AWS CloudHSM Setup

CBOM Secure · HSM Sensor (Discover_HSM)

Overview

This guide configures AWS CloudHSM so the CBOM Secure HSM Sensor can discover cryptographic material in your CloudHSM cluster:

- Symmetric keys (AES, 3DES) managed within HSM partitions
- Asymmetric key pairs (RSA, EC) stored on the HSM
- X.509 certificates and associated metadata
- Key attributes such as key type, length, label, ID, and usage flags

It covers creating and initializing a cluster, provisioning a read-only Crypto User (CU), configuring the CloudHSM client daemon, and wiring the sensor configuration.

Prerequisites

- An AWS account with permissions to manage CloudHSM (e.g. AWSCloudHSMFullAccess).
- A VPC with at least one private subnet in the target region.
- The AWS CLI installed and configured.
- The CloudHSM Client SDK 5 (cloudhsm-cli) installed on the sensor host.
- Crypto Officer (CO) credentials for an existing cluster, or you will initialize one here.
- The sensor host can reach the cluster on port 2223 (HSM ENI) and 2224 (client daemon).

Step-by-Step Guide

Step 1: Create and Initialize a CloudHSM Cluster

```
aws cloudhsmv2 create-cluster \
  --hsm-type hsm1.medium \
  --subnet-ids subnet-0abc12345def67890

aws cloudhsmv2 create-hsm \
  --cluster-id cluster-0123456789abcdef0 \
  --availability-zone us-east-1a
```

Download the cluster CSR from the console, sign it with your CA (or a self-signed issuing cert for lab use), then upload and initialize:

```
openssl req -new -x509 -days 3652 -key customerCA.key \
  -out customerCA.crt -subj "/CN=CustomerCA"

openssl x509 -req -days 3652 \
  -in cluster-0123456789abcdef0_ClusterCsr.csr \
  -CA customerCA.crt -CAkey customerCA.key -CAcreateserial \
  -out cluster-0123456789abcdef0_CustomerHsmCertificate.crt
```

Step 2: Configure the CloudHSM Client Daemon

```
# Amazon Linux 2 / RHEL
sudo yum install -y cloudhsm-client
# Ubuntu / Debian
sudo apt-get install -y cloudhsm-client

sudo /opt/cloudhsm/bin/configure-client -a <HSM_ENI_IP_ADDRESS>
sudo cp customerCA.crt /opt/cloudhsm/etc/customerCA.crt

sudo systemctl enable cloudhsm-client
sudo systemctl start cloudhsm-client
ls -lh /opt/cloudhsm/lib/libcloudhsm_pkcs11.so
```

Step 3: Create a Dedicated Crypto User (CU)

Authenticate as Crypto Officer, then create a read-only Crypto User for the sensor. Do not use CO credentials for the sensor.

```
/opt/cloudhsm/bin/cloudhsm-cli interactive
aws-cloudhsm> login --username admin --role crypto-officer
aws-cloudhsm> user create --username cbom-sensor --role crypto-user
aws-cloudhsm> user list
aws-cloudhsm> logout
```

Note A Crypto User can enumerate and use keys shared with it but cannot create, delete, or wrap other users' keys — satisfying the sensor's read-only requirement.

Step 4: Enumerate Slots

```
pkcs11-tool --module /opt/cloudhsm/lib/libcloudhsm_pkcs11.so --list-slots
```

Step 5: Configure the CBOM Secure Sensor

```
sensors:
  - name: Discover_HSM
    type: hsm_pkcs11
    enabled: true
    schedule: "0 2 * * *"          # daily at 02:00 UTC
    connection:
      hsm_provider: aws_cloudhsm
      pkcs11_library: /opt/cloudhsm/lib/libcloudhsm_pkcs11.so
      cluster_id: cluster-0123456789abcdef0
      hsm_ip: 10.0.1.42            # HSM ENI IP
      slot: 0
    credentials:
      username: cbom-sensor
      password: "${CBOM_CLOUDHSM_CU_PASSWORD}"   # from secrets manager
      role: crypto-user
    discovery:
      include_key_types: [AES, RSA, EC, DES3]
      include_certificates: true
      include_secret_keys: true
      include_private_keys: true   # metadata only; material never extracted
    output:
      tags: { environment: production, region: us-east-1 }
      notify_on_expiry_days: 30
```

Step 6: Validate

```
cbom sensor run --name Discover_HSM --dry-run
cbom sensor run --name Discover_HSM
```

Confirm output such as Discovered 14 key objects (AES: 8, RSA: 4, EC: 2) and 3 certificate objects, then verify assets under Assets → HSM filtered by region.

Common Errors

CKR_PIN_INCORRECT during PKCS#11 login

Cause: The CU password in the config/variable does not match CloudHSM.

Resolution: Reset it from a CO session (user change-password --username cbom-sensor) and update CBOM_CLOUDHSM_CU_PASSWORD.

Failed to connect to CloudHSM client daemon

Cause: The cloudhsm-client daemon is not running, or the HSM ENI IP is wrong.

Resolution: Restart the daemon, verify the configured server_ip, and reconfigure with configure-client -a <CORRECT_HSM_ENI_IP> if needed.

CKR_SLOT_ID_INVALID

Cause: The slot index in the YAML does not match the PKCS#11 library (e.g. after failover).

Resolution: Re-enumerate slots with pkcs11-tool --list-slots and update the slot value.

Security Recommendations

- Use a dedicated CU account; never share credentials with other applications.
- Rotate the CU password regularly via a CO session and update the vault simultaneously.
- Restrict the HSM ENI security group to inbound 2223 from the sensor host only.
- Keep log level at INFO/WARN — never DEBUG, which may log attribute values.
- Enable CloudTrail and CloudHSM audit logs for CU login and key-list operations.

Conclusion

With the cluster initialized, the client daemon running, and a dedicated Crypto User provisioned, the HSM Sensor enumerates all cryptographic assets in the CloudHSM cluster on a schedule through PKCS#11 with read-only CU privileges — no key material is ever extracted. For multi-cluster environments, add one Discover_HSM block per cluster with a distinct name.