Oracle Key Vault Setup

CBOM Secure · KMIP Sensor (Discover_KMIP)

Overview

This guide connects the CBOM Secure Discover_KMIP sensor to an Oracle Key Vault (OKV) instance using KMIP 1.2+ over mutually authenticated TLS on port 5696, for read-only discovery of:

- Symmetric keys (AES, 3DES) and asymmetric key pairs (RSA, EC) in key wallets
- X.509 certificates associated with managed objects
- Secret data objects and opaque data blobs
- Key metadata: algorithm, length, state, activation/expiration dates, and usage masks

The sensor uses only KMIP Locate, Get Attributes, and Get Attribute List operations - it never writes, rotates, deletes, or retrieves raw key material.

Prerequisites

- Oracle Key Vault 21.3 or later, deployed and network-accessible.
- OKV web UI access with Key Administrator or System Administrator privileges.
- The okv command-line utility (from the OKV 21.x client package on Oracle Support) on the enrollment host.
- Outbound TCP from the sensor host to the OKV server on port 5696.
- OpenSSL 1.1.1 or later on the sensor host.
- The OKV server CA certificate (downloadable from System > Oracle Key Vault CA Certificate).
- A dedicated key wallet whose contents the sensor is authorized to discover.

Step-by-Step Guide

Step 1: Create a Dedicated Read-Only Endpoint

In the OKV web UI, go to Endpoints > Create and complete the form:

Field	Value
Endpoint Name	CBOM-Sensor-ReadOnly
Endpoint Type	Other
Description	Read-only KMIP endpoint for CBOM Discover_KMIP sensor
Platform	Linux x86-64 (match your sensor host)

Save the endpoint (it enters Pending Enrollment) and record the single-use Enrollment Token (default 24-hour validity).

Step 2: Grant Read-Only Wallet Access

- Navigate to Keys & Wallets > Wallets and open the target wallet (e.g. Production-TDE-Wallet).
- On the Access tab, click Grant Access.
- Select the CBOM-Sensor-ReadOnly endpoint as grantee.
- Set Access Level to Read Only, then Save.
- Repeat for each additional wallet in scope.

Note Read Only permits listing and attribute-metadata retrieval - not raw key value bytes. The sensor uses Get Attributes only.

Step 3: Enroll the Endpoint (mutual TLS bundle)

```
mkdir -p /opt/okv-client && tar -xzf okvclient.tar.gz -C /opt/okv-client && cd /opt/okv-client
java -jar okvclient.jar -d /opt/okv-client/install -o <okv-host>:5696
export OKV_HOME=/opt/okv-client/install
$OKV_HOME/bin/okv admin client-config enroll --server <okv-host>:5696 --token
<enrollment-token>
# verify server fingerprint, type yes; on success:
ls -lh $OKV_HOME/ssl/ewallet.p12
```

Copy the client bundle and CA cert to the sensor config directory and lock permissions:

```
mkdir -p /etc/cbom/sensors/kmip/oracle-kv
cp $OKV_HOME/ssl/ewallet.p12 /etc/cbom/sensors/kmip/oracle-kv/
cp ~/okv-ca.pem /etc/cbom/sensors/kmip/oracle-kv/
chmod 600 /etc/cbom/sensors/kmip/oracle-kv/ewallet.p12
chmod 644 /etc/cbom/sensors/kmip/oracle-kv/okv-ca.pem
```

Step 4: Configure the CBOM Secure Sensor

```
sensors:
  - name: Discover_KMIP
    type: kmip
    enabled: true
    schedule: "0 2 * * *"
    connection:
      host: okv.example.internal
      port: 5696
      protocol: kmip
      kmip_version: "1.2"
      tls:
        enabled: true
        mutual_tls: true
        client_cert_bundle: /etc/cbom/sensors/kmip/oracle-kv/ewallet.p12
        client_cert_password: "${OKV_WALLET_PASSWORD}"
        ca_cert: /etc/cbom/sensors/kmip/oracle-kv/okv-ca.pem
        verify_hostname: true
        min_tls_version: "TLSv1.2"
    discovery:
      operations: [locate, get_attributes, get_attribute_list]
      object_types: [symmetric_key, asymmetric_key, certificate, secret_data]
      max_items_per_request: 200
      include_deactivated: false
```

Export OKV_WALLET_PASSWORD from your secrets manager - do not hard-code the PKCS#12 password.

Step 5: Validate

```
sudo systemctl restart cbom-agent
sudo cbom sensor run --name Discover_KMIP --debug
# confirm: TLS handshake complete / Mutual TLS accepted / KMIP 1.2 negotiated
# confirm: KMIP Locate returned N object UIDs
```

Verify output JSON exists and, in OKV, that Reports > Audit > Endpoint Activity shows read operations from CBOM-Sensor-ReadOnly.

Common Errors

TLS Handshake Failure - Certificate Verify Error

Cause: The ca_cert points to the wrong file, misses the OKV intermediate, or the server cert was issued by an untrusted CA.

Resolution: Re-download the full-chain CA PEM from the OKV UI, verify it with openssl verify against the server cert, and update ca_cert.

KMIP Locate Returns Empty

Cause: The endpoint has no wallet access, or the access cache has not refreshed since the grant.

Resolution: Confirm at least one wallet shows Read Only under the endpoint's Access tab, wait up to 2 minutes for cache refresh, and confirm the wallet holds active objects.

ewallet.p12 Password Incorrect (MAC verify failure)

Cause: OKV_WALLET_PASSWORD is wrong, or the bundle was regenerated with a different password.

Resolution: Test with openssl pkcs12 -in ewallet.p12 -noout -passin pass:...; if lost, re-enroll and set a known password.

Security Recommendations

- Rotate the enrollment certificate annually or whenever the OKV CA rotates; revoke the old endpoint immediately.
- Store the ewallet.p12 password in a secrets manager - never in plaintext config or shell history.
- Restrict the bundle to the CBOM service account (chmod 600, chown cbom-svc).
- Grant the endpoint access only to in-scope wallets - not the system wallet or TDE master-key wallets unless required.
- Alert via OKV if the endpoint performs any operation other than Locate or Get Attributes.

Conclusion

With a dedicated read-only endpoint, scoped wallet grants, a mutual-TLS enrollment bundle, and KMIP 1.2 over port 5696, CBOM Secure continuously inventories the symmetric keys, key pairs, certificates, and secret objects managed in Oracle Key Vault - without write access or exposure of raw key material. For multi-cluster OKV, repeat the endpoint and sensor steps per cluster.