Microsoft SQL Server Setup

CBOM Secure · Database TDE Sensor (Discover_DB)

Overview

This guide configures the CBOM Secure Discover_DB sensor to discover cryptographic assets managed by Microsoft SQL Server's Transparent Data Encryption (TDE). The sensor connects with a read-only login and queries system catalog views, read-only:

- Database encryption keys (DEKs) via sys.dm_database_encryption_keys
- Server-level certificates that protect DEKs via sys.certificates
- Asymmetric keys (RSA/ECC) used for DEK protection via sys.asymmetric_keys
- Symmetric keys at server or database scope via sys.symmetric_keys

For each asset it records name, algorithm, thumbprint, key length, and creation/expiry dates. It never decrypts data, exports key material, or performs writes.

Prerequisites

- SQL Server 2008 or later (any edition supporting TDE); 2019+ recommended for full catalog coverage.
- Network access from the sensor host to the instance on TCP 1433 (or the custom port).
- SQL Server Authentication (Mixed Mode), or a mapped Windows login if using Windows Authentication.
- sysadmin or securityadmin rights to create the login and grant permissions.
- CBOM administrator access to create a Discover_DB sensor.
- At least one TDE-enabled database for meaningful DEK results (certs/keys enumerate regardless).

Step-by-Step Guide

Step 1: Create a Read-Only Login

Create a dedicated login in the master database (do not reuse application or admin accounts):

```
USE master;
GO
CREATE LOGIN cbom_reader
   WITH PASSWORD = 'R3pl@ceW1thStr0ngP@ssword!',
       DEFAULT_DATABASE = master,
       CHECK_EXPIRATION = ON,
       CHECK_POLICY = ON;
GO
```

Step 2: Grant the Three Required Server Permissions

Only server-level read permissions are needed — no database roles or data-read access:

```
USE master;
GO
GRANT VIEW SERVER STATE    TO cbom_reader;    -- sys.dm_database_encryption_keys and
DMVs
GRANT VIEW ANY DEFINITION TO cbom_reader;    -- sys.certificates, sys.asymmetric_keys,
sys.symmetric_keys
GRANT VIEW ANY DATABASE    TO cbom_reader;    -- enumerate databases and encryption
state
GO
```

Verify with a query against sys.server_permissions joined to sys.server_principals — expect GRANT rows for all three permissions.

Step 3: Verify Catalog View Access

Connected as cbom_reader, confirm each required view returns results (or an empty set) with no permission errors:

```
SELECT name, subject, thumbprint, algorithm_desc, expiry_date FROM sys.certificates;
SELECT name, algorithm_desc, thumbprint FROM sys.asymmetric_keys;
SELECT DB_NAME(database_id) AS database_name, encryption_state, encryptor_type,
      key_algorithm, key_length FROM sys.dm_database_encryption_keys;
SELECT name, algorithm_desc, key_length, create_date FROM sys.symmetric_keys;
```

Step 4: Configure the CBOM Secure Sensor

```
sensors:
  - name: mssql-tde-prod
    type: Discover_DB
    enabled: true
    schedule: "0 2 * * *"
    connection:
      driver: mssql
      host: "sql-prod-01.corp.example.com"
      port: 1433
      database: master              # server-scope catalog queries
      username: "cbom_reader"
      password: "${CBOM_MSSQL_PASSWORD}"
      encrypt: true                 # enforce TLS
      trust_server_certificate: false
    discovery:
      catalogs: [sys.certificates, sys.asymmetric_keys, sys.dm_database_encryption_keys,
sys.symmetric_keys]
      include_system_databases: false
    output:
      asset_types: [certificate, asymmetric_key, symmetric_key, tde_dek]
      tags: { environment: production, data_classification: confidential }
    notifications:
      on_expiring_cert_days: 30
```

Store the password referenced by ${CBOM_MSSQL_PASSWORD} in a secrets manager; never hard-code it.

Step 5: Validate

```
cbom sensor run --name mssql-tde-prod
```

Confirm the run connects and reports discovered certificate/asymmetric-key/DEK/symmetric-key counts. In the CBOM inventory filtered by source = mssql-tde-prod, verify certificates and TDE database encryption keys appear with thumbprint, algorithm, expiry, and subject populated. If SQL auditing is enabled, confirm cbom_reader issued only SELECTs against catalog views.

Common Errors

Login failed for user 'cbom_reader'

Cause: The login is missing/disabled, the password is wrong, or the instance is in Windows-Authentication-only mode.

Resolution: Enable Mixed Mode (and restart the service), confirm the login is not disabled, and reset the password with ALTER LOGIN if needed.

The user does not have permission to perform this action

Cause: One of VIEW SERVER STATE / VIEW ANY DEFINITION / VIEW ANY DATABASE was not granted or was revoked.

Resolution: Re-run the permission verification query and re-grant any missing permission.

SSL Provider — certificate chain not trusted

Cause: trust_server_certificate is false but the SQL Server TLS certificate is self-signed or from an untrusted CA.

Resolution: Install the SQL Server CA into the sensor host trust store (preferred) or issue a certificate from a trusted CA. Only use trust_server_certificate: true as a temporary non-production workaround.

Security Recommendations

- Grant the login only the three listed permissions — never sysadmin, db_owner, or data-access roles.
- Rotate the password on schedule (CHECK_EXPIRATION = ON) and store it in a secrets manager.
- Enforce encrypt: true and trust_server_certificate: false; set Force Encryption = Yes on the server.
- Enable SQL Server Auditing for cbom_reader to record read-only SELECTs against catalog views.
- Restrict inbound TCP/1433 to the sensor host IP via firewall or network security groups.

Conclusion

With a least-privilege cbom_reader login and three server-level read permissions, the Discover_DB sensor gives CBOM Secure continuous, read-only visibility into SQL Server TDE certificates, asymmetric keys, symmetric keys, and database encryption keys — without exposing or decrypting protected data — and, with expiry notifications, timely awareness of certificate lifecycle events.