Crypto4A QxHSM Setup

CBOM Secure · HSM Sensor (Discover_HSM)

Overview

This guide enables the CBOM Secure HSM Sensor to discover cryptographic assets on a Crypto4A QxHSM appliance over PKCS#11:

- Symmetric keys (AES, 3DES) in token partitions
- Asymmetric key pairs (RSA, EC, Ed25519, ML-KEM, ML-DSA) and their attributes
- X.509 certificates associated with key objects
- Secret and data objects tagged within partitions
- Token metadata (slot ID, token label, firmware version, mechanism list)

The sensor authenticates as a read-only CryptoUser and never extracts key material.

Prerequisites

- A QxHSM appliance powered on, network-accessible, running firmware 2.x or later.
- QxAdmin installed on the management workstation.
- The Security Officer (SO) PIN for the target partition (or access to someone who has it).
- The QxHSM PKCS#11 client library on the sensor host: /usr/local/lib/libqx_pkcs11.so.
- Connectivity from the sensor host to the QxHSM PKCS#11 daemon port (default 2348/tcp).
- pkcs11-tool (from opensc) on the sensor host for validation.

Step-by-Step Guide

Step 1: Initialize the Token Partition (if needed)

In QxAdmin, add the device, expand the appliance to view slots, right-click an uninitialized slot → Initialize Token. Set a Token Label (e.g. CBOM_PROD_01), an SO PIN (stored in your vault), and a User PIN (the CryptoUser PIN used by the sensor). Record the Slot ID. CLI alternative:

```
pkcs11-tool \
  --module /usr/local/lib/libqx_pkcs11.so \
  --init-token \
  --slot 0 \
  --label "CBOM_PROD_01" \
  --so-pin "<SO_PIN>"
```

Step 2: Set the Read-Only CryptoUser PIN

In QxAdmin, right-click the token → Set User PIN, authenticate with the SO PIN, and set a strong User PIN. CLI alternative:

```
pkcs11-tool \
```

```
  --module /usr/local/lib/libqx_pkcs11.so \
  --slot 0 --init-pin \
  --so-pin "<SO_PIN>" --new-pin "<USER_PIN>"
```

Verify the CryptoUser can log in and enumerate objects:

```
pkcs11-tool \
  --module /usr/local/lib/libqx_pkcs11.so \
  --slot 0 --login --pin "<USER_PIN>" --list-objects
```

Step 3: Enforce Read-Only Behavior (Recommended)

If policy prohibits the sensor credential from creating objects, apply a Usage Policy in QxAdmin: select the token → Policies → Key Management Policy, and under CryptoUser permissions uncheck C_GenerateKey, C_GenerateKeyPair, C_CreateObject, and C_DestroyObject, then Save Policy.

Step 4: Confirm the Library and Collect Values

```
ls -l /usr/local/lib/libqx_pkcs11.so
# optional, if your tooling reads it:
export PKCS11_MODULE_PATH=/usr/local/lib/libqx_pkcs11.so
```

Collect: library path /usr/local/lib/libqx_pkcs11.so, Slot ID (e.g. 0), token label (CBOM_PROD_01), and the User PIN (in your vault).

Step 5: Configure the CBOM Secure Sensor

```
sensors:
  - name: Discover_HSM
    type: pkcs11_hsm
    enabled: true
    description: "Crypto4A QxHSM - Production Token Partition"
    schedule: "0 2 * * *"          # daily at 02:00 UTC
    connection:
      library_path: /usr/local/lib/libqx_pkcs11.so
      slot_id: 0
      token_label: CBOM_PROD_01
    auth:
      user_type: user              # CryptoUser role (non-SO)
      pin_secret: cbom/qxhsm/prod/user-pin    # reference to secrets vault
    discovery:
      include_object_types: [CKO_PRIVATE_KEY, CKO_PUBLIC_KEY, CKO_SECRET_KEY,
        CKO_CERTIFICATE, CKO_DATA]
      include_attributes: [CKA_LABEL, CKA_ID, CKA_KEY_TYPE,
CKA_KEY_GEN_MECHANISM,
        CKA_VALUE_LEN, CKA_MODULUS_BITS, CKA_EC_PARAMS, CKA_START_DATE,
CKA_END_DATE,
        CKA_EXTRACTABLE, CKA_SENSITIVE, CKA_ENCRYPT, CKA_DECRYPT, CKA_SIGN,
CKA_VERIFY,
        CKA_WRAP, CKA_UNWRAP]
      enumerate_mechanisms: true
    tags: { environment: production, vendor: Crypto4A, appliance_model: QxHSM }
```

Note pin_secret is a vault path (HashiCorp Vault, AWS Secrets Manager, etc.) resolved at runtime — the PIN is never stored in plaintext.

Step 6: Validate

```
cbom-sensor run --config cbom-sensors.yaml --sensor Discover_HSM --log-level debug
```

```
pkcs11-tool --module /usr/local/lib/libqx_pkcs11.so --slot 0 --list-mechanisms
```

Confirm CryptoUser login successful and an enumerated-object count, then check Assets → HSM filtered by vendor: Crypto4A. Cross-reference the mechanism list against CBOM's token metadata view.

Common Errors

CKR_PIN_INCORRECT on startup

Cause: The vault-stored User PIN does not match the token PIN, or pin_secret is wrong.

Resolution: Verify the vault path; test login with pkcs11-tool --login --pin; reset the PIN via QxAdmin (SO PIN required) and update the vault entry.

CKR_SLOT_ID_INVALID / slot not found

Cause: slot_id no longer matches the actual slot (numbering can shift after restart or partition changes).

Resolution: Re-enumerate with pkcs11-tool --list-slots and update slot_id, or use token_label-based discovery if supported.

Cannot open library / dlopen failure

Cause: The Crypto4A PKCS#11 client package is not installed, or the library is at a non-default path.

Resolution: Install the crypto4a-pkcs11-client package; confirm /usr/local/lib/libqx_pkcs11.so or update library_path.

Security Recommendations

- Never use the SO PIN for the sensor — use only the read-only CryptoUser PIN.
- Store the User PIN in a secrets manager and reference it by path in the YAML.
- Apply a QxHSM Usage Policy removing C_GenerateKey/KeyPair, C_CreateObject, and C_DestroyObject from CryptoUser.
- Restrict the PKCS#11 daemon port (default 2348/tcp) to the sensor host IP via firewall or QxHSM allowlist.
- Rotate the CryptoUser PIN quarterly and audit PKCS#11 session logs in QxAdmin.

Conclusion

With a read-only CryptoUser credential, confirmed PKCS#11 connectivity, and an optional hardware Usage Policy, the Discover_HSM sensor inventories keys, certificates, and cryptographic objects on the QxHSM on schedule — providing continuous visibility into key types, algorithm usage, attribute flags, and certificate lifecycles without privileged access.