Apache Tomcat

11.0.21

Integration Guide

u.trust GP HSM

6.4.0

utimaco®

Imprint

Copyright 2026	Utimaco IS GmbH
	Krefelder Straße 220
	52070 Aachen
	Germany
Phone	AMERICAS +1-844-UTIMACO (+1 844-884-6226)
	EMEA +49 800-627-3081
	APAC +81 800-919-1301
Internet	https://support.hsm.utimaco.com/
e-mail	support@utimaco.com
Document Version	2.0.0
Date	2026-05-04
Status	PUBLISHED
Document No.	IG-2026-0003

Table of Contents

1 Introduction

This guide is part of the support provided by Utimaco. Additional documentation produced to support your Utimaco u.trust General Purpose (GP) Hardware Security Module (HSM) product can be found in the document directory of the Utimaco u.trust GP HSM product bundle.

All Utimaco u.trust GP HSM product documentation is available from Utimaco's website at *https://utimaco.com/*.

1.1 About This Guide

This guide describes how to integrate an Utimaco u.trust GP HSM with Apache Tomcat. The Utimaco HSM securely stores the private key for SSL and offloads the cryptographic operations to the HSM.

1.2 Target Audience for This Guide

This guide is intended for Apache Tomcat and Utimaco HSM administrators.

1.3 Purpose of the Integration

The integration of Apache Tomcat with u.trust GP HSM serves a critical role in enhancing the security, compliance, and performance of cryptographic operations by utilizing a centrally managed, cloud-based HSM service.

1.4 Document Conventions

The following conventions are used in this guide:

Convention	Use	Example
Bold	Items of the Graphical User Interface (GUI), e.g., menu options	Click **Add** button

Convention	Use	Example
`Monospaced`	Code that is given for explanation or as an example, file paths	`./p11tool2 LoginUser=12345678 GetSlotInfo`
Italic	References and important terms	Visit the official *Utimaco Portal*.

Table 1: Document Conventions

We use special icons to highlight the most important notes and information.

Here you will find important safety information that should be followed.

Here you will find additional notes or supplementary information.

This message marks the result expected after the successful execution of an instruction.

1.5 Abbreviations

The following abbreviations are used in this guide:

Abbreviation	Meaning
CA	Certificate Authority
CD	Compact Disc
CSADM	CryptoServer Command-line Administration Tool

Abbreviation	Meaning
CSR	Certificate Signing Request
GUI	Graphical User Interface
HSM	Hardware Security Module
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
JDK	Java Development Kit
LAN	Local Area Network
MBK	Master Backup Key
P11CAT	PKCS#11 CryptoServer Administration Tool
PCIe	PCI Express Interface
PIN	Personal Identification Number
PKCS#11	Public-Key Cryptography Standard #11
RSA	Rivest-Shamir-Adleman
SSL	Secure Socket Layer
SO	Security Officer

Abbreviation	Meaning
URL	Uniform Resource Locator
PID	Process Identifier
XML	Extensible Markup Language
HTTP/2	Hypertext Transfer Protocol Version 2
NIO	Non-blocking I/O
JVM	Java Virtual Machine

Table 2: List of Abbreviations

2 Overview

2.1 Apache Tomcat

The Apache Tomcat software powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations. The Apache Tomcat software is an open-source implementation of the Jakarta Servlet, Jakarta Server Pages, Jakarta Expression Language, Jakarta WebSocket, Jakarta Annotations and Jakarta Authentication specifications. These specifications are part of the Jakarta EE platform.

2.2 Utimaco u.trust GP HSM

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

3 Integration Requirements and Prerequisites

Ensure the system environment you will be using meets the following hardware and software requirements.

3.1 Tested Versions

The integrations that have been successfully tested with the Utimaco HSM with Apache Tomcat.

Operating System	Apache Tomcat	JAVA	Utimaco Security Server Version	Utimaco HSM
RHEL 9	11.0.21	Java 17	SecurityServer V6.4.0	u.trust GP HSM CSe-Series/Se-Series

Table 3: List of Tested Versions

3.2 Software Requirements

Software	Software Requirements
HSM Interfaces	CryptoServer PKCS 11 configured
JDK	17.0.11
Host VM	Host machine Operating System: Redhat 9 and above
HSM software	Utimaco CryptoServer Software 6.4.0
Tomcat	Tomcat version 11.0.21

Table 4: List of Software Requirements

3.3 Hardware Requirements

Hardware	Hardware Requirements
Utimaco LAN HSM	u.trust GP HSM CSe-Series/Se-Series LAN with firmware SecurityServer 6.4.0 or higher
Utimaco PCI-e HSM	u.trust GP HSM CSe-Series/Se-Series PCI-e with firmware SecurityServer 6.4.0 or higher

Table 5: List of Hardware Requirements

Set up an account on the Utimaco support portal and request download access at the following URL: https://support.hsm.utimaco.com/.

3.4 Prerequisites

Before you begin, please ensure that you have:

- u.trust GP HSM set up and configured. Refer to the u.trust GP HSM documentation to set up the HSM.
- u.trust GP HSM Default Admin replaced with a new admin user.
- The MBK created and stored onto each HSM. Refer to the u.trust GP HSM documentation to set up the MBK.
- The operating system listed in *Tested Versions*.
- SecurityServer as listed in *Tested Versions*.
- Familiarized yourself with the Apache Tomcat documents and setup process.
- The admin user for installing software on the Apache Tomcat server.
- Allowed port 443 through the firewall.

4 Installing and Configuring Utimaco SecurityServer Software

4.1 Downloading and Installing Utimaco Software

If you have not already done so, please create and request an Utimaco Support Portal Account. This will allow you to download the software components needed for this installation.

1. Copy the downloaded software at the appropriate location on the Apache Tomcat Server.
2. Create Utimaco folder under `/opt` directory and further create 2 directories: `/opt/utimaco/bin` and `/opt/utimaco/lib`.

›_ Console

```
# mkdir -p  /opt/utimaco/bin

# mkdir /opt/utimaco/lib
```

3. Copy pkcs11 library file `libcs_pkcs11_R3.so` from Utimaco CryptoServer software to the `/opt/utimaco/lib` directory.

›_ Console

```
# cp ~/path_to_application_folder/lib/libcs_pkcs11_R3.so /opt/utimaco/lib
```

4. Copy the `csadm` and `p11tool2` files from Utimaco CryptoServer software to `/opt/utimaco/bin` directory and make both the files executable.

›_ Console

```
# cd ~/path_to_application_folder

# cp csadm p11tool2 /opt/utimaco/bin

# chmod +x /opt/utimaco/bin/csadm /opt/utimaco/bin/p11tool2
```

4.2 u.trust GP HSM PKCS#11 Configuration

1. Create the directory `/etc/utimaco`. Locate the Utimaco PKCS#11 configuration file in your SecurityServer directory, `Linux/x86-64/Crypto_APIS/PKCS11_R3/sample`. Copy the Utimaco PKCS#11 configuration file `cs_pkcs11_R3.cfg` into `/etc/utimaco` directory.

›_ Console

```
# mkdir /etc/utimaco

# cd <install directory>/Software/Linux/x86-64/Crypto_APIs/PKCS11_R3/sample # cp
cs_pkcs11_R3.cfg /etc/utimaco # cd /etc/utimaco
```

2. Edit the `cs_pkcs11_R3.cfg` file and make the appropriate changes to the file.

cs_pkcs11_R3.cfg

```
[Global]
# For unix:
Logpath = /tmp
# Loglevel (0 = NONE; 1 = ERROR; 2 = WARNING; 3 = INFO; 4 = TRACE)
Logging = 1
Keepalive = true
# Set the Device to connect with
[CryptoServer]
# Device specifier
Device = <HSM_IP>
```

For more information regarding the commands and command parameters please check the u.trust GP HSM documentation. The device may be a u.trust GP HSM (PCIe or LAN) device.

The device line will follow one of these patterns, based on the HSM form-factor:

Device = 288@<HSM IP address> Hardware (LAN) HSM

OR

Device = /dev/cs2.0 Hardware (PCIe) HSM

To make your testing easier, it would be good to enable the PKCS#11 log file.

That can be enabled by editing the **Logging** Loglevel. Set the **LogPath** and Logging **Loglevel** to **1**.

For testing you may want to increase it to **4**.

The added **LogPath** points to a writable directory, not to a file.

If you encounter problems, check the log file named **cs_pkcs11_R3.log** in the **LogPath** defined directory. When you are done testing, you should change Logging to **1** or **2**.

This will limit the logging to only critical and important messages.

4.3 Create SO User and Initialize a Slot

You must initialize a slot with a custom label using `p11tool2`.

First using `p11tool2` create, the SO or Security Officer and then using `p11tool2` command initialize the Slot that you want to use, and the slot user as shown below.

›_ Console

```
# ./p11tool2 slot=<slot_no> Label=<token_label> Login=ADMIN,ADMIN.key
InitToken=<SO_PIN>

# ./p11tool2 slot=<slot_no> LoginSO=<SO_PIN> InitPin=<CryptoUser_PIN>
```

```
[admin@master-node bin]$ # ./p11tool2 Slot=0 Label=tomcat Login=ADMIN,ADMIN.key InitToken=87654321

#

# ./p11tool2 Slot=0 LoginSO=87654321 InitPin=12345678
```

Figure 1 : Slot Initialization Output

4.4 Create a pkcs11.cfg at /etc/utimaco/

Create a file `/etc/utimaco/pkcs11.cfg` and add the below contents to it.

›_ Console

```
name=CryptoServer
library=/opt/utimaco/lib/libcs_pkcs11_R3.so
slotListIndex=0
attributes=compatibility
attributes(*,*,*) = {
CKA_TOKEN = true
}
```

This file will be used by the SunPKCS11 provider to perform cryptographic operations on the Utimaco HSM.

Specify the correct library path and slot index.

5 Downloading and Installing Apache Tomcat

To install Apache Tomcat:

1. (Optional) It is recommended to update the system with the latest security patch.

›_ Console

```
# dnf -y update
```

2. Install OpenJDK for Java 17.

›_ Console

```
# dnf -y install java-17-openjdk java-17-openjdk-devel
```

3. Create a non-root user and set its password.

›_ Console

```
# useradd tomcat
# passwd tomcat
```

4. Download Tomcat 11.

›_ Console

```
# wget https://dlcdn.apache.org/tomcat/tomcat-11/v11.0.21/bin/apache-
tomcat-11.0.21.tar.
```

5. Create a directory.

›_ Console

```
# mkdir -p /opt/tomcat
```

6. Extract the archived file to the `/opt/tomcat` directory.

›_ Console

```
# tar -xvf apache-tomcat-11.0.21.tar.gz -C /opt/tomcat --strip-components=1
```

7. Change ownership of the `/opt/tomcat` directory to the Tomcat user.

›_ Console

```
# chown -R tomcat:tomcat /opt/tomcat
```

8. Set executable permissions to scripts.

›_ Console

```
# chmod +x /opt/tomcat/bin/*.sh
```

9. Create Apache Tomcat `systemd` file `/etc/systemd/system/tomcat.service` to manage the Tomcat service through `systemctl` and add below lines.

›_ Console

```
[Unit]
Description=Apache Tomcat Web Application Container
Wants=network.target
After=network.target

[Service]
Type=forking

Environment=JAVA_HOME=/usr/lib/jvm/jre

Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat
Environment='CATALINA_OPTS=-Xms512M -Xmx1G -Djava.net.preferIPv4Stack=true'
Environment='JAVA_OPTS=-Djava.awt.headless=true'

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh
SuccessExitStatus=143

User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always

[Install]
WantedBy=multi-user.target
```

Change the values according to your system configuration.

```
[admin@master-node ~]$ vi /etc/systemd/system/tomcat.service
[Unit]
Description=Apache Tomcat Web Application Container
Wants=network.target
After=network.target
[Service]
Type=forking
Environment=JAVA_HOME=/usr/lib/jvm/jre
Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat
Environment='CATALINA_OPTS=-Xms512M -Xmx1G -Djava.net.preferIPv4Stack=true'
Environment='JAVA_OPTS=-Djava.awt.headless=true'
ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh
SuccessExitStatus=143
User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always
[Install]
WantedBy=multi-user.target
~
```

Figure 2 : tomcat.service File Output

10. Reload the daemon using:

›_ Console

```
# systemctl daemon-reload
```

11. Start and enable the Tomcat service.

›_ Console

```
# systemctl start tomcat
# systemctl enable tomcat
```

12. Confirm that the Tomcat status is `running` using:

›_ Console

```
# systemctl status tomcat
```

```
admin@master-node tomcat]$ sudo systemctl start tomcat
admin@master-node tomcat]$ sudo systemctl status tomcat
 tomcat.service - Apache Tomcat Web Application Container
    Loaded: loaded (/etc/systemd/system/tomcat.service; enabled; preset: disabled)
    Active: active (running) since Tue 2026-04-14 00:57:39 PDT; 2min 26s ago
  Main PID: 3723328 (java)
     Tasks: 40 (limit: 48895)
    Memory: 109.5M
       CPU: 11.491s
    CGroup: /system.slice/tomcat.service
            └─3723328 /usr/lib/jvm/jre/bin/java -Djava.util.logging.config.file=/opt/tomcat/conf/logging.properties -Djava.util.logging.manager=org.apache.

pr 14 00:57:38 master-node systemd[1]: Starting Apache Tomcat Web Application Container...
pr 14 00:57:39 master-node startup.sh[3723321]: Tomcat started.
pr 14 00:57:39 master-node systemd[1]: Started Apache Tomcat Web Application Container.
ines 1-13/13 (END)
```

Figure 3 : Tomcat Service Status

13. Open http://<apache_tomcat_server-ip>:8080, in any web browser and verify that the Apache Tomcat page is visible.

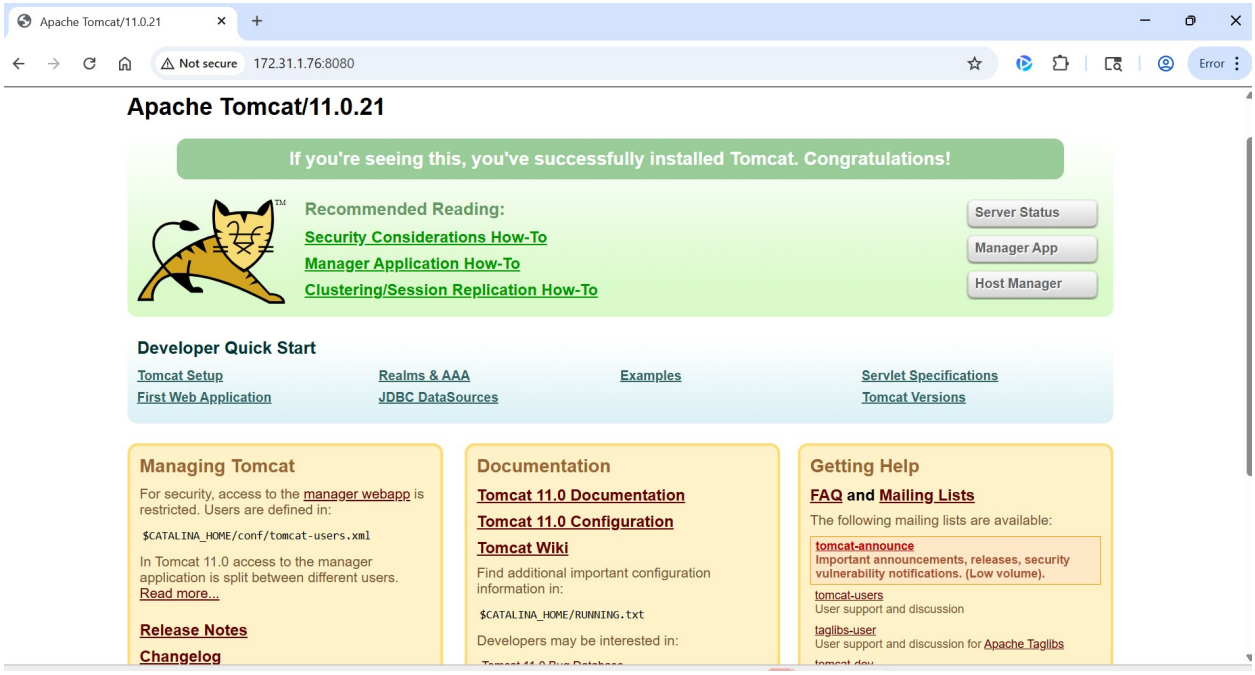


Figure 4 : Browser Output Over Page 8080

6 Java Configuration to Use Utimaco HSM

6.1 Update the java.security File to Use the Utimaco HSM for JDK17

1. Go to the `<JDK_Installation_directory> conf/security` directory.

›_ Console

```
# cd /usr/lib/jvm/java-17-openjdk/conf/security
```

2. Edit the `java.security` configuration file to add the SunPKCS11 provider.

›_ Console

```
security.provider.1=SUN
security.provider.2=SunRsaSign
security.provider.3=SunEC
security.provider.4=SunJSSE
security.provider.5=SunJCE
security.provider.6=SunJGSS
security.provider.7=SunSASL
security.provider.8=XMLDSig
security.provider.9=SunPCSC
security.provider.10=JdkLDAP
security.provider.11=JdkSASL
security.provider.12=SunPKCS11 /etc/utimaco/pkcs11.cfg
```

Specify the correct provider number and path for the `pkcs11.cfg` file.

7 SSL Key & Certificate Generation for Apache Tomcat on Utimaco HSM

7.1 Generating the CA-Signed SSL Certificate

7.1.1 For OpenJDK17 with an RSA Key

1. Generate an RSA keypair on Utimaco HSM.

›_ Console

```
# keytool -genkey -keyalg RSA -keysize 2048 -keystore NONE -storetype PKCS11
storepass 12345678  -providername SunPKCS11-CryptoServer -alias tomcatrsa
```

Provide information when prompted here:

- RSA is the key algorithm
- 2048 is the key size
- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name
- tomcatrsa is the key name that will be generated on Utimaco HSM

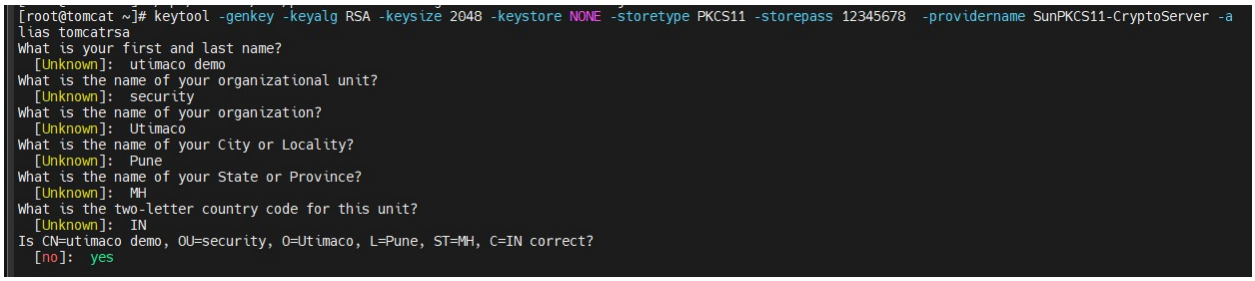


Figure 5 : Key Generation Using Keytool Command

2. Verify that the keys have been generated.

›_ Console

```
# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-
CryptoServer -storepass 12345678 -v
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name

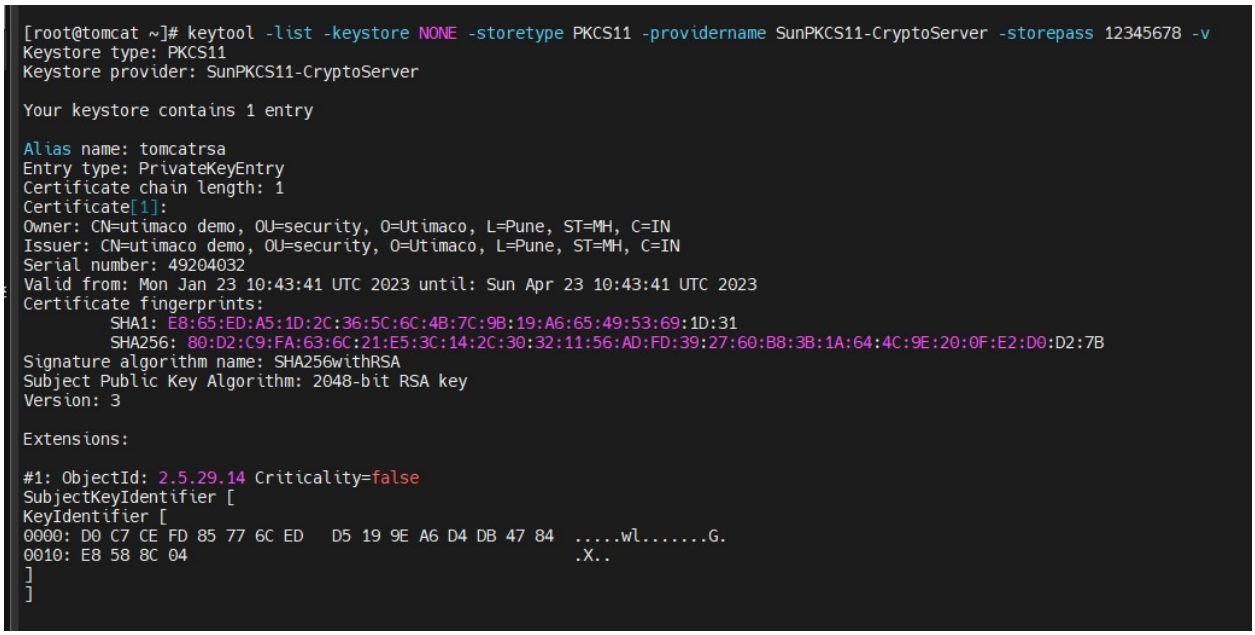


Figure 6 : Listkeys Output

3. List the keys using `p11tool2`.

›_ Console

```
# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=12345678 ListObjects
```

```
[root@tomcat ~]# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=12345678 ListObjects

CKO_CERTIFICATE:

+ 1.1
  CKA_CERTIFICATE_TYPE          = CKC_X_509
  CKA_UNIQUE_ID                 = 3074A9E8-C4EF-4B4F-B614-599D4E6F1FE2
  CKA_LABEL                     = tomcatrsa
  CKA_ID                        =
               0x746F6D63 61747273 61               |tomcatrsa       |

  CKA_SUBJECT                   =
               0x3065310B 30090603 55040613 02494E31 |0e1 0   U    IN1|
                 0B300906 03550408 13024D48 310D300B | 0   U    MH1 0 |
                 06035504 07130450 756E6531 10300E06 |  U    Pune1 0  |
                 0355040A 13075574 696D6163 6F311130 | U    Utimaco1 0|
                 0F060355 040B1308 73656375 72697479 |   U    security|
                 31153013 06035504 03130C75 74696D61 |1 0   U    utima|
                 636F2064 656D6F                     |co demo         |


CKO_PUBLIC_KEY:

+ 2.1
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = 8A22FF77-82ED-428A-B354-F2F343471FA3
  CKA_LABEL                     =
  CKA_ID                        =


CKO_PRIVATE_KEY:

+ 3.1
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = 541F5F6C-AD49-4018-AFAE-8B5AF9F663DD
  CKA_SENSITIVE                 = CK_FALSE
  CKA_EXTRACTABLE               = CK_TRUE
  CKA_LABEL                     =
  CKA_ID                        =
               0x746F6D63 61747273 61               |tomcatrsa       |
```

Figure 7 : List Keys Output Using p11tool2

4. Generate a CSR using the `keytool` command.

›_ Console

```
# keytool -certreq -keystore NONE -storetype PKCS11 -storepass 12345678
providername SunPKCS11-CryptoServer -alias tomcatrsa -file tomcatrsa.csr
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name
- tomcatrsa is the key name
- tomcatrsa.csr is the CSR file name that will be generated

5. Get this CSR signed by the CA.
6. Copy the signed certificate along with the root CA certificate chain on the Tomcat server.
7. Import the signed certificate chain reply using the command below.

›_ Console

```
# keytool -importcert -trustcacerts -alias tomcatrsa -file /root/tomcatrsa.p7b
-storetype PKCS11 -keystore NONE -providername SunPKCS11-CryptoServer storepass
12345678
```

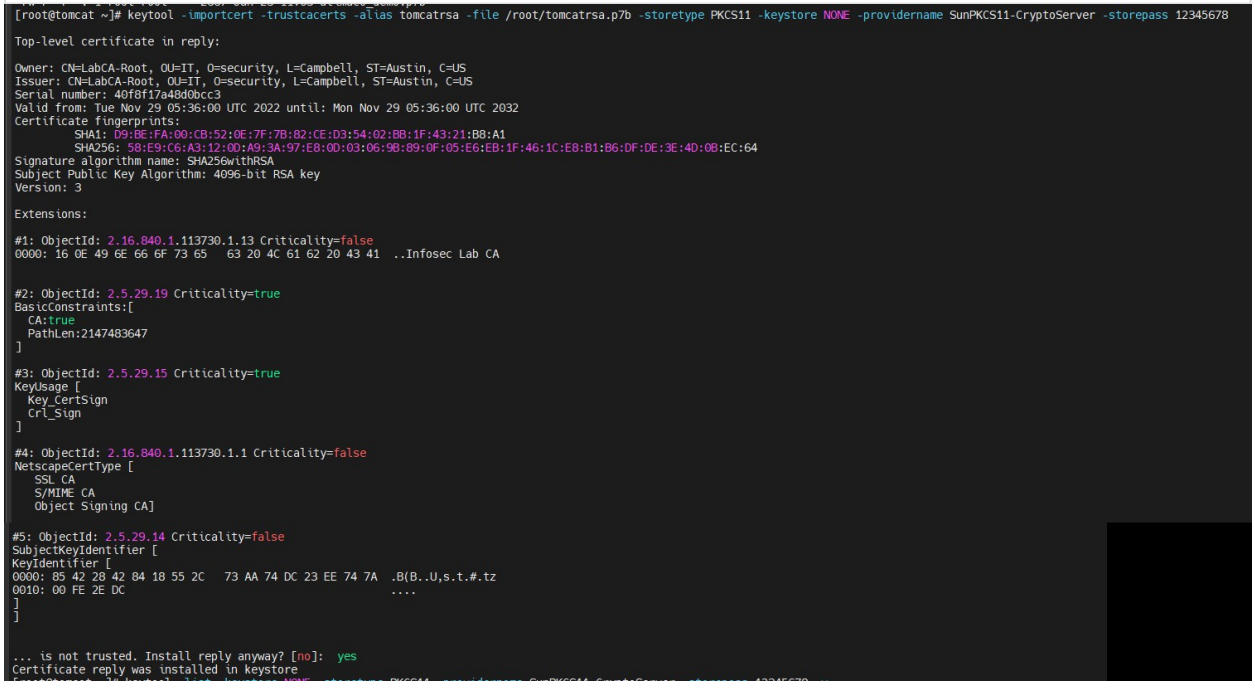


Figure 8 : Import User Certificate Into Keystore

The signed certificate must also contain the certificate chain.

8. Verify that the `keytool` command shows the signed certificate as well as the root CA certificate.

›_ Console

```
# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-
CryptoServer -storepass 12345678 -v
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name

```
[root@tomcat ~]# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-CryptoServer -storepass 12345678 -v
Keystore type: PKCS11
Keystore provider: SunPKCS11-CryptoServer

Your keystore contains 1 entry

Alias name: tomcatrsa
Entry type: PrivateKeyEntry
Certificate chain length: 2
Certificate[1]:
Owner: CN=utimaco demo, OU=security, O=Utimaco, L=Pune, ST=MH, C=IN
Issuer: CN=LabCA-Root, OU=IT, O=security, L=Campbell, ST=Austin, C=US
Serial number: 6a8e2b2c65363ed8
Valid from: Mon Jan 23 10:47:00 UTC 2023 until: Tue Jan 23 10:47:00 UTC 2024
Certificate fingerprints:
         SHA1: 3B:3A:7A:85:84:CA:2A:92:22:A8:39:F0:E8:C7:9D:DE:5D:97:CC:ED
         SHA256: 15:A8:AC:23:A4:F9:BD:8C:62:67:07:F9:1F:2F:0F:A5:64:36:D2:F2:18:63:37:E9:99:C8:C7:A2:84:1A:32:F9
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: D0 C7 CE FD 85 77 6C ED   D5 19 9E A6 D4 DB 47 84  .....wl.......G.
0010: E8 58 8C 04                                        .X..
]
]

Certificate[2]:
Owner: CN=LabCA-Root, OU=IT, O=security, L=Campbell, ST=Austin, C=US
Issuer: CN=LabCA-Root, OU=IT, O=security, L=Campbell, ST=Austin, C=US
Serial number: 40f8f17a48d0bcc3
Valid from: Tue Nov 29 05:36:00 UTC 2022 until: Mon Nov 29 05:36:00 UTC 2032
Certificate fingerprints:
         SHA1: D9:BE:FA:00:CB:52:0E:7F:7B:82:CE:D3:54:02:BB:1F:43:21:B8:A1
         SHA256: 58:E9:C6:A3:12:0D:A9:3A:97:E8:0D:03:06:9B:89:0F:05:E6:EB:1F:46:1C:E8:B1:B6:DF:DE:3E:4D:0B:EC:64
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 4096-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.16.840.1.113730.1.13 Criticality=false
0000: 16 0E 49 6E 66 6F 73 65   63 20 4C 61 62 20 43 41  ..Infosec Lab CA


#2: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
  Key_CertSign
  Crl_Sign
]

#4: ObjectId: 2.16.840.1.113730.1.1 Criticality=false
NetscapeCertType [
   SSL CA
   S/MIME CA
   Object Signing CA]

#5: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 85 42 28 42 84 18 55 2C   73 AA 74 DC 23 EE 74 7A  .B(B..U,s.t.#.tz
0010: 00 FE 2E DC                                        ....
]
]



*******************************************
*******************************************


[root@tomcat ~]#
```

Figure 9 : Keytool List Output

7.1.2 For OpenJDK17 with an EC Key

1. Generate an EC keypair on Utimaco HSM.

›_ Console

```
# keytool -genkey -alias tomssleckey -keyalg EC -keystore NONE -storetype

PKCS11 -storepass 12345678 -providername SunPKCS11-CryptoServer -v
```

Provide information when prompted here:

- EC is the key algorithm
- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name
- tomssleckey is the key name that will be generated on Utimaco HSM

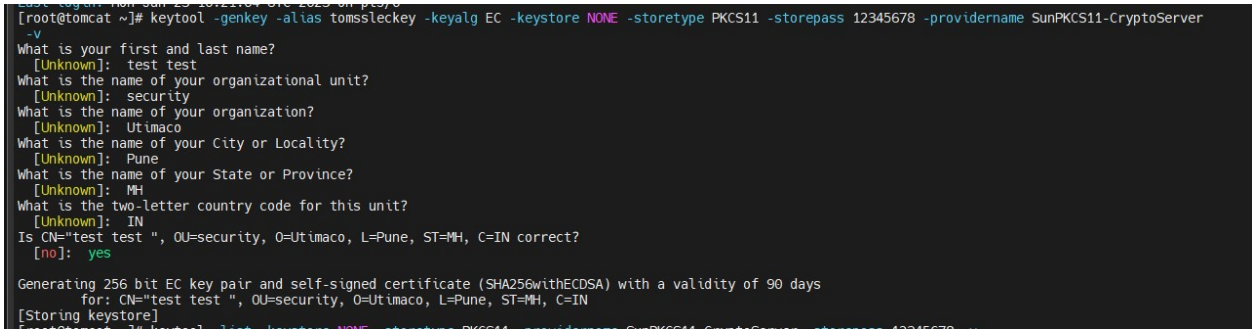


Figure 10 : Key Generation Using the Keytool Command

2. Verify that the keys have been generated.

›_ Console

```
# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-
CryptoServer -storepass 12345678 -v
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype

- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name

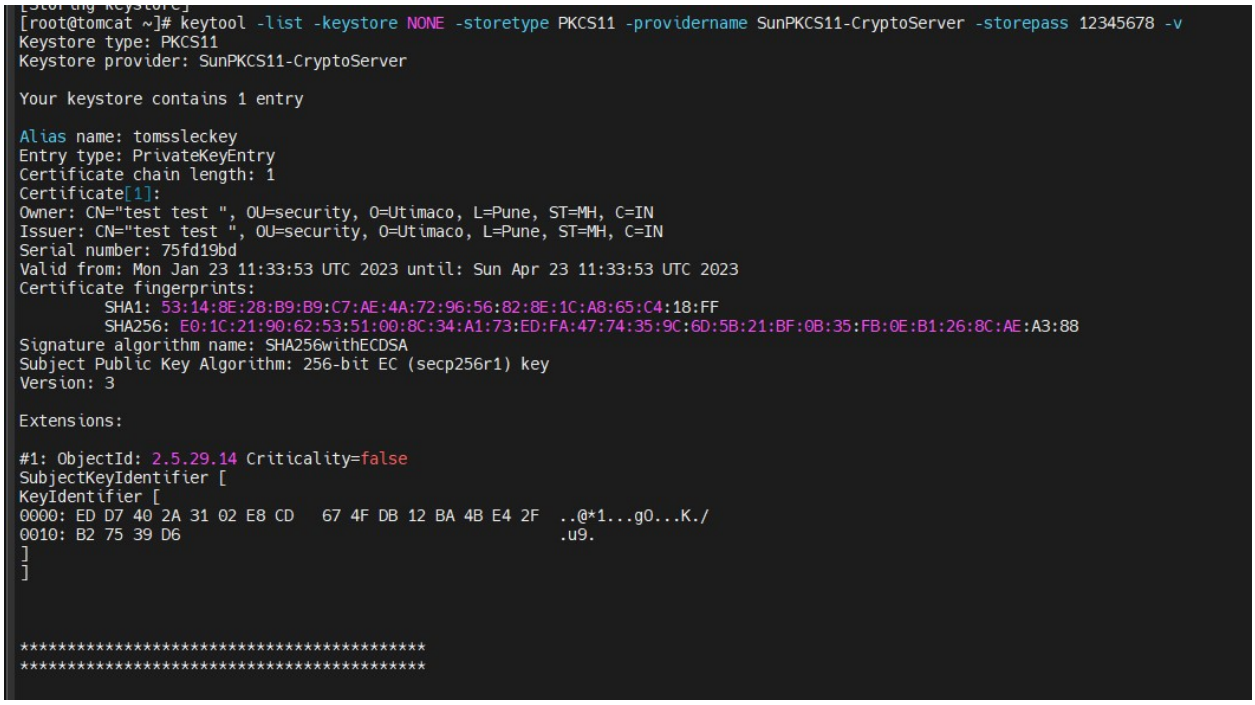


Figure 11 : Listkeys Output

3. List the keys using `p11tool2`.

›_ Console

```
# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=12345678 ListObjects
```

```
[root@tomcat ~]# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=12345678 ListObjects

CKO_CERTIFICATE:

+ 1.1
  CKA_CERTIFICATE_TYPE          = CKC_X_509
  CKA_UNIQUE_ID                 = C8600964-25B8-469F-882C-41DDD865588B
  CKA_LABEL                     = tomssleckey
  CKA_ID                        =
                0x746F6D73 736C6563 6B6579          |tomssleckey     |

  CKA_SUBJECT                   =
                0x3063310B 30090603 55040613 02494E31 |0c1 0   U    IN1|
                  0B300906 03550408 13024D48 310D300B | 0   U    MH1 0 |
                  06035504 07130450 756E6531 10300E06 |  U    Pune1 0  |
                  0355040A 13075574 696D6163 6F311130 | U    Utimaco1 0|
                  0F060355 040B1308 73656375 72697479 |   U    security|
                  31133011 06035504 03130A74 65737420 |1 0   U    test |
                  74657374 20                         |test           |


CKO_PUBLIC_KEY:

+ 2.1
  CKA_KEY_TYPE                  = CKK_ECDSA
  CKA_UNIQUE_ID                 = 4307A902-5708-4DC5-B39E-24308E4A43E8
  CKA_LABEL                     =
  CKA_ID                        =


CKO_PRIVATE_KEY:

+ 3.1
  CKA_KEY_TYPE                  = CKK_ECDSA
  CKA_UNIQUE_ID                 = BA833B57-8882-4CF7-84F3-B1F0C928CEDE
  CKA_SENSITIVE                 = CK_FALSE
  CKA_EXTRACTABLE               = CK_TRUE
  CKA_LABEL                     =
  CKA_ID                        =
                0x746F6D73 736C6563 6B6579          |tomssleckey     |
```

Figure 12 : List Keys Output Using p11tool2

4. Generate a CSR using the `keytool` command.

›_ Console

```
# keytool -certreq -keystore NONE -storetype PKCS11 -storepass 12345678
providername SunPKCS11-CryptoServer -alias tomssleckey -file tomcateckey.csr
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name
- tomssleckey is the key name
- tomcateckey.csr is the CSR file name that will be generated

5. Get this CSR signed by the CA.
6. Copy the signed certificate along with the root CA certificate chain on the Tomcat server.
7. Import the signed certificate chain reply using the command below.

›_ Console

```
 #keytool -importcert -trustcacerts -alias tomssleckey -file

/root/test_test.p7b -storetype PKCS11 -keystore NONE -providername SunPKCS11-
CryptoServer -storepass 12345678
```

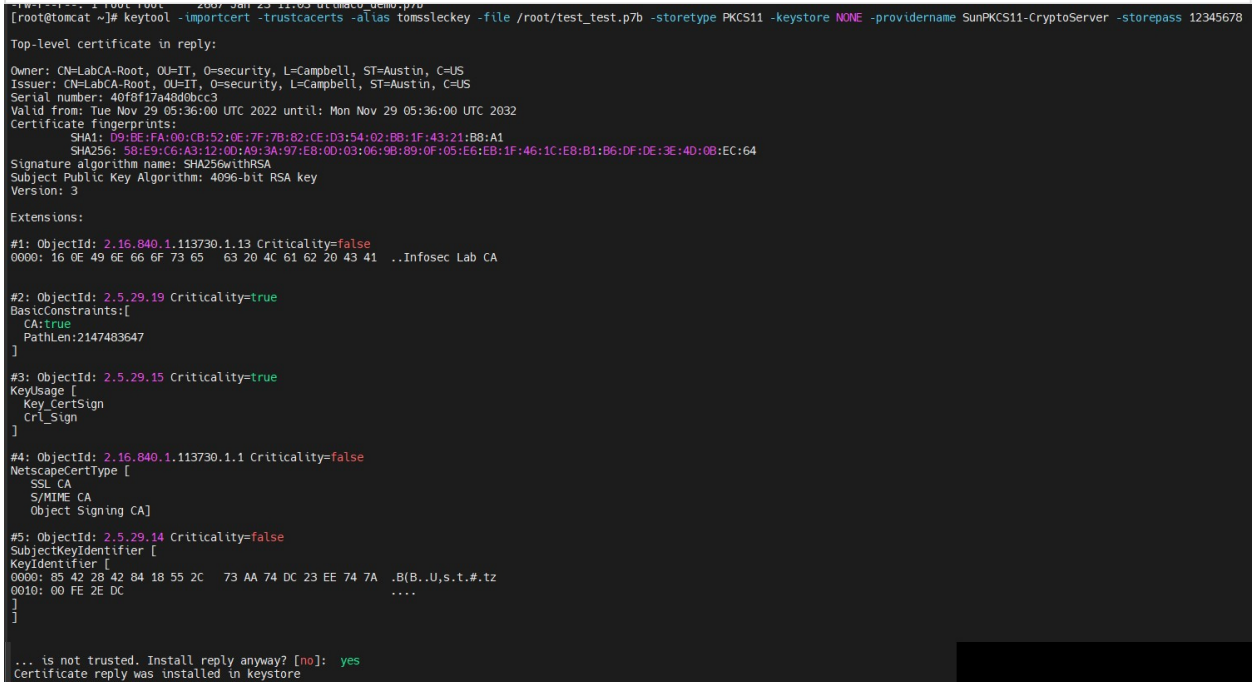


Figure 13 : Signed Certificate Imported

The signed certificate must also contain the certificate chain.

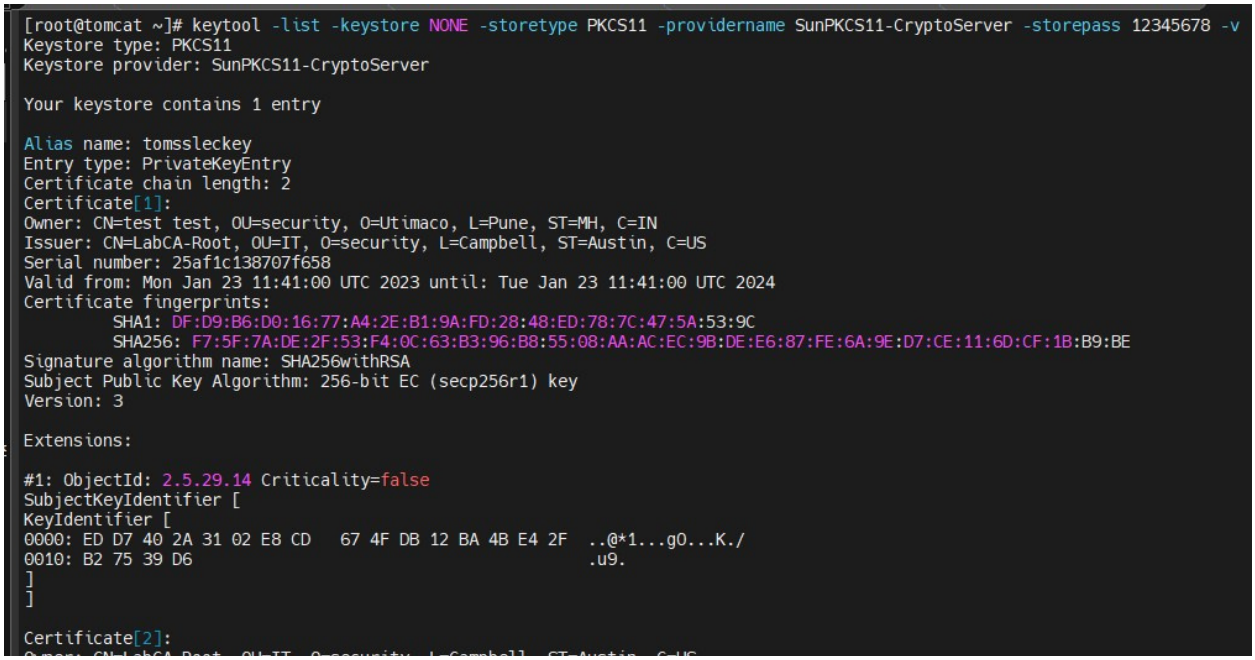
8. Verify that the `keytool` command shows the signed certificate as well as the root CA certificate.

›_ Console

```
# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-
CryptoServer -storepass 12345678 -v
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- SunPKCS11-CryptoServer is the provider name
- 12345678 is the slot PIN



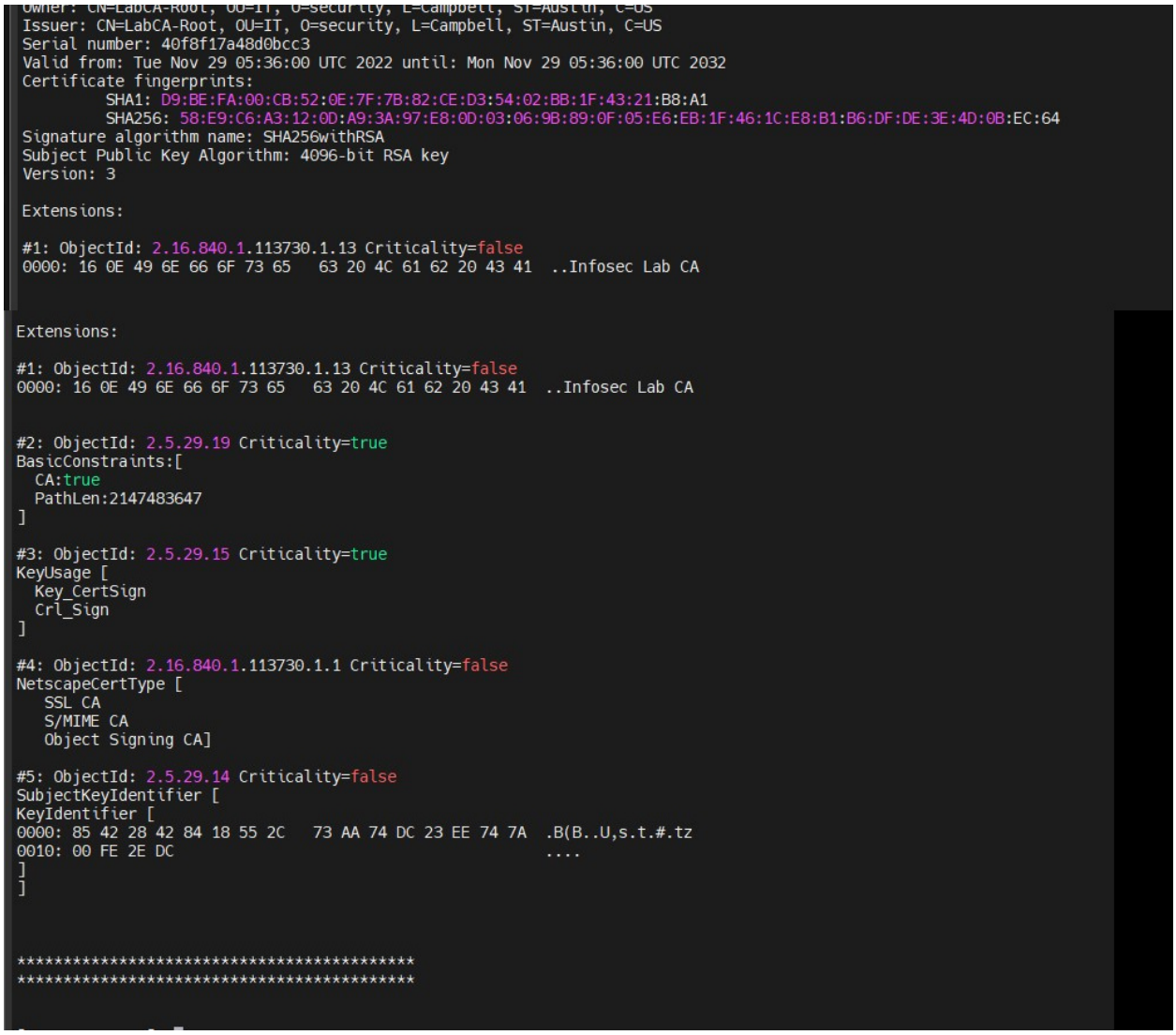


Figure 14 : Keytool List Output

7.2 Using a Self-Signed Certificate

7.2.1 For OpenJDK17 with an RSA Key Using a Self-Signed Certificate

1. Generate an RSA keypair on the Utimaco HSM.

›_ Console

```
# keytool -genkey -keyalg RSA -keysize 2048 -keystore NONE -storetype PKCS11
storepass 12345678  -providername SunPKCS11-CryptoServer -alias tomcatrsa17
```

Provide information when prompted here:

- RSA is the key algorithm
- 2048 is the key size
- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name
- tomcatrsa is the key name that will be generated on the Utimaco HSM

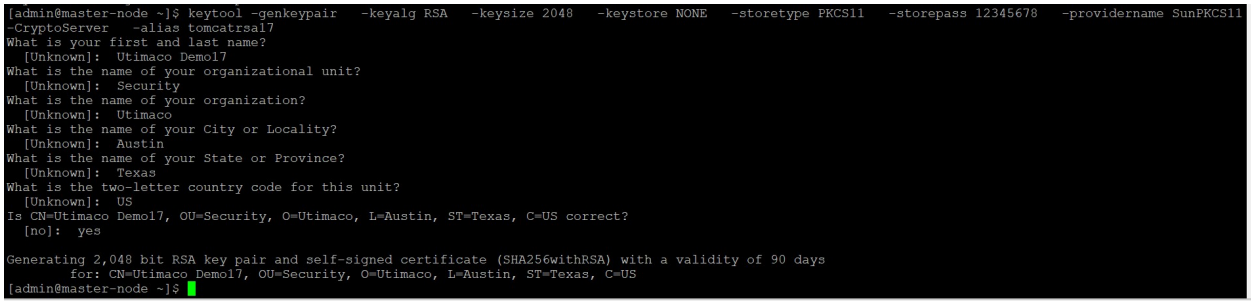


Figure 15 : RSA Key Generation Using a Self-Signed Certificate

It is recommended to use a CA-signed certificate for the production environment.

2. Verify that the keys have been generated by `p11tool2`.

›_ Console

```
# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=<passcode> ListObjects
```

```
[admin@master-node ~]$ /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=12345678 ListObjects


CKO_CERTIFICATE:

+ 1.1
  CKA_CERTIFICATE_TYPE          = CKC_X_509
  CKA_UNIQUE_ID                 = 537F67D4-9404-42BF-BE48-8ED4AEB41F18
  CKA_LABEL                     = tomcatrsa17
  CKA_ID                        =
               0x746F6D63 61747273 613137           |tomcatrsa17     |

  CKA_SUBJECT                   =
               0x306C310B 30090603 55040613 02555331 |011 0   U    US1|
                 0E300C06 03550408 13055465 78617331 | 0   U    Texas1|
                 0F300D06 03550407 13064175 7374696E | 0   U    Austin|
                 3110300E 06035504 0A130755 74696D61 |1 0   U    Utima|
                 636F3111 300F0603 55040B13 08536563 |co1 0   U    Sec|
                 75726974 79311730 15060355 0403130E |urity1 0   U    |
                 5574696D 61636F20 44656D6F 3137     |Utimaco Demo17  |
```

Figure 16 : Certificate Details

```
CKO_PRIVATE_KEY:

+ 2.1
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = C6BD433C-CD81-4CD5-B22A-5718A7DAA8B4
  CKA_SENSITIVE                 = CK_TRUE
  CKA_EXTRACTABLE               = CK_FALSE
  CKA_LABEL                     =
  CKA_ID                        =
               0x746F6D63 61747273 613137           |tomcatrsa17     |
```

Figure 17 : Key Details

3. List the keys using the `keytool` command.

›_ Console

```
# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-
CryptoServer -storepass 12345678 -v
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- SunPKCS11-CryptoServer is the provider name
- 12345678 is the slot PIN

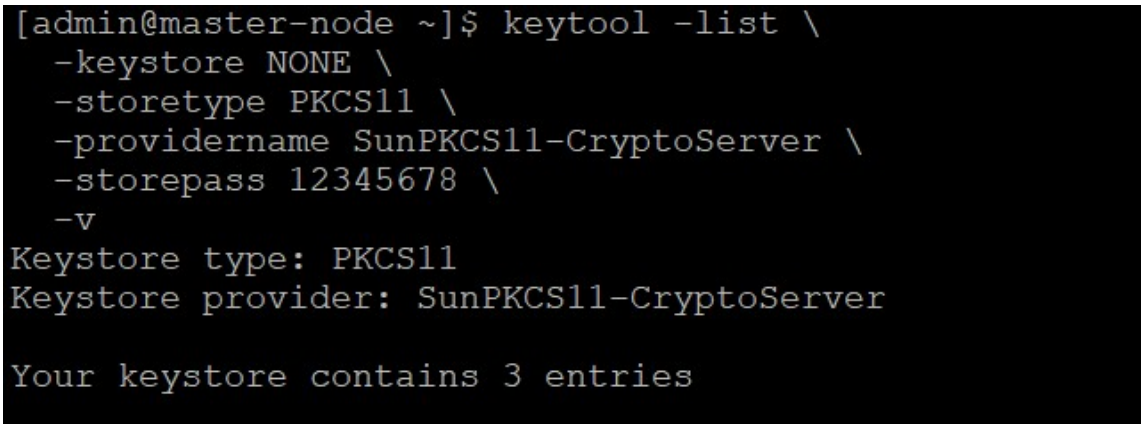


Figure 18 : Keytool List Command

```
*******************************************
*******************************************


Alias name: tomcatrsa17
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Utimaco Demo17, OU=Security, O=Utimaco, L=Austin, ST=Texas, C=US
Issuer: CN=Utimaco Demo17, OU=Security, O=Utimaco, L=Austin, ST=Texas, C=US
Serial number: 5ff940066936bc14
Valid from: Thu Apr 16 14:46:21 PDT 2026 until: Wed Jul 15 14:46:21 PDT 2026
Certificate fingerprints:
         SHA1: FD:04:A5:DB:C2:B1:74:85:29:17:16:7D:09:32:BB:1F:B4:57:A6:B9
         SHA256: 76:41:44:2A:48:9F:BC:C4:27:E5:27:20:44:B8:C9:35:26:59:23:14:E4:10:84:87:C2:81:12:F7:2B:ED:81:33
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: D3 30 F1 C3 5C 18 EC 7C   05 60 B9 75 6E DC FB F1  .0..\....`.un...
0010: 4D A4 EA 5C                                        M..\
]
]


*******************************************
*******************************************
```

Figure 19 : Keytool List Output

7.2.2 For OpenJDK17 with an EC Key Using a Self-Signed Certificate

1. Generate an EC keypair on the Utimaco HSM.

›_ Console

```
# keytool -genkey -alias tomssleckey -keyalg EC -keystore NONE -storetype
PKCS11 -storepass 12345678 -providername SunPKCS11-CryptoServer -v
```

Provide information when prompted here:

- EC is the key algorithm
- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN
- SunPKCS11-CryptoServer is the provider name

- tomssleckey is the key name that will be generated on the Utimaco HSM

```
[admin@master-node ~]$
keytool -genkeypair \
  -alias tomssleckey17 \
  -keyalg EC \
  -keysize 256 \
  -keystore NONE \
  -storetype PKCS11 \
  -storepass 12345678 \
  -providername SunPKCS11-CryptoServer \
  -v

Warning:
Specifying -keysize for generating EC keys is deprecated, please use "-groupname secp256r1" instead.

What is your first and last name?
  [Unknown]:  Utimaco DemoEC17
What is the name of your organizational unit?
  [Unknown]:  Security
What is the name of your organization?
  [Unknown]:  Utimaco
What is the name of your City or Locality?
  [Unknown]:  Plano
What is the name of your State or Province?
  [Unknown]:  Texas
What is the two-letter country code for this unit?
  [Unknown]:  US
Is CN=Utimaco DemoEC17, OU=Security, O=Utimaco, L=Plano, ST=Texas, C=US correct?
  [no]:  yes

Generating 256 bit EC key pair and self-signed certificate (SHA256withECDSA) with a validity of 90 days
        for: CN=Utimaco DemoEC17, OU=Security, O=Utimaco, L=Plano, ST=Texas, C=US
[Storing keystore]
[admin@master-node ~]$
```

Figure 20 : Keytool Command to Generate Keys

It is recommended to use a CA-signed certificate for the production environment.

2. Verify that the keys have been generated by `p11tool2`.

›_ Console

```
# /opt/utimaco/bin/p11tool2 Slot=0 LoginUser=<passcode> ListObjects
```

```
+ 2.4
  CKA_KEY_TYPE                    = CKK_ECDSA
  CKA_UNIQUE_ID                   = E535EFF4-48EC-46D3-ABE5-591AA189E4DC
  CKA_SENSITIVE                   = CK_TRUE
  CKA_EXTRACTABLE                 = CK_FALSE
  CKA_LABEL                       =
  CKA_ID                          =
                0x746F6D73 736C6563 6B657931 37       |tomssleckey17   |
```

Figure 21 : Created Key Details

```
+ 1.5
  CKA_CERTIFICATE_TYPE            = CKC_X_509
  CKA_UNIQUE_ID                   = 776DE1B7-81F7-4757-A6DF-72D88C130BD9
  CKA_LABEL                       = tomssleckey17
  CKA_ID                          =
                0x746F6D73 736C6563 6B657931 37       |tomssleckey17   |

  CKA_SUBJECT                     =
                0x306D310B 30090603 55040613 02555331 |0m1 0   U    US1|
                  0E300C06 03550408 13055465 78617331 | 0   U    Texas1|
                  0E300C06 03550407 1305506C 616E6F31 | 0   U    Plano1|
                  10300E06 0355040A 13075574 696D6163 | 0   U    Utimac|
                  6F311130 0F060355 040B1308 53656375 |o1 0   U    Secu|
                  72697479 31193017 06035504 03131055 |rity1 0   U    U|
                  74696D61 636F2044 656D6F45 433137   |timaco DemoEC17 |
```

Figure 22 : List Keys Output Using p11tool2

3. List the keys using the `keytool` command.

›_ Console

```
# keytool -list -keystore NONE -storetype PKCS11 -providername SunPKCS11-
CryptoServer -storepass 12345678 -v
```

Here:

- NONE is the keystore for HSM
- PKCS11 is the storetype
- 12345678 is the slot PIN

- SunPKCS11-CryptoServer is the provider name

```
[admin@master-node ~]$ keytool -list \
  -keystore NONE \
  -storetype PKCS11 \
  -providername SunPKCS11-CryptoServer \
  -storepass 12345678 \
  -v
Keystore type: PKCS11
Keystore provider: SunPKCS11-CryptoServer

Your keystore contains 4 entries
```

Figure 23 : Keytool Command to List Keys

```
*******************************************
*******************************************


Alias name: tomssleckey17
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Utimaco DemoEC17, OU=Security, O=Utimaco, L=Plano, ST=Texas, C=US
Issuer: CN=Utimaco DemoEC17, OU=Security, O=Utimaco, L=Plano, ST=Texas, C=US
Serial number: 2efe33a44fa866b1
Valid from: Thu Apr 16 15:07:55 PDT 2026 until: Wed Jul 15 15:07:55 PDT 2026
Certificate fingerprints:
         SHA1: B4:16:70:56:38:55:3B:F9:D4:44:89:5C:2C:01:DA:88:16:FA:57:68
         SHA256: 7A:B9:6F:B6:96:79:FB:0F:EE:BC:77:93:48:A8:E8:73:86:73:BD:32:63:29:20:EA:59:1A:89:17:C4:F8:40:3A
Signature algorithm name: SHA256withECDSA
Subject Public Key Algorithm: 256-bit EC (secp256r1) key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: DD 80 82 E2 84 7B F6 B7   13 E0 B5 63 8D F6 49 58  ...........c..IX
0010: 39 65 B9 AD                                        9e..
]
]
```

Figure 24 : Keytool List Output

7.3 Update the server.xml File for the SSL Configuration

1. Open the `server.xml` file.

›_ Console

```
# vi /opt/tomcat/conf/server.xml
```

2. Add the following entries to the `connector` section for SSL.

›_ Console

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
           maxThreads="150" SSLEnabled="true">
        <UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />
        <SSLHostConfig>
            <Certificate certificateKeystoreFile=""
                  certificateKeystoreType="PKCS11"
                  certificateKeystoreProvider="SunPKCS11-CryptoServer"
                  certificateKeyAlias=" tomcatsslkey "
                  certificateKeystorePassword="123456"
                  type="RSA" />
        </SSLHostConfig>
    </Connector>
```

Here:

- certificateKeystoreFile is blank as the HSM is being used
- certificateKeystoreType is pkcs11 as the keystore is being used
- certificateKeystoreProvider is SunPKCS11-CryptoServer
- certificateKeyAlias is the name of the key generated using the `keytool` command
- certificateKeystorePassword is the password of the HSM keystore
- type is the key algorithm to use (RSA/EC)

3. Reload the daemon using:

›_ Console

```
# systemctl daemon-reload
```

4. Restart the Tomcat service using:

›_ Console

```
# systemctl restart tomcat
```

5. Confirm that the Tomcat status is `running` using:

›_ Console

```
# systemctl status tomcat
```

6. The below output shows it is running:

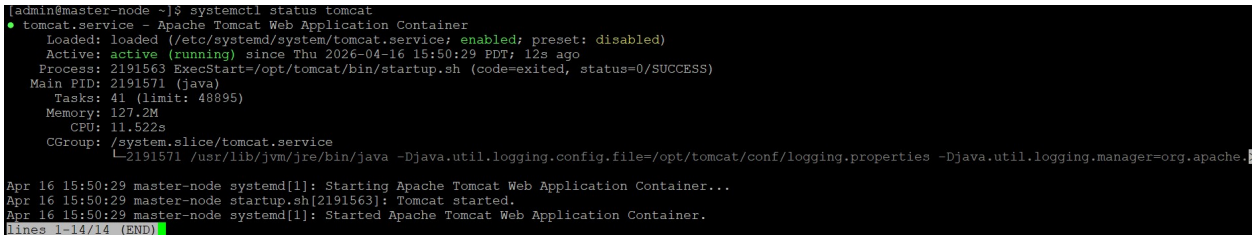


Figure 25 : Tomcat Service Status Output

7. Now access the page over https using *https://<ip>:8443*.

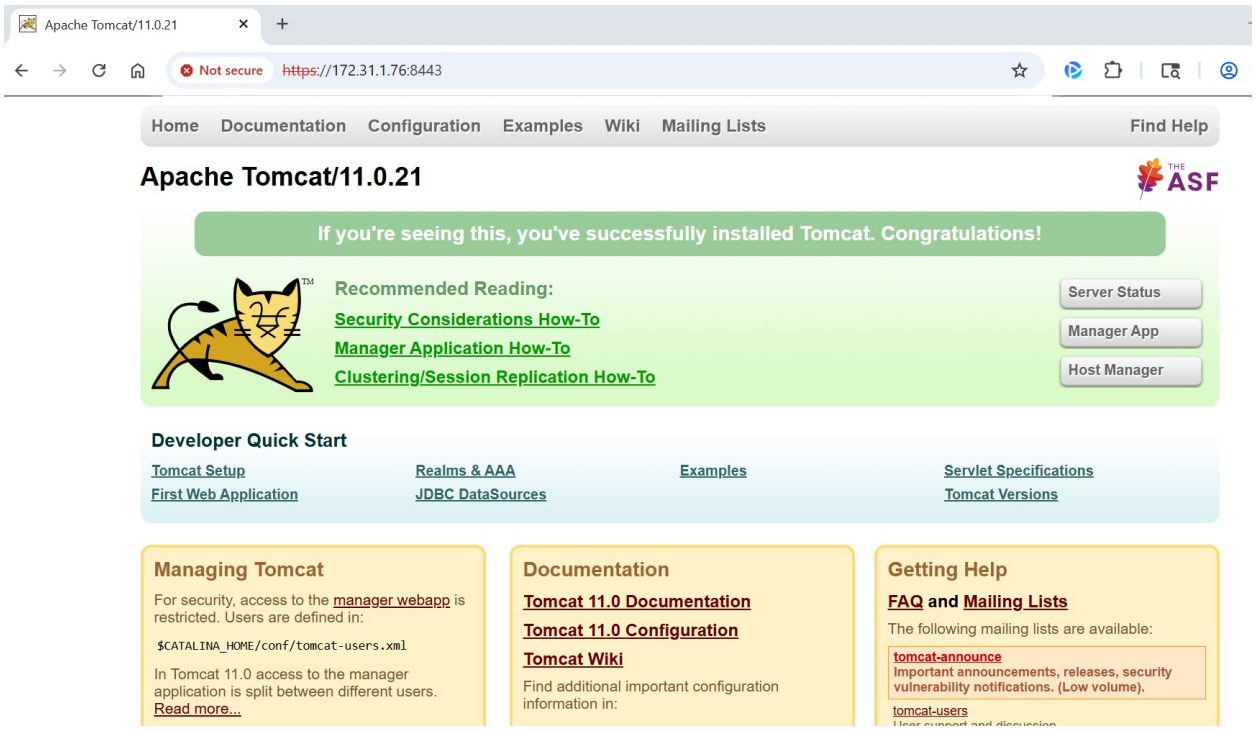


Figure 26 : Tomcat Service Status Output - Browser

This completes the integration of Apache Tomcat with the Utimaco HSM using the SunPKCS11 security provider.

8 Troubleshooting

Error	Diagnosis
LoginUser= failed: 05.12.2021 23:45:45 src/p11adm_R2.c[429] p11_login: C_Login [type=1] returned Error 0x00000102 (CKR_USER_PIN_NOT_INITIALIZED)	PKCS#11 Slot is not initialized.
The CryptoServer PKCS#11 Library R3 is not initialized. Error CKR_CRYPTOKI_NOT_INITIALIZED occurred	PKCS#11 Slot is not initialized.

Table 6: List of Errors and their Diagnoses

9 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 u.trust GP HSM product documentation is also available at the Utimaco IS GmbH website: https://utimaco.com/.

10 Contact and Support Information

You can reach us from Monday to Friday, 09.00 a.m. to 05.00 p.m., Central European Time (CET).

Utimaco IS GmbH
Krefelder Str. 220
52070 Aachen
Germany

RMA Query

If you need to send the device back to Utimaco IS GmbH, please open a new RMA case (Return Merchandise Authorization). We request that you use the following web address. RMA cases cannot be opened by email or phone.

https://support.hsm.utimaco.com/support/rma/new

Other Support Queries

- Mail (preferred contact method)
  support@utimaco.com
  Attach the diagnostic information to your email.
- Web portal
  https://support.hsm.utimaco.com/support/cases/new/
  The diagnostic information will be requested in our response if necessary.
- By phone
  AMERICAS +1-844-UTIMACO (+1 844-884-6226)
  EMEA +49 800-627-3081
  APAC +81 800-919-1301
  The diagnostic information will be requested in our response if necessary.

11 Appendices

11.1 References

Reference	Title/Company	Document No.
[CSADMIN]	CryptoServer – csadm Manual/Utimaco IS GmbH	2009-0003
[CSTrSh]	CryptoServer Troubleshooting/Utimaco IS GmbH	M011-0008-en
[CSADMIN2]	CryptoServer_csadm_Manual_Systemadministrators.pdf	2009-0003
[CSP11Tool2]	CryptoServer_p11tool2_Manual.pdf	2012-0004
[CSPKCSM]	CryptoServer - PKCS#11 P11CAT Manual	M013-0001-en
[CSLAN5]	CryptoServerLAN_Manual_Systemadministrators.pdf	2018-0004

Table 7: References

11.2 Command Summary

Commands Used	Purpose
`dnf -y update`	To update system packages.
`dnf -y install java-17-openjdk java-17-openjdk-devel`	To install OpenJDK 17.
`useradd tomcat`	To create a non-root Tomcat user.

Commands Used	Purpose
`mkdir -p /opt/tomcat`	To create a Tomcat installation directory.
`tar -xvf apache-tomcat-11.0.21.tar.gz -C /opt/tomcat --strip-components=1`	To extract Tomcat binaries.
`chown -R tomcat:tomcat /opt/tomcat`	To set ownership to a Tomcat user.
`chmod +x /opt/tomcat/bin/*.sh`	To allow execution of Tomcat scripts.
`mkdir -p /opt/utimaco/bin /opt/utimaco/lib`	To create Utimaco directories.
`cp libcs_pkcs11_R3.so /opt/utimaco/lib`	To ready Place PKCS#11 library.
`vi /etc/utimaco/pkcs11.cfg`	To configure the PKCS#11 provider.
`cd /usr/lib/jvm/java-17-openjdk/conf/security`	To navigate to the `java.security` location.
`keytool -genkey -keyalg RSA -keystore NONE -storetype PKCS11`	To generate an RSA key in the HSM.
`keytool -genkey -keyalg EC -keystore NONE -storetype PKCS11`	To generate an EC key in the HSM.

Commands Used	Purpose
`keytool -certreq -keystore NONE -storetype PKCS11`	To generate a certificate signing request.
`keytool -importcert -keystore NONE -storetype PKCS11`	To import a signed certificate.
`systemctl restart tomcat`	To restart the Tomcat service.

Table 8: List of Commands Used