MariaDB Setup

CBOM Secure · Database TDE Sensor (Discover_DB)

Overview

This guide configures a read-only MariaDB user and connects the CBOM Secure Discover_DB sensor to a MariaDB instance using Data-at-Rest Encryption (TDE). The sensor discovers, read-only:

- Active encryption key identifiers and versions (file_key_management or aws_key_management plugins)
- Tablespace-level encryption status via information_schema.INNODB_TABLESPACES_ENCRYPTION
- Plugin state and key management backend configuration
- Per-table and per-tablespace encryption metadata

The sensor never modifies database state, keys, or encryption configuration.

Prerequisites

- MariaDB 10.1.3 or later, running with Data-at-Rest Encryption enabled.
- At least one key management plugin active: file_key_management or aws_key_management.
- A MariaDB administrative account (e.g. root) able to create users and grant privileges.
- The Discover_DB sensor module available, with connectivity to the MariaDB port (default 3306).
- For TLS: a valid server certificate and CA certificate.
- InnoDB compiled with encryption support so information_schema.INNODB_TABLESPACES_ENCRYPTION is accessible.

Step-by-Step Guide

Step 1: Create the CBOM Reader User

Connect as an administrator and create a dedicated account (scope the host to the sensor IP in production):

```
mariadb -u root -p -h <mariadb-host> -P 3306

CREATE USER 'cbom_reader'@'%' IDENTIFIED BY 'StrongPassword123!';
-- or restrict to the sensor host:
CREATE USER 'cbom_reader'@'192.168.10.50' IDENTIFIED BY 'StrongPassword123!';

SELECT User, Host FROM mysql.user WHERE User = 'cbom_reader';
```

Step 2: Grant Required Privileges

Grant read access to the InnoDB encryption metadata and the PROCESS privilege for plugin/thread state:

```
GRANT SELECT ON information_schema.INNODB_TABLESPACES_ENCRYPTION TO
'cbom_reader'@'%';
GRANT PROCESS ON *.* TO 'cbom_reader'@'%';
FLUSH PRIVILEGES;

SHOW GRANTS FOR 'cbom_reader'@'%';
```

Step 3: Verify the Encryption Plugin and Tablespaces

```
SHOW PLUGINS SONAME LIKE '%key_management%';   -- expect Status ACTIVE

SELECT SPACE, NAME, ENCRYPTION_SCHEME, KEY_ID
FROM information_schema.INNODB_TABLESPACES_ENCRYPTION
LIMIT 10;
```

An ACTIVE plugin and returned tablespace rows confirm the sensor will be able to discover the key IDs in use.

Step 4: Configure the CBOM Secure Sensor

```
sensors:
  - name: mariadb-tde-production
    type: Discover_DB
    enabled: true
    schedule: "0 */6 * * *"        # every 6 hours
    connection:
      vendor: mariadb
      host: mariadb-prod.internal.example.com
      port: 3306
      database: information_schema
      username: cbom_reader
      password: "{{ vault.mariadb_cbom_reader_password }}"
      tls:
        enabled: true
        ca_cert: /etc/cbom/certs/mariadb-ca.pem
        verify_server_cert: true
    discovery:
      targets: [innodb_tablespaces_encryption, plugin_status]
      key_management_plugin: file_key_management   # or aws_key_management
      include_unencrypted_tablespaces: false
    output:
      cbom_asset_types: [symmetric_key, encryption_scheme]
      tag_with: { environment: production, data_sensitivity: high }
```

Reference the cbom_reader password from a secrets manager, set key_management_plugin to match your deployment, and point ca_cert at the CA that signed the MariaDB server certificate.

Step 5: Validate

```
cbom sensor run --name mariadb-tde-production

mariadb -u cbom_reader -p -h mariadb-prod.internal.example.com \
  -e "SELECT COUNT(*) FROM information_schema.INNODB_TABLESPACES_ENCRYPTION;"
```

Confirm the run logs a successful connection, an ACTIVE plugin, and a discovered tablespace/KEY_ID count. In the CBOM inventory filtered by source mariadb-tde-production, verify symmetric-key and encryption-scheme records appear.

Common Errors

Access denied for INNODB_TABLESPACES_ENCRYPTION (ERROR 1142)

Cause: The SELECT privilege was not granted, or it was applied to a different host pattern than the sensor connects from.

Resolution: Re-run the GRANT SELECT for the exact host (or '%') and FLUSH PRIVILEGES.

Encryption plugin not found or inactive

Cause: The key management plugin is not loaded, or plugin_load_add is missing from my.cnf.

Resolution: Add plugin_load_add = file_key_management (plus key file settings) under [mysqld], restart MariaDB, and re-check SHOW PLUGINS.

TLS handshake failure — certificate verify failed

Cause: ca_cert does not match the CA that signed the MariaDB server certificate, or the path is wrong.

Resolution: Identify the issuer with openssl s_client -starttls mysql, export the correct CA, update ca_cert, and restart the sensor.

Security Recommendations

- Use a dedicated least-privilege account — never grant SUPER, FILE, or any write privilege; scope the host to the sensor IP in production.
- Store the cbom_reader password in a secrets manager and rotate it on schedule.
- Enforce TLS with verify_server_cert: true for all sensor connections.
- Enable MariaDB audit logging for cbom_reader and confirm only SELECT queries against information_schema occur.
- Use the CBOM inventory to identify key IDs overdue for rotation, and confirm records update to the new key version after rotation.

Conclusion

With a scoped cbom_reader user and TLS-secured credentials, the Discover_DB sensor continuously discovers and tracks MariaDB encryption key usage and tablespace encryption status without impacting database operations — keeping CBOM Secure's inventory aligned with your cryptographic governance policies.