Practice

Using Database Authentication Methods

Practice Target

In this practice you will study the OS authentication and the password file authentication in Oracle databases.

Specifically, you will perform the following:

- Examine the OS authentication for administrative users
- Create and use password files
- Examine the OS authentication for non-administrative Users

Practice Assumptions

This practice assumes that you have the virtual machine `srv1` up and running from its **non-CDB** snapshot.

Examining the OS Authentication for Administrative Users

In this section of the practice, you will examine how SYS can use OS authentication to login to the database. In the concepts lecture, we learnt that in Linux systems any user who is member of dba group can access the database as SYS user. In the following steps, you will examine this concept.

1. Open Putty and connect to srv1 as root

2. Create the following testing user and add it to the dba group.

```
useradd testuser -g dba
```

3. Login as the testing user then add the values of the environment variables required to connect to the database to the user bash shell profile. Source the bash shell profile.

The variables are set in the bash shell profile so that each time the user logs on to the system, the variables are automatically set.

```
su - testuser
echo "export ORACLE_HOME=/u01/app/oracle/product/19.0.0/db_1" >> .bash_profile
echo "export ORACLE_SID=oradb" >> .bash_profile
echo "export TNS_ADMIN=/u01/app/oracle/product/19.0.0/db_1/network/admin" >>
.bash_profile
echo "export PATH=\$PATH:/u01/app/oracle/product/19.0.0/db_1/bin" >>
.bash_profile

source .bash_profile
```

4. Login to the database as SYS

Observe that the user connects to the database as sys because it is a member of the dba group (by default). This means that the database in its current configuration can be accessed as SYS by the system administrator because the OS administrator can add any OS user to the dba group.

Furthermore, observe that the OS user connects to the database without creating a corresponding database user for it. This is true for any OS user that logs on to the database using any administrative privilege.

```
sqlplus / as sysdba
```

5. In SQL*Plus session, display the current user.

For any user who logs on to the database as sysdba, the database considers the logged on database user is SYS, regardless of the user used to login to the database. This is called the default schema.

```
show user
```

6. Try connecting to the database as sysoper using the OS authentication.

The connection fails because testuser is not a member of the oper group.

```
conn / as sysoper
```

7. Exit from the SQL*Plus and from current user shell to make `root` the current user

```
exit
exit
```

Let's examine the database behavior for the OS users who are granted the `sysoper` privilege.

8. Add `testuser` to `oper` group.

If `oper` group is not there, create it.

```
usermod testuser -G oper
```

9. As `testuser`, login to the database as `SYS`.

The connection succeeds because the user is a member of the `oper` group.

```
su - testuser
sqlplus / as sysoper
```

10. Display the current user.

When we connect to the database as `sysoper`, Oracle database considers the logged on database user is `public`.

```
show user
```

11. Exit from the SQL*Plus

```
exit
```

12. Exit from `testuser` session and change the current user to `oracle`

```
exit
su - oracle
```

Creating and Using Password Files

In the following steps, you will create a new password file and examine how it is used by the database.

Let's first see which users' passwords are saved in the password file.

13. Login as `SYS` to the database.

```
sqlplus / as sysdba
```

14. List all the users saved in the password file.

When the database was created, it saved only `SYS` user in the password file.

Observe that `testuser` is not included in the output, although it can log on to the database as `sysdba`. `testuser` can use OS authentication to login to the database, not the password file authentication.

```
col USERNAME for a15
SELECT USERNAME, SYSDBA, SYSOPER FROM V$PWFILE_USERS;
```

15. Exit from SQL*Plus.

```
exit
```

In the following steps, you will examine how the database behaves when both the OS authentication and the password file authentication are used.

16. Issue the following command and see if the login is successful.

The login is successful because the database actually uses the OS authentication to login to the database, not the password file authentication. OS authentication supersedes the password file authentication.

```
sqlplus sys/wrongpassword as sysdba
```

17. Issue the following connect attempt.

The login is unsuccessful because this format of connection uses the network Listener to connect to the database. Because the connection is not secure, it cannot be established using the OS authentication. The password must be correct to login to the database with this connection method.

```
conn sys/wrongpassword@oradb as sysdba
```

18. Issue the following connection attempt.

This connection attempt uses the password file authentication. The login is successful because the provided `sys` password is correct

```
conn sys/ABcd##1234@oradb as sysdba
```

19. Exit from SQL*Plus.

```
exit
```

Now, let's examine the password file itself.

20. Examine the permissions set for the password file in the operating system.

Only `oracle` user has the write/read permissions on the password file. `oinstall` group members are able to read the file. Other users are unable to access the file.

Note: As with all file names in Linux, password file name is case-sensitive. If the password file name does not match the case-sensitivity with the `$ORACLE_SID` value, the database is unable to see the password file.

```
ls -al $ORACLE_HOME/dbs/orapw$ORACLE_SID
```

21. Rename the existing password file to any other name.

Once the password file name is changed from its default name, the database is unable to see it anymore.

```
mv /u01/app/oracle/product/19.0.0/db_1/dbs/orapworadb
/u01/app/oracle/product/19.0.0/db_1/dbs/orapworadb.bak
```

22. Try to connect to the database as `SYS` using the password file authentication.

The connection attempt fails because the password file does not exist.

```
sqlplus sys/ABcd##1234@oradb as sysdba
```

23. Issue the following command to create a new password file.

The provided password is for `SYS` user. If you do not provide it to the command, you will be asked to enter the password when you run the command.

```
orapwd file=$ORACLE_HOME/dbs/orapw$ORACLE_SID password=ABcd##4321
```

24. Try to connect to the database as `SYS` using username/password credential.

The connection attempt succeeds because the provided password is saved in the password file for the `SYS` user.

```
sqlplus sys/ABcd##4321@oradb as sysdba
```

25. Exit from SQL*Plus

```
exit
```

Examining the OS Authentication for non-Administrative Users

In the following steps, you will create a new OS user and allow it to connect to the database as a non-administrative user.

Note: In production systems, users normally use database authentication for connecting to the database. Use OS authentication only if you need to.

26. Login to the database as `sys`

```
sqlplus / as sysdba
```

27. Display the value of `OS_AUTHENT_PREFIX`

The parameter is set to its default value, which is "`ops$`". When we create the OS user (to be used for connecting to the database), we do not include this prefix to it. But its corresponding database user must include this prefix.

Note: You can change this prefix value or disable it by setting it to `''`.

```
show parameter OS_AUTHENT_PREFIX
```

28. Create the database user that corresponds to the created OS user. Grant the required privileges to it.

Observe that in case of using OS authentication for administrative users, we did not need to create a corresponding database user accounts for it. The reason is that whenever a user logs on to the database using an administrative privilege, Oracle database automatically sets the current schema as the default schema of the administrative privilege.

```
CREATE USER OPS$DBUSER IDENTIFIED EXTERNALLY;
GRANT DBA, CONNECT TO OPS$DBUSER;

-- verify that user was created with the required authentication type:
SELECT AUTHENTICATION_TYPE FROM DBA_USERS WHERE USERNAME='OPS$DBUSER';
```

29. Exit from SQL*Plus and from `oracle` shell so that the current OS user is `root`

```
exit
exit
```

30. Create the OS user that corresponds to the created database user.

Observe the OS username does not include the OPS$ part.

Observe that the username in this command is in lower-case. But the username we used with the `CREATE USER` was in upper-case. This works fine because Oracle database deals with usernames in case-insensitivity manner.

```
useradd dbuser
```

31. Login as the OS user and try connecting to the database as `sysdba`.

The connection fails because it uses the administrative OS authentication method. This OS user is not a member of any administrative groups and therefore cannot login to the database using the administrative OS authentication method.

```
su - dbuser
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/db_1
export ORACLE_SID=oradb
export PATH=$PATH:/u01/app/oracle/product/19.0.0/db_1/bin
export TNS_ADMIN=/u01/app/oracle/product/19.0.0/db_1/network/admin
sqlplus / as sysdba
```

32. Try connecting to the database as a non-administrative user.

The connection is successful and we do not provide a password to the connection attempt. The user was authenticated externally (by OS in this case).

```
sqlplus /
```

33. Display the current user.

The current user is `OPS$DBUSER`, which is the database user that we created fort this OS user.

```
show user
```

Clean up

34. Exit from the SQL*Plus and from current user shell to make `root` the current user

```
exit
exit
```

35. Delete the testing users.

```
userdel -r dbuser
userdel -r testuser
```

36. (optional) Shutdown `srv1` and restore it from its **non-CDB** snapshot.

Summary

- We can allow OS users to connect to Oracle databases using administrative privileges using OS authentication. We just need to add the OS user to the specific OS group associated with the require administrative privilege.
- Password files are used to save administrative user passwords in it. In some scenarios, administrative users must use the password file authentication to login to the database.
- DBAs can create new password files for the databases.
- We can allow OS users to login to Oracle databases as non-administrative users. However, it real life scenario, whenever possible, this should be avoided.