(TDE) Oracle Database Encryption Setup Guide

TBD - FOR NEXT IMPLEMENTATION UPDATE STEPS TO USE AES256 ENCRYPTION FOR WALLET

Introduction

Setting up encryption for your Oracle database is essential for protecting sensitive information from unauthorized access. This guide outlines the necessary steps to enable encryption and secure your database environment.

Other documentation on TDE can be found here: Key Management (TDE)

Setup Steps

Step 1: Generate Parameter File

- **Action:** Create a parameter file (pfile) from the server parameter file (spfile).
- **Command:**

```
 CREATE PFILE='/<Service Group>/ora/admin/<Database Name>/pfile/init<Database Name>.ora_bkp' FROM SPFILE;
```

Step 2: Configure Encryption Wallet Location

- **Action:** Define the location of the encryption wallet in the `sqlnet.ora` file.
- **File:** `$ORACLE_HOME/network/admin/sqlnet.ora`
- **Content:**

```
 ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=FILE)(METHOD_DATA=(DIRECTORY=/<Service Group>/ora/admin
/<Database Name>/wallet)))
```

Step 3: Set Keystore Location and Type

- **Action:** Configure the keystore location and type, and restart the database.
- **Commands:**

```
ALTER SYSTEM SET WALLET_ROOT="/<Service Group>/ora/admin/<Database Name>/wallet/" SCOPE=SPFILE;
SHUTDOWN IMMEDIATE;
STARTUP;
ALTER SYSTEM SET TDE_CONFIGURATION="KEYSTORE_CONFIGURATION=FILE" SCOPE=SPFILE;
```

Step 4: Create Software Keystore

- **Action:** Initialize the software keystore.
- **Command:**

```
ADMINISTER KEY MANAGEMENT CREATE KEYSTORE '/<Service Group>/ora/admin/<Database Name>/wallet/tde'
IDENTIFIED BY <password>;
```

Step 5: Verify Wallet Status

- **Action:** Check the encryption wallet's status to confirm proper configuration.
- **Commands:**

```
 SELECT STATUS FROM V$ENCRYPTION_WALLET; SELECT WRL_TYPE, WRL_PARAMETER, STATUS, CON_ID FROM
V$ENCRYPTION_WALLET;
```

Step 6: Enable Auto Login Keystore

- **Action:** Set up auto-login for the keystore for easier access.
- **Command:**

```
ADMINISTER KEY MANAGEMENT CREATE AUTO_LOGIN KEYSTORE FROM KEYSTORE '/<Service Group>/ora/admin/<Database
Name>/wallet/tde' IDENTIFIED BY <password>;
```

Step 7: Open Software Keystore

- **Action:** Make the keystore accessible for use.
- **Command:**

```
 ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN FORCE KEYSTORE IDENTIFIED BY <password>;
```

Step 8: Set TDE Master Key

- **Action:** Define the Transparent Data Encryption (TDE) master key.
- **Command:**

```
 ADMINISTER KEY MANAGEMENT SET KEY FORCE KEYSTORE IDENTIFIED BY <password> WITH BACKUP;
```

Step 9: Create Encrypted Tablespace

- **Action:** Establish a new tablespace with encryption enabled for data storage.
- **Command:**

```
CREATE TABLESPACE <Tablespace Name> DATAFILE '/<Service Group>/ora/data01/<Database Name>/<Tablespace
Name>_0001.dbf' SIZE 1000M EXTENT MANAGEMENT LOCAL AUTOALLOCATE SEGMENT SPACE MANAGEMENT AUTO ENCRYPTION
DEFAULT STORAGE (ENCRYPT);
```

Step 10: Manage Table Movement and Index Rebuilding (!! MAKE SURE ONLINE CLAUSE IS USED !!)

- **Action:** Migrate tables and rebuild indexes in the encrypted tablespace.
- **Commands:**

```
-- To minimize human error, please generate move/rebuild script using below:
set linesize 600
set heading off
-- Move tables
SELECT 'ALTER TABLE '||OWNER||'.'||TABLE_NAME||' MOVE ONLINE TABLESPACE <Tablespace Name>;' from
dba_tables where TABLE_NAME IN ('<TABLE1>','<TABLE2>','<TABLE_N>');
--Rebuild indexes
SELECT 'ALTER INDEX '||OWNER||'.'|| INDEX_NAME ||' REBUILD TABLESPACE <Tablespace Name> ONLINE;' from
dba_indexes where  INDEX_NAME IN ('<INDEX1>','<INDEX2>','<INDEX_N>');
--Rebuild index partitions
SELECT 'ALTER INDEX '||index_owner||'.'|| INDEX_NAME ||' rebuild PARTITION ' || PARTITION_NAME ||'
TABLESPACE <Tablespace Name> ONLINE;' from dba_ind_partitions where  INDEX_NAME IN
('<INDEX1>','<INDEX2>','<INDEX_N>');

-- ALTER TABLE <Schema>.<Table> MOVE ONLINE TABLESPACE <Tablespace Name>;
-- ALTER INDEX <Schema>.<Index> REBUILD TABLESPACE <Tablespace Name> ONLINE;
```

Crontab Entry for Wallet Backups

Schedule regular wallet backups using the following crontab entry:

```
 30 04 * * * [ -f /ora/rman/scripts/backup_wallets.sh ] && /ora/rman/scripts/backup_wallets.sh > /tmp
/backup_wallets.out 2>&1
```

Add the Password into Sky Vault

Login to sky vault using the following link :
https://skyvault.bskyb.com/PasswordVault/v10/logon/radius

```
Accounts View (Classic UI) --> add account
store in safe --> Oracle Wallets
Device type --> Database
Platform Name --> Sky Oracle Database
Username --> U01_WALLET_MER (env+Wallet+Application)
Address --> /<service Group>/ora/admin/<database name>/wallet/tde (Wallet location)
Database Name --> <database name>
Port --> As per your database port
password --> please set the same password which you used for configuration
Tick the check box for disable automatic management for this account
save
```

Summary

Setting up encryption for your Oracle database is essential for safeguarding sensitive data against unauthorized access. Following the steps outlined in this guide ensures that your database is encrypted and secure. Here is a brief overview of the key steps:

- **Create pfile from spfile:** Generate a parameter file from the server parameter file to prepare for encryption configuration.
- **Set the ENCRYPTION_WALLET_LOCATION:** Define the location for encryption wallets in the `sqlnet.ora` file.
- **Configure the Keystore Location and Type:** Establish the keystore location, restart the database, and set the keystore type.
- **Create software Keystore:** Initialize a software keystore with a specified password.
- **Check the status of the wallet:** Confirm the encryption wallet's status to ensure it's operational.
- **Configure Auto Login Keystore:** Set up an auto-login keystore for seamless access.
- **Open the software Keystore:** Access the software keystore using the specified password.
- **Set the Keystore TDE Encryption Master Key:** Assign the Transparent Data Encryption (TDE) master key with a backup option for enhanced security.
- **Tablespace creation:** Generate tablespaces with encryption enabled to secure data at rest.
- **Table mv and Crontab Entry:** Implement additional steps for table movements and schedule wallet backups for ongoing maintenance.

By adhering to these steps, you will bolster the security measures of your Oracle database and protect sensitive information from potential security threats.