Quintessence Labs

APPLICATION NOTE

Trusted Security Foundation® (TSF®)

Using TSF Key and Policy Manager as an HSM for Hashicorp Vault

Overview

This guide describes the procedure to configure Hashicorp Vault's PKCS#11 seal mechanism to use a QuintessenceLabs Trusted Security Foundation (TSF) key and policy manager appliance as its HSM.

Intended audience: Hashicorp Vault administrators
Assumptions: Familiarity with QuintessenceLabs TSF and Hashicorp Vault Enterprise Edition products
Versions:

- QuintessenceLabs TSF version 1.8 or later
- QuintessenceLabs PKCS#11 provider
- Hashicorp Vault Enterprise Edition with HSM support

There are two main tasks: performing a TSF pre-setup and configuring Hashicorp Vault.

Task 1: TSF Pre-setup

Step 1: Creating and Downloading a TSF KMIP Connection Pack

The QuintessenceLabs PKCS#11 provider requires SSL credentials to communicate with the TSF. This comes in the form of a connection pack that can be created via the web management interface of the TSF.

To create a client connection pack you will have to do the following at the TSF web interface:

1. Create a new KMIP client credential
2. Associate the KMIP client credential with a KMIP client entity

This can be done by:

1. Login to the web interface as the 'root' user or a user with enough privilege to create a KMIP client and KMIP client credential.
2. Create a new credential for a KMIP client:
   a. Navigate to the credential generation page by clicking PKI Management → Credentials → generate.
   b. Fill in the entries of the 'Generate New Credential' selecting 'Role' to be 'client'. If using with the qRand™ quantum entropy enhancer do not provide a private key password.
   c. Click generate.
3. Create a new KMIP client (using the previously created credential):
   a. Navigate to the 'Add Client' page by clicking KMIP Clients → Clients → add a client.
   b. Fill in form ensuring to:
      i. Not provide a password
      ii. Select the newly created credential
      iii. Setting 'Action Policy' to 'Default All Allowed'
   c. Click 'Add Client' button to add.
4. Download the client connection pack for distribution by clicking 'download connection pack' of the corresponding client shown in the KMIP Clients list view. You will be shown a form. Download the credential by clicking 'Download Connection Pack'.
5. Securely move the downloaded client connection pack to the Hashicorp Vault instance.

Step 2: Install the QuintessenceLabs PKCS#11 Provider

For Vault to use TSF as its virtual HSM, the QuintessenceLabs PKCS#11 provider needs be installed.

1. Extract the QuintessenceLabs PKCS#11 provider

```
$ tar -xf <package-name>.tar.gz
```

2. To install the PKCS#11 provider, run the `install.sh` script inside the package. We will be installing to a run directory `/opt/qlabs/p6r`. Provide a token password.

```
$ python install.py -p /opt/qlabs/p6r/ -c <connection-pack>.tar
```

3. In the run directory, run the `config-test.py` utility to test the connection.

Task 2: Configure Vault

Step 1: Configuring Vault

After this step Vault will be configured to use TSF to protect its Seal key.

1. Create a Vault config file containing the example entries shown in the example config below.

   Example snippet of the stanza:

```
# Provide the QuintessenceLabs PKCS#11 connection information
seal "pkcs11" {
   lib = "/opt/qlabs/p6r/libp6pkcs11.so"
   slot = "0"
   pin = "password"
   key_label = "vault_seal_hsm_demo"
   hmac_key_label = "vault_seal_hsm_hmac_demo"
   generate_key = "true"
}

# Configure the storage backend for Vault
storage "file" {
   path = "/tmp/vault"
}

# Disable mlock (optional)
Disable_mlock = true
```

The Vault's configuration file must have the PKCS#11 seal stanza defined to provide the necessary connection information for TSF, specifically:

1. 'slot' id of the PKCS#11 token slot (should be '0' if default QuintessenceLabs PKCS#11 provider configuration is used)
2. 'pin' password for the PKCS#11 token (assigned during installation of the QuintessenceLabs PKCS#11 provider)
3. 'key_label' for the encryption key
4. 'hmac_key_label' for the HMAC key
5. 'generate_key' to generate key

For demonstration purposes the backend storage will be a file (as defined in the 'storage' stanza).

Step 2: Initializing Vault

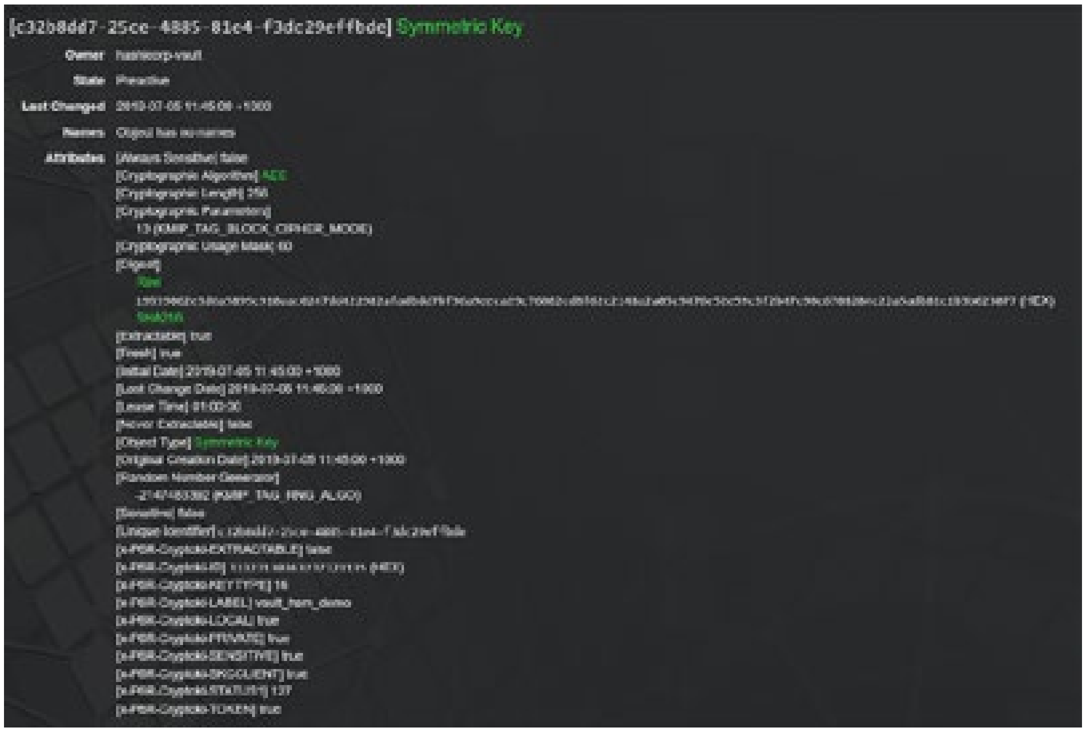
For this demonstration we will initialize Vault with a key-share of 1 and a quorum of 1. For production deployment you will want to select an appropriate quorum configuration to satisfy your requirements (e.g. key-shares=5 and key-threshold=3).

```
$ vault operator init -key-shares=1 -key-threshold=1

Recovery Key 1: hU+HaJaO/Cr42q+hxEXLe62bQne8X4YeZu5aiuOHFnFx

Initial Root Token: 471UTM7RQXF3pkASVAsBaIU4

Success! Vault is initialized

Recovery key initialized with 1 key shares and a key threshold of 1.

Please securely distribute the key shares printed above.
```

Check that the Vault has been `Initialized` by running `Vault Status`.
Notice that the `Initialized` is *true* and `Sealed` is *false*.

```
$ vault status

Key                         Value
---                         -----
Recovery Seal Type          shamir
Initialized                 true
Sealed                      false
Total Recovery Shares       1
Threshold                   1
Version                     0.11.5+ent.hsm
Cluster Name                vault-cluster-52aaa569
Cluster ID                  fa9d3fd0-e006-5a79-ae2f-600c1ce869b4
HA Enabled                  false
```

You can also confirm the creation of the key on the TSF web interface. Note down the created KMIP key id shown in the log file. Navigate to TSF's managed object page and search for object names created by Vault. A screenshot of TSF's managed object page for Vault's key is shown below. Vault's object names will be identified by the x-attribute of an object under [x-P6R-Cryptoki-LABEL].

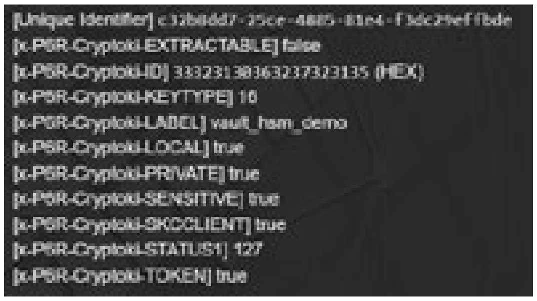


```
[c32b8dd7-25ce-4885-81c4-f3dc29effbde] Symmetric Key
        Owner  hashicorp-vault
        State  Preactive
 Last Changed  2019-07-05 11:45:08 +1000
        Names  Object has no names
   Attributes  [Always Sensitive] false
               [Cryptographic Algorithm] AES
               [Cryptographic Length] 256
               [Cryptographic Parameters]
                  13 (KMIP_TAG_BLOCK_CIPHER_MODE)
               [Cryptographic Usage Mask] 60
               [Digest]
                  Raw
                  [illegible] (HEX)
                  SHA256
               [Extractable] true
               [Fresh] true
               [Initial Date] 2019-07-05 11:45:00 +1000
               [Last Change Date] 2019-07-05 11:45:08 +1000
               [Lease Time] 01:00:00
               [Never Extractable] false
               [Object Type] Symmetric Key
               [Original Creation Date] 2019-07-05 11:45:00 +1000
               [Random Number Generator]
                  -2147483392 (KMIP_TAG_RNG_ALGO)
               [Sensitive] false
               [Unique Identifier] c32b8dd7-25ce-4885-81c4-f3dc29effbde
               [x-P6R-Cryptoki-EXTRACTABLE] false
               [x-P6R-Cryptoki-ID] [illegible] (HEX)
               [x-P6R-Cryptoki-KEYTYPE] 16
               [x-P6R-Cryptoki-LABEL] vault_hsm_demo
               [x-P6R-Cryptoki-LOCAL] true
               [x-P6R-Cryptoki-PRIVATE] true
               [x-P6R-Cryptoki-SENSITIVE] true
               [x-P6R-Cryptoki-SKCCLIENT] true
               [x-P6R-Cryptoki-STATUS1] 127
               [x-P6R-Cryptoki-TOKEN] true
```

You can inspect the attributes of both the Encryption and Integrity Key in their corresponding details page. An example table showing the attributes of the keys is shown below.

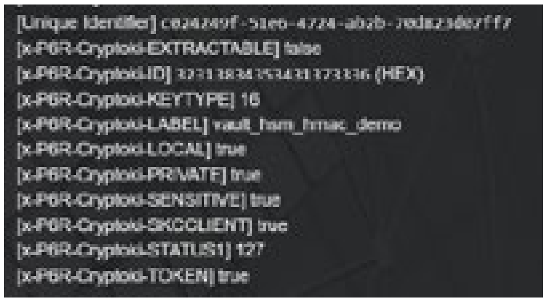
Attributes

Encryption Key



```
[Unique Identifier] c32b8dd7-25ce-4885-81c4-f3dc29effbde
[x-P6R-Cryptoki-EXTRACTABLE] false
[x-P6R-Cryptoki-ID] 3332313036323737323135 (HEX)
[x-P6R-Cryptoki-KEYTYPE] 16
[x-P6R-Cryptoki-LABEL] vault_hsm_demo
[x-P6R-Cryptoki-LOCAL] true
[x-P6R-Cryptoki-PRIVATE] true
[x-P6R-Cryptoki-SENSITIVE] true
[x-P6R-Cryptoki-SKCCLIENT] true
[x-P6R-Cryptoki-STATUS1] 127
[x-P6R-Cryptoki-TOKEN] true
```

Integrity Key



```
[Unique Identifier] c624249f-51e6-4724-a62b-7ed823de7ff7
[x-P6R-Cryptoki-EXTRACTABLE] false
[x-P6R-Cryptoki-ID] 3231383835343131373336 (HEX)
[x-P6R-Cryptoki-KEYTYPE] 16
[x-P6R-Cryptoki-LABEL] vault_hsm_hmac_demo
[x-P6R-Cryptoki-LOCAL] true
[x-P6R-Cryptoki-PRIVATE] true
[x-P6R-Cryptoki-SENSITIVE] true
[x-P6R-Cryptoki-SKCCLIENT] true
[x-P6R-Cryptoki-STATUS1] 127
[x-P6R-Cryptoki-TOKEN] true
```

You can inspect Vault operations history on each object in their corresponding details page (Object Operation History section).

The operations history for the Encryption key will include the following operations:

- Create
- Get Attributes
- Activate
- Encrypt
- Decrypt

The operations history for the MAC key will include the following operations:

- Create
- Get Attributes
- Activate
- MAC
- MAC Verify

Step 3: Seal and Unseal Vault

You can seal the Vault by running:

```
$ vault operator seal
```

Unsealing a Vault in the sealed state will result in Vault sending a request to the TSF to decrypt Vault's seal key.

```
$ vault operator unseal
```

Troubleshooting

Error initializing core: Failed to lock memory: cannot allocate memory

```
Error initializing core: Failed to lock memory: cannot allocate memory

This usually means that the mlock syscall is not available.
Vault uses mlock to prevent memory from being swapped to
disk. This requires root privileges as well as a machine
that supports mlock. Please enable mlock on your system or
disable Vault from using it. To disable Vault from using it,
set the `disable_mlock` configuration option in your configuration
file.
```

Set the `disable_mlock=true` in the Hashicorp configuration file to disable Vault's use of memory lock.

About QuintessenceLabs

QuintessenceLabs' portfolio of modular products addresses the most difficult security challenges, helping implement robust security strategies to protect data today and in the future. For more information on QuintessenceLabs' data protection solutions, please visit **www.quintessencelabs.com**.

CANBERRA | SYDNEY | SAN JOSE | WASHINGTON D.C. | LONDON

Quintessence Labs

AUSTRALIA
Unit 11, 18 Brindabella Circuit
Brindabella Business Park
Canberra Airport ACT 2609
+61 2 6260 4922

UNITED STATES
175 Bernal Road
Suite 220
San Jose CA 95119
+1 650 870 9920

www.quintessencelabs.com

Document ID: 5028