Apache HTTP Server

httpd-2.4.66

Integration Guide

SecurityServer

v6.3.0

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	3.0.0
Date	2026-03-30
Status	PUBLISHED
Document No.	IG-2025-0001

Table of Contents

1 Introduction

This guide is part of the information and support provided by Utimaco. Additional documents produced to support your Utimaco SecurityServer product can be found in the document directory of the Utimaco SecurityServer product bundle. All Utimaco SecurityServer product documentation are available on Utimaco's web site at https://utimaco.com/.

1.1 About This Guide

This guide provides an integration explaining how to integrate an Utimaco CryptoServer Hardware Security Module (HSM) with Apache HTTP Server. Utimaco HSM is used to secure the private keys for SSL certificate and offload cryptographic operations on the HSM.

1.2 Target Audience for This Guide

This guide is intended for administrators of Apache HTTP Server and of Utimaco HSMs.

1.3 Purpose of this Integration

The purpose of this document is to describe the integration of Apache HTTP Server with an Utimaco CryptoServer Hardware Security Module (HSM). It provides guidance on configuring Apache to securely store and use SSL/TLS private keys within the HSM and to offload cryptographic operations from the web server to the HSM.

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	Press the **OK** button.

Convention	Use	Example
`Monospaced`	File names, folder and directory names, commands, file outputs, programming code samples	You will find the file example.conf in the `/exmp/demo/` directory.
Italic	References and important terms	See Chapter 3, "Sample Chapter", in the *CryptoServer - csadm Manual* or [CSADMIN].

Table 1: Document conventions

Special icons are used to highlight the most important notes and information.

Here you find important safety information that should be followed.

Here you 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

Abbreviation	Meaning
CSADM	CryptoServer Command-line Administration Tool
CSR	Certificate Signing Request
GUI	Graphical User Interface
HSM	Hardware Security Module
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
LAN	Local Area Network
PCIe	PCI Express Interface
PIN	Personal Identification Number
PKCS#11	Public-Key Cryptography Standard #11
RSA	Rivest-Shamir-Adleman
SO	Security Officer
SSL	Secure Socket Layer
TLS	Transport Layer Security

Abbreviation	Meaning
URL	Uniform Resource Locator

Table 2: List of abbreviations

2 Overview

2.1 Apache HTTP Server

Apache HTTP Server is one of the most popular WebServer across the globe. Apache is very flexible for use and is very handy for Website Owners, Developers, Hosting Providers.

Apache HTTP web server has modules which add more functions to its software, such as mod_ssl for enabling SSL v3 and TLS support.

2.2 Utimaco SecurityServer HSM

CryptoServer is a hardware security module developed by Utimaco IS GmbH. CryptoServer 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.
This guide assumes that the user has already installed and configured required software.

3.1 Tested Versions

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

Operating System	Apache	Utimaco Security Server Version	Utimaco HSM
RHEL 9.6	httpd-2.4.66	SecurityServer 6.3.0 p11tool2 from product package Utimaco SecurityServer	SecurityServer CSe-Series/Se-Series

Table 3: List of tested versions

3.2 Software Requirements

Software	Software Requirements
OpenSSL	OpenSSL 3.5.1
Apache HTTP Server	httpd-2.4.66
Libp11 (Linux)	0.4.18
HSM Interface	SecurityServer PKCS#11 Provider

Table 4: List of software requirements

3.3 Hardware Requirements

Hardware	Hardware Requirements
Utimaco LAN HSM	CryptoServer CSe-Series/Se-Series LAN with firmware SecurityServer 6.3.0
Utimaco PCI-e HSM	CryptoServer CSe-Series/Se-Series PCI-e with firmware SecurityServer 6.3.0

Table 5: List of hardware requirements

Setup 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:

- Installed / configured CryptoServer. Refer to the CryptoServer documentations to setup the HSM.
- Replaced the CryptoServer Default Admin with a new admin user.
- Created and stored the MBK onto each HSM. Refer to the CryptoServer documentations to setup the MBK.
- Installed / set up the operating system listed in Tested Versions.
- Installed / set up SecurityServer as listed in Tested Versions.
- Set up a user with admin privileges. This is required for installing few packages on to the Apache HTTP server.
- Familiarized yourself with the Apache HTTP and OpenSSL documents and setup process.
- Allowed port 80 and 443 through Firewall.

4 Installing and Configuring Utimaco SecurityServer Software

4.1 Download and Install 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 HTTP Server.
2. Create `utimaco` folder under `/opt` directory and further create 2 directories.

`/opt/utimaco/bin and /opt/utimaco/lib`

```
# 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 and make the file executable.

```
# 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.

```
# cd ~/path_to_application_folder
# cp csadm p11tool2 /opt/utimaco/bin
# chmod +x /opt/utimaco/bin/csadm /opt/utimaco/bin/p11tool2
```

4.2 CryptoServer 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.

```
# 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.

```
[Global]

# For unix:

Logpath = /tmp

# Loglevel (0 = NONE; 1 = ERROR; 2 = WARNING; 3 = INFO; 4 = TRACE)

 Logging = 1

Keepalive = true

MultiInitReturnsCKR_OK = 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 Utimaco CryptoServer documentation. The device may be a CryptoServer (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 should 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.

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

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

Figure 1 : Slot initialization

5 Integrating Apache HTTP Server with Utimaco HSM

5.1 Installing OpenSSL

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

```
For RHEL

# dnf upgrade
```

If you are using existing or pre-installed openssl then skip step 2, 3, 4, and 5.

2. Install dependent packages for openssl.

```
For RHEL

# dnf install make gcc perl pcre-devel zlib-devel
```

3. Download the latest version of openssl on Linux machine from *https://www.openssl.org*.

```
# wget https://www.openssl.org/source/openssl-3.5.1.tar.gz
```

4. Extract the downloaded file.

```
# tar xvf openssl-3.5.1.tar.gz
```

5. Go to openssl directory and run the following commands to build and install openssl.

```
# cd openssl-3.5.1
# ./config --prefix=/usr/local/openssl
# make
# make test
# make install
# export LD_LIBRARY_PATH=/usr/local/openssl/lib:$LD_LIBRARY_PATH
# export PATH=/usr/local/openssl/bin:$PATH
```

```
# openssl version -a
```

Figure 2 : OpenSSL version output

After rebooting the server, export the library path every time.

```
# export LD_LIBRARY_PATH=/usr/local/openssl/lib:$LD_LIBRARY_PATH  # export PATH=/usr/local/openssl/bin:$PATH
```

5.2 Installing Libp11

1. Download the latest libp11 package from Releases · OpenSC/libp11 · GitHub.

```
# wget https://github.com/OpenSC/libp11/releases/download/libp110.4.12/libp11-0.4.18.tar.gz
```

2. Extract the file.

```
# tar -xvf libp11-0.4.18.tar.gz
```

3. Go to libp11 directory, build and install libp11using the following commands.

```
# cd libp11-0.4.18
# ./configure OPENSSL_CFLAGS="-I/usr/local/openssl/include/openssl"

OPENSSL_LIBS="-L/usr/local/openssl/lib -lcrypto" prefix="/usr/local/libp11/"
```

```
# make
# make install # export LD_LIBRARY_PATH=/usr/local/openssl/lib: /usr/local/libp11/
lib/:$LD_LIBRARY_PATH
```

If you are using existing or pre-installed openssl then change the value of `OPENSSL_CFLAGS` and `OPENSSL_LIBS` to their correct path.

Make a note of " `Engine Directory` " path while running the configure command as the pkcs11.so file is generated inside this directory after running "make install" command.

After rebooting the server, export the library path every time.

`# export LD_LIBRARY_PATH=/usr/local/openssl/lib:`

`/usr/local/libp11/lib/:$LD_LIBRARY_PATH and # export PATH=/usr/local/openssl/bin:$PATH`

5.3 Configuring OpenSSL to Use Utimaco HSM

5.3.1 Setting Up Utimaco CryptoServer Library in OpenSSL Configuration File

1. Open the file `/usr/local/openssl/ssl/openssl.cnf` and enter the following line in the first line of the file.

```
openssl_conf = openssl_init
```

2. Enter the following lines under last section of `openssl.cnf` file.

```
[openssl_init] engines=engine_section
[engine_section] pkcs11 = pkcs11_section
[pkcs11_section] engine_id = pkcs11
dynamic_path = /usr/local/libp11/lib/pkcs11.so MODULE_PATH = /opt/utimaco/lib/
libcs_pkcs11_R3.so init = 0
```

Dynamic path and Module path will get changed according to the user environment.

5.3.2 Verify PKCS#11 Engine

Run the command below to verify the OpenSSL Engine is available or not.

```
# openssl engine pkcs11 -t
```

Figure 3 : Verification of pkcs11 engine

5.4 Install Apache HTTP Server

1. Install the dependent packages for Apache HTTP server.

```
# dnf install apr apr-devel apr-util apr-util-devel expat pcre pcre-devel make gcc
perl -y
```

2. Download the Apache Open-Source Library files from Apache Website.

```
# wget https://dlcdn.apache.org/httpd/httpd-2.4.56.tar.gz
```

If you are installing through `dnf install httpd` command, then skip step 3 and 4.

3. Extract the downloaded packages.

```
# tar -xzvf httpd-2.4.54.tar.gz
```

4. Go to httpd-2.4.54 directory, build and install Apache HTTP using the following commands.

```
# cd httpd-2.4.54.tar.gz

#./configure  --prefix=/usr/local/apache --enable-ssl --withssl=/usr/local/openssl

# make

# make install
```

5. Start the Apache HTTP Service with below command. If you have installed from source code:

```
# /usr/bin/apache/bin/apachectl -k start
```

If you have installed from dnf:

```
# systemctl start httpd.service
```

6. Open any browser and run the http://<apache_server_ip> to verify that apache service is running successfully.

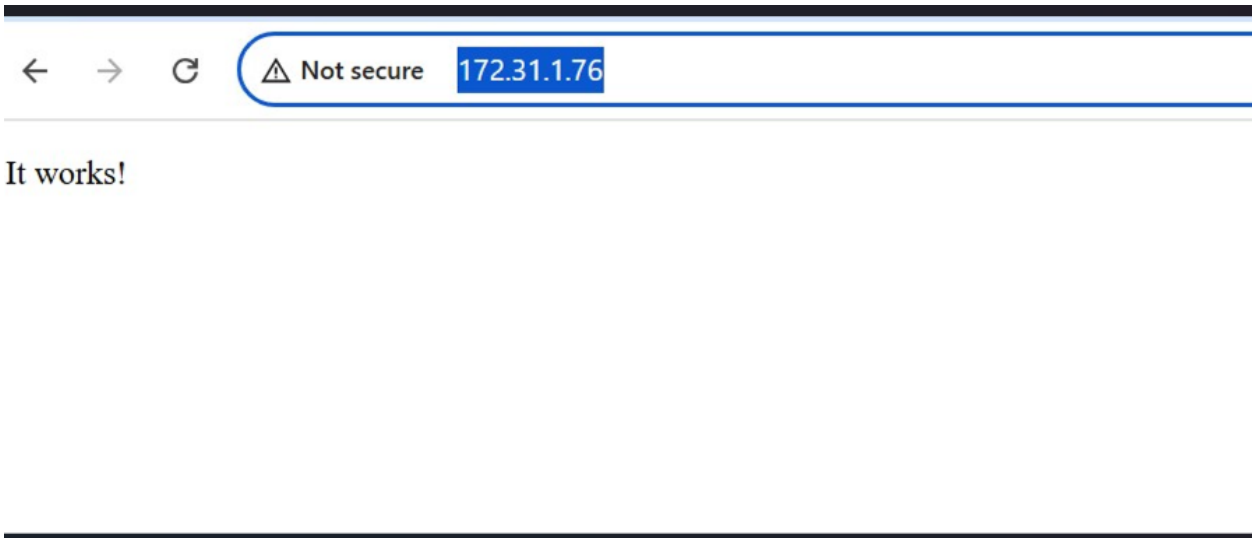

Figure 4 : Apache web page

5.5 Generate Keys and Certificate for SSL

1. Generate the RSA key-pair using p11tool2.

```
# p11tool2 slot=<slot_no>  LoginUser=<cryptouser_password>
PubKeyAttr=CKA_LABEL="RSAKey",CKA_ID=0x45
 PrvKeyAttr=CKA_LABEL="RSAKey",CKA_ID=0x45  GenerateKeyPair=RSA
```

2. Verify that the keys are generated onto the HSM using following command.

```
# p11tool2 slot=<slot_no> LoginUser=<cryptouser_password> ListObjects
```

```
[admin@master-node bin]$ ./p11tool2 LoginUser=12345678 ListObjects


CKO_PUBLIC_KEY:

+ 1.1
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = 77CA304E-2759-419A-993A-65E441BA49D1
  CKA_LABEL                     = rsa_public_key
  CKA_ID                        = 0x3031 (01)

+ 1.2
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = 1594064D-17FA-4E0E-99A0-F78328843E3D
  CKA_LABEL                     = RSAKey
  CKA_ID                        = 0x45 (E)


CKO_PRIVATE_KEY:

+ 2.1
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = 67229BDA-4098-4437-8457-9621A5679832
  CKA_SENSITIVE                 = CK_TRUE
  CKA_EXTRACTABLE               = CK_FALSE
  CKA_LABEL                     = RSAKey
  CKA_ID                        = 0x45 (E)
```

Figure 5 : List objects output

3. Generate a certificate request.

```
# openssl req -engine pkcs11 -new -key "pkcs11:token=HTTPSERVER;object=RSAKey"
 -keyform engine -out apache.csr
```

Here `HTTPSERVER` is the token label and `RSAKey` is the key on the HSM. Provide `Cryptouser PIN` when prompted. `apache.csr` is the certificate signing request file. Also provide other required information for certificate when prompted.

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:CA
Locality Name (eg, city) [Default City]:Cambell
Organization Name (eg, company) [Default Company Ltd]:Utimaco
Organizational Unit Name (eg, section) []:Security
Common Name (eg, your name or your server's hostname) []:Utim
Email Address []:test@utimc.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Figure 6 : Generate CSR certificate output

4. Get this CSR signed by your CA and copy the signed certificate to Apache server.
5. Alternatively, you can create the self-signed certificate based on the generated key.

```
# openssl req -engine pkcs11 -new -x509 -days 365 -key
"pkcs11:token=HTTPSERVER;object=RSAKey" -keyform engine -out SSL.crt
```

Here `HTTPSERVER` is the token label and `RSAKey` is the key on the HSM. Provide `Cryptouser PIN` when prompted. Also provide other required information for certificate when prompted.

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:CA
Locality Name (eg, city) [Default City]:Campbell
Organization Name (eg, company) [Default Company Ltd]:Utimaco
Organizational Unit Name (eg, section) []:Security
Common Name (eg, your name or your server's hostname) []:Utim
Email Address []:test@utim.com
```

Figure 7 : Generate Self-Signed Certificate Output

```
[admin@master-node bin]$ cat SSL.cert
-----BEGIN CERTIFICATE-----
MIID3zCCAsegAwIBAgIUQ5d9fx+AaM5fmR5RUvVkkqgwelYwDQYJKoZIhvcNAQEL
BQAwfzELMAkGA1UEBhMCVVMxCzAJBgNVBAgMAkNBMREwDwYDVQQHDAhDYW1wYmVs
bDEQMA4GA1UECgwHVXRpbWFjbzERMA8GA1UECwwIU2VjdXJpdHkxDTALBgNVBAMM
BFV0aW0xHDAaBgkqhkiG9w0BCQEWDXRlc3RAdXRpbS5jb20wHhcNMjYwMzE5MDcx
MTI0WhcNMjcwMzE5MDcxMTI0WjB/MQswCQYDVQQGEwJVUzELMAkGA1UECAwCQ0Ex
ETAPBgNVBAcMCENhbXBiZWxsMRAwDgYDVQQKDAdVdGltYWNvMREwDwYDVQQLDAhT
ZWN1cml0eTENMAsGA1UEAwwEVXRpbTEcMBoGCSqGSIb3DQEJARYNdGVzdEB1dGlt
LmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAIMQGK9Lz3ZdZKVd
w9nVcH/IHIiXNx1NmGJCQTUoGrqGFTWrQR01Mtmyfk5BgtD66bbD2GLEQLDfiCKf
sjoF0C+NN2bS0KzBzvYXPrZYe0WguCCYaSzBqGufuH8AGW8Oxh4Flv2LbusQbrRu
+JAhz/q6msXOiH4krY3KsMj0KWParJS32U2jpK2t/xMNElG9S/ZrpApUJckq6aP0
4mVqLGzji7Um9a5Xnp2pumHKsBMHESIk9isMIqYrFa8AtaGIl6YbuG2WdN/nCQdJ
ALAvw91FS0AP5awuyDcgeb8S7CXod7NS16b8g170/K5HGQjq5zJpcMplLSHyXCT3
Ceu0y+ECAwEAAaNTMFEwHQYDVR0OBBYEFDXpsQpyoVRCgvbOPl1Sf10dHd7nMB8G
A1UdIwQYMBaAFDXpsQpyoVRCgvbOPl1Sf10dHd7nMA8GA1UdEwEB/wQFMAMBAf8w
DQYJKoZIhvcNAQELBQADggEBAEwUMKZiiN3I/S8ZBOCDrvY5Izh3IkxcSrFYf12Z
12rL65raywtrLYYZAbwTaiXAwQ6XH2XOGC4I4utEALc5fwSFcQmcmD/zk8W0SbgO
WvTQBZTpbYCR0t3lpzgo+hGJKmvu1BESPbFOUqX+wTE/7I3NEcaBaqIe9hPSU3/j
tLOItNDfxx7/q3gP2NOa8D6H+tQkdDxI/ho7jkC9jkdD6P4wI6CLuCBY7yaj5xvg
h1jvfj2kwNwWF2/LoVBSwUxx+z1uqyPbANNAIKcy4AHfKiKYOmYjkn4I5z38EuyL
wgSGa46QR71DHYcYnA42E2auutQWQIv5TycsdF0qltC5JD4=
-----END CERTIFICATE-----
```

Figure 8 : Certificate File Output

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

5.6 Configuring Apache to Use Utimaco HSM

To configure the Apache HTTP Server to use Utimaco HSM, first we need to enable SSL-TLS support on the Apache and then configure the above created Keys and Certificate in the Configuration file.

1. Navigate to the Apache configuration file `/usr/local/apache/conf/httpd.conf` and uncomment the below mentioned configuration lines.

```
LoadModule ssl_module modules/mod_ssl.so
LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
```

2. Open the Apache SSL Configuration file `/usr/local/apache/conf/extra/httpd-ssl.conf` and add the below entries.

```
SSLCryptoDevice pkcs11
Listen 443
<VirtualHost *:443>
ServerName <Name of the Server>:443
SSLEngine on
SSLCertificateFile "/<folder path>/SSL.cert"
SSLCertificateKeyFile "pkcs11:token=HTTPSERVER;object=RSAKey?pin-
value=<cryptouser_pin>"
</VirtualHost>
```

Where, pkcs11 is engine, `HTTPSERVER` is the name/label of token and `RSAKey` is the key inside the HSM.

The path of httpd-ssl.conf file will change based on your installation method.

3. Restart the Apache Service with the below command.
   If you have installed from source code:

```
# /usr/bin/apache/bin/apachectl -k stop
# /usr/bin/apache/bin/apachectl -X
```

```
[root@ ~]#
[root@ ~]#  /usr/local/apache/bin/apachectl -X
```

Figure 9 : Start Apache Service Output

If you are restarting the service through systemctl make sure to edit the systemd file and update the ExecStart with single worker apachectl command as described in point a, b and c as shown below. This integration is not supported with multiple workers.

If you have installed from dnf:

a) Create the backup of the Apache httpd service file `/usr/lib/systemd/system/httpd.service` as described below

```
# cp /usr/lib/systemd/system/httpd.service /usr/lib/systemd/system/
httpd.service.bkp
```

b) Open the Apache `httpd` service file `/usr/lib/systemd/system/httpd.service` and make sure it contains the below entries

```
[Unit]
Description=The Apache HTTP Server
[Service]
Type=simple
RemainAfterExit=yes
Environment="PATH=/usr/local/openssl/bin:$PATH"
Environment="LD_LIBRARY_PATH=/usr/local/libp11/lib:/usr/local/openssl/lib"
ExecStart=/usr/local/apache/bin/httpd -X
ExecStop=/usr/local/apache/bin/apachectl -k graceful-stop
User=root
[Install]
WantedBy=multi-user.target
```

The path of openssl and libp11 library files will change based on your installation method.

c) Now, reload the `systemd` daemon and restart the httpd service with below commands

```
# systemctl daemon-reload
# systemctl restart httpd.service
```

4. Open <https://<apache_server_ip>> in any browser and verify that you are able to load the page.

Figure 10 : Apache Web Page

5.7 Migrating Existing Keys to Utimaco HSM

If you have Apache running over SSL with the existing keys locally somewhere on the directory you can migrate those keys to Utimaco HSM and securely store them.
To migrate the existing key to Utimaco HSM follow these steps

1. Make sure to complete the steps for installing and configuring openssl and libp11 according to the sections *Installing OpenSSL*, *Installing Libp11*, and *Configuring OpenSSL to Use Utimaco HSM* before proceeding ahead.
2. Convert a private key to pkcs12 format with openssl.

```
openssl pkcs12 -export -nocerts -inkey private.pem -out myprivatekey.p12
```

3. Import converted private key to HSM using p11tool2.

```
# p11tool2 slot=<slot#> loginuser=<cryptouser_password>
PubKeyAttr=CKA_LABEL="Imported_PublicKey",CKA_ID=0x25
 PrvKeyAttr=CKA_LABEL="Imported_PrivateKey",CKA_ID=0x25
 ImportP12=myprivatekey.p12,ask
```

Only private key is required to import to Utimaco HSM, as certificate contains only public key information.

4. List the generated imported keys using p11tool2.

```
# p11tool2 slot=<slot_no> LoginUser=<cryptouser_password> ListObjects
```

```
CKO_PUBLIC_KEY:

+ 2.1
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = D2BF655B-4D53-431C-AB9C-8940EF2A77A9
  CKA_LABEL                     = Imported_PublicKey
  CKA_ID                        = 0x25 (%)
  CKA_SUBJECT                   =


+ 2.2
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = 77CA304E-2759-419A-993A-65E441BA49D1
  CKA_LABEL                     = rsa_public_key
  CKA_ID                        = 0x3031 (01)

+ 2.3
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = 1594064D-17FA-4E0E-99A0-F78328843E3D
  CKA_LABEL                     = RSAKey
  CKA_ID                        = 0x45 (E)


CKO_PRIVATE_KEY:

+ 3.1
  CKA_KEY_TYPE                  = CKK_RSA
  CKA_UNIQUE_ID                 = 5682CD7B-8EEE-4308-848D-418F319E68CE
  CKA_SENSITIVE                 = CK_TRUE
  CKA_EXTRACTABLE               = CK_TRUE
  CKA_LABEL                     = Imported_PrivateKey
  CKA_ID                        = 0x25 (%)
  CKA_SUBJECT                   =
```

Figure 11 : List Keys Output

5. Open the Apache SSL Configuration File `/usr/local/apache/conf/extra/httpd-ssl.conf` and update the Certificate and Private Key Objects as below.

```
SSLCryptoDevice pkcs11
Listen 443
<VirtualHost *:443>
ServerName <Name of the Server>:443
SSLEngine on
SSLCertificateFile "/<path>/SSL.cert"
SSLCertificateKeyFile "pkcs11:token=HTTPSERVER;object=Imported_PrivateKey?pin-value=<cryptouser_pin>"
</VirtualHost>
```

Where, pkcs11 is engine, `HTTPSERVER` is the name/label of token and `Imported_PrivateKey` is the key inside the HSM. `SSL.cert` is the existing cert that you have been using.

The path of `httpd-ssl.conf` file will change based on your installation method.

6. Restart the Apache Service with the below command.
   If you have installed from source code:

```
# /usr/local/apache/bin/apachectl -k stop
# /usr/local/apache/bin/apachectl -X
```

If you are restarting the service through systemctl make sure to edit the systemd file and update the ExecStart with single worker apachectl command as described in point a, b and c as shown below. This integration is not supported with multiple workers.

If you have installed from dnf:

a) Create the backup of the Apache httpd service file `/usr/lib/systemd/system/httpd.service` as described below.

```
# cp /usr/lib/systemd/system/httpd.service /usr/lib/systemd/system/
httpd.service.bkp
```

b) Open the Apache httpd service file `/usr/lib/systemd/system/httpd.service` and make sure it contains the below entries.

```
[Unit]
```

```
Description=The Apache HTTP Server
[Service]
Type=simple
RemainAfterExit=yes
Environment="PATH=/usr/local/openssl/bin:$PATH"
Environment="LD_LIBRARY_PATH=/usr/local/libp11/lib:/usr/local/openssl/lib"
ExecStart=/usr/local/apache/bin/httpd -X
ExecStop=/usr/local/apache/bin/apachectl -k graceful-stop
User=root
[Install]
WantedBy=multi-user.target
```

The path of openssl and libp11 library files will change based on your installation method.

c) Now, reload the `systemd` daemon and restart the `httpd` service with the below commands.

```
# systemctl daemon-reload
# systemctl restart httpd.service
```

7. Open <https://<apache_server_ip>> in any browser and verify that you are able to load the page.

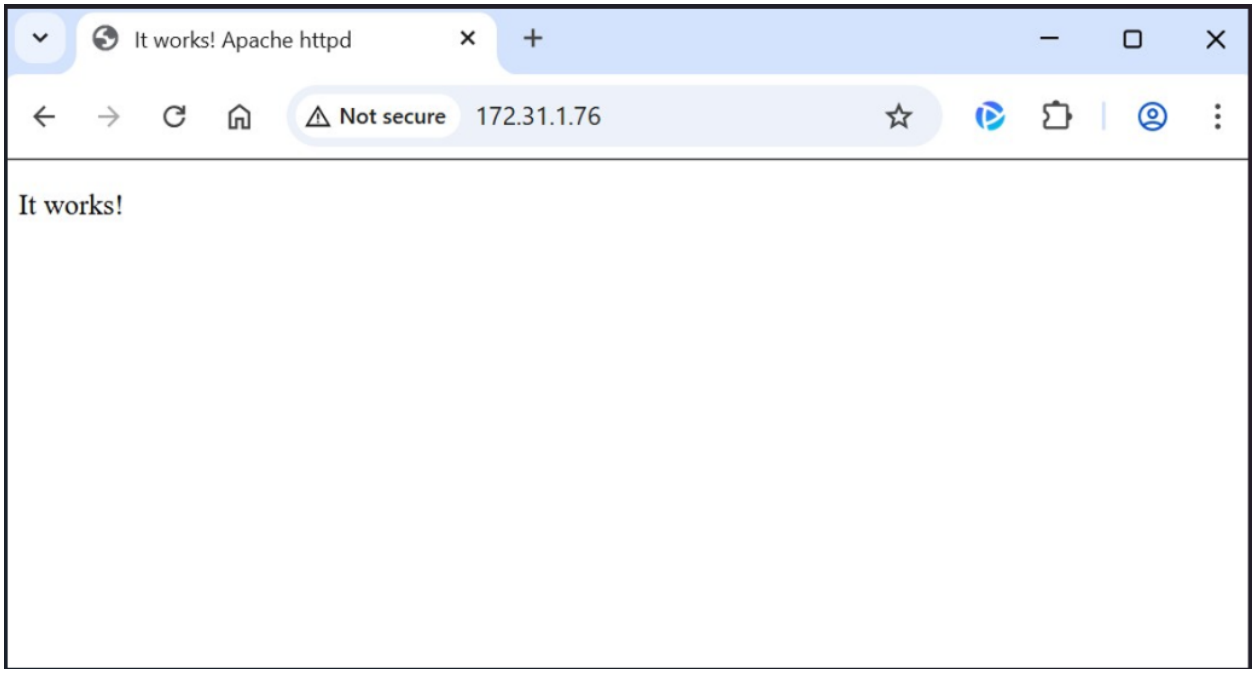

Figure 12 : Apache Web Page

Once Apache is running successfully after migrating the keys on Utimaco HSM you can delete the private key from the software location.

6 Troubleshooting

Error	Diagnosis
LoginUser= failed: 05.12.2021 23:45:45 src/p11adm_R3.c[429] p11_login: C_Login [type=1] returned Error 0x00000102 (CKR_USER_PIN_NOT_INITIALIZED)	PKCS#11 Slot is not initialized. Refer Initialize a Slot
The CryptoServer PKCS#11 Library R3 is not initialized. Error CKR_CRYPTOKI_NOT_INITIALIZED occurred	PKCS#11 Slot is not initialized. Refer Initialize a Slot
OpenSSL> engine -t dynamic -pre SO_PATH:/usr/lib64/openssl/engines/pkcs11.so -pre ID:pkcs11 -pre LIST_ADD:1 -pre LOAD -pre MODULE_PATH:/opt/utimaco/lib/libcs_pkcs11_R3.soengine: Cannot mix flags and engine names. engine: Use -help for summary. error in engine	Install updated libp11 library on host machine

Error	Diagnosis
openssl req -engine pkcs11 -new -key 4F70656E73736C4B6579 -keyform engine -out req.pem -text -x509 -subj "CN=Utimaco" invalid engine "pkcs11" 139703122831248:error:25066067:DSO support routines:DLFCN_LOAD:could not load the shared library:dso_dlfcn.c:187:filename(/usr/lib64/openssl/engines/libpkcs11.so): libcrypto.so.1.1: cannot open shared object file: No such file or directory 139703122831248:error:25070067:DSO support routines:DSO_load:could not load the shared library:dso_lib.c:233: 139703122831248:error:260B6084:engine routines:DYNAMIC_LOAD:dso not found:eng_dyn.c:467: 139703122831248:error:2606A074:engine routines:ENGINE_by_id:no such engine:eng_list.c:392:id=pkcs11 139703122831248:error:25066067:DSO support routines:DLFCN_LOAD:could not load the shared library:dso_dlfcn.c:187:filename(libpkcs11.so): libpkcs11.so: cannot open shared object file: No such file or directory 139703122831248:error:25070067:DSO support routines:DSO_load:could not load the shared library:dso_lib.c:233: 139703122831248:error:260B6084:engine routines:DYNAMIC_LOAD:dso not found:eng_dyn.c:467: no engine specified unable to load Private Key	Export the below value of LD_LIBRARY_PATH and Path for Openssl to avoid the above error. export LD_LIBRARY_PATH=/usr/local/libp11/lib:/usr/local/openssl/lib:$LD_LIBRARY_PATH export PATH=/usr/local/openssl/bin:$PATH
AH00526: Syntax error on line 92 of /usr/local/apache/conf/extra/httpd-ssl.conf: SSLSessionCache: 'shmcb' session cache not supported (known names:). Maybe you need to load the appropriate socache module (mod_socache_shmcb?).	Make sure you have uncommented the required configuration line from the Apache Configuration line

Table 6: List of errors and their diagnoses

7 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.

8 Contact Address for Support Queries

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

Utimaco IS GmbH
Germanusstr. 4
52080 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.

9 Appendices

9.1 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

9.2 Command Summary

Command	Purpose
`./p11tool2 slot=<slot_no> Label=<token_label> Login=ADMIN,ADMIN.key InitToken=<SO_PIN>`	Initializes the HSM slot with a new token label and Security Officer (SO) PIN; creates a fresh PKCS#11 token.
`./p11tool2 slot=<slot_no> LoginSO=<SO_PIN> InitPin=<CryptoUser_PIN>`	Creates/initializes the Crypto User (CU) PIN for the slot, allowing key generation and usage.

Command	Purpose
`p11tool2 slot=<slot#> loginuser=<cryptouser_password> PubKeyAttr=CKA_LABEL="<PublicKey>",CKA_ID=0x25 PrvKeyAttr=CKA_LABEL="<PrivateKey>",CKA_ID=0x25 ImportP12=myprivatekey.p12,ask`	Imports an existing PKCS#12 private key into the HSM with new labels and attributes.
`p11tool2 slot=<slot_no> LoginUser=<cryptouser_password> ListObjects`	Lists all cryptographic objects (keys, certificates) stored in the HSM slot.
`dnf upgrade`	Updates OS packages and security patches.
`wget`	Downloads the source package.
`systemctl start httpd.service`	Starts Apache HTTP Server when installed via package manager (DNF).
`/usr/bin/apache/bin/apachectl -k start`	Starts Apache when installed from source under */usr/local/apache/*.
`openssl req -engine pkcs11 -new -key "pkcs11:token=<token name>;object=<key name>" -keyform engine -out apache.csr`	Generates a Certificate Signing Request (CSR) using a private key stored inside the HSM via PKCS#11.
`openssl req -engine pkcs11 -new -x509 -days 365 -key "pkcs11:token=<token name>;object=<keyname>" -keyform engine -out SSL.crt`	Creates a self-signed certificate using the HSM-stored key .

Command	Purpose
`systemctl daemon-reload`	Reloads systemd service files after modifying Apache service configuration.
`systemctl restart httpd.service`	Restarts Apache HTTP Server (DNF installation).
`/usr/local/apache/bin/apachectl -k stop`	Stops Apache when running in single-process mode (source installation).
`/usr/local/apache/bin/apachectl -X`	Starts Apache in single-worker/debug mode.

Table 8: Command Summary