[3] If the file /opt/nfast/cknfastrc does not yet exist, create it