OpenStack Barbican

Dalmatian

v19.0.0

Integration Guide

u.trust GP HSM

v6.1.1

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	2026-02-19
Status	PUBLISHED
Document No.	IG-2025-0032

Table of Contents

1 Introduction

This guide is part of the information and support provided by Utimaco. Additional documentation 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 is available from Utimaco's website at https://utimaco.com/.

1.1 About This Guide

This guide describes how to enable HSM integration with OpenStack Dalmatian Barbican. Utimaco HSM securely stores the MKEK and HMAC used by Barbican.

1.2 Target Audience for This Guide

This guide is intended for OpenStack Barbican and HSM administrators.

1.3 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 **OK**
`Monospaced`	Code that is given for explanation or as an example, file paths	`chsm-create`
Italic	References and important terms	See *Sample Chapter* in the *CryptoServer - Sample Manual*

Table 1: Document conventions

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

Here you will find important safety information.

Here you will find additional notes or supplementary information.

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

1.4 Abbreviations

The following abbreviations are used in this guide:

Abbreviation	Meaning
API	Application Programming Interface
CSADM	CryptoServer Command-line Administration Tool
CSP	Cryptographic Service Provider
CXI	Cryptographic eXtended Services Interface
DB	Database
GUI	Graphical User Interface
HMAC	Hash-based message authentication code
HSM	Hardware Security Module

IP	Internet Protocol
LAN	Local Area Network
MBK	Master Backup Key
MKEK	Master Key Encryption Key
PCIe	PCI Express Interface
URL	Uniform Resource Locator

Table 2: Abbreviations

2 Overview

2.1 OpenStack Barbican

Barbican is the OpenStack Key Manager service. It provides secure storage, provisioning, and management of secret data. This includes keying material such as Symmetric Keys, Asymmetric Keys, Certificates, and raw binary data.

2.2 Utimaco u.trust General Purpose HSM Se-Series

Utimaco u.trust General Purpose HSM Se-Series is a hardware security module developed by Utimaco IS GmbH. It is a physically protected, specialized computer unit designed to perform sensitive cryptographic tasks and securely manage and 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 that 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 the required Software.

3.1 Tested Versions

The integrations that have been successfully tested with the Utimaco HSM and OpenStack Barbican:

Operating System	OpenStack Version	Utimaco Security Server Version	Utimaco HSM
RHEL 9	OpenStack Dalmatian	SecurityServer 6.1.1	u.trust General Purpose HSM Se-Series

Table 3: List of Tested Versions

3.2 Software Requirements

Software	Software Requirements
HSM Interfaces	PKCS11

Table 4: List of Software Requirements

3.3 Hardware Requirements

Hardware	Hardware Requirements
Utimaco LAN HSM	u.trust General Purpose HSM Se-Series LAN with firmware SecurityServer 6.1.1

Hardware	Hardware Requirements
Utimaco PCI-e HSM	u.trust General Purpose HSM Se-Series PCIe card with firmware SecurityServer 6.1.1

Table 5: List of Hardware Requirements

3.4 Prerequisites

Before you begin, please ensure that you have installed/setup:

- SecurityServer is set up and configured. Refer to the SecurityServer documentation to set up the HSM.
- MBK must be created and stored on each HSM. Refer to the SecurityServer documentation to set up the MBK.
- SecurityServer Default Admin should be replaced with a new admin user.
- Operating system listed in Tested Versions.
- SecurityServer listed in Tested Version.
- An admin user is required to install software.

4 Installing and Configuring Utimaco SecurityServer Software

4.1 Download and Install Utimaco SecurityServer Software

1. Copy the downloaded software to the appropriate location on the OpenStack Barbican Server.
2. Create a 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 SecurityServer 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 SecurityServer software to /opt/utimaco/bin directory and make both the files executable.

```
›_ Console
```

```
# cd ~/path_to_application_folder/Software/Linux/Administration

# cp csadm p11tool2 gladm /opt/utimaco/bin

# chmod +x /opt/utimaco/bin/csadm /opt/utimaco/bin/p11tool2  /opt/utimaco/bin/gladm
# chmod +x /opt/utimaco/lib/libcs_pkcs11_R3.so
```

4.2 SecurityServer PKCS#11 Configuration

1. Create the directory /opt/utimaco/lib. 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 p11_R3.cfg into /opt/utimaco/lib directory.

```
›_ Console
```

```
# mkdir /etc/utimaco

# cd <install directory>/Software/Linux/Crypto_APIs/PKCS11_R3/sample

# cp cs_pkcs11_R3.cfg /etc/

# cd /etc
```

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 = 4

Keepalive = true

# Set the Device to connect with

[HSMCluster]

# Device specifier

Device = <HSM_IP>
```

For more information regarding the commands and command parameters, please see the Utimaco SecurityServer documentation. The device may be a SecurityServer (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/cs2.0.1 (last number represents cHSM slot (1)) PCIe card`

To make your testing easier, enable the PKCS#11 log file. You can enable it by editing the logging Loglevel. Set the LogPath and Loglevel to 1. For testing, you may want to increase them 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 under 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 the p11tool2 command, initialize the slot you want to use and the slot user, as shown below.

```
›_ Console
```

```
#  export CS_PKCS11_R3_CFG="/etc/cs_pkcs11_R3.cfg"
#./p11tool2 [Slot=<slot_id>] [Label=<label>] \
            [Force=<force>] [Login=<admin_name>,<admin_auth_token>] \
            InitToken=<so_pin>
#./p11tool2 [Slot=<slot_id>]  LoginSO=12345678 setpin=12345678,123456789
# ./p11tool2 slot=<slot_no> LoginSO=12345678 InitPin=123456789
# ./p11tool2 slot=<slot_id> Login=123456789 setpin=123456789,1234567890
```

5 (Optional) Installing OpenStack Barbican

Skip this section if OpenStack Barbican has already been installed. The steps below are only for demonstration purposes and will change based on the actual requirement.

5.1 Install OpenStack Barbican

Use the steps below to install OpenStack Dalmatian via Packstack.

1. Disable SeLinux using the command below:

```
›_ Console
```

```
# setenforce 0
```

Alternatively, open the /etc/selinux/config file and change permissive to disabled as shown below. Then reboot your machine.

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Figure 1 : SELINUX Config File

This is for demonstration purposes only. Add the OpenStack Barbican files and directories in the SELinux context in a production environment.

2. Change the hostname to controller by using the below command and reboot.

```
›_ Console
```

```
# hostnamectl set-hostname controller

# reboot
```

3. Map barbican and controller to IP in /etc/hosts file.

```
›_ Console
```

```
# vi /etc/hosts

<Host_IP>    barbican        barbican.localdomain

<Host_IP>    controller       controller.localdomain
```

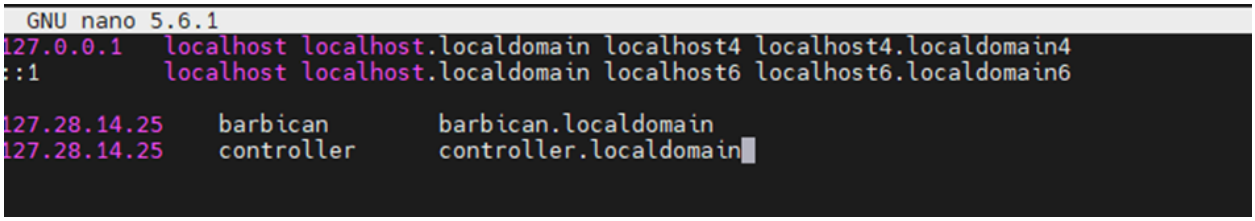

Figure 2 : Hosts File

4. Stop the firewall and NetworkManager using the below command.

This is for demonstration purpose only. In production environment allow openstack barbican services through firewall.

```
›_ Console
```

```
# systemctl stop firewalld NetworkManager

# systemctl disable firewalld NetworkManager
```

5. Set up the timezone.

```
›_ Console

# timedatectl set-timezone <timezone name>
```

6. Enable the required repositories.

```
›_ Console

# subscription-manager repos --enable=rhel-9-for-x86_64-baseos-rpms --enable=rhel-9-for-x86_64-appstream-rpms --enable=rhel-9-for-x86_64-supplementary-rpms --enable=codeready-builder-for-rhel-9-x86_64-rpms
```

7. Install OpenStack Packstack by using the following commands.

```
›_ Console

# dnf install https://trunk.rdoproject.org/rdo_release/rdo-release.el9s.rpm

# dnf upgrade

# dnf install -y openstack-packstack
```

8. Install the OpenStack instance.

›_ Console

```
# packstack --allinone
```

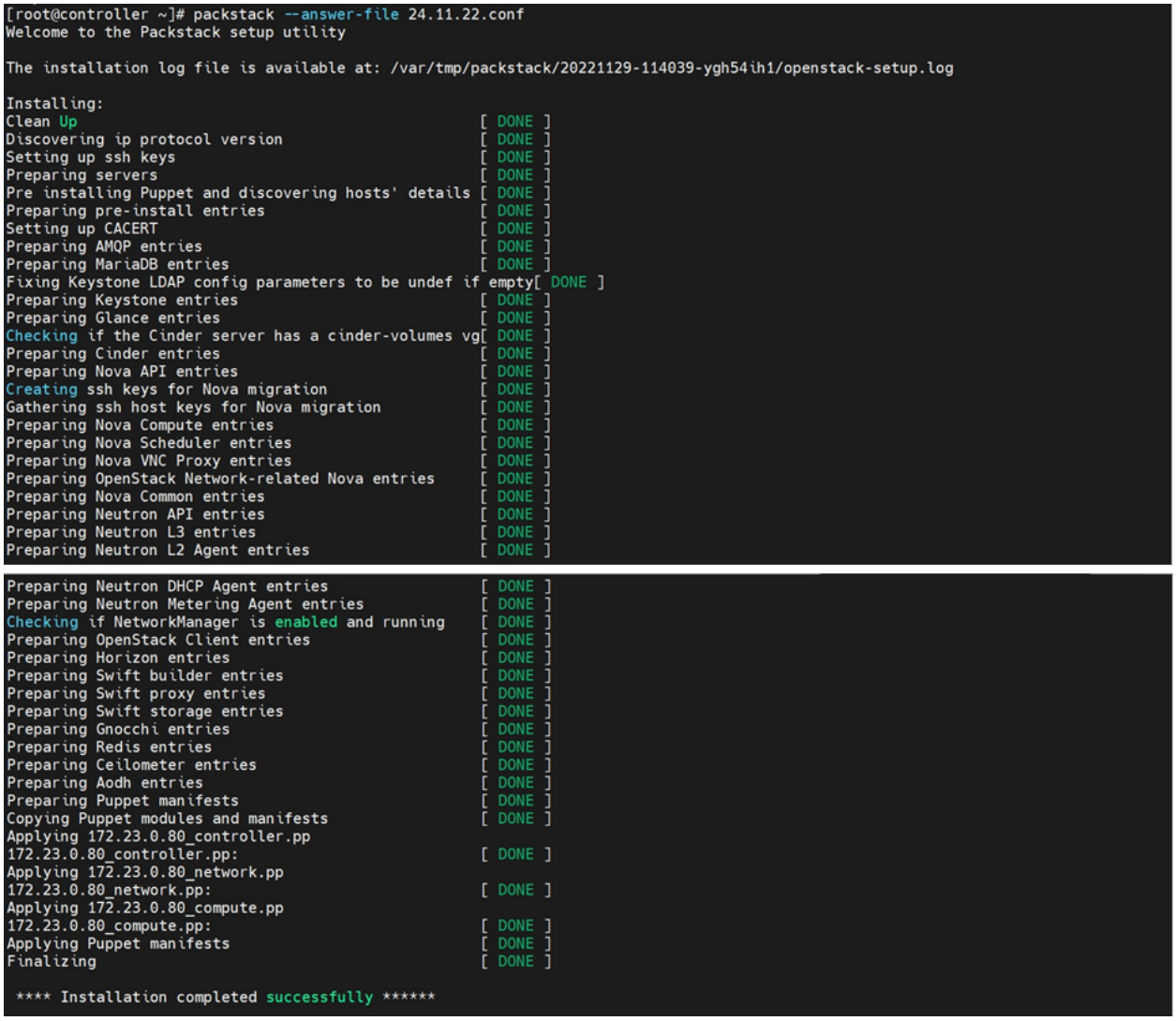

Figure 3 : PackStack Configuration

An answer file will be generated in /root/. This file contains the passwords automatically generated by Packsack and used by MariaDB and other services.

9. Open mysql as root.

```
›_ Console
```

```
# mysql -u root -p
```

10. Create a database Barbican and allow full permission to the Barbican user.

```
›_ Console
```

```
> CREATE DATABASE barbican;

> GRANT ALL PRIVILEGES ON barbican.* TO 'barbican'@'localhost' IDENTIFIED
BY '123456';

>GRANT ALL PRIVILEGES ON barbican.* TO 'barbican'@'%' IDENTIFIED BY

'123456';

> exit;
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON barbican.* TO 'barbican'@'localhost' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.004 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON barbican.* TO 'barbican'@'%' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> exit;
Bye
[root@controller ~(keystone_admin)]#
```

Figure 4 : Mysql command output

11. Restart the http daemon.

```
›_ Console
```

```
 # systemctl restart httpd
```

12. Set the environment variable for OpenStack.

›_ Console

```
# source /root/keystonerc_admin
```

13. Create a user barbican in OpenStack.

›_ Console openstack

```
# openstack user create --domain default --password 123456 barbican
```

```
[root@controller user(keystone_admin)]# openstack user create --domain default --password 123456 barbican
Could not load 'message_list': module 'zaqarclient.queues.v2.cli' has no attribute 'OldListMessages'
Could not load 'message_post': module 'zaqarclient.queues.v2.cli' has no attribute 'OldPostMessages'
+---------------------+----------------------------------+
| Field               | Value                            |
+---------------------+----------------------------------+
| default_project_id  | None                             |
| domain_id           | default                          |
| email               | None                             |
| enabled             | True                             |
| id                  | c6426a7dbe1345bfa67f8d7db4fdcc2e |
| name                | barbican                         |
| description         | None                             |
| password_expires_at | None                             |
```

14. Add the admin role to the user barbican in project services.

›_ Console

```
# openstack role add --project services --user barbican admin
```

15. Add the creator role.

›_ Console

```
# openstack role create creator
```

```
[root@controller user(keystone_admin)]# openstack role create creator
Could not load 'message_list': module 'zaqarclient.queues.v2.cli' has no attribute 'OldListMessages'
Could not load 'message_post': module 'zaqarclient.queues.v2.cli' has no attribute 'OldPostMessages'
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| id          | 4640be74da5441788609bfeacea5a0dc |
| name        | creator                          |
| domain_id   | None                             |
| description | None                             |
+-------------+----------------------------------+
```

Figure 5 : OpenStack Role Creation

16. Add the creator role to the barbican user.

›_ Console

```
# openstack role add --project services --user barbican creator
```

```
[root@controller bin(keystone_admin)]# openstack role add --project services --user barbican creator
[root@controller bin(keystone_admin)]#
```

Figure 6 : Add Creator Role

17. Create the barbican service entities.

›_ Console

```
# openstack service create --name barbican --description "key manager" key-manager
```

```
[root@controller user(keystone_admin)]# openstack service create --name barbican --description "key manager" key-manager
Could not load 'message_list': module 'zaqarclient.queues.v2.cli' has no attribute 'OldListMessages'
Could not load 'message_post': module 'zaqarclient.queues.v2.cli' has no attribute 'OldPostMessages'
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| id          | f11eda1aee1248bc95789d7142b847e9 |
| name        | barbican                         |
| type        | key-manager                      |
| enabled     | True                             |
| description | key manager                      |
+-------------+----------------------------------+
```

Figure 7 : OpenStack Service Creation

18. Create the Key Manager service API public endpoint.

›_ Console

openstack endpoint create --region RegionOne key-manager public http://172.28.14.25:9311

```
[root@controller ~(keystone_admin)]# openstack endpoint create --region RegionOne   key-manager public http://controller:9311
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | 78b2f48139914f659f4c6216dac6cd27 |
| interface    | public                           |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | 856011803d674e678cc15e7f3caf94ad |
| service_name | barbican                         |
| service_type | key-manager                      |
| url          | http://controller:9311           |
+--------------+----------------------------------+
```

Figure 8 : OpenStack Public Endpoint Creation

19. Create the Key Manager service API Internal endpoint.

›_ Console

```
# openstack endpoint create --region RegionOne key-manager internal http://172.28.14.25:9311
```

```
[root@controller ~(keystone_admin)]# openstack endpoint create --region RegionOne key-manager internal http://controller:9311
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | 0c755c2012c748a6ac64047794051507 |
| interface    | internal                         |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | 856011803d674e678cc15e7f3caf94ad |
| service_name | barbican                         |
| service_type | key-manager                      |
| url          | http://controller:9311           |
+--------------+----------------------------------+
```

Figure 9 : OpenStack Internal Endpoint Creation

20. Create the Key Manager service API Admin endpoint.

›_ Console

```
# openstack endpoint create --region RegionOne key-manager admin http://172.28.14.25:9311
```

```
[root@controller ~(keystone_admin)]# openstack endpoint create --region RegionOne key-manager admin http://controller:9311
+--------------+----------------------------------+
| Field        | Value                            |
+--------------+----------------------------------+
| enabled      | True                             |
| id           | 93e0798fef324e61a8a262bb683d7947 |
| interface    | admin                            |
| region       | RegionOne                        |
| region_id    | RegionOne                        |
| service_id   | 856011803d674e678cc15e7f3caf94ad |
| service_name | barbican                         |
| service_type | key-manager                      |
| url          | http://controller:9311           |
+--------------+----------------------------------+
```

Figure 10 : OpenStack Admin Endpoint Creation

21. List the Key Manager service API endpoints created.

›_ Console

```
# openstack endpoint list --service barbican
```

```
[root@controller ~(keystone_admin)]# openstack endpoint list --service barbican
+----------------------------------+-----------+--------------+--------------+---------+-----------+-----------------------+
| ID                               | Region    | Service Name | Service Type | Enabled | Interface | URL                   |
+----------------------------------+-----------+--------------+--------------+---------+-----------+-----------------------+
| 0c755c2012c748a6ac64047794051507 | RegionOne | barbican     | key-manager  | True    | internal  | http://controller:9311 |
| 78b2f48139914f659f4c6216dac6cd27 | RegionOne | barbican     | key-manager  | True    | public    | http://controller:9311 |
| 93e0798fef324e61a8a262bb683d7947 | RegionOne | barbican     | key-manager  | True    | admin     | http://controller:9311 |
+----------------------------------+-----------+--------------+--------------+---------+-----------+-----------------------+
```

Figure 11 : Endpoint List

22. Install OpenStack Barbican:

›_ Console

```
# dnf install -y openstack-barbican
```

23. Add the below information to the /etc/barbican/barbican.conf file. You can replace the username and password with the ones you set earlier during installation.

›_ Console

```
# vi /etc/barbican/barbican.conf

[DEFAULT]

host_href = http://barbican:9311

debug = true

transport_url = rabbit://guest:guest@controller

log_file = /tmp/logfile

[keystone_authtoken]

www_authenticate_uri = http://controller:5000

auth_url = http://controller:5000/v3

auth_version = v3

insecure = true

region_name = RegionOne

memcached_servers = localhost:11211

auth_type = password

project_name = services

user_domain_id = default

project_domain_id = default

username = barbican

password = 123456

[database]

connection = mysql+pymysql://barbican:123456@localhost/barbican
```

24. Populate the Key Manager service database.

›_ Console

```
# su -s /bin/sh -c "barbican-manage db upgrade" barbican
```

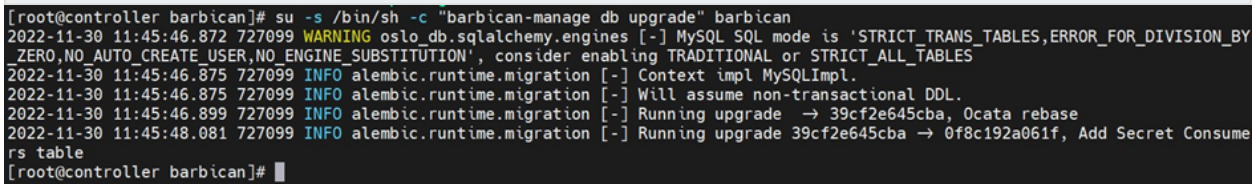

Figure 12 : Database Upgrade

25. Restart Barbican API service.

›_ Console

```
# systemctl restart openstack-barbican-api.service
```

26. Restart and enable the httpd service.

›_ Console

```
# systemctl restart httpd

# systemctl enable httpd
```

6 Integrating OpenStack Barbican with Utimaco SecurityServer

6.1 Configure Barbican to use Utimaco HSM

1. Add the below information to the barbican.conf file.

›_ Console

```
# vi /etc/barbican/barbican.conf

[secretstore]
namespace = barbican.secretstore.plugin
enabled_secretstore_plugins = store_crypto

[crypto]
enabled_crypto_plugins = p11_crypto

[p11_crypto_plugin]
# Path to Utimaco PKCS11 library
library_path = /opt/utimaco/lib/libcs_pkcs11_R3.so

# CryptoUser PIN to login to PKCS11
login = <PKCS11 Slot User PIN>

# Master KEK label as stored in the HSM
mkek_label = mkek_utimaco

# Master KEK length in bytes. (integer value)
mkek_length = 32

# Master HMAC Key label (as stored in the HSM) (string value)
hmac_label = hmac_utimaco

# HSM Slot ID (integer value)
slot_id = 3

encryption_mechanism = CKM_AES_CBC
```

mkek_utimaco and hmac_utimaco keys will be generated on the Utimaco HSM in slot 3 in the next section of this document.

6.2 Generating MKEK and HMAC Key on Utimaco HSM

1. Generate the MKEK using the below command.

```
›_ Console

# chmod 666  /tmp/cs_pkcs11_R3.log
# su -s /bin/sh -c "barbican-manage hsm gen_mkek --library-path '/opt/utimaco/lib/libcs_pkcs11_R3.so' --passphrase 1234567890 --slot-id 3 --label 'mkek_utimaco' --length 32" barbican
```

Figure 13 : MKEK Key Generation

2. Generate the HMAC Key using the below command.

```
›_ Console

# su -s /bin/sh -c "barbican-manage hsm gen_hmac --library-path '/opt/utimaco/lib/libcs_pkcs11_R3.so' --passphrase 1234567890  --slot-id 3 --label 'hmac_utimaco' --length 32" barbican
```

Figure 14 : HMAC Key Generation

3. Verify that the keys are generated on the Utimaco HSM using the p11tool2 command.

```
›_ Console
```

```
#./p11tool2 slot=<slot_id> LoginUser=<Crypto_User_PIN> ListObjects
```

4. Restart the OpenStack-Barbican-API and httpd services.

```
›_ Console
```

```
# systemctl restart openstack-barbican-api.service

# systemctl restart httpd
```

6.3 Encrypting and Decrypting Secrets

1. Create a secret or password.

```
›_ Console
```

```
# openstack secret store --name utimaco123 --payload password
```

```
[root@rhelserver user(keystone_admin)]# openstack secret store --name utimaco123 --payload password
Could not load 'message_list': module 'zaqarclient.queues.v2.cli' has no attribute 'OldListMessages'
Could not load 'message_post': module 'zaqarclient.queues.v2.cli' has no attribute 'OldPostMessages'
+---------------+-------------------------------------------------------------------------------+
| Field         | Value                                                                         |
+---------------+-------------------------------------------------------------------------------+
| Secret href   | http://172.28.14.25:9311/v1/secrets/20d68ada-e4fd-4e33-9f6b-e08927ab5d8f |
| Name          | utimaco123                                                                    |
| Created       | None                                                                          |
| Status        | None                                                                          |
| Content types | None                                                                          |
| Algorithm     | aes                                                                           |
| Bit length    | 256                                                                           |
| Secret type   | opaque                                                                        |
| Mode          | cbc                                                                           |
| Expiration    | None                                                                          |
+---------------+-------------------------------------------------------------------------------+
```

Figure 15 : Creating a secret with OpenStack Barbican

Here utimaco123 is the secret name and its value is password. This secret is stored in an encrypted form in openstack barbican.

2. You can also verify the encryption operation logging in PKCS11 log file cs_pkcs11_R3.log during secret generation as shown below.

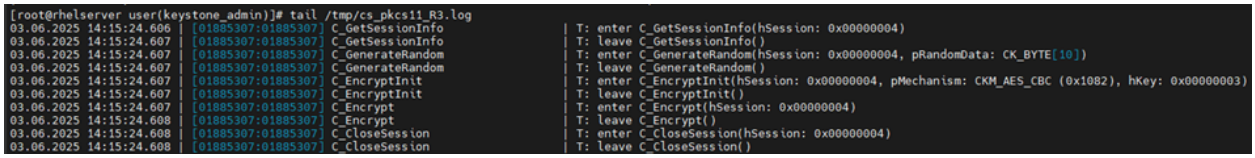

Figure 16 : pkcs#11 Logs

3. Fetch the secret that was created without its value.

›_ Console

```
# openstack secret get http://barbican:9311/v1/secrets/8ac7918d-36ce4cae-b1d9-a2a818ea30a0
```

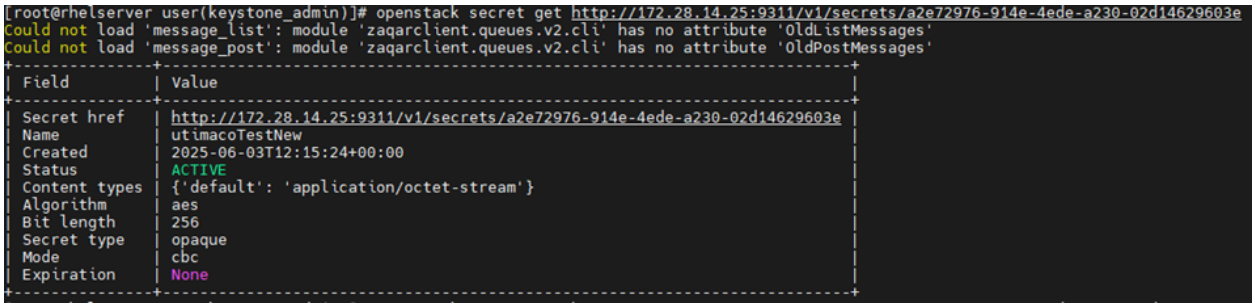

Figure 17 : Get secret without payload

4. Fetch the secret that was created with its value.

›_ Console

```
# openstack secret get http://barbican:9311/v1/secrets/8ac7918d-36ce-4cae-b1d9-a2a818ea30a0 --payload
```

```
[root@rhelserver user(keystone_admin)]# openstack secret get http://172.28.14.25:9311/v1/secrets/a2e72976-914e-4ede-a230-02d14629603e --payload
Could not load 'message_list': module 'zaqarclient.queues.v2.cli' has no attribute 'OldListMessages'
Could not load 'message_post': module 'zaqarclient.queues.v2.cli' has no attribute 'OldPostMessages'
+---------+----------+
| Field   | Value    |
+---------+----------+
| Payload | password |
+---------+----------+
```

Figure 18 : Get secret with payload

The secret is decrypted first and displayed.

5. You can also verify the decryption operation logging in PKCS11 log file cs_pkcs11_R3.log during secret retrieval as shown below.

```
05.01.2023 10:29:56.105 | [00002946:00002946] C_OpenSession                | T: leave C_OpenSession()
05.01.2023 10:29:56.105 | [00002946:00002946] C_GetSessionInfo             | T: enter C_GetSessionInfo(hSession: 0x00000004)
05.01.2023 10:29:56.105 | [00002946:00002946] C_GetSessionInfo             | T: leave C_GetSessionInfo()
05.01.2023 10:29:56.105 | [00002946:00002946] C_DecryptInit                | T: enter C_DecryptInit(hSession: 0x00000004, pMechanism: CKM_AES_CBC (0
x1082), hKey: 0x00000004)
05.01.2023 10:29:56.105 | [00002946:00002946] C_DecryptInit                | T: leave C_DecryptInit()
05.01.2023 10:29:56.108 | [00002946:00002946] C_Decrypt                    | T: enter C_Decrypt(hSession: 0x00000004)
05.01.2023 10:29:56.109 | [00002946:00002946] C_Decrypt                    | T: leave C_Decrypt()
05.01.2023 10:29:56.109 | [00002946:00002946] C_CloseSession               | T: enter C_CloseSession(hSession: 0x00000004)
05.01.2023 10:29:56.109 | [00002946:00002946] C_CloseSession               | T: leave C_CloseSession()
```

Figure 19 : pkcs#11 Logs

6.4 Generating a symmetric key in Barbican

1. Generate a new 256-bit key using OpenStack order create and store it in Barbican.

```
›_ Console

# openstack secret order create --name app_key --algorithm aes --mode ctr
--bit-length 256 --payload-content-type=application/octet-stream key
```

```
[root@rhelserver user(keystone_admin)]# openstack secret order create --name app_key --algorithm aes --mode ctr --bit-length 256 --payload-content-type=application/octet-stream key
Could not load 'message_list': module 'zaqarclient.queues.v2.cli' has no attribute 'OldListMessages'
Could not load 'message_post': module 'zaqarclient.queues.v2.cli' has no attribute 'OldPostMessages'
+---------------+-------------------------------------------------------------------------------+
| Field         | Value                                                                         |
+---------------+-------------------------------------------------------------------------------+
| Order href    | http://172.28.14.25:9311/v1/orders/42ee7b6d-6c31-49ef-b77d-305ba04a6b79 |
| Type          | Key                                                                           |
| Container href | N/A                                                                           |
| Secret href   | None                                                                          |
| Created       | None                                                                          |
| Status        | None                                                                          |
| Error code    | None                                                                          |
| Error message | None                                                                          |
+---------------+-------------------------------------------------------------------------------+
```

Figure 20 : Symmetric Key Generation

2. You can also verify the encryption operation logging in the PKCS11 log file cs_pkcs11_R3.log during secret generation, as shown below.

```
05.01.2023 10:44:20.024 | [00002941:00002941] C_OpenSession                | T: leave C_OpenSession()
05.01.2023 10:44:20.024 | [00002941:00002941] C_GetSessionInfo             | T: enter C_GetSessionInfo(hSession: 0x00000003)
05.01.2023 10:44:20.024 | [00002941:00002941] C_GetSessionInfo             | T: leave C_GetSessionInfo()
05.01.2023 10:44:20.028 | [00002941:00002941] C_GenerateRandom             | T: enter C_GenerateRandom(hSession: 0x00000003, pRandomData: CK_BYTE[20
])
05.01.2023 10:44:20.028 | [00002941:00002941] C_GenerateRandom             | T: leave C_GenerateRandom()
05.01.2023 10:44:20.031 | [00002941:00002941] C_GenerateRandom             | T: enter C_GenerateRandom(hSession: 0x00000003, pRandomData: CK_BYTE[10
])
05.01.2023 10:44:20.031 | [00002941:00002941] C_GenerateRandom             | T: leave C_GenerateRandom()
05.01.2023 10:44:20.031 | [00002941:00002941] C_EncryptInit                | T: enter C_EncryptInit(hSession: 0x00000003, pMechanism: CKM_AES_CBC (0
x1082), hKey: 0x00000003)
05.01.2023 10:44:20.031 | [00002941:00002941] C_EncryptInit                | T: leave C_EncryptInit()
05.01.2023 10:44:20.034 | [00002941:00002941] C_Encrypt                    | T: enter C_Encrypt(hSession: 0x00000003)
05.01.2023 10:44:20.034 | [00002941:00002941] C_Encrypt                    | T: leave C_Encrypt()
05.01.2023 10:44:20.034 | [00002941:00002941] C_CloseSession               | T: enter C_CloseSession(hSession: 0x00000003)
05.01.2023 10:44:20.034 | [00002941:00002941] C_CloseSession               | T: leave C_CloseSession()
```

Figure 21 : pkcs#11 Logs

3. View the details of the order to identify the location of the generated key, which is shown here as the Secret href value.

`›_ Console`

```
# openstack secret order get http://barbican:9311/v1/orders/fb007366-
1965-49b0-97b6-e1306d103a73
```

```
[root@rhelserver user(keystone_admin)]# openstack secret order get http://172.28.14.25:9311/v1/orders/9faf4691-5f71-424c-bd0e-9242bd0ccb3c
Could not load 'message_list': module 'zaqarclient.queues.v2.cli' has no attribute 'OldListMessages'
Could not load 'message_post': module 'zaqarclient.queues.v2.cli' has no attribute 'OldPostMessages'
+----------------+------------------------------------------------------------------------+
| Field          | Value                                                                  |
+----------------+------------------------------------------------------------------------+
| Order href     | http://172.28.14.25:9311/v1/orders/9faf4691-5f71-424c-bd0e-9242bd0ccb3c |
| Type           | Key                                                                    |
| Container href | N/A                                                                    |
| Secret href    | http://172.28.14.25:9311/v1/secrets/9ad5dd8e-d14e-4b4b-a234-a50da3fc81b2 |
| Created        | 2025-06-03T12:21:16+00:00                                              |
| Status         | ACTIVE                                                                 |
| Error code     | None                                                                   |
| Error message  | None                                                                   |
+----------------+------------------------------------------------------------------------+
```

```
[root@rhelserver user(keystone_admin)]# openstack secret order list
Could not load 'message_list': module 'zaqarclient.queues.v2.cli' has no attribute 'OldListMessages'
Could not load 'message_post': module 'zaqarclient.queues.v2.cli' has no attribute 'OldPostMessages'
+-----------------------------------------+------+----------------+-----------------------------------------+----------------------------+--------+------------+---------------+
| Order href                              | Type | Container href | Secret href                             | Created                    | Status | Error code | Error message |
+-----------------------------------------+------+----------------+-----------------------------------------+----------------------------+--------+------------+---------------+
| http://172.28.14.25:9311/v1/orders/b77ca54d- | Key  | N/A            | http://172.28.14.25:9311/v1/secrets/b81b887b | 2025-06-03T12:19:15+00:00 | ACTIVE | None       | None          |
| 8f6c-4b54-b33a-75eff312a1c1             |      |                | -bec8-424b-bddd-d4cfb6474c0b            |                            |        |            |               |
| http://172.28.14.25:9311/v1/orders/42ee7b6d- | Key  | N/A            | http://172.28.14.25:9311/v1/secrets/363e84a5 | 2025-06-03T12:20:37+00:00 | ACTIVE | None       | None          |
| 6c31-49ef-b77d-305ba04a6b79             |      |                | -0d63-413f-b3c5-297dca051274            |                            |        |            |               |
| http://172.28.14.25:9311/v1/orders/9faf4691- | Key  | N/A            | http://172.28.14.25:9311/v1/secrets/9ad5dd8e | 2025-06-03T12:21:16+00:00 | ACTIVE | None       | None          |
| 5f71-424c-bd0e-9242bd0ccb3c             |      |                | -d14e-4b4b-a234-a50da3fc81b2            |                            |        |            |               |
+-----------------------------------------+------+----------------+-----------------------------------------+----------------------------+--------+------------+---------------+
```

Figure 22 : Details of the Order and the order list

4. Retrieve the details of the secret.

`›_ Console`

```
# openstack secret get http://barbican:9311/v1/secrets/731068e6-87834efc-b937-34002224aff6
```

```
[root@controller bin(keystone_admin)]# openstack secret get http://barbican:9311/v1/secrets/731068e6-8783-4efc-b937-34002224aff6
+---------------+-------------------------------------------------------------------------+
| Field         | Value                                                                   |
+---------------+-------------------------------------------------------------------------+
| Secret href   | http://barbican:9311/v1/secrets/731068e6-8783-4efc-b937-34002224aff6 |
| Name          | app_key                                                                 |
| Created       | 2022-12-12T11:54:44+00:00                                               |
| Status        | ACTIVE                                                                  |
| Content types | {'default': 'application/octet-stream'}                                 |
| Algorithm     | aes                                                                     |
| Bit length    | 256                                                                     |
| Secret type   | symmetric                                                               |
| Mode          | ctr                                                                     |
| Expiration    | None                                                                    |
+---------------+-------------------------------------------------------------------------+
```

Figure 23 : Retrieve the secret details

5. Alternatively, you can list the symmetric key that has been generated by the command below.

```
›_ Console
```

```
# openstack secret list
```

```
[root@controller bin(keystone_admin)]# openstack secret list
+------------------------------------------------------------------------+-----------+---------------------------+--------+--------
--------------------------------+-----------+------------+-------------+------+------------+
| Secret href                                                            | Name      | Created                   | Status | Content
 types                         | Algorithm | Bit length | Secret type | Mode | Expiration |
+------------------------------------------------------------------------+-----------+---------------------------+--------+--------
--------------------------------+-----------+------------+-------------+------+------------+
| http://barbican:9311/v1/secrets/8ac7918d-36ce-4cae-b1d9-a2a818ea30a0 | utimaco123 | 2022-12-06T06:38:56+00:00 | ACTIVE | {'defau
lt': 'application/octet-stream'} | aes       |        256 | opaque      | cbc  | None       |
| http://barbican:9311/v1/secrets/731068e6-8783-4efc-b937-34002224aff6 | app_key    | 2022-12-12T11:54:44+00:00 | ACTIVE | {'defau
lt': 'application/octet-stream'} | aes       |        256 | symmetric   | ctr  | None       |
+------------------------------------------------------------------------+-----------+---------------------------+--------+--------
--------------------------------+-----------+------------+-------------+------+------------+
```

Figure 24 : Secret listing

6.5 Storing Public Key, Private Key and Certificate in OpenStackBarbican

1. Create a self-signed certificate using the command below.

```
›_ Console
```

```
# openssl req -x509 -newkey rsa:4096 -keyout private.pem -out cert.pem
-sha256 -days 365 -nodes
```

```
[root@controller ~(keystone_admin)]# openssl req -x509 -newkey rsa:4096 -keyout private.pem -out cert.pem -sha256 -days 365 -nodes
Generating a RSA private key
.....................................................................................++++
.....................................++++
writing new private key to 'private.pem'
-----
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]:IN
State or Province Name (full name) []:MH
Locality Name (eg, city) [Default City]:Pune
Organization Name (eg, company) [Default Company Ltd]:utimaco.com
Organizational Unit Name (eg, section) []:utimaco
Common Name (eg, your name or your server's hostname) []:test.utimaco.com
Email Address []:support@utimaco.com
```

Figure 25 : Create a self-signed certificate

You can generate a key and a certificate by using other utilities as well.

2. Verify that the private key and certificate file are generated.

```
›_ Console
```

```
# ls
```

```
[root@controller ~(keystone_admin)]# ls
24.11.22.conf  cert.pem  cxi.log  key.pem  keystonerc_admin  private.pem  public.pem
[root@controller ~(keystone_admin)]#
```

Figure 26 : File listing

3. Generate a public key from a private key.

```
[root@controller ~]# ll
total 84
-rw-------  1 root root 51310 Dec  2 16:02 24.11.22.conf
-rw-r--r--  1 root root  2143 Dec 12 18:29 cert.pem
-rw-r--r--. 1 root root  9792 Dec  2 10:23 cxi.log
-rw-------  1 root root  3272 Dec 12 18:18 key.pem
-rw-------. 1 root root   361 Nov 29 11:48 keystonerc_admin
-rw-------  1 root root  3272 Dec 12 18:28 private.pem
-rw-r--r--  1 root root   800 Dec 12 18:29 public.pem
[root@controller ~]#
```

Figure 27 : Creation of Public Key

4. Store the public key in OpenStack Barbican.

›_ Console

```
# openssl rsa -in private.pem -out public.pem -pubout

# openstack secret store --algorithm rsa --secret-type public -payloa-
dcontent-type application/octet-stream --payload-content-encoding base64 -

-payload "$(base64 < public.pem)" --bit-length 2048 --name pubtest
```

```
[root@controller ~(keystone_admin)]# openstack secret store --algorithm rsa --secret-type public --payload-content-type applicatio
n/octet-stream --payload-content-encoding base64 --payload "$(base64 < public.pem)" --bit-length 2048 --name pubtest
+---------------+------------------------------------------------------------------+
| Field         | Value                                                            |
+---------------+------------------------------------------------------------------+
| Secret href   | http://barbican:9311/v1/secrets/85c202ff-05b6-4923-9ce6-a6cee3cde2d9 |
| Name          | pubtest                                                          |
| Created       | None                                                             |
| Status        | None                                                             |
| Content types | {'default': 'application/octet-stream'}                          |
| Algorithm     | rsa                                                              |
| Bit length    | 2048                                                             |
| Secret type   | public                                                           |
| Mode          | cbc                                                              |
| Expiration    | None                                                             |
+---------------+------------------------------------------------------------------+
```

Figure 28 : Store Public Key in Openstack Barbican

5. You can also verify the encryption operation logging in the PKCS11 log file cs_pkcs11_R3.log during public secret generation, as shown below.

```
05.01.2023 11:05:46.315 | [00002944:00002944] C_OpenSession                | T: leave C_OpenSession()
05.01.2023 11:05:46.315 | [00002944:00002944] C_GetSessionInfo             | T: enter C_GetSessionInfo(hSession: 0x00000003)
05.01.2023 11:05:46.316 | [00002944:00002944] C_GetSessionInfo             | T: leave C_GetSessionInfo()
05.01.2023 11:05:46.319 | [00002944:00002944] C_GenerateRandom             | T: enter C_GenerateRandom(hSession: 0x00000003, pRandomData: CK_BYTE[10
])
05.01.2023 11:05:46.319 | [00002944:00002944] C_GenerateRandom             | T: leave C_GenerateRandom()
05.01.2023 11:05:46.319 | [00002944:00002944] C_EncryptInit                | T: enter C_EncryptInit(hSession: 0x00000003, pMechanism: CKM_AES_CBC (0
x1082), hKey: 0x00000003)
05.01.2023 11:05:46.319 | [00002944:00002944] C_EncryptInit                | T: leave C_EncryptInit()
05.01.2023 11:05:46.322 | [00002944:00002944] C_Encrypt                    | T: enter C_Encrypt(hSession: 0x00000003)
05.01.2023 11:05:46.322 | [00002944:00002944] C_Encrypt                    | T: leave C_Encrypt()
05.01.2023 11:05:46.323 | [00002944:00002944] C_CloseSession               | T: enter C_CloseSession(hSession: 0x00000003)
05.01.2023 11:05:46.323 | [00002944:00002944] C_CloseSession               | T: leave C_CloseSession()
```

Figure 29 : pkcs#11 Logs

6. Get the value of the public key.

›_ Console

```
# openstack secret get -p -c Payload -f value

http://barbican:9311/v1/secrets/85c202ff-05b6-4923-9ce6-a6cee3cde2d9
```

```
[root@controller ~(keystone_admin)]# openstack secret get -p -c Payload -f value http://barbican:9311/v1/secrets/85c202ff-05b6-492
3-9ce6-a6cee3cde2d9
-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAyoaYwpbZSLC9+xCvUWpN
RpTxBiu0f2ln2LOrGmxTfrK8se3CX0n3ZwWHsaBDrKfi6hNF0pgaGlIBQ/F85IE1
hVxjVHBS7NYEFZ1lVG6pHi00WMcP1WOGvoZBPSHwYuq2MciB78PaIC3Yn/IjASy9
BcUrD2JTaZ2YppDHgbGuWd7/mmc9TKB0n5M5SFiqfjNXeHvIYzlvxg4teWMrdGdT
v1lIzJqHFFLIncQOZbkMS0ZoobsDW9suNC7oyeKMX01LqWIb1sT2PfTYXcYLZG4q
/3ogZ69Doox8T/RjFKRlq0JJvj47EzmAoJs9ZkZMA5rAdN/rnkdEnB6E/UPWrhSv
ydqJWtIrmJ50lsFRWHfV0R55KXjgsqKKK+lCFMVZB//uzV+KiJTD9e68+8tknxOY
0GSX/gk2xk1HAupq5Iae7/7KDGrUwpS/JcDW7o4zDZSgIhZZlwlv735fHDn+oOlU
818uOgCKHROfCaGcvYhaGDlJBNx5dH9VxWUxMBzFrdrMIPm0SnAOs+TgAg+Fw8F+
CMAR0UfC3U4oGu6ibtVYi/QAzxaRlzMaW712EAN5QH7dYhSpH3jNwnhfvXaXB5Sd
sw45fjWB7R9DZ/pVhva608s4RXOBFOzDMg6oNsrzI8pr8gyNSFT2MvzTguIv66IU
Isy/c8KSay/t3QbXtOj9hsUCAwEAAQ==
-----END PUBLIC KEY-----
```

Figure 30 : Get Public Key from Openstack Barbican

7. Store the Private key in OpenStack Barbican.

›_ Console

```
# openstack secret store --algorithm rsa --secret-type private -payload-
content-type application/octet-stream --payload-content-encoding base64 -

-payload "$(base64 < private.pem)" --bit-length 2048 --name privatekeytest
```

```
[root@controller ~(keystone_admin)]# openstack secret store --algorithm rsa --secret-type private --payload-content-type applicati
on/octet-stream --payload-content-encoding base64 --payload "$(base64 < private.pem)" --bit-length 2048 --name privatekeytest
+---------------+------------------------------------------------------------------+
| Field         | Value                                                            |
+---------------+------------------------------------------------------------------+
| Secret href   | http://barbican:9311/v1/secrets/39e9bf4e-638a-4e4c-9cb3-2caebfb88b11 |
| Name          | privatekeytest                                                   |
| Created       | None                                                             |
| Status        | None                                                             |
| Content types | {'default': 'application/octet-stream'}                          |
| Algorithm     | rsa                                                              |
| Bit length    | 2048                                                             |
| Secret type   | private                                                          |
| Mode          | cbc                                                              |
| Expiration    | None                                                             |
+---------------+------------------------------------------------------------------+
```

Figure 31 : Store Private Key in Openstack Barbican

8. Get the value of the Private key.

›_ Console

```
# openstack secret get -p -c Payload -f value

http://barbican:9311/v1/secrets/39e9bf4e-638a-4e4c-9cb3-2caebfb88b11
```

```
[root@controller ~(keystone_admin)]# openstack secret get -p -c Payload -f value http://barbican:9311/v1/secrets/39e9bf4e-638a-4e4
c-9cb3-2caebfb88b11
-----BEGIN PRIVATE KEY-----
MIIJQwIBADANBgkqhkiG9w0BAQEFAASCCS0wggkpAgEAAoICAQDKhpjCltlIsL37
EK9Rak1GlPEGK7R/aWfYs6sabFN+sryx7cJfSfdnBYexoEOsp+LqE0XSmBoaUgFD
8XzkgTWFXGNUcFLs1gQVnWVUbqkeLTRYxw/VY4a+hkE9IfBi6rYxyIHvw9ogLdif
8iMBLL0FxSsPYlNpnZimkMeBsa5Z3v+aZz1MoHSfkzlIWKp+M1d4e8hjOW/GDi15
Yyt0Z1O/WUjMmocUUsidxA5luQxLRmihuwNb2y40LujJ4oxfTUupYhvWxPY99Nhd
xgtkbir/eiBnr0OijHxP9GMUpGWrQkm+PjsTOYCgmz1mRkwDmsB03+ueR0ScHoT9
Q9auFK/J2ola0iuYnnSWwVFYd9XRHnkpeOCyooor6UIUxVkH/+7NX4qIlMP17rz7
y2SfE5jQZJf+CTbGTUcC6mrkhp7v/soMatTClL8lwNbujjMNlKAiFlmXCW/vfl8c
Of6g6VTzXy46AIodE58JoZy9iFoYOUkE3Hl0f1XFZTEwHMWt2swg+bRKcA6z5OAC
D4XDwX4IwBHRR8LdTiga7qJu1ViL9ADPFpGXMxpbvXYQA3lAft1iFKkfeM3CeF+9
dpcHlJ2zDjl+NYHtH0Nn+lWG9rrTyzhFc4EU7MMyDqg2yvMjymvyDI1IVPYy/NOC
4i/rohQizL9zwpJrL+3dBte06P2GxQIDAQABAoICACaUJR1MJu/V2xqsPvk/SqEb
Vh5azPNIXOPGtd0+rVkBKziplEZAeu5/fM0GDAXev4j1bUcDDzfSiZrJeNRSaSx
yKoZC3LaAcUsSlUfD1AURFh08usvHkfEw0Cl6mitVS9+hmHp6O8gTJeLRGfA4fDr
chxoGoOP5bg09AwckJ3GAOKkr+Sc3BpZrRu+6BtKJGtrCOz9z/DmWns5DNoLpMRa
eaef+ANDEbwb0LJH4bOlIeFNAoLI6bA/D1EFur+ctcD6U1tCAbYjoeObCiiDY+Vu
miWmGTgpkrSYVRdliYhqBCodKJus5HezmqqU0X4OSIL69KMd6E/k8GheN9LJVgLC
wlogv93O3QOAE4pI8wvXigPMLNbESgQiAPcJhsgJlQ7kw++4SYGVl0BR+gx/DJ7S
kMv4AvdUgxF1YREJR5RftwNatTQaKb/vognZaRsJvPrb8LAqTym5nEvEjcbCpnEq
eX9ElMX3k98agTjC6BVuCZ+FrhERb6mqOjcERtzjgevjYCVSjl+EB0/Pj/7RvDX7
S3hshh0ebSeAWyo3FavLqno+O5WyPcfm0OasfeMbOkctCio4eieHkKhG83ZqnA8p
1UgiUGfFnj+y0dhXFI/tBo05X7gFXkj/ouXka+T25RShyEF+T85cwDlD/sFCArYh
LqoUHQS1cd+tsM3XwEqBAoIBAQDkwwxAj7mF8GbdGRtg+CSlvAXyxqswSHw9c3iW
Gwln/RlFUzJansvY4pXMmQKP2l7M3k5oFlvMutP6V8R2lgQjQGRqawx7r/fd6yAS
PRhSsmktQ5MgxhBNKIxcLZXi1251T4sBaU3OgDFism9jkjaNGC3KfVGJvKDMAJuk
7LGkBRuMaN7xZLSuxh6CiFVNCbLww6cKoQl1jAdP5R9rE6lhvTAT0IXcdnk4us18
9GHQ9ArBU5BlGFe9f6ctR/6rrV0Pob8vbSuobefw7ffdjIR61n7zc4l21jqRWO2j
9Sq8YnFw5ZaOWR7XVzCCMwiR1Ubf0BCs3hM6cotVWdGxf3qtAoIBAQDio9aVq1Jd
7wRlWMEOt9gOUHgg86nAKg3q3J1yIyPoTmHLQAWCjFA+2TX8STkT68PhZD7+FTEt
aDdLkBTw70utXsa5xymFIViC1FEpTQ8zfej1USczPPEjzTawGNQ5Uja5laVGaDN/
Ug4rYvqjvPceopOo9dinqNQC1yWu7gqvDOs7/zRRjnnfGL5qoB7F7NXxZeBAm2/Y
uiPpfdonv8gMw0t+qA0zIlLMZ6VWRt4AEnfnR6E7e6vSk6SblctZQyURnQmXvxQv
mEhR4god63rWMieei/+q8tLy74f/Vl4Na8csPzWejwnhwxFoG57ijumgObeDePDN
kIphq+nEcxd5AoIBAERwkj7SgQewV6X4N/e35YbsDL+IbnjghsL/w+jsp9DDqQbT
kDIKZiQ6DeULNfgBtfEG5WBAqDBFCNxR28Y6tXwybJhl9HMgv0BWct7iBl/fXpYb
LG5Ws1PSOMPvI3OdTj+sj4LZlVydf6juh2K7wLRPsottGLLjxvdZrNd2PugR5Cn
Y2PG5EORpFkbHqIQr8dWUahda24OebWOb+riE//Y0QRsk0/U5glx4b5rgsQ37uvN
61pDgSSBh2uFqYyFLl1Z38/5kmEumIC9QTW4L35so8iEHXE/QIBxoY827xUzQosr
NPLbeN2mCS46xGqU60JUhlihjBElacgcOh8na3UCggEBAJ+SWaCxqmGmy0KMFgnF
2WJit+IDjU5WjcwCHvzgzwReQ662WerL3igSCVejOoPTKOyGB3Af/asJyHz0ajol
UfQ0Wdkt6AbrLoWW9boNnplTmnU2+VGzI9lgQJQGN05Qm2VZfq+u6CtJ294iZ3Rn
tNql/y8kQbHyRavmTYr34R/VCb8JdY3Csf4pTtCGAoS6tvurDhSPHFXM4UIhsTrx
ZenOi1YW/ow1K3TtuNShOud0GWYG4feeoKC4AVrmQfGMPApFth5Yk97a6nh0nUjp
pmgvkgOocI+tPmd7zRYzei8a5nY53wfPkFabYKMbZ2f+1XFRkesIeQtTLIuOXh5K
JokCggEBAIzt6n3QlBssMMKfmdS1JG2l3iWfuVZcXUkVNza/zqkhR7hv0A4AoAcc
SbNxQOsHtJKrXlRSH6kFu22RFywPk7siBy8rb5jIjNIl6esrqsfCxhN4CvldRYKt
Wf1SGqacJAiVfcvf9JYfB7ywibDScXORg2LOqetQy2dXmqmvRy+ynGZz9dALj5KO
ZMe7Tp7YCLYwYwyChG77Jlqhrg666zrQkoNc/2Q8Fdt15s7io637+eRtPtuFy7bE
BnxbcoFt1j33pjBPdCfhHGLIP1wPjPJSURwrU6TjDOB7cTzNn4G+bf/FCBBnY6DH
wULb12pZMGyrnXjr724ETQeP1z4vvKc=
-----END PRIVATE KEY-----
```

Figure 32 : Get Private key from Openstack Barbican

9. Store the certificate on OpenStack Barbican.

```
›_ Console
```

```
# openstack secret store --algorithm rsa --bit-length 2048  --secret-type certificate --payload-content-type application/octet-stream -payload-content-encoding base64 --payload "$(base64 < cert.pem)” --name certificatetestutimaco
```

```
[root@controller ~(keystone_admin)]# openstack secret store --algorithm rsa --bit-length 2048  --secret-type certificate --payload
-content-type application/octet-stream --payload-content-encoding base64 --payload "$(base64 < cert.pem)"  --name certificatetestu
timaco
+---------------+------------------------------------------------------------------+
| Field         | Value                                                            |
+---------------+------------------------------------------------------------------+
| Secret href   | http://barbican:9311/v1/secrets/9bba64eb-c875-4d72-9058-2b30fa907c17 |
| Name          | certificatetestutimaco                                           |
| Created       | None                                                             |
| Status        | None                                                             |
| Content types | {'default': 'application/octet-stream'}                          |
| Algorithm     | rsa                                                              |
| Bit length    | 2048                                                             |
| Secret type   | certificate                                                      |
| Mode          | cbc                                                              |
| Expiration    | None                                                             |
+---------------+------------------------------------------------------------------+
```

Figure 33 : Store certificate in OpenStack Barbican

10. Get the value of the certificate from OpenStack Barbican.

```
›_ Console

# openstack secret get -p -c Payload -f value

http://barbican:9311/v1/secrets/9bba64eb-c875-4d72-9058-2b30fa907c17
```

```
[root@controller ~(keystone_admin)]# openstack secret get -p -c Payload -f value http://barbican:9311/v1/secrets/9bba64eb-c875-4d7
2-9058-2b30fa907c17
-----BEGIN CERTIFICATE-----
MIIGATCCA+mgAwIBAgIUJH3q+N4OCpMYdwBOn2DHVDuaSRwwDQYJKoZIhvcNAQEL
BQAwgY8xCzAJBgNVBAYTAklOMQswCQYDVQQIDAJNSDENMAsGA1UEBwwEUFVORTEQ
MA4GA1UECgwHVXRpbWFjbzEQMA4GA1UECwwHVGVzdGluZzEcMBoGA1UEAwwTdGVz
dGluZy51dGltYWNvLmNvbTEiMCAGCSqGSIb3DQEJARYTc3VwcG9ydEB1dGltYWNv
LmNvbTAeFw0yMjEyMTIxMjU5MTFaFw0yMzEyMTIxMjU5MTFaMIGPMQswCQYDVQQG
EwJJTjELMAkGA1UECAwCTUgxDTALBgNVBAcMBFBVTkUxEDAOBgNVBAoMB1V0aW1h
Y28xEDAOBgNVBAsMB1Rlc3RpbmcxHDAaBgNVBAMME3Rlc3RpbmcudXRpbWFjby5j
b20xIjAgBgkqhkiG9w0BCQEWE3N1cHBvcnRAdXRpbWFjby5jb20wggIiMA0GCSqG
SIb3DQEBAQUAA4ICDwAwggIKAoICAQDKhpjCltlIsL37EK9Rak1GlPEGK7R/aWfY
s6sabFN+sryx7cJfSfdnBYexoEOsp+LqE0XSmBoaUgFD8XzkgTWFXGNUcFLs1gQV
nWVUbqkeLTRYxw/VY4a+hkE9IfBi6rYxyIHvw9ogLdif8iMBLL0FxSsPYlNpnZim
kMeBsa5Z3v+aZz1MoHSfkzlIWKp+M1d4e8hjOW/GDi15Yyt0Z1O/WUjMmocUUsid
xA5luQxLRmihuwNb2y40LujJ4oxfTUupYhvWxPY99Nhdxgtkbir/eiBnr0OijHxP
9GMUpGWrQkm+PjsTOYCgmz1mRkwDmsB03+ueR0ScHoT9Q9auFK/J2ola0iuYnnSW
wVFYd9XRHhkpeOCyooor6UIUxVkH/+7NX4qIlMP17rz7y2SfE5jQZJf+CTbGTUcC
6mrkhp7v/soMatTClL8lwNbujjMNlKAiFlmXCW/vfl8cOf6g6VTzXy46AIodE58J
oZy9iFoYOUkE3Hl0f1XFZTEwHMWt2swg+bRKcA6z5OACD4XDwX4IwBHRR8LdTiga
7qJu1ViL9ADPFpGXMxpbvXYQA3lAft1iFKkfeM3CeF+9dpcHlJ2zDjl+NYHtH0Nn
+lWG9rrTyzhFc4EU7MMyDqg2yvMjymvyDI1IVPYy/NOC4i/rohQizL9zwpJrL+3d
Bte06P2GxQIDAQABo1MwUTAdBgNVHQ4EFgQUncuKAgvd4sifk62YgOfw9CrDzUQw
HwYDVR0jBBgwFoAUncuKAgvd4sifk62YgOfw9CrDzUQwDwYDVR0TAQH/BAUwAwEB
/zANBgkqhkiG9w0BAQsFAAOCAgEADj65v/A+Y9XYASuCF8GITkR31OOBOC27I4Ug
7v+GaoaDynZsvF+qyaF+DRYXs4hEPuWAzyAO+U3LP8wwkzzUzhg5a+jvbTwxSZZl
xtQ+rCe/ipx0KHlcqmv3OGJxp4Ps+dtz9CZ1jHwPw3ehUaPOFyeKPdRVQxrvbORu
AnPp+dHeRRIosQG+d6514xRCGuvWKSGA1kmMq80FLCYL61HJtbY1DzJohWAYkCO0
zcHd+G4UNMqht2rrjAjOJxohc25pqp7uBQABLdpuD440+/pE1b7KDPhW6S4vqo1S
blFNf6JMwOJEH+w1Eq/6JvJyyorWkzsOeTggbrxLqw5G2qr37yEnjjmNPX8kuTp4
1tKsdZtv22pBYDZ+SXJd+aggAqpSbfzqDkikIvWZxA4wkyyDhO5MXuQLfwomHNtb
0DeJ1rOZTdxLmS0CojLWrGqaHmcce/a1+ddOaiWb6OE1L08S5SCZP1SpowZ+LN65
3ZZJyKvHu1HR3GWYotje0pg6mfY+mAZVaf+idnwzwEF6269P8TnbwycPNFAgkGhE
/1hoWP1QBRS0sf8UjVxPEtulFN2FMlf8woHVt4Ou4ShaqQ4sHNn15KwvL5KP0fV1
eZEwOUP2DSoXGYw9Nt+eveCiT1GRaWhe6yYsBzQ1yFbEnOLhSBApzzzy+/DjJSi+
50wEtTM=
-----END CERTIFICATE-----
```

Figure 34 : Get Certificate from Openstack Barbican

11. You can verify all the secrets.

›_ Console

```
# openstack secret list
```

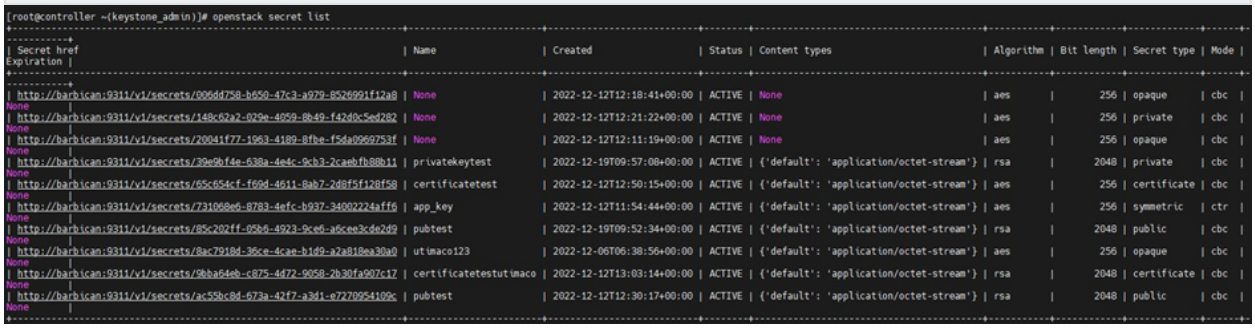

Figure 35 : Secret List

6.6 Keys Rotation/Migration

Use the steps below to rotate the keys.

1. Create a new MKEK Key by using the command below.

›_ Console

```
# su -s /bin/sh -c "barbican-manage hsm gen_mkek --library-path '/opt/
utimaco/lib/libcs_pkcs11_R3.so' --passphrase 1234567890  --slot-id 3 --
label 'mkek_utimaco1234' --length 32" barbican

'/etc/libcs_pkcs11_R3.so' --passphrase <Crypto_User_PIN>  -slot-id
<slot_id>  --label 'mkek_utimaco1234' --length 32
```

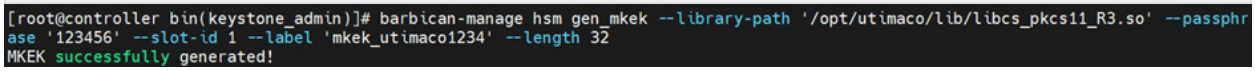

Figure 36 : MKEK Key Generation

2. Create a New HMAC Key by using the below command.

›_ Console

```
# su -s /bin/sh -c "barbican-manage hsm gen_hmac --library-path '/opt/utimaco/lib/libcs_pkcs11_R3.so' --passphrase 1234567890 --slot-id 3  --label 'hmac_utimaco1234' --length 32" barbican
```

```
[root@controller bin(keystone_admin)]# barbican-manage hsm gen_hmac --library-path '/opt/utimaco/lib/libcs_pkcs11_R3.so' --passphrase '123456' --slot-id 1 --label 'hmac_utimaco1234' --length 32
HMAC successfully generated!
```

Figure 37 : HMAC Key Generation

3. Verify that the keys are generated on the Utimaco HSM using the p11tool2 command.

›_ Console

```
#./p11tool2 slot=<slot_id> LOGINUSER=<Crypto_User_PIN> ListObjects
```

```
[root@controller bin(keystone_admin)]# ./p11tool2 slot=1 LOGINUSER=123456 ListObjects


CKO_SECRET_KEY:

+ 1.1
  CKA_KEY_TYPE                  = CKK_AES
  CKA_UNIQUE_ID                 = CK_UNAVAILABLE_INFORMATION
  CKA_SENSITIVE                 = CK_TRUE
  CKA_EXTRACTABLE               = CK_FALSE
  CKA_LABEL                     = mkek_utimaco1234
  CKA_ID                        =


+ 1.2
  CKA_KEY_TYPE                  = CKK_AES
  CKA_UNIQUE_ID                 = CK_UNAVAILABLE_INFORMATION
  CKA_SENSITIVE                 = CK_TRUE
  CKA_EXTRACTABLE               = CK_FALSE
  CKA_LABEL                     = hmac_utimaco1234
  CKA_ID                        =

[root@controller bin(keystone_admin)]#
```

Figure 38 : Key Listing

4. Add a new label to the Barbican config file.

```
# Master KEK label (as stored in the HSM) (string value)
mkek_label = mkek_utimaco1234

# Master KEK length in bytes. (integer value)
mkek_length = 32

# Master HMAC Key label (as stored in the HSM) (string value)
hmac_label = hmac_utimaco1234
```

Figure 39 : Entries in barbican.conf file

5. Restart OpenStack-Barbican-API and the https service.

›_ Console

```
# systemctl restart openstack-barbican-api.service

# systemctl restart httpd
```

6. Run the rewrap_pkek command to rewrap pkek with the new mkek.

›_ Console

```
# su -s /bin/sh -c "barbican-manage hsm rewrap_pkek" barbican
```

```
[root@controller bin(keystone_admin)]# barbican-manage hsm rewrap_pkek
2022-12-09 11:31:12.007 1222190 WARNING oslo_db.sqlalchemy.engines [-] MySQL SQL mode is 'STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION', consider enabling TRADITIONAL or STRICT_ALL_TABLES
Retrieving all available projects
Retrieving KEKs for Project 4a60eab7-d73a-43f2-8837-9b6d1310d7b5
Post-change IV: b'7KfVkPss/qYh2pjeAoByKw==', Wrapped Key: b'6VlxgtIS+FR9zjBCu5PfQ3d7AmsjqPl9cnLgs+fsptCXjSwvzAE+fW+lEsDSpVwW'
```

Figure 40 : Key rewrap Output

7. Restart the OpenStack-Barbican-API and httpd services.

›_ Console

```
# systemctl restart openstack-barbican-api.service

# systemctl restart httpd
```

8. Verify that you can get the secret generated earlier.

```
›_ Console

# openstack secret get http://barbican:9311/v1/secrets/8ac7918d-36ce-

4cae-b1d9-a2a818ea30a0 --payload
```

```
[root@controller ~(keystone_admin)]# openstack secret get http://barbican:9311/v1/secrets/8ac7918d-36ce-4cae-b1d9-a2a818ea30a0 --p
ayload
+---------+----------+
| Field   | Value    |
+---------+----------+
| Payload | password |
+---------+----------+
[root@controller ~(keystone_admin)]#
```

Figure 41 : Get secret with payload

This completes the integration of OpenStack Barbican and Utimaco HSM.

6.7 Using Different Encryption Mechanisms

The encryption algorithm used to encrypt secret payloads before they are stored in the database is configurable and depends on the PKCS#11 mechanism supported by the connected HSM.

By default, AES in CBC mode (CKM_AES_CBC) is used to encrypt the payloads. However, Barbican and u.Trust GP HSM also support additional encryption mechanisms, such as AES in GCM mode (CKM_AES_GCM).

Follow these steps to change the encryption mechanism, and to store a secret, and verify its use:

1. Edit the Barbican configuration file `barbican.conf` and update the `encryption_mechanism` parameter under `p11_crypto_plugin]`.

barbican.conf

```
[p11_crypto_plugin]
# Path to Utimaco PKCS11 library

library_path = /opt/utimaco/lib/libcs_pkcs11_R3.so
# CryptoUser PIN to login to PKCS11

login = <PKCS11 Slot User PIN>
# Master KEK label as stored in the HSM

mkek_label = mkek_utimaco
# Master KEK length in bytes. (integer value)

mkek_length = 32
# Master HMAC Key label (as stored in the HSM) (string value)

hmac_label = hmac_utimaco
# HSM Slot ID (integer value)

slot_id = 3

encryption_mechanism = CKM_AES_GCM
```

Save the file after making the change.

It is important to note that changing the encryption mechanism affects only newly created secrets. Existing secrets remain encrypted using the mechanism that was active at the time of their creation and can not be decrypted with a different mechanism.

2. Restart the OpenStack-Barbican-API and httpd services.

›_ Console

```
# systemctl restart openstack-barbican-api.service

# systemctl restart httpd
```

3. Generate a new secret.

```
›_ Console

# openstack secret store --name gcm_test_secret --payload testingGCM
```

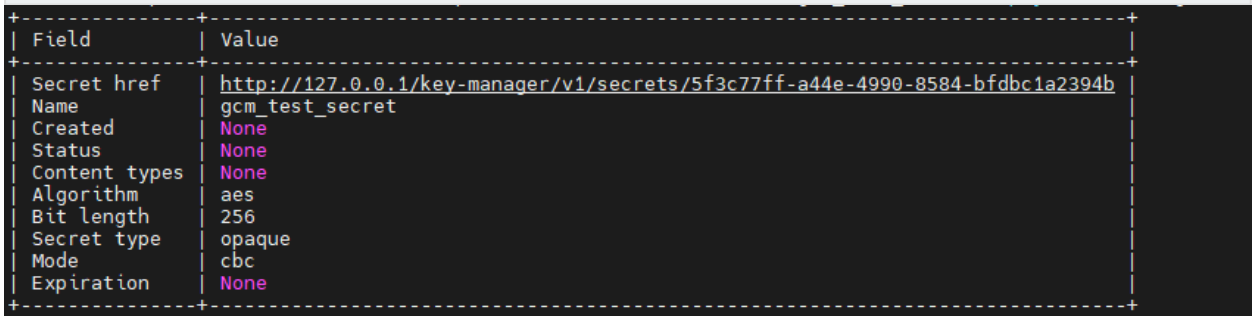

Figure 42 : Storing a secret after changing encryption mechanism

4. Verify the encryption mechanism used via Barbican Database.

```
›_ Console

# mysql
```

```
›_ mysql console

# USE barbican;

# SELECT secret_id, kek_meta_extended FROM encrypted_data ORDER BY
created_at DESC;
```

The newly created secret will appear at the top of the table.

```
mysql> USE barbican;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT secret_id, kek_meta_extended FROM encrypted_data ORDER BY created_at DESC;
+--------------------------------------+-------------------------------------------------------------+
| secret_id                            | kek_meta_extended                                           |
+--------------------------------------+-------------------------------------------------------------+
| 5f3c77ff-a44e-4990-8584-bfdbc1a2394b | {"iv":"YXdePkGhtGK6iBxB","mechanism":"CKM_AES_GCM"}         |
| 4d29326d-303d-44b7-bc2f-3b09bd81df46 | {"iv":"Af96wzNBDWzGO693fYk7Ew==","mechanism":"CKM_AES_CBC"} |
| 5dc2bbe1-1745-470c-b98d-5d82a9eec7eb | {"iv":"5ndybKl6P3CFjlV7LmnuYg==","mechanism":"CKM_AES_CBC"} |
| 284c94aa-47b7-48f6-90d9-98327bd3843a | {"iv":"hSEaD78v+f9onBvVrR9Iqg==","mechanism":"CKM_AES_CBC"} |
| e5b9436e-e16b-4973-8b3f-5571a5359914 | {"iv":"MkSw31S/OT+e27t/","mechanism":"CKM_AES_GCM"}         |
| 60dc47f7-635b-407f-a075-339e48149bff | {"iv":"/tSKKvYc3SAYdx7P","mechanism":"CKM_AES_GCM"}         |
| 2b1a448e-bfd4-4852-9c06-b310025f5773 | {"iv":"2kWNuYP+z/+sBvxW","mechanism":"CKM_AES_GCM"}         |
| 1f1e1b57-c31f-4fae-af30-49040acd69b0 | {"iv":"71qgkEExU6IlUBqm","mechanism":"CKM_AES_GCM"}         |
| 4427b632-5861-4585-a8f2-9cb14e7f6aa0 | {"iv":"LYxrcAXLoK6eMJrz","mechanism":"CKM_AES_GCM"}         |
| 86b7bced-7f7e-41a3-9d69-ea03ae47b9d7 | {"iv":"9sMQbCX65ORprveb","mechanism":"CKM_AES_GCM"}         |
| 994fc99d-2371-4aa3-830c-f0fcc6b31875 | {"iv":"DQfZG7ZkmhcKddKn","mechanism":"CKM_AES_GCM"}         |
| 57da9dc6-7e1e-4499-bbbb-90337b60d54a | {"iv":"aemgqjGow+M0M2N8elH6oA==","mechanism":"CKM_AES_CBC"} |
| 56f06d5d-3d5f-4fe1-a3b3-1f4b85b77f32 | {"iv":"icMlZQ25cgJxSNBJ2NbIYQ==","mechanism":"CKM_AES_CBC"} |
| 4f9c6e45-6e8b-4d82-b626-2da021fc0af5 | {"iv":"b53K5zuuI5QNefq/di9rYw==","mechanism":"CKM_AES_CBC"} |
| 63134ef8-8e24-4225-8349-244695439a20 | {"iv":"tlQfOTpuC/Z4y7mA2iC4Jg==","mechanism":"CKM_AES_CBC"} |
| 9ab07141-e417-4b0c-b832-07b4a298ff13 | {"iv":"SIFmyB7At25zuab9","mechanism":"CKM_AES_GCM"}         |
| 602ff4da-4183-44e1-868e-406c8cfa54fe | {"iv":"6kmNQjxZU0hUADlI","mechanism":"CKM_AES_GCM"}         |
+--------------------------------------+-------------------------------------------------------------+
17 rows in set (0.00 sec)
```

Figure 43 : Table showing encryption mechanism used

5. Confirm that the secret payload can be retrieved.

```
›_ Console
```

```
# openstack secret get <secret_href> --payload
```

```
+---------+------------+
| Field   | Value      |
+---------+------------+
| Payload | testingGCM |
+---------+------------+
```

Figure 44 : Retrieving secret payload

6. Change the `encryption_mechanism` parameter in the `barbican.conf` back to `CKM_AES_CBC` and restart the Barbican and httpd services.
7. Attempting to retrieve the secret payload again results in an internal server error.

```
5xx Server error: Internal Server Error: Secret payload retrieval failure seen - please contact site administrator.
Internal Server Error: Secret payload retrieval failure seen - please contact site administrator.
```

Figure 45 : Internal server error after attempting to retrieve payload with wrong mechanism

8. An error also appears in the PKCS#11 logs that indicates that the decryption operation was attempted using parameters that do not match the original encryption mechanism. In this case the parameter is the initialization vector (IV) format and length as CKM_AES_CBC requires a 16-byte IV and CKM_AES_GCM requires a GCM parameter structure containing an IV (typically 12 bytes). If an IV does not match the requirements of the selected mechanism, the HSM rejects the operation.

```
stack@ub22-openstack-barbican:/root$ tail  /tmp/cs_pkcs11_R3.log
15.02.2026 16:35:25.066 | [00452591:00452591] C_Decrypt                    | E: Error CKR_GENERAL_ERROR occurred.
15.02.2026 16:35:25.369 | [00452591:00452591] C_Decrypt                    | E: Utimaco::HSM::DeviceException(error_code = 0xb0680032)
                                                                                 thrown in execute
                                                                                    Error occurred on device 3001@localhost:
                                                                                Error B0680032
                                                                                      CryptoServer module CXI
                                                                                      invalid IV length

15.02.2026 16:35:25.370 | [00452591:00452591] mapToP11                    | E: error 0xb0680032 mapped.
15.02.2026 16:35:25.370 | [00452591:00452591] C_Decrypt                    | E: Error CKR_GENERAL_ERROR occurred.
```

Figure 46 : Invalid IV length error in PKCS#11 logs

7 Further Information

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

All SecurityServer product documentation is also available at the Utimaco IS GmbH website:

https://utimaco.com/

7.1 References

Reference	*Title/Company*
[CSADMIN]	utrust_Anchor_cHSM_Manual_Administrators.pdf