Marvell LiquidSecurity HSM Setup

CBOM Secure · HSM Sensor (PKCS#11)

Overview

This guide enables the CBOM Secure HSM Sensor to discover cryptographic material in a Marvell LiquidSecurity HSM over the PKCS#11 interface, read-only:

- Symmetric keys (AES, 3DES) and asymmetric key pairs (RSA, EC, Ed25519)
- X.509 certificates stored within HSM partitions
- Key metadata: key type, size, label, ID, usage flags, and creation date
- Slot and partition inventory: slot IDs, token labels, and firmware versions

Prerequisites

- An active Marvell LiquidSecurity subscription with LiquidSecurity Management Console (LSMC) access at HSM Administrator level.
- The LiquidSecurity client software (including libliquidpkcs11.so) installed on the sensor host.
- Network connectivity from the sensor host to the LiquidSecurity service endpoint (default port 2225).
- An initialized HSM instance with at least one partition, or Crypto Officer credentials to create one.
- pkcs11-tool (opensc) or the LiquidSecurity CLI (lscli) for validation.
- Python 3.8+ and the CBOM sensor agent installed on the host.

Step-by-Step Guide

Step 1: Provision and Initialize a LiquidSecurity Instance

In the LSMC (https://console.liquidsecurity.io), under HSM Instances then Create Instance, choose the region and instance type and place it in the VPC/VNet where the sensor host resides (allow outbound TCP 2225). Once Active, click Initialize HSM and set an HSM Administrator PIN (store it in your vault). Record the Service Endpoint hostname (e.g. lsm-abc123.us-east-1.liquidsecurity.io).

Step 2: Create an HSM Partition

Under the instance Partitions tab then Create Partition, set a Partition Label (e.g. CBOM_Partition) and a Crypto Officer PIN; note the auto-assigned Slot ID. CLI alternative:

```
lscli partition create \
  --hsm-endpoint lsm-abc123.us-east-1.liquidsecurity.io \
  --admin-pin <HSM_ADMIN_PIN> \
  --label CBOM_Partition \
  --crypto-officer-pin <CO_PIN>
```

```
pkcs11-tool --module /usr/local/lib/libliquidpkcs11.so --list-slots
```

Step 3: Create a Read-Only CryptoUser

Under Partitions then CBOM_Partition then Users then Add User, create cbom_reader with Role CryptoUser and a strong User PIN. CLI alternative:

```
lscli user create \
  --hsm-endpoint lsm-abc123.us-east-1.liquidsecurity.io \
  --partition CBOM_Partition \
  --co-pin <CO_PIN> \
  --username cbom_reader \
  --role CryptoUser \
  --user-pin <CBOM_USER_PIN>
```

Note The CryptoUser role is read-only by design: it permits C_Login, C_FindObjects, and C_GetAttributeValue, but denies C_GenerateKey, C_CreateObject, C_DestroyObject, and C_Wrap/UnwrapKey. No extra permission changes are needed.

Step 4: Install and Configure the PKCS#11 Client

```
ls -la /usr/local/lib/libliquidpkcs11.so
# if missing, install the client package:
sudo dpkg -i liquidsecurity-client_<version>_amd64.deb    # or rpm -ivh on RHEL

# /etc/liquidsecurity/client.conf
[Network]
ServerAddress = lsm-abc123.us-east-1.liquidsecurity.io
ServerPort    = 2225
TLSVerify     = true
[Client]
CertFile = /etc/liquidsecurity/client-cert.pem
KeyFile  = /etc/liquidsecurity/client-key.pem
CAFile   = /etc/liquidsecurity/ls-ca.pem
```

Grant the sensor user read access to the client key (chown root:cbom, chmod 640) and export LIQUIDSECURITY_PKCS11_LIB for runtime reference.

Step 5: Configure the CBOM Secure Sensor

```
sensor:
  name: Discover_HSM
  type: pkcs11_hsm
  enabled: true
  schedule: "0 2 * * *"
connection:
  pkcs11_library: /usr/local/lib/libliquidpkcs11.so
  slot_id: 0                    # from Step 2
  token_label: CBOM_Partition
credentials:
  user_type: CKU_USER
  username: cbom_reader
  pin:
    secret_ref: cbom/marvell-ls/cbom_reader_pin
discovery:
  object_types: [CKO_PRIVATE_KEY, CKO_PUBLIC_KEY, CKO_SECRET_KEY,
CKO_CERTIFICATE, CKO_DATA]
```

```
  attributes: [CKA_LABEL, CKA_ID, CKA_KEY_TYPE, CKA_VALUE_LEN,
CKA_MODULUS_BITS,
    CKA_EC_PARAMS, CKA_START_DATE, CKA_END_DATE, CKA_SIGN, CKA_VERIFY,
    CKA_ENCRYPT, CKA_DECRYPT, CKA_WRAP, CKA_UNWRAP, CKA_EXTRACTABLE,
CKA_SENSITIVE]
output:
  cbom_server: https://cbom.internal.example.com
  api_key: { secret_ref: cbom/api/ingest_key }
  tags: { hsm_vendor: marvell, hsm_model: liquidsecurity, region: us-east-1 }
```

Store the CryptoUser PIN and CBOM API key in a secrets manager and reference them via secret_ref, never plaintext in the YAML.

Step 6: Validate

```
pkcs11-tool \
  --module /usr/local/lib/libliquidpkcs11.so \
  --slot 0 --login --login-type user --pin <CBOM_USER_PIN> --list-objects

cbom-sensor run --config /etc/cbom/sensors/marvell-liquidsecurity.yaml --dry-run --log-level
debug
```

Confirm the library connects, the slot opens (token: CBOM_Partition), and objects are enumerated. Run live and verify objects appear under Assets then HSM in the marvell / liquidsecurity tag group.

Common Errors

CKR_PIN_INCORRECT on login

Cause: The CryptoUser PIN does not match the partition PIN, or secret_ref resolves to a stale value.

Resolution: Reset the PIN via the LSMC (or lscli user set-pin) and update the secret in your secrets manager.

CKR_SLOT_ID_INVALID / library init failure

Cause: slot_id does not match the partition, or the client is not connected to the HSM endpoint (network/TLS/client.conf).

Resolution: Re-run --list-slots and update slot_id; verify client.conf and that the client certificate has not expired (openssl x509 -noout -dates).

CKR_USER_NOT_LOGGED_IN during enumeration

Cause: Missing/incorrect user_type, or the session dropped on an idle timeout.

Resolution: Confirm user_type: CKU_USER (not CKU_SO); increase the LiquidSecurity session timeout for long-running scans.

Security Recommendations

- Use the CryptoUser role only, never a Crypto Officer or HSM Administrator account.
- Rotate the CryptoUser PIN on schedule (e.g. every 90 days) and update the secrets manager entry.

- Reference PINs and API keys via secret_ref, never plaintext YAML or shell profiles.
- Keep TLSVerify = true in client.conf; monitor the mTLS client certificate expiry (alert 30 days out).
- Enable HSM audit logging, ship to your SIEM, and restrict port 2225 to the sensor host IP.

Conclusion

The HSM Sensor authenticates to the Marvell LiquidSecurity partition as a least-privilege CryptoUser and enumerates keys and certificates over PKCS#11 (libliquidpkcs11.so), submitting the inventory to CBOM Secure on schedule. For additional partitions or instances, duplicate the sensor block with a unique name, slot_id, and token_label.