securosys

SWISS SECURITY TECHNOLOGIES
FOR COMMUNICATIONS SYSTEMS

Primus PKCS#11 Integration into P11-Kit

To use latest OpenSSL, Apache/Nginx...

Application Note

Primus HSM Integration Guide for RHEL8
to secure OpenSSL, Apache, Nginx based on
p11-kit, Primus PKCS#11 Provider using Primus HSM

Securosys SA, Förrlibuckstrasse 70
CH-8005 Zürich, Switzerland
Tel. +41 44 552 31 00 • www.securosys.ch
info@securosys.ch

Document Information and Revision Control

Version	Date	Author	Description, Changes
1	04.05.2019	PM	Initial draft document
2	04.07.2020	PM	NGINX, URI updates, several updates

Table of Contents

1 Introduction

This application note describes the integration of Primus Hardware Security Module with PKCS#11 provider as p11-kit module, to be used by applications like OpenSSL, Apache, Nginx supporting such modules.

RedHat Enterprise Linux 8 onwards improves support for storing secrets on external hardware like smartcards and HSMs via PKCS#11. Additional libraries (p11-kit, openssl-pkcs11), application extensions and standardization of key references (URI) provide the necessary mechanisms to integrate HSMs based on manufacturer PKCS#11 providers.

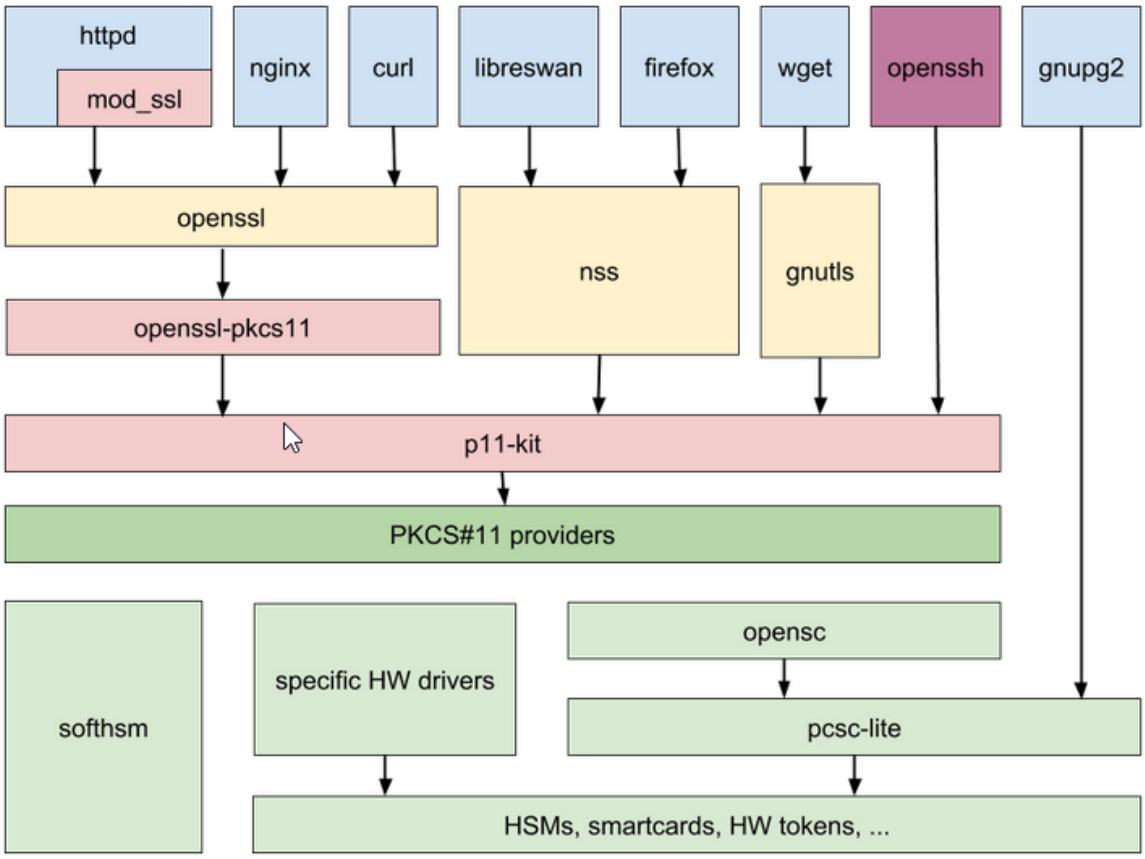

Fig 1-1, Architecture graphics from RHEL 8

1.1 References and More Information

- https://www.redhat.com/en/blog/consistent-pkcs-11-support-red-hat-enterprise-linux-8
- https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/security_hardening/configuring-applications-to-use-cryptographic-hardware-through-pkcs-11_security-hardening
- https://p11-glue.github.io/p11-glue/p11-kit/manual/
- https://tools.ietf.org/html/rfc7512

1.2 Requirements

Software packages depending on Linux distribution:

Modules on RHEL/CentOS 8	**Ubuntu 20**	**Debian 10**
Package Manager: YUM	*Package Manager: APT*	
Primus HSM PKCS#11 Provider (1.5.6+)		
p11-kit (0.23+)		
opensc (0.19+)		
		libccid opensc-pkcs11 pcscd
For OpenSSL		
openssl (1.1.1+)		
openssl-pkcs11 (0.4.10+)	libengine-pkcs11-openssl (0.4.10+)	
For Apache/httpd		
httpd (2.4.37+)	apache2 2.4.42 or later req.	
mod_ssl (2.4.37+)		
For Nginx		
nginx (1.14.1+)		

1.3 Software and Hardware Used for this Setup

- Primus HSM with firmware 2.8 or higher
- Virtualization Software (VMware Desktop 15.5.6)
- CentOS Linux release 8.2.2004
- Primus HSM PKCS#11 Provider, PrimusAPI_PKCS11-1.5.6-rhel8-x86_64.tar.gz
- opensc-0.19.0-7.el8.src.rpm
- p11-kit-0.23.14-5.el8_0.src.rpm
- openssl-pkcs11-0.4.10-2.el8.src.rpm
- openssl-1.1.1c-15.el8.src.rpm
- httpd-2.4.37-21.module_el8.2.0+382+15b0afa8.src.rpm
- nginx-1.14.1-9.module_el8.0.0+184+e34fea82.src.rpm

2 Installation Procedures

This procedure provides a straightforward integration process, which has been tested. Please take notice that there may be other ways to achieve it. This guide assumes that you are familiar with the Primus HSM, Linux operating system installation and configuration procedures, OpenSSL configuration etc. and does not cover every step of the hardware and software setup process. The application note does not cover firewall nor SELinux configuration.

The following installation and configuration description are based on CentOS 8 Linux (see 1.3) and may slightly differ for other Linux distributions (see chapter 1.2).

2.1 Setup the HSM

Setting-up the Primus HSM hardware or your Clouds HSM partition is not part of this application note. Please refer to the corresponding Quick Start Guide and the Primus HSM User Guide.

2.2 Installing the Primus HSM PKCS#11 Provider

Installation of Primus PKCS#11 provider according PKCS#11 Provider User Guide (PrimusAPI_PKCS11-UserGuide_UG-Enn.pdf), downloadable from the Securosys Support Portal.

Change the Primus environment variables to exclude Primus OpenSSL and Apache[1]:

```
export PRIMUS_HOME=/usr/local/primus
export PATH=$PRIMUS_HOME/bin:$PATH
export LD_LIBRARY_PATH=$PRIMUS_HOME/lib:$LD_LIBRARY_PATH
```

Ensure that the latest OpenSSL from your OS is now used and not the one included in the Primus provider package (1.0.2.m) with the following command:

```
openssl version
OpenSSL 1.1.1c FIPS  28 May 2019
```

Ensure that the latest Apache/httpd from your OS is now used and not the one included in the Primus provider package (2.4.29) with the following command:

```
httpd -v
Server version: Apache/2.4.37 (centos)
```

2.3 P11-Kit / OpenSC

2.3.1 Installation of Necessary Packages

Package manager and modules may differ depending on your Linux distributions (see chapter 1.2).

Install the opensc package:

```
sudo yum install opensc
```

Install p11-kit package:

```
sudo yum install p11-kit
```

[1] Included best in shell initialization files (e.g. .bash_profile or /etc/environment)

2.3.2 Register Primus HSM PKCS#11 Provider with p11-kit Tool

Each configured PKCS#11 module has its own configuration file. Depending on the OS, these configuration files may reside at the following locations:

- the p11-kit installation folder (`modules` folder)
- `/etc/pkcs11/` folder (`modules` folder)
- `.pkcs11/modules` folder. Each user can optionally provide additional configuration or override the system configuration. Note that user configuration files are not loaded from the home directory if running inside a setuid or setgid program

Create the file `primus.module` for your OS in

- RHEL/CentOS: `/etc/pkcs11/modules/` or `/usr/share/p11-kit/modules/`
- Ubuntu/Debian: `/usr/share/p11-kit/modules/`

containing the following lines:

```
# This file describes how to load the primus pkcs11 provider module
# For details see p11-kit and pkcs11.conf
# and https://p11-glue.github.io/p11-glue/p11-kit/manual/

# This is a relative path, which means it will be loaded from
# the p11-kit default path which is usually $(libdir)/pkcs11.
# Doing it this way allows for packagers to package primus for
# 32-bit and 64-bit and make them parallel installable
module: libprimusP11.so
managed: true
priority: 2
#critical: true

# A comma and/or space separated list of names of programs that this module should only
# be loaded in. The module will not be loaded for other programs using p11-kit. The base
# name of the process executable should be used here, for example seahorse, ssh.
# do not use enable-in and disable-in simultaneously
enable-in: p11-kit, openssl, httpd, nginx
#disable-in:
#remote:
#log-calls: true
```

Add a symbolic link for libprimusP11.so module in

- RHEL/CentOS: `/usr/lib64/pkcs11/` `/usr/lib64/`
- Ubuntu 20, Debian 10: `/usr/lib/x86_64-linux-gnu/pkcs11/` `/usr/lib/x86_64-linux-gnu/`

e.g. for RHEL/CentOS 8:

```
sudo ln -s /usr/local/primus/lib/libprimusP11.so /usr/lib64/pkcs11/libprimusP11.so
sudo ln -s /usr/local/primus/lib/libprimusP11.so /usr/lib64/libprimusP11.so
```

Use the following command to check that Primus HSM module is registered:

```
p11-kit list-modules
primus: libprimusP11.so
    library-description: PKCS#11 Library
    library-manufacturer: Securosys SA
    library-version: 1.5
    token: DEMO-TEST
        manufacturer: Securosys SA
        model: GRIMSEL
        serial-number: a7176edc3c192207
        hardware-version: 2.9
        firmware-version: 1.5
        flags:
               rng
               login-required
               user-pin-initialized
               restore-key-not-needed
               clock-on-token
               token-initialized
               secondary-authentication
p11-kit-trust: p11-kit-trust.so
    library-description: PKCS#11 Kit Trust Module
    library-manufacturer: PKCS#11 Kit
    library-version: 0.23
…
opensc: opensc-pkcs11.so
    library-description: OpenSC smartcard framework
    library-manufacturer: OpenSC Project
    library-version: 0.19
…
```

2.4 OpenSSL

2.4.1 Installation of Necessary Packages

Package manager and modules may differ depending on your Linux distributions (see 1.2).

Install the openssl-pkcs11 package:

```
sudo yum install openssl-pkcs11
```

2.4.2 OpenSSL Configuration

To use another engine, OpenSSL requires engine settings in the `openssl.cnf` file. Some OpenSSL commands allow specifying a dedicated configuration file (`-config myssl.cnf`) and some do not. Setting the environment variable `OPENSSL_CONF` is a workaround, but be sometimes the default `openssl.cnf` file contains entries that are needed by some commands (e.g. `openssl req`).

You may add the engine entries to your default OpenSSL config file or add other requirements for your OpenSSL command into the config file.

We suggest that you create a separate config file for interactions with the HSM in order to prevent conflicts with previous settings or defaults.

Allocate were OpenSSL and it's default configuration file is installed:

```
openssl version -d
OPENSSLDIR: "/etc/pki/tls"
```

Create a new configuration file `/etc/pki/tls/primusopenssl.cnf` with the engine section and other necessary controls:

```
    openssl_conf = openssl_init

    [openssl_init]
    engines = engine_section

    [engine_section]
    pkcs11 = pkcs11_section

    [pkcs11_section]
    engine_id = pkcs11
#   dynamic_path is not required if you have installed
#   the appropriate pkcs11 engines to your openssl directory
#   dynamic_path = /usr/local/primus/lib/libprimusP11.so
    MODULE_PATH = /usr/local/primus/lib/libprimusP11.so
#   Insert your PKCS11 password, otherwise interactively or command line option
    PIN = "PRIMUSDEV"
#   it is not recommended to use "debug" for production use
#   INIT_ARGS = connector=http://127.0.0.1:12345 debug
    init = 0
```

Include above file in `/etc/pki/tls/openssl.cnf`, the main OpenSSL configuration file:

```
# Note that you can include other files from the main configuration
# file using the .include directive.
.include /etc/pki/tls/primusopenssl.cnf
```

Alternatively, you may also create an alias to use a dedicated config file:

```
alias primusssl='OPENSSL_CONF=/etc/pki/tls/myprimusopenssl.cnf openssl'
```

2.4.3 OpenSSL – HSM Key Reference

Keys of the Primus HSM can be referenced according RFC7512 via URI after the `-key` option: `-key "pkcs11:identifier1;identifier2;…"` using the following identifiers:

- `token=<token label>` Token Label (partition name)
- `serial=<serial number>` Serial Number of the partition
- `object=<object label>` Key Label
- `id=<key id>` Key ID, note that pkcs11-tool writes the key id in hexadecimal notation to the HSM (e.g. "%10%01")
- `type=<cert|public|private>` Key type
- `pin-value=<PKCS#11 PIN>` to pass the PKCS11 PIN via command line

e.g. generating a CSR using the private key "myeckey" on partition "DEMO-TEST" in slot 1[2]:

```
openssl req -engine pkcs11 -new -key "pkcs11:token=DEMO-TEST;object=myeckey;
type=private;pin-value=MYPIN" -keyform engine -out ec.hsm.csr -sha384 -verify
```

[2] See examples in chapter 3.1

In some older implementations, the keys of the Primus HSM can be referenced alternatively according OpenSC definitions:

- Key ID: `<key-id>` or `id_<key-id>`
- Key label: `label_<key-label>`
- Slot and Key ID: `<slot>:<key-id>` or `slot_<slot>-id_<key-id>`
- Slot and key label: `slot_<slot-number>-label_<key-label>`

Below are some examples for using a key via card slot 0 with key id "1234" and label "myeckey":

```
openssl req -engine pkcs11 -new -key 1234 -keyform engine …
openssl req -engine pkcs11 -new -key id_1234 -keyform engine …
openssl req -engine pkcs11 -new -key label_myeckey -keyform engine …
openssl req -engine pkcs11 -new -key 0:1234 -keyform engine …
openssl req -engine pkcs11 -new -key slot_0-id_1234 -keyform engine …
openssl req -engine pkcs11 -new -key slot_0-label_myeckey -keyform engine …
```

Note: in case the slot is not specified the key must reside on the first slot!

e.g. generating a CSR using the private key "myeckey" on partition "DEMO-TEST" in slot 1:

```
openssl req -engine pkcs11 -new -key slot_1-label_myeckey -keyform engine
-out ec.hsm.csr -sha384 -verify
```

2.5 Apache/Httpd Web Server

2.5.1 Installation Necessary Packages

Package manager and modules may differ depending on your Linux distributions (see 1.2).

The httpd webserver requires above P11-Kit and OpenSSL modules and configuration with Primus integrated plus following modules:

```
yum install httpd mod_ssl
```

2.5.2 Httpd/SSL Configuration

Add at the end of the main httpd configuration file `/etc/httpd/conf/httpd.conf` the following lines:

```
<IfModule ssl_module>
    SSLRandomSeed startup builtin
    SSLRandomSeed connect builtin
</IfModule>
```

Adapt the mod_ssl configuration file `/etc/httpd/conf.d/ssl.conf`

```
SSLPassPhraseDialog builtin
```

And reference the used Certificate and Private Key within the VirtualHost section using the object (=label) reference:

```
SSLCertificateFile "pkcs11:token=<partition-name>;object=myrsakey;type=cert"
SSLCertificateKeyFile "pkcs11: token=<partition-name>;object=myrsakey;type=private"
```

2.5.3 Httpd/SSL – HSM Key Reference

HSM objects are referenced in general according RFC7512 URI scheme "pkcs11:identifier1;identifier2;…".

Currently only the following URI specifier and forms are supported:

- token=<token label> Token Label (partition name)
- serial=<serial number> Serial Number of the partition
- object=<object label> Key Label
- id=<key id> Key ID, note that pkcs11-tool writes the key id in hex notation to the HSM (e.g. 1010 = "%10%01")
- type=<cert|public|private> Key type
- pin-value=<PKCS#11 PIN> to pass the PKCS11 PIN via command line

2.6 Nginx Web Server

Nginx (Engine X) is a popular, powerful and high-performance open-source HTTP web server and reverse proxy server with a scalable event-drive (asynchronous) architecture. It can also be used as a load balancer, mail proxy, and HTTP cache due to its speed, stability, feature-rich set, easy configuration, and low resource utilization.

Because Nginx also uses the OpenSSL for cryptographic operations, support for PKCS #11 must go through the openssl-pkcs11 engine. Nginx currently supports only loading private keys from an HSM, and a certificate must be provided separately as a regular file.

2.6.1 Installation Necessary Packages

The Nginx webserver requires above P11-Kit and OpenSSL modules and configuration with Primus integrated plus following modules

```
yum install nginx
```

2.6.2 Nginx/SSL Configuration

Nginx currently supports only loading private keys from an HSM, and a certificate must be provided separately as a regular file. Modify the nginx configuration file /etc/nginx/nginx.conf by activating TLS and referencing the used Certificate as regular file and Private Key within the server section using the object (=label) reference:

```
ssl_certificate "/path/to/cert.pem"
ssl_certificate_key "engine:pkcs11:pkcs11:token=<partition-name>;pin-value=<pkcs11
PIN>;object=myrsakey;type=private"
```

2.6.3 Nginx/SSL – HSM Key Reference

HSM objects are referenced in general according RFC7512 URI scheme "engine:pkcs11:pkcs11:identifier1;identifier2;…".

Currently only the following URI specifier and forms are supported:

- `token=<token label>` Token Label (partition name)
- `serial=<serial number>` Serial Number of the partition
- `object=<object label>` Key Label
- `id=<key id>` Key ID, note that pkcs11-tool writes the key id in hexadecimal notation to the HSM (e.g. "%10%01")
- `type=<cert|public|private>` Key type
- `pin-value=<PKCS#11 PIN>` to pass the PKCS11 PIN via command line

3 Examples

The examples below are using the `openssl` command, using customized configuration for Primus HSM PKCS#11 provider as shown above. HSM partition is mounted as slot 0. For your requests you probably have to adapt the openssl configuration file or define all your CSR parameters on the command line.

3.1 Generating Key Pair and Certificate Signing Request (CSR) and Import Certificate

This chapter describes how to generate a keypair using `pkcs11-tool`, generating a CSR using `openssl` and importing the certificate using `pkcs11-tool` (you cannot create an RSA key inside the HSM with genrsa/genpkey).

3.1.1 RSA Key Pair, CSR and Certificate Import

Generating a new key pair using pkcs11-tool: 2048-bit RSA, CKA_SENSITIVE set, using slot 0:

```
pkcs11-tool --module /usr/local/primus/lib/libprimusP11.so --slot=0 --login
--pin PRIMUSDEV --keypairgen --key-type RSA:2048 --id 1001 --label myrsakey
--sensitive

Using slot 0 with a present token (0x0)
Key pair generated:
Private Key Object; RSA
  label:      myrsakey
  ID:         1001
  Usage:      decrypt, sign, unwrap
Public Key Object; RSA 2048 bits
  label:      myrsakey
  ID:         1001
  Usage:      encrypt, verify, wrap
```

Generate the CSR interactively, using OpenSSL by referencing the key ID on the HSM by the key ID:

```
openssl req -engine pkcs11 -new -key "pkcs11:token=<partition-
name>;id=%10%01;type=private" -keyform engine -out rsa.hsm.csr -sha256 -verify
or alternatively
openssl req -engine pkcs11 -new -key slot_0-id_1001 -keyform engine -out
rsa.hsm.csr -sha256 -verify

engine "pkcs11" set.
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]:CH
State or Province Name (full name) []:ZH
Locality Name (eg, city) [Default City]:Zuerich
Organization Name (eg, company) [Default Company Ltd]:Securosys SA
Organizational Unit Name (eg, section) []:IT
Common Name (eg, your name or your server's hostname) []:cos8.securosys.ch
Email Address []:cos8@securosys.ch

…
verify OK
```

Alternatively, you can issue a self signed certificate using OpenSSL:

```
openssl req -engine pkcs11 -new -x509 -key "pkcs11:token=<partition-
name>;id=%10%01;type=private" -keyform engine -out rsa.hsm.crt -sha256 -days
365 -nodes
```

If required, import the signed certificate rsa.hsm.crt (DER format), using pkcs-11 tool:

```
pkcs11-tool --module /usr/local/primus/lib/libprimusP11.so -l -p PRIMUSDEV
--slot 0 --write-object rsa.hsm.crt --type cert --id 1001 --label myrsakey

Using slot with ID 0x0
Created certificate:
Certificate Object; type = X.509 cert
  label:      myrsakey
  subject:    DN: C=CH, O=Securosys SA, OU=IT, CN=cos8.securosys.ch
  ID:         1001
```

3.1.2 EC Key Pair, CSR and Certificate Import

Generating a new key pair using pkcs11-tool: Elliptic Curve secp384r1, using slot 0:

```
pkcs11-tool --module /usr/local/primus/lib/libprimusP11.so –slot=0 --login
--pin PRIMUSDEV --keypairgen --key-type EC:secp384r1 --id 1002 --label myeckey

Using slot 0 with a present token (0x0)
Key pair generated:
Private Key Object; EC
  label:      myeckey
  ID:         1002
  Usage:      sign, derive
Public Key Object; EC  EC_POINT 384 bits
  EC_POINT:
046104fb05c1978b6d700e4adb629f81194f5a9c4a06ba4ffe50c41c9165630021fbecd5e91a72db8e48f5d0
5c81a8dd5d55eca1722587f1f13b9fc4e622eb54b07a15d18f77d774326c3f35abe69ca3a7de53a8e92e2cf2
c44e1a9a34f52fdbacf71b
  EC_PARAMS:  06052b81040022
  label:      myeckey
  ID:         1002
  Usage:      verify, derive
```

Generate the CSR interactively, using OpenSSL by referencing the key on the HSM by the key Label (`id_<id-number>`) or label:

```
openssl req -engine pkcs11 -new -key "pkcs11:token=<partition-
name>;object=myeckey;type=private" -keyform engine -out ec.hsm.csr -sha384 -verify
or alternatively
openssl req -engine pkcs11 -new -key slot_0-label_myeckey -keyform engine -
out ec.hsm.csr -sha384 -verify

engine "pkcs11" set.
```

```
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]:CH
$State or Province Name (full name) []:ZH
Locality Name (eg, city) [Default City]:Zuerich
Organization Name (eg, company) [Default Company Ltd]:Securosys SA
Organizational Unit Name (eg, section) []:IT
Common Name (eg, your name or your server's hostname) []:cos8.securosys.ch
Email Address []:cos8@securosys.ch
…
verify OK
```

Alternatively, you can issue a self signed certificate using OpenSSL:

```
openssl req -engine pkcs11 -new -x509 -key "pkcs11:token=<partition-
name>;object=myeckey;type=private" -keyform engine -out ec.hsm.crt -sha384 -days
365 -nodes
```

If required, import the signed certificate ec.hsm.cer (DER format), using pkcs-11 tool:

```
pkcs11-tool --module /usr/local/primus/lib/libprimusP11.so -l -p PRIMUSDEV
--slot 0 --write-object ec.hsm.crt --type cert --id 1002 --label myeckey
```