Cryptomathic CKMS Setup

CBOM Secure · KMIP Sensor (Discover_KMIP)

Overview

This guide configures Cryptomathic CKMS so the CBOM Secure Discover_KMIP sensor can discover cryptographic assets via KMIP 1.2+ over TLS on port 5696, in read-only mode:

- Symmetric and asymmetric managed keys (AES, RSA, EC)
- Key attributes including algorithm, length, state, and lifecycle dates
- Certificates stored or linked within the CKMS object store
- Key metadata including usage masks, sensitive flags, and cryptographic parameters

No key material, plaintext secrets, or sensitive values are extracted or transmitted.

Prerequisites

- Cryptomathic CKMS 5.x+ with KMIP 1.2+ support enabled in the server license.
- A CKMS system administrator account able to create roles, users, and manage the KMIP endpoint.
- The KMIP service module licensed and installed on the CKMS instance.
- Network connectivity from the sensor host to TCP port 5696.
- A CKMS server TLS certificate signed by a CA the sensor trusts, or the CKMS CA certificate exported.
- A running CBOM deployment and OpenSSL on the sensor host for verification.

Step-by-Step Guide

Step 1: Create a Read-Only Operator Role

In the CKMS Administration Console (https://<ckms-host>:8443/ckms-admin), under Administration → Access Control → Roles → Create Role, create cbom-kmip-readonly with only these permissions enabled:

```
key:list — enumerate managed cryptographic objects
key:get — retrieve key attributes and metadata (not key material)
```

Leave all write, delete, rotate, activate, and revoke permissions disabled, and Save Role.

Step 2: Create the Operator User Account

Under Access Control → Users → Create User, set Username cbom-sensor, assign the cbom-kmip-readonly role, and select Certificate (mutual TLS) as the authentication method. Leave password authentication disabled where policy permits certificate-only auth.

Step 3: Enable the KMIP Service Endpoint

Under Services → KMIP Service, confirm the service is Running with Protocol KMIP 1.2+, Port 5696, TLS 1.2 minimum, and Client Authentication set to Mutual TLS (required). Optionally restrict cipher suites to ECDHE with AES-GCM or ChaCha20-Poly1305.

Step 4: Generate a Client Certificate

Under Certificate Management → Issue Certificate, select Client Certificate with CN cbom-sensor, linked user cbom-sensor, key RSA 2048 or EC P-256, and 365-day validity. Download the client certificate, private key, and CKMS CA chain, then restrict permissions:

```
chmod 600 /etc/cbom/certs/cbom-sensor.key
chmod 644 /etc/cbom/certs/cbom-sensor.crt
chmod 644 /etc/cbom/certs/ckms-ca.crt
```

Verify the mTLS handshake from the sensor host:

```
openssl s_client \
  -connect <ckms-host>:5696 \
  -cert /etc/cbom/certs/cbom-sensor.crt \
  -key /etc/cbom/certs/cbom-sensor.key \
  -CAfile /etc/cbom/certs/ckms-ca.crt \
  -verify_return_error
# Expected: Verify return code: 0 (ok)
```

Step 5: Configure the CBOM Secure Sensor

```
sensors:
  - name: Discover_KMIP
    type: kmip
    enabled: true
    schedule: "0 2 * * *"       # daily at 02:00 UTC
    connection:
      host: ckms.example.internal
      port: 5696
      protocol_version: "1.2"
      tls:
        enabled: true
        min_version: "TLSv1.2"
        client_cert: /etc/cbom/certs/cbom-sensor.crt
        client_key: /etc/cbom/certs/cbom-sensor.key
        ca_cert: /etc/cbom/certs/ckms-ca.crt
        verify_server_cert: true
    authentication:
      method: certificate
    discovery:
      object_types: [SymmetricKey, PublicKey, PrivateKey, Certificate]
      include_attributes: [Cryptographic_Algorithm, Cryptographic_Length, State,
        Initial_Date, Activation_Date, Deactivation_Date, Compromise_Date,
        Usage_Masks, Name, Sensitive]
    output:
      tags: { source: cryptomathic-ckms, environment: production }
systemctl restart cbom-sensor
```

Step 6: Validate

```
journalctl -u cbom-sensor -f --since "5 minutes ago"
```

```
cbom-sensor run --sensor Discover_KMIP --dry-run
```

Confirm a TLS handshake and a line such as Discovered 142 cryptographic objects. In the CKMS audit log (KMIP Operations), only Locate and Get operations should appear for cbom-sensor.

Common Errors

TLS handshake failure — certificate unknown

Cause: The client certificate is not linked to a valid CKMS user, or the issuing CA is not trusted by the KMIP service.

Resolution: Confirm the certificate is associated with cbom-sensor under Client Certificates, add the issuing CA under KMIP Service → Trusted CAs, restart the service, and retry the s_client test.

KMIP operation not permitted — insufficient privileges

Cause: The role is missing key:list or key:get, or was not saved.

Resolution: Confirm both permissions on cbom-kmip-readonly and the role assignment on cbom-sensor, then retry.

Connection refused / timeout on 5696

Cause: The KMIP service is not running, the port is firewalled, or the host value is wrong.

Resolution: Confirm the service is Running, test with nc -zv <host> 5696, open TCP 5696, and verify DNS resolution.

Security Recommendations

- Keep the cbom-kmip-readonly role limited to key:list and key:get; audit it periodically.
- Store cbom-sensor.key with mode 600 or inject it from a secrets manager.
- Rotate the client certificate before expiry (set a 30-day reminder).
- Enforce TLS 1.2 minimum (1.3 preferred); reject TLS 1.0/1.1.
- Forward CKMS audit logs to your SIEM and alert on any KMIP write operations by cbom-sensor.

Conclusion

With a dedicated operator role limited to key:list and key:get and a scoped client certificate, the Discover_KMIP sensor continuously inventories all managed keys and certificates in Cryptomathic CKMS over mutual TLS — without access to sensitive key material.