Managing TDE wallets in a RAC environment [ID 567287.1]

Modified 11-AUG-2011 *Type* HOWTO *Status* PUBLISHED

In this Document

Applies to:

Advanced Networking Option - Version: 10.2.0 to 11.2.0.2 [Release: to 11.2]
Information in this document applies to any platform.
Checked for relevance on 11-Aug-2011

Goal

What special considerations exist when managing TDE wallets in a RAC environment?

Solution

Before Oracle RDBMS 11gR2 TDE wallet management in RAC is not automated. Operations like opening the wallet, closing the wallet, and changing the Master Key are restricted to a single RAC instance, where they were executed. These operations do not have a RAC-wide effect; they have only a local instance effect.

On manually managing TDE wallets in RAC, the "Design and Deployment Techniques" chapter of the *Real Application Clusters Administration and Deployment Guide* includes the section "Transparent Data Encryption and Wallets" on how to setup the TDE wallet in a RAC environment:

> "Wallets used by Oracle RAC instances for Transparent Database Encrypion may be a local copy of a common wallet shared by multiple nodes or a shared copy residing on a network file system that all of the nodes can access. A deployment with a single wallet on a shared disk requires no additional configuration to use Transparent Data Encryption. Deployments where no shared storage exists require that each Oracle RAC node maintain its own local wallet. Details about creating and provisioning a wallet can be found in the Database Security Guide.
>
> "After you create and provision a wallet a single node, you must copy the wallet and make it available to all of the other nodes. For systems using Transparent Data Encryption with encrypted wallets, you can use any standard file transport protocol. For systems using Transparent Data Encryption with obfuscated wallets, file transport through a secured channel is recommended. The wallet must reside in the directory specified by the setting for the ENCRYPTION_WALLET_LOCATION or WALLET_LOCATION parameter in sqlnet.ora. The local copies of the wallet need not be synchronized for the duration of Transparent Data Encryption usage until the server key is re-keyed though the ALTER SYSTEM SET KEY SQL statement. Each time you run the ALTER SYSTEM SET KEY statement at a database instance, you must again copy the wallet residing on that node and make it available to all of the other nodes. To avoid unnecessary administrative overhead, reserve re-keying for exceptional cases where you are certain that the server master key is compromised and that not re-keying it would cause a serious security problem."

The following points should also be taken into consideration:

1) The sqlnet.ora of each instance should have the location of a local copy of the RAC instance's wallet.

2) Care should be taken that only one instance sets the Master Key. Once the Master Key is set the wallet has to be manually copied to all the instances of the cluster to the locations as indicated by the sqlnet.ora of that instance.

The wallet has to be reopened on all the instances after the copying of wallet is completed. Only after all the instances have re-opened the wallet can a Master Key re-key command be executed.

3) Care should be taken that only one instance issues the Master Key re-key at any given time. After the instance has issued the Master Key re-key command the wallet has to be copied to all the instances of the RAC database. Again, all instances would have to re-open the wallet after it has been copied. Only after all the instances have re-opened the wallet can another Master Key re-key command can be executed.

4) Do not issue any wallet open or close command while setting up or changing the Master Key.

5) Do not perform any TDE operations while setting up or changing the Master Key.

6) A wallet needs to be opened on all instances of RAC databases in order to have the wallet open in RAC.

7) A wallet needs to be closed on all instances of RAC databases in order to have the wallet closed in RAC.

Starting with Oracle Database 11g Release 2 Oracle recommends to store the Oracle Wallet in a centralized location. This centralized location can be an ACFS directory, a directory on a third party clustered file system or a HSM device. When the wallet is stored in a centralized location the commands to open or close the Wallet or re-key the unified master encryption key are propagated automatically to all the other instances.

The steps to do this are:

1) Identify a directory accessible from all the nodes. You can create an ACFS file system in ASM using „asmca" (ASM Configuration Assistant) and store the wallet there.

2) Add the ENCRYPTION_WALLET_LOCATION parameter in the sqlnet.ora file of all the nodes:

```
ENCRYPTION_WALLET_LOCATION=
    (SOURCE =
      (METHOD = FILE)
       (METHOD_DATA = (DIRECTORY = /opt/oracle/acfsmounts/tdevolume/wallet/)))
```

This file system is mounted automatically when the instances start. Opening and closing the wallet, as well as commands to set or rekey/rotate the TDE master encryption key are synchronized between all nodes.

It is recommended to enable TDE with a shared wallet during a planned downtime; if the changes to the sqlnet.ora file are applied on all instances of a running RAC system, restarting the individual instances one by one (except the first one where the wallet was created) is necessary to initially synchronize the wallet status and to update the view gv$encryption_wallet with the DIRECTORY entry from sqlnet.ora.

If the Oracle Wallet cannot be stored on a centralized location, it needs to be copied to all instances:
1) Create the wallet and the master key on the first instance
2) Copy the wallet to all other instances.

In this scenario the wallet open/close commands are synchronized between all the RAC instances, even if the wallet is not stored in a central location. The master key rekey operations however are not synchronized and as such it is necessary to copy the wallet to all the other nodes after recreating the master key.

References

Related

Products

- Oracle Database Products > Oracle Database > Net Services > Advanced Networking Option

Keywords

ENCRYPTION; LOCAL INSTANCE; REAL APPLICATION CLUSTER; REAL APPLICATION CLUSTERS; SQLNET.ORA; TDE; TRANSPARENT DATA ENCRYPTION; WALLET_LOCATION

Back to top