Practice 10

Using RMAN-Encrypted Backups

Practice Target

In this practice you will gain hands-on experience on implementing all the encryption modes in RMAN backups.

Practice Overview

In this practice, you will perform the following tasks:

- Configure Transparent Data Encryption
- Produce backupsets encrypted with the following methods:
  - Transparent mode encryption
  - Password mode encryption
  - Dual mode encryption

Assumptions

This practice assumes the `srv1` appliance is up and running and its database `ORADB` is running in `OPEN` state.

Note

I recommend taking a snapshot of the appliance `srv1` before you start implementing the practice.

A. Configuring Transparent Data Encryption

Configuring Transparent Data Encryption (TDE) is needed, if you want to use it for encrypting backup files in RMAN. In the following steps, you will configure TDE in `ORADB` database so that it can later be used by RMAN for encrypting its backups.

1. Open Putty and login to `srv1` as `oracle`.

2. Create a directory for saving the keystore in it.

```
mkdir /u01/app/oracle/oradata/ORADB/keystore
```

3. Specify the keystore location in `sqlnet.ora`

Do **not** copy the code from the PDF file into your `sqlnet.ora` file. Obtain it from the sqlnet.ora file added to the lecture downloadable resources.

```
vi $TNS_ADMIN/sqlnet.ora
```

```
# add the following to it:
ENCRYPTION_WALLET_LOCATION =
(SOURCE =
 (METHOD = FILE)  (METHOD_DATA =
 (DIRECTORY = /u01/app/oracle/oradata/ORADB/keystore)))
```

4. Invoke SQL*Plus and login as sysdba

```
sqlplus / as sysdba
```

5. Create the software keystore file:

```
ADMINISTER KEY MANAGEMENT CREATE KEYSTORE
 '/u01/app/oracle/oradata/ORADB/keystore' IDENTIFIED BY oracle;
```

6. Verify the created file (`ewallet.p12`)

```
! ls -al /u01/app/oracle/oradata/ORADB/keystore
```

7. Open the software keystore file:

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY oracle;
```

8. Create the master encryption key.

```
ADMINISTER KEY MANAGEMENT SET KEY
 IDENTIFIED BY oracle WITH BACKUP USING 'for_rman' ;
```

9. (optional) Retrieve the key identifier

```
SELECT KEY_ID FROM V$ENCRYPTION_KEYS;
```

Note: Not only RMAN can use this created Keystore encryption keystore. It can be used for encrypting the data in SQL as well.

B. Producing Transparent-mode Encrypted Backups

In the following steps, you will create backups that are encrypted in TDE mode.

10. Invoke RMAN and connect to the database `ORADB` as target.

```
rman target /
```

11. Display the configured encryption algorithm. List of available algorithms can be obtained from `V$RMAN_ENCRYPTION_ALGORITHMS`

```
SHOW ENCRYPTION ALGORITHM;
```

12. Execute the following commands to produce a backupset that is encrypted in TDE-mode.

The example below is using `SET` command to enable the encryption. You can use `CONFIGURE` command to make all the produced backups encrypted by default.

```
SET ENCRYPTION ON;
BACKUP TABLESPACE USERS TAG 'ENCRYPTED_USERS';
```

13. Issue the following command to list the generated backupset.

Observe that the output does not tell if the backup set is encrypted.

```
LIST BACKUPSET TAG 'ENCRYPTED_USERS';
```

14. Issue the following query to obtain information about the produced backupset. Observe that the query informs that the backup piece is encrypted.

```
SELECT S.RECID AS "BS_REC", P.RECID AS "BP_REC", P.ENCRYPTED
FROM   V$BACKUP_PIECE P, V$BACKUP_SET S
WHERE  P.SET_STAMP = S.SET_STAMP
  AND  P.SET_COUNT = S.SET_COUNT
  AND  P.TAG ='ENCRYPTED_USERS';
```

Note: You will learn how to restore from encrypted backup later in the course.

15. Restart the database.

```
SHUTDOWN IMMEDIATE
STARTUP
```

16. Trying taking another encrypted backup for the `users` tablespace.

You will receive the error "`ORA-28365: wallet is not open`". With this type of keystore, every time you start up the database, you cannot use it until you open it first. When you use the auto-login software keystore, encrypted backup operations can be performed at any time, because the auto-login keystore is always open.

```
SET ENCRYPTION ON;
BACKUP TABLESPACE users TAG 'ENCRYPTED_USERS';
```

17. Open the software keystore file.

```
sql 'ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY oracle';
```

18. Take an encrypted backup of the `users` tablespace.

```
SET ENCRYPTION ON;
BACKUP TABLESPACE users TAG 'ENCRYPTED_USERS';
```

Using Auto-login Software Keystore

In the following steps you will create an auto-login software keystore and test it.

Note: Auto-login software keystore is used in unattended scenarios.

19. Exit from RMAN, invoke SQL*Plus, and login as sysdba to `oradb`

```
sqlplus / as sysdba
```

20. Create auto-login keystore.

```
ADMINISTER KEY MANAGEMENT CREATE AUTO_LOGIN KEYSTORE FROM KEYSTORE
'/u01/app/oracle/oradata/ORADB/keystore' IDENTIFIED BY oracle;
```

21. Verify the created files (`ewallet*.p12` and `cwallet.sso`).

All those files should not be removed from that location.

```
! ls -al /u01/app/oracle/oradata/ORADB/keystore
```

22. Invoke RMAN and login to `ORADB` as target

```
rman target /
```

23. Restart the database.

```
SHUTDOWN IMMEDIATE
STARTUP
```

24. Trying taking another encrypted backup for the `users` tablespace.

The command should **not** return "`ORA-28365: wallet is not open`" error.

```
SET ENCRYPTION ON;
BACKUP TABLESPACE users TAG 'ENCRYPTED_USERS';
```

Note: To restore this encrypted backupset, the Keystore files must exist in the destination system. Otherwise, you cannot restore it.

Note: Do not include the auto-login Keystore files with the database backup files. Otherwise, if the backup files are accessed by unauthorized person, the person can restore them.

C. Producing Password-based Encrypted Backups

When password encryption is used, the DBA should provide a password when creating and restoring encrypted backups. In the next step, you will create backups that are encrypted with password.

25. Execute the following commands to produce password-encrypted backups for the `users` tablespace.

```
SET ENCRYPTION ON IDENTIFIED BY MyPassword ONLY;
BACKUP TABLESPACE USERS TAG 'PENCRYPTED_USERS';
```

- After creating an encrypted backupset, is there a way to know which mode has been used to create the encrypted backupset?

  Personally, I don't know any method to know directly on which mode an encrypted backupset has been encrypted. If you know one, please drop me a line. For this reason, I added to the TAG value the letter 'P' in the beginning to indicate that this backup has been encrypted with a password.

D. Producing Dual-mode Encrypted Backups

When dual-mode encryption is used, the DBA should provide a password when creating and restoring encrypted backups and, in addition to that, the Keystore must be opened.

26. Execute the following commands to produce dual-mode encrypted backups for the `users` tablespace.

The only difference between the code below with the code in the previous step is the `ONLY` keyword.

```
SET ENCRYPTION ON IDENTIFIED BY MyPassword;
BACKUP TABLESPACE USERS TAG 'DENCRYPTED_USERS';
```

Note: when restoring a dual-mode encrypted backup, you can use either the Oracle keystore or a password for decryption.

Clean Up

27. Delete the produced backupsets.

```
DELETE NOPROMPT BACKUPSET TAG 'ENCRYPTED_USERS';
DELETE NOPROMPT BACKUPSET TAG 'PENCRYPTED_USERS';
DELETE NOPROMPT BACKUPSET TAG 'DENCRYPTED_USERS';
```

28. Delete the appliance snapshot, if you have created one.

Summary

In this practice, you performed the following tasks:

- Configure Transparent Data Encryption
- Produce backupsets encrypted with the following methods:
  - Transparent mode encryption
  - Password mode encryption
  - Dual mode encryption