Managing the Secure External Password Store for storing Oracle db credentials

Theory (from oracle docs)

Oracle wallet authentication (password store) can be used in the following scenarios:

a. To avoid storing clear text passwords in password files on local hosts or remote hosts.
b. Avoid enabling remote os-authentication on the database

About the Secure External Password Store

You can store password credentials for connecting to databases by using a client-side Oracle wallet. An Oracle wallet is a secure software container that stores authentication and signing credentials.

This wallet usage can simplify large-scale deployments that rely on password credentials for connecting to databases. When this feature is configured, application code, batch jobs, and scripts no longer need embedded user names and passwords. This reduces risk because the passwords are no longer exposed, and password management policies are more easily enforced without changing application code whenever user names or passwords change.

Note: The external password store of the wallet is separate from the area where public key infrastructure (PKI) credentials are stored. Consequently, you cannot use Oracle Wallet Manager to manage credentials in the external password store of the wallet. Instead, use the command-line utility `mkstore` to manage these credentials.

However, when clients are configured to use the secure external password store, applications can connect to a database with the following `CONNECT` statement syntax, without specifying database login credentials:

```
CONNECT /@db_connect_string

CONNECT /@db_connect_string AS SYSDBA

CONNECT /@db_connect_string AS SYSOPER
```

NOTE: In this specification, `db_connect_string` is a valid connection string to access the intended database, such as the service name, URL, or alias as shown in the earlier examples. Each user account must have its own unique connection string. You cannot create one connection string for multiple users.

In this case, the database credentials, user name and password, are securely stored in an Oracle wallet created for this purpose. The autologin feature of this wallet is turned on, so the system does not need a password to open the wallet. From the wallet, it gets the credentials to access the database for the user they represent.

The examples given below shows the steps to setup a batch account to be used on the 2 nodes of a db server that hosts a RACONE edition database that can be active on any one of the nodes in the cluster. This example holds good for single instance database, RAC and RACONE database since a service_name is used to connect to the db.

Unix batch accounts setup on both nodes looks like the ones given below. Only one of these accounts are used in the examples below for demonstrating the process.

```
[oracle@tslinrac01 ~]$ cat /etc/group |grep orabatch
orabatch:x:1302:orabatch1,orabatch2                    → Unix group "orabatch"

[oracle@tslinrac01 ~]$ cat /etc/passwd | grep orabatch      → unix account used
orabatch1:x:1102:1302:oracle batchid 1:/home/orabatch1:/bin/ksh
orabatch2:x:1103:1302:oracle batchid 2:/home/orabatch2:/bin/ksh

[oracle@tslinrac01 ~]$ ll /home |grep orabatch              → Home directory of users
drwx------  2 orabatch2 orabatch  4096 Jan  9 10:59 orabatch2/
drwx------  2 orabatch1 orabatch  4096 Jan  9 11:13 orabatch1/
```

Step 1 : Create a batch specific TNS_ADMIN folder

Login to the os account that owns the oracle binary on the first node. In this case 'oracle'

```
[oracle@tslinrac01 dbhome_1]$ hostname
tslinrac01
[oracle@tslinrac01 dbhome_1]$ whoami
oracle
[oracle@tslinrac01 dbhome_1]$ cd $ORACLE_HOME/network/admin
[oracle@tslinrac01 admin]$ pwd
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin
```

--Create a new directory to hold a separate wallet, tnsnames.ora and sqlnet.ora specific to the account 'orabatch1'

[oracle@tslinrac01 admin]$ mkdir TNS_orabatch1
[oracle@tslinrac01 admin]$ cd TNS_orabatch1
[oracle@tslinrac01 TNS_orabatch1]$ pwd
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/TNS_orabatch1

Create a new sqlnet.ora file in this directory that looks like below. DIRECTORY should be having the full path name of the newly created batch specific TNS_<batchid> directory:

```
NAMES.DEFAULT_DOMAIN=dba.com
adr_base=/u01/app/oracle
diag_adr_enabled=on
SQLNET.WALLET_OVERRIDE = TRUE
SSL_CLIENT_AUTHENTICATION = FALSE
WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
  (DIRECTORY =
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/TNS_orabatch1 )
  )
```

Place a new tnsnames.ora file that contains the tns alias of the database that this batch id needs to connect to. This tnsnames.ora can have multiple tns entries. Tns entries can be for remote databases to which this batch id will be using a wallet to connect and it can also contain tns entries for databases on the current server. Once wallet authentication is enabled for this batch account, OS authentication to local databases will not work. Wallet authentication needs to be setup to connect to local databases also.
Sample file is :

```
# tnsnames.ora Network Configuration File:
# Generated by Oracle configuration tools.
racdb.dba.com =
(DESCRIPTION =
  (FAILOVER=on)
  (ADDRESS = (PROTOCOL = TCP)(HOST = tslinrac01-vip.dba.com)(PORT = 1521))
  (ADDRESS = (PROTOCOL = TCP)(HOST = tslinrac02-vip.dba.com)(PORT = 1521))
  (CONNECT_DATA =
  (SERVICE_NAME = racdbservice)
   (FAILOVER_MODE=
     (TYPE=select)
     (METHOD=basic)
      (RETRIES=20)(DELAY=5)
    )
  )
)
```

Step 2 : Create a Oracle wallet and password store

Steps shown below shows how to create a wallet and add database credentials to the password store for connecting to a local or remote database.

[oracle@tslinrac01 TNS_orabatch1]$ mkstore -wrl /u01/app/oracle/product/11.2.0/dbhome_1/network/admin/TNS_orabatch1 -create
Oracle Secret Store Tool : Version 11.2.0.1.0 - Production

Enter password:
Enter password again:

Note: Password used for the wallet is “test123!”. This is the password that is set for the wallet and this password needs to be specified every time an operation needs to be done on the wallet.

Create a PASSWORD authenticated oracle account to match with the os-batch account “orabatch1” . The password should be complicated enough since the mkstore command will check the password complexity.

```
SQL> create user orabatch1 identified by Orabatch1
  2  default tablespace users
  3  temporary tablespace temp;

User created.

SQL> grant create session to orabatch1;

Grant succeeded.
```

Test the newly created account via sqlnet.

```
[oracle@tslinrac01 TNS_orabatch1]$ sqlplus orabatch1/Orabatch1@racdb

SQL*Plus: Release 11.2.0.1.0 Production on Sat Jan 9 12:49:29 2010

Copyright (c) 1982, 2009, Oracle.  All rights reserved.
```

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, Real Application Clusters, Automatic Storage Management, Oracle Label Security,
OLAP, Data Mining, Oracle Database Vault and Real Application Testing options

Store the credentials into the wallet.

[oracle@tslinrac01 TNS_orabatch1]$ mkstore -wrl /u01/app/oracle/product/11.2.0/dbhome_1/network/admin/TNS_orabatch1 -createCredential racdb orabatch1
Oracle Secret Store Tool : Version 11.2.0.1.0 - Production
Copyright (c) 2004, 2009, Oracle and/or its affiliates. All rights reserved.

Your secret/Password is missing in the command line
Enter your secret/Password: → oracle password for account orabatch1
Re-enter your secret/Password:
Enter wallet password:
Create credential oracle.security.client.connect_string1

PASSWORD_POLICY : Passwords must have a minimum length of eight characters and contain alphabetic characters combined with numbers or special characters.

Step 3: Test the wallet/password store using the account 'oracle'

[oracle@tslinrac01 TNS_orabatch1]$ pwd
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/TNS_orabatch1
[oracle@tslinrac01 TNS_orabatch1]$ export TNS_ADMIN=/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/TNS_orabatch1
[oracle@tslinrac01 TNS_orabatch1]$ tnsping racdb

TNS Ping Utility for Linux: Version 11.2.0.1.0 - Production on 10-JAN-2010 11:15:50

Copyright (c) 1997, 2009, Oracle. All rights reserved.

Used parameter files:
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/TNS_orabatch1/sqlnet.ora

Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (FAILOVER=on) (ADDRESS = (PROTOCOL = TCP)(HOST = tslinrac01-vip.dba.com)(PORT = 1521)) (ADDRESS = (PROTOCOL = TCP)(HOST = tslinrac02-vip.dba.com)(PORT = 1521)) (CONNECT_DATA = (SERVICE_NAME = racdbservice) (FAILOVER_MODE= (TYPE=select) (METHOD=basic) (RETRIES=20)(DELAY=5))))
OK (10 msec)
[oracle@tslinrac01 TNS_orabatch1]$ whoami
oracle
[oracle@tslinrac01 TNS_orabatch1]$ sqlplus /@racdb

SQL*Plus: Release 11.2.0.1.0 Production on Sun Jan 10 11:15:59 2010

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, Real Application Clusters, Automatic Storage Management, Oracle Label Security,
OLAP, Data Mining, Oracle Database Vault and Real Application Testing options

SQL> show user
USER is "ORABATCH1"
SQL>

Change the permission on the directory so that unix account orabatch1 can read it.

[oracle@tslinrac01 admin]$ pwd
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin
[oracle@tslinrac01 admin]$ chmod -R 555 TNS_orabatch1/
[oracle@tslinrac01 admin]$ ll TNS_orabatch1/
total 16
-r-xr-xr-x 1 oracle oinstall 3880 Jan 9 12:51 ewallet.p12*
-r-xr-xr-x 1 oracle oinstall 3957 Jan 9 12:51 cwallet.sso*
-r-xr-xr-x 1 oracle oinstall 510 Jan 9 12:58 tnsnames.ora*
-r-xr-xr-x 1 oracle oinstall 300 Jan 9 13:03 sqlnet.ora*

This is to make sure that account orabatch1 can read the wallet and connect to the target db. Once that test is done, we will change the ownership of the directory TNS_orabatch1 to the unix account orabatch1.

Example above shows that while connected as unix user “oracle”, we can connect to the database racdb on a different server using “/@racdb” syntax. This will use the password stored in the wallet and connect to the target as that user. Show user command validates this.

Step 4: Configure the the .profile entry of the batch account to use the new TNS_ADMIN location

TNS_ADMIN has to be set in the batch accounts shell environment so that it can use the oracle wallet/password store to connect to the databases without having to specify the credentials.

```
$ pwd
/home/orabatch1
$ whoami
orabatch1
$ ls -ltr .profile
-rw------- 1 orabatch1 orabatch 5694 Jan 10 11:19 .profile
$ cat .profile | grep TNS
export TNS_ADMIN=/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/TNS_orabatch1
# TNS_ADMIN
$ tnsping racdb

TNS Ping Utility for Linux: Version 11.2.0.1.0 - Production on 10-JAN-2010 11:22:31

Copyright (c) 1997, 2009, Oracle.  All rights reserved.

Used parameter files:
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/TNS_orabatch1/sqlnet.ora


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (FAILOVER=on) (ADDRESS = (PROTOCOL
= TCP)(HOST = tslinrac01-vip.dba.com)(PORT = 1521)) (ADDRESS = (PROTOCOL =
TCP)(HOST = tslinrac02-vip.dba.com)(PORT = 1521)) (CONNECT_DATA =
(SERVICE_NAME = racdbservice) (FAILOVER_MODE= (TYPE=select)
(METHOD=basic) (RETRIES=20)(DELAY=5))))
OK (0 msec)
$ sqlplus /@racdb

SQL*Plus: Release 11.2.0.1.0 Production on Sun Jan 10 11:22:37 2010
```

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, Real Application Clusters, Automatic Storage Management, Oracle Label Security,
OLAP, Data Mining, Oracle Database Vault and Real Application Testing options

SQL> show user
USER is "ORABATCH1"
SQL>

Step 5: Change the permission and the ownership of the new TNS_ADMIN directory

[oracle@tslinrac01 admin]$ ls -ltr
total 16
-rw-r--r-- 1 oracle oinstall 187 May 9 2007 shrept.lst
drwxr-xr-x 2 oracle oinstall 4096 Dec 21 00:09 samples/
-rw-r--r-- 1 oracle oinstall   0 Jan 8 21:58 sqlnet.ora
-rw-r----- 1 oracle oinstall 502 Jan 8 22:50 tnsnames.ora
dr-xr-xr-x 2 oracle oinstall 4096 Jan 9 13:03 TNS_orabatch1/
[oracle@tslinrac01 admin]$ chmod -R 750 TNS_orabatch1/
[oracle@tslinrac01 admin]$ ll TNS_orabatch1/
total 16
-rwxr-x--- 1 oracle oinstall 3880 Jan 9 12:51 ewallet.p12*
-rwxr-x--- 1 oracle oinstall 3957 Jan 9 12:51 cwallet.sso*
-rwxr-x--- 1 oracle oinstall 510 Jan 9 12:58 tnsnames.ora*
-rwxr-x--- 1 oracle oinstall 300 Jan 9 13:03 sqlnet.ora*

--Change ownership of the directory and files to orabatch1.

```
[oracle@tslinrac01 admin]$ su -
Password:
[root@tslinrac01 ~]# cd
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin
[root@tslinrac01 admin]# chown -R orabatch1:oinstall TNS_orabatch1/
[root@tslinrac01 admin]# ll
total 16
drwxr-xr-x 2 oracle    oinstall 4096 Dec 21 00:09 samples
```

```
-rw-r--r-- 1 oracle    oinstall  187 May  9  2007 shrept.lst
-rw-r--r-- 1 oracle    oinstall    0 Jan  8 21:58 sqlnet.ora
-rw-r----- 1 oracle    oinstall  502 Jan  8 22:50 tnsnames.ora
drwxr-x--- 2 orabatch1 oinstall 4096 Jan  9 13:03 TNS_orabatch1
[root@tslinrac01 admin]# ll TNS_orabatch1
total 16
-rwxr-x--- 1 orabatch1 oinstall 3957 Jan  9 12:51 cwallet.sso
-rwxr-x--- 1 orabatch1 oinstall 3880 Jan  9 12:51 ewallet.p12
-rwxr-x--- 1 orabatch1 oinstall  300 Jan  9 13:03 sqlnet.ora
-rwxr-x--- 1 orabatch1 oinstall  510 Jan  9 12:58 tnsnames.ora
```

NOTE: Any os-user who can read the wallet can now connect to the db if they know the TNS alias used for storing the passwords in the wallet.

Step 6: Test the db account after logging in as the unix batch account.

This step is mandatory to make sure the setup is working. Need to do this even in production so that it works before the job kicks in.

```
[root@tslinrac01 admin]# su - orabatch1
$ tnsping racdb

TNS Ping Utility for Linux: Version 11.2.0.1.0 - Production on 10-JAN-2010 11:26:01

Copyright (c) 1997, 2009, Oracle.  All rights reserved.

Used parameter files:
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/TNS_orabatch1/sqlnet.ora


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (FAILOVER=on) (ADDRESS = (PROTOCOL
= TCP)(HOST = tslinrac01-vip.dba.com)(PORT = 1521)) (ADDRESS = (PROTOCOL =
TCP)(HOST = tslinrac02-vip.dba.com)(PORT = 1521)) (CONNECT_DATA =
(SERVICE_NAME = racdbservice) (FAILOVER_MODE= (TYPE=select)
(METHOD=basic) (RETRIES=20)(DELAY=5))))
OK (0 msec)
$ sqlplus /@racdb

SQL*Plus: Release 11.2.0.1.0 Production on Sun Jan 10 11:26:05 2010

Copyright (c) 1982, 2009, Oracle.  All rights reserved.
```

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, Real Application Clusters, Automatic Storage Management, Oracle Label Security,
OLAP, Data Mining, Oracle Database Vault and Real Application Testing options

SQL> show user
USER is "ORABATCH1"

Step 7: One time setup for dba's

Change the permission on the "mkstore" command so that its accessible to only dba/oinstall group.

[oracle@tslinrac01 ~]$ ls -ltr /u01/app/oracle/product/11.2.0/dbhome_1/bin/mkstore
-rwxr-xr-x 1 oracle oinstall 2824 Dec 21 23:27
/u01/app/oracle/product/11.2.0/dbhome_1/bin/mkstore*
[oracle@tslinrac01 ~]$ chmod 750
/u01/app/oracle/product/11.2.0/dbhome_1/bin/mkstore
[oracle@tslinrac01 ~]$ ls -ltr /u01/app/oracle/product/11.2.0/dbhome_1/bin/mkstore
-rwxr-x--- 1 oracle oinstall 2824 Dec 21 23:27 /u01/app/oracle/product/11.2.0/dbhome_1/bin/mkstore*

Step 8: Test the database connectivity in a RACONE database

[oracle@tslinrac01 ~]$ whoami
oracle
[oracle@tslinrac01 ~]$ su - orabatch1
Password:
$ echo $TNS_ADMIN
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/TNS_orabatch1
$ ll $TNS_ADMIN
total 16
-rwxr-x--- 1 orabatch1 oinstall 3880 Jan 9 12:51 ewallet.p12*
-rwxr-x--- 1 orabatch1 oinstall 3957 Jan 9 12:51 cwallet.sso*
-rwxr-x--- 1 orabatch1 oinstall 510 Jan 9 12:58 tnsnames.ora*
-rwxr-x--- 1 orabatch1 oinstall 300 Jan 9 13:03 sqlnet.ora*
$ sqlplus /@racdb

```
SQL*Plus: Release 11.2.0.1.0 Production on Mon Jan 11 22:18:00 2010

Copyright (c) 1982, 2009, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, Real Application Clusters, Automatic Storage Management, Oracle
Label Security,
OLAP, Data Mining, Oracle Database Vault and Real Application Testing options

SQL> show user
USER is "ORABATCH1"
SQL> !cat 1.sql
set line 120
column host_name format A15
select instance_number,instance_name,host_name,version,status ,logins
from v$instance
/

SQL> @1

INSTANCE_NUMBER INSTANCE_NAME    HOST_NAME       VERSION           STATUS       LOGINS
--------------- ---------------- --------------- ----------------- ------------ ---------
-
              2 racdb_2          tslinrac01      11.2.0.1.0        OPEN         ALLOWED
```

NOTE: Use Omotion to relocate the instance to the second node (omotion details given below)

```
Result shown below is AFTER executing Omotion to relocate the instance to the second
node.

SQL> /

INSTANCE_NUMBER INSTANCE_NAME    HOST_NAME       VERSION           STATUS       LOGINS
--------------- ---------------- --------------- ----------------- ------------ ---------
-
              1 racdb_1          tslinrac02      11.2.0.1.0        OPEN         ALLOWED

SQL>
```

Result of executing Omotion to relocate the instance

Omotion

RAC One Node databases on this cluster:

#	Database	Server	Fix Required
[1]	racdb	tslinrac01	N

Enter number of the database to migrate [1]:

Specify maximum time in minutes for migration to complete (max 30) [30]: 2

Available Target Server(s) :

#	Server	Available
[1]	tslinrac02	Y

Enter number of the target node [1]:

Omotion Started...
Starting target instance on tslinrac02...
Migrating sessions...
Stopping source instance on tslinrac01...
Omotion Completed...

=== Current Status ===
Database racdb is running on node tslinrac02

[oracle@tslinrac02 ~]$

Limitations:

1. TWO_TASK wont work with this setup for the target db.
2. If the account need to connect to a db on the local server (server where the account is created), it requires configuring a password account into the wallet for local databases also. This is required since setting SQLNET.WALLET_OVERRIDE = TRUE in the sqlnet.ora does not allow os-authenticated connections to the local db.
3. Any os-user that can read the wallet files and knows the connect string can connect to the db. So, the os-permission on the wallet has to be controlled tightly.