Integration Guide

PrimeKey Enterprise Java Bean Certificate Authority (EJBCA)
Red Hat Enterprise Linux

utimaco®

Imprint

copyright 2016	Utimaco IS GmbH Germanusstrasse 4 D-52080 Aachen Germany
phone	+49 (0)241 / 1696-200
fax	+49 (0)241 / 1696-199
web	http://hsm.utimaco.com
email	support-cs@utimaco.com
document version	1.1.1
date	January 2016
author	System Engineering HSM
document no.	IG_EJBCA_6.3.1.1_RHEL6.4

Contents

1 Introduction

This article provides an integration guide for configuring *EJBCA (Enterprise Java Bean Certification Authority)* with the *CryptoServer* using the PKCS#11 R2 API. In the course of this guide it is focused on the integration procedures necessary to run *EJBCA* with the *CryptoServer*. Further information details to setup *EJBCA* as Certificate Authority can be found on *EJBCA* website.

2 Overview

PrimeKey Enterprise Java Bean Certificate Authority (EJBCA) is an open source application which is designed to manage the digital keys and certificates that make up the digital identities required to transparently automate all security-related processes in an organization. As the organization's CA system, *PrimeKey EJBCA Security Manager* software enables the use of digital signatures, digital receipts, encryption and permissions management services across a wide variety of applications and solutions. To enhance security of keys and certificates generated and used by a CA, *EJBCA* can be configured to use a Hardware Security Module (HSM). Enabling the use of a hardware security module with *EJBCA* not only strengthens protection of keys and certificates but also might be a necessary step towards legal conformity and certification.

CryptoServer is a hardware security module developed by *Utimaco IS GmbH*, i.e. a physically protected specialized computer unit designed to perform sensitive cryptographic tasks and to securely manage and store cryptographic keys and data. It can be used as a universal, independent security component for heterogeneous computer systems.

3 Requirements

You should have prepared an installed *Red Hat Enterprise Linux* operating system. If you are using a PCI(e) card you also have to compile and install the necessary driver for that card. We use in this guide our LAN appliance.

Software- and Hardware Requirements

HSM Model	CryptoServer CSe-Series/Se-Series LAN CryptoServer Simulator
HSM Firmware	SecurityServer 3.30.0
Software	JBoss Red Hat Enterprise Linux Server 7.2 JBoss EAP 6.4.0 EJBCA 6.3.1.1 CE Apache Ant 1.9.2 OpenJDK 1.7.0

4 Components

In this section, we are going to describe which components are required and how to install and configure all the components those are necessary for setting up an *EJBCA*.

4.1 Download and Install the JDK

The JDK used in this guide was *OpenJDK* and one can install the JDK by using following command in a terminal window.

CONSOLE

```
# yum install java-1.7.0-openjdk.x86_64
```

4.2 Download and Install Apache Ant

Apache Ant can be downloaded and installed by typing the below command into a terminal window.

CONSOLE

```
# yum install ant ant-optional
```

4.3 Download JBoss Enterprise Application Platform

For your reference, we used *JBoss EAP 6.4.0* as application server in this guide. Download *JBoss EAP* and unzip the downloaded file to a directory of your choice. The path `/opt` is our installation path which we will use throughout this guide. Let us assume that we installed the application server under `/opt/jboss`. When installation is done, create a user and group named as *jboss* with a working directory `/opt/jboss`.

CONSOLE

```
# groupadd jboss
# useradd -d /opt/jboss -g jboss jboss
# passwd jboss
```

If you use different installation path other than above path, then modify the path value whenever needed according to your path.

4.4 Download EJCBA

Download *EJBCA* from the *EJBCA* website and unzip it to the directory of your choice. In this article, we used the path `/opt/ejbca` as our installation path.

4.5 Setup Utimaco PKCS11 R2 Library and Tools

Install and configure the software for PKCS11 R2 of our Product CD as described in *QuickStart Guide — PKCS#11 (R2) — Linux* (CS_QSG_PKCS11.pdf).

5 Installation of EJBCA

We are going to install *EJBCA* with only a management Certificate Authority first. As next the configuration extended to include PKCS#11 token in *EJBCA*. PKCS#11 token enables the usage of the CryptoServer as key storage for further Certificate Authorities instances in *EJBCA*.

5.1 Setup the Environment Variables

To run all installed tools properly, a series of environment variables has to be created first. Create a file named as a `ejbca.sh` in `/etc/profile.d` and add the following lines to this file. Always remember that provided paths in this article depend on your installation and needs to be adjusted when your installation differs from this article. This script will be executed whenever a bash login shell is entered.

```
#!/bin/sh
export JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk
export JBOSS_HOME=/opt/jboss
export PATH=$JAVA_HOME/bin:$ANT_HOME/bin:$PATH
```

5.2 Configure and Start JBoss EAP

To enable PKCS#11 token support for *EJBCA*, *JBoss* application server needs to be configured to allow *Sun PKCS#11 JCE* provider to be included into *JBoss* classpath and accessible by *EJBCA*. Adjust the `modules.xml` (`$JAVA_HOME/modules/system/layers/base/sun/jdk/main/module.xml`) file in *JBoss* modules path and add these lines

```
<path name="sun/security/x509"/>
<path name="sun/security/pkcs11"/>
<path name="sun/security/pkcs11/wrapper"/>
```

into *paths* section of XML structure.

```
<?xml version="1.0" encoding="UTF-8"?>
...
```

```
<module xmlns="urn:jboss:module:1.1" name="sun.jdk">
    <resources>
        <!-- currently jboss modules has not way of importing services from
        classes.jar so we duplicate them here -->
        <resource-root path="service-loader-resources"/>
    </resources>
    <dependencies>
        <system export="true">
            <paths>
                <path name="sun/security/x509"/>
                <path name="sun/security/pkcs11"/>
                <path name="sun/security/pkcs11/wrapper"/>
                <path name="com/sun/script/javascript"/>
...
            </paths>
...
        </system>
    </dependencies>
</module>
```

Now we can start the *JBoss* Server. To start up a *JBoss* managed domain, execute the `$JBOSS_HOME/bin/domain.sh` script. To start up a standalone server, execute the `$JBOSS_HOME/bin/standalone.sh`. For detailed information on how to start *JBoss* and the provided command line parameters we refer to the documentation of *JBoss*. In this guide we start up a standalone server with no restrictions for accessibility.

CONSOLE

```
# $JBOSS_HOME/bin/standalone.sh -b 0.0.0.0 -bmanagement 0.0.0.0 &
```

5.3 Configuration for Installation of EJBCA

This article installs *EJBCA* without an external database or other extra enabled features of *EJBCA*. Basically four files needs to adjusted: `ejbca.properties`, `web.properties`, `catoken.properties` and `install.properties`. You find them in `$EJBCA_HOME/conf`. In the following we describe only

the necessary instructions to start *EJBCA* with the CryptoServer. For other necessary and security-related configuration steps we refer to the *EJBCA* documentation.

Adjust in `ejbca.properties` the following attributes. For test puproses only switch *ejbca.productionmode* to false to indicate test mode.

```
# Application server home directory used during development. The path can not
# end with a slash or backslash.
appserver.home=/opt/jboss

# Which application server is used? Normally this is auto-detected from
# 'appserver.home' and should not be configured.
# Possible values: jboss, glassfish (, weblogic)
# Default: <auto-detect>
appserver.type=jboss

# To prevent accidental runs of tests or deploying the wrong thing in a
# production environment, we
# could prevent this by setting this variable to either "true" or "false".
# Setting this value to 'false' will allow system tests to alter the
# configuration of the running
# EJBCA instance.
# Default: true
ejbca.productionmode=false
```

In `web.properties` edit the following attributes.

```
# The password used to protect the generated super administrator P12
# keystore (to be imported in browser).
# Choose a good password here.
superadmin.password=ejbca

# The CA servers DNS host name, must exist on client using the admin GUI.
httpsserver.hostname=10.17.72.201
```

```
# The private port JBoss will listen to https on, client cert required
# Default 8443
httpserver.privhttps=8443

# Available PKCS#11 CryptoToken libraries and their display names
# If a library file's presence is not detected it will not show up in the
# Admin GUI.
# Default values (see src/java/defaultvalues.properties for most up to date
# values):
cryptotoken.p11.lib.30.name=Utimaco
cryptotoken.p11.lib.30.file=/usr/lib/libcs_pkcs11_R2.so
```

Also the attributes in `install.properties` must be adjusted. The value for *ca.tokenpassword* is the password of the cryptographic user of the initialized PKCS#11 slot.

```
# ------------ Administrative CA configuration ---------------------
# This installation will create a first administrative CA. This CA will be used
# to create the first
# superadministrator and for the SSL server certificate of administrative web
# server.
# When the administrative web server have been setup you can create other CAs
# and administrators.
# This is only used for administrative purposes,
# Enter a short name for the administrative CA.
ca.name=HSM-EJBCA

# The Distinguished Name of the administrative CA.
# This is used in the CA certificate to distinguish the CA.
# Note, you can not use DC components for the initial CA, you can create CAs
# using DC components later on once the admin GUI is up and running.
ca.dn=CN=HSM-EJBCA,O=HSM,C=DE
```

```
# The token type the administrative CA will use.
# Use soft for software generated keys (default) or enter a class path for the
# HSM class.
# Normally the HSM class should be the PKCS11CryptoToken.
#
# Possible values are:
# soft
# org.cesecore.keys.token.PKCS11CryptoToken
# se.primeKey.caToken.card.PrimeCAToken
# Note: If you use JBoss 7/EAP 6 and want to use PKCS#11 you have to configure
# JBoss to permit this.
#       See instructions in the Install Guide.
#
# Default: soft
ca.tokentype=org.cesecore.keys.token.PKCS11CryptoToken

# Password for the administrative CA token.
# With soft token, use password null.
# To prompt for the password on the terminal, don't set, i.e. comment out the
# line below.
# If no password should be used (for example nCipher module protected), use
# password '' (nothing).
ca.tokenpassword=123456

# Configuration file were you define key name, password and key alias for the
# HSM used
# by the administrative CA. Same as the Hard CA Token Properties in Admin gui.
# Remove everything in the file and add your own configuration.
# Note that this must be a full path.
# On windows use / instead of \
ca.tokenproperties=/opt/ejbca/conf/catoken.properties
```

The file `catoken.properties` must be configured as follows. You can choose via *slotLabelValue* your favored slot for the Crypto Token which will used during the installation of *EJBCA*. Be sure, that this port is initialized via *PKCS#11 CAT* or *p11tool2*.

```
# Utimaco HSM Crypto Token example:
sharedLibrary=/usr/lib/libcs_pkcs11_R2.so
slotLabelType=SLOT_NUMBER
slotLabelValue=1

# CA key configuration
defaultKey=defaultKey
certSignKey=signKey
crlSignKey=signKey
testKey=testKey
```

Before we can deploy and start *EJBCA* it needs (via the *Java PKCS#11 provider*) two object on the HSM, a private key and a certificate (this is simply a holder of the public key used by java, and not the real certificate of a CA). We can create these keys, which we defined in the `catoken.properties` file, with the command

CONSOLE

```
# $EJBCA_HOME/dist/clientToolBox/ejbcaClientToolBox.sh PKCS11HSMKeyTool \
generate <pkcs11-lib path> <keysize> <keylabel> <slot number>
```

Each CA should have its own slot and each slot must have been initialized before keys could be generated on the them. Here follows an example on how to generate keys to e used by EJBCA.

CONSOLE

```
# $EJBCA_HOME/dist/clientToolBox/ejbcaClientToolBox.sh PKCS11HSMKeyTool \
generate /usr/lib/libcs_pkcs11_R2.so 4096 defaultKey 1
Using Slot Reference Type: Slot Number.
PKCS11 Token [SunPKCS11-libcs_pkcs11_R2.so-slot2] Password:
Created certificate with entry signKey.
# $EJBCA_HOME/dist/clientToolBox/ejbcaClientToolBox.sh PKCS11HSMKeyTool \
generate /usr/lib/libcs_pkcs11_R2.so 4096 signKey 1
Using Slot Reference Type: Slot Number.
PKCS11 Token [SunPKCS11-libcs_pkcs11_R2.so-slot2] Password:
```

```
Created certificate with entry signKey.
# $EJBCA_HOME/dist/clientToolBox/ejbcaClientToolBox.sh PKCS11HSMKeyTool \
generate /usr/lib/libcs_pkcs11_R2.so 4096 testKey 1
Using Slot Reference Type: Slot Number.
PKCS11 Token [SunPKCS11-libcs_pkcs11_R2.so-slot2] Password:
Created certificate with entry signKey.
```

You can view the pkcs11 objects created with the following command.

CONSOLE

```
# p11tool2 Slot=1 LoginUser=ask ListObjects
```

To test keys on the HSM for use by EJBCA you can use the EJBCA Client Toolbox.

CONSOLE

```
# cd $EJBCA_HOME
# ant clientToolBox
```

5.4 Install EJBCA

- Build and deploy *EJBCA* application.

  CONSOLE

  ```
  # cd $EJBCA_HOME
  # ant deploy
  ```

  

  Wait until *EJBCA* application has been deployed by *JBoss* application server

- Create initial Management CA and TLS keystores.

  CONSOLE

  ```
  # ant install
  ```

- Copy `$EJBCA_HOME/p12/superadmin.p12` file to your admin desktop machine and import the `superadmin.p12` file in the certificate store of your web browser. You need for it the password that you have defined in `web.properties` file.

- Open *EJBCA* application using the web browser with this URL `https://{yourserverip}:8443/ejbca` to check if installation was successful

Check the firewall configuration if the website is not accessible.

5.5 Setup PKCS#11 Token

After *EJBCA* application has been deployed to *JBoss* application server and has been checked that *EJBCA* application is accessible via web browser, there is only the management Certificate Authority available at this moment. To create another Certificate Authority based on a PKCS#11 token you need to create another Crypto Token in *EJBCA*. Follow the next steps to create a new Crypto Token.

- Connect to your *EJBCA* instance via web browser (e.g. `https://{yourserverip}:8443/ejbca`)
- Authenticate with *superadmin.p12* certificate
- Select Administration menu item *Miscellaneous -> Administration*
- Select Crypto Tokens menu item *CA Functions -> Crypto Tokens*
- Click *Create new...*
- Enter name for the new Crypto Token
- Select token type PKCS#11
- Enter PKCS#11 user password for *Authentication Code/Repeat Authentication Code*
- Check *Auto-activation*
- Select *Utimaco* as PKCS#11 library
- Enter slot number for PKCS#11 *Reference*
- Press *Save* button to finish Crypto Token creation

The new Crypto Token now appears in the list of available Crypto Tokens. Switch now to *CA Functions -> Certification Authorities* to create a new Certificate Authority. When creating a new Certificate Authority select the new Crypto Token based on PKCS#11. For further details on how to setup a Certificate Authority we refer to the user manual of *EJBCA*.

6 Further Information

This document forms a part of the information and support which is provided by the Utimaco IS GmbH . Additional documentation can be found on the product CD in the documentation directory.

All CryptoServer product documentation is also available at the Utimaco IS GmbH website: http://hsm.utimaco.com

Contact

Utimaco IS GmbH
Germanusstraße 4
D - 52080 Aachen
Germany

phone +49 241 1696 - 200
fax +49 241 1696 - 199

web https://hsm.utimaco.com
email support-cs@utimaco.com