Managing PDB Lockdown Profiles

By Ahmed Baraka

Objectives

By the end of this lecture, you should be able to:

- Describe when you should use Lockdown profiles
- How to use PDB lockdown profiles
- How to configure an OS user for a PDB

About PDB Lockdown Profiles

- Used to define custom security policies for a PDB: which feature, option and/or statement you want to enable/disabled in a PDB.
- Categories of the features and operations you can restrict:
  - **Network access**: using UTL_TCP, UTL_HTTP, UTL_MAIL, UTL_SNMP, UTL_INADDR
  - **Operations on common users**: like adding objects in a common schema, granting privileges to common objects, etc.
  - **Operating System access**: like using UTL_FILE
  - **Connections**: common user to connect as SYSOPER

Using PDB Lockdown Profiles

1. Create a PDB Lockdown Profile (from the CDB$ROOT):

```
CREATE LOCKDOWN PROFILE cdb1_profile;
```

2. Set the restriction:

```
ALTER LOCKDOWN PROFILE cdb1_profile ..
```

3. Enable the a PDB Lockdown Profile:

```
ALTER SYSTEM SET PDB_LOCKDOWN = cdb1_profile;
```

4. To obtain information about created Lockdown Profiles:

```
SELECT  PROFILE_NAME, RULE_TYPE, RULE, STATUS
FROM DBA_LOCKDOWN_PROFILES ;
```

Using PDB Lockdown Profiles Example: Disable Database Options

- Possible options to enable/disable:
  - DATABASE QUEUING
  - PARTITIONING
- Disabling partitioning option:

```
ALTER LOCKDOWN PROFILE cdb1_profile
 DISABLE OPTION = ('PARTITIONING');
```

- When the partitioning is tried in the PDB:

```
CREATE TABLE SALES ( ID NUMBER ..)  PARTITION BY ..;


ORA-00439: feature not enabled: Partitioning
```

Using PDB Lockdown Profiles Example: Disable specific SQL Statement Clause

- Possible statements to enable/disable:
  - ALTER DATABASE
  - ALTER SESSION
  - ALTER PLUGGABLE DATABASE
  - ALTER SYSTEM

```
ALTER LOCKDOWN PROFILE cdb1_profile DISABLE STATEMENT=('ALTER
SYSTEM') CLAUSE ALL EXCEPT = ('KILL SESSION');
```

```
ALTER LOCKDOWN PROFILE cdb1_profile DISABLE
STATEMENT=('ALTER SYSTEM') CLAUSE=('SET');

ALTER LOCKDOWN PROFILE cdb1_profile ENABLE STATEMENT=('ALTER
SYSTEM') CLAUSE=('SET') OPTION=('undo_retention','heat_map');
```

Using PDB Lockdown Profiles Example: Control Values in SQL Statement

```
ALTER LOCKDOWN PROFILE cdb1_profile
 DISABLE STATEMENT = ('ALTER SYSTEM')
 CLAUSE = ('SET')
 OPTION = ('CPU_COUNT')
 MINVALUE = '2'
 MAXVALUE = '6';
```

Using PDB Lockdown Profiles Example: Disable specific Database Feature

- Feature category examples:
  - AWR
  - Connections
  - Common Schema access
  - Network access
- Refer to documentation for full list.

```
ALTER LOCKDOWN PROFILE cdb1_profile DISABLE
FEATURE=('UTL_HTTP','UTL_SMTP');
```

```
ALTER LOCKDOWN PROFILE cdb1_profile
DISABLE FEATURE = ('NETWORK_ACCESS');
```

About Configuring an OS User for a PDB

- PDB operations that interact with OS:
  - External jobs
  - External table
  - PL/SQL library executions
- Configure a separate OS user for each PDB using the parameter PDB_OS_CREDENTIAL
- Datafiles creation is not affected by this parameter

Configuring an OS User for a PDB

1. Login to the root with EXECUTE privilege on DBMS_CREDENTIAL
2. Create an Oracle credential for the operating system user

```
BEGIN
 DBMS_CREDENTIAL.CREATE_CREDENTIAL (
  CREDENTIAL_NAME => 'PDB1_OSU',
  USERNAME => 'pdb1_osuser', PASSWORD => 'password');
END;
```

3. Login to the PDB and set the PDB_OS_CREDENTIAL

```
ALTER SYSTEM SET PDB_OS_CREDENTIAL = PDB1_OSU SCOPE = SPFILE;
```

4. Restart the PDB

Summary

In this lecture, you should have learnt how to:

- Describe when you should use Lockdown profiles
- How to use PDB lockdown profiles
- How to configure an OS user for a PDB