Security in and outside of the database vault

Michelle Malcher

Keywords: Security, Database Vault, Testing Security

The work day starts out with a question," if we had some super sensitive data that only a very small group of people should have access to, can that be done in the database?" It continues, "the other data is needed by others to manage their services so they need to see summaries of some data, complete information about another area. So, how can we make this happen?" This is seems to me to be a typical question when looking at security in the database. Security is not just about access to schemas and tables, but it is looking at what data is being stored and the rules and policies that need to go around the data to protect from information getting into the wrong hands, handling compliance and regulations as well as allowing the applications to do what they need to do.

Security Planning

It would seem at this point the first step is to understand more of the requirements, gather what data is sensitive and what applications are going to be accessing the data. Also are there any regulations around the data, such as PCI or HIPA. DBAs are also in a position to see all of the data in the database with the administrator privileges, so is the data so sensitive that the DBAs should be restricted. All of these requirements point to security options that can be implemented in the database or application.

Providing a secure Oracle database environment is the baseline, and then the classification of the data will add requirements to general securing of the database. As a DBA part of the planning is knowing this baseline, and then knowing enough about the advance security options to know if the requirements can be met with database security or if there is security being handled in the application.

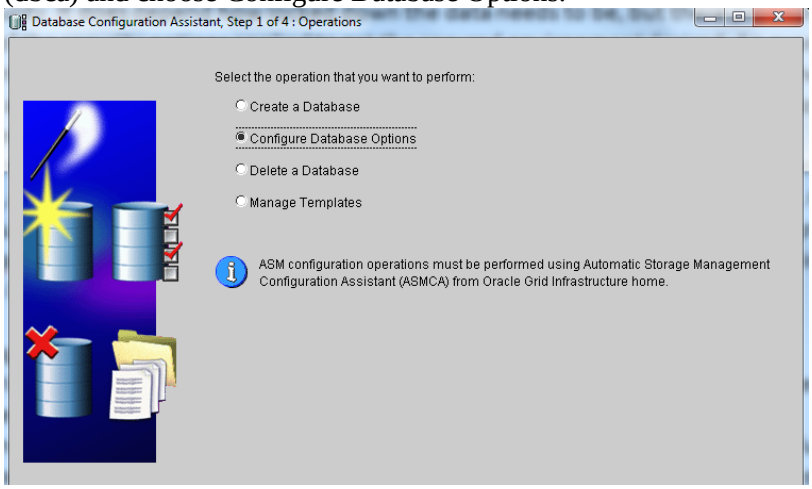
Database security can be easy to implement, and there are several layers that obviously allow different layers of security. It does depend how locked down the data needs to be, but there probably needs to be a couple of the security options applied to get the secured environment desired. An example would be implementing Oracle Database Vault will allow for creating realms that will even limit the access to the DBA and administrators, but if the data is not encrypted, backups and datafiles can possibly be hacked to view the data.

Back to the initial question, in looking at the sensitivity of the data, and how the application needs to use the data, a couple of security options can be implemented to satisfy the requirements. Oracle Database Vault to restrict the access of the system administrators, transparent data encryption to encrypt the tablespaces without having to worry about encryption in the application and virtual private databases to limit the access inside the application.

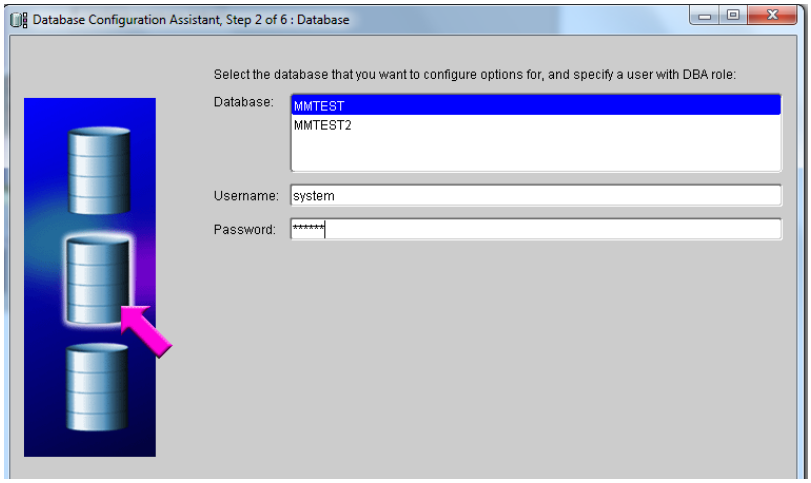
Implementing

Overall securing of the database will be considered here with implementing some of the best practices for a secure environment. This would include using least privilege, and installing only the components needed (but don't forget to install the advanced security option), making sure the security patches are applied. The base securing of database is a whole other topic and there are papers and presentations out there regarding that, but for this paper the focus is on some of the advanced security options and how to test that implementation.

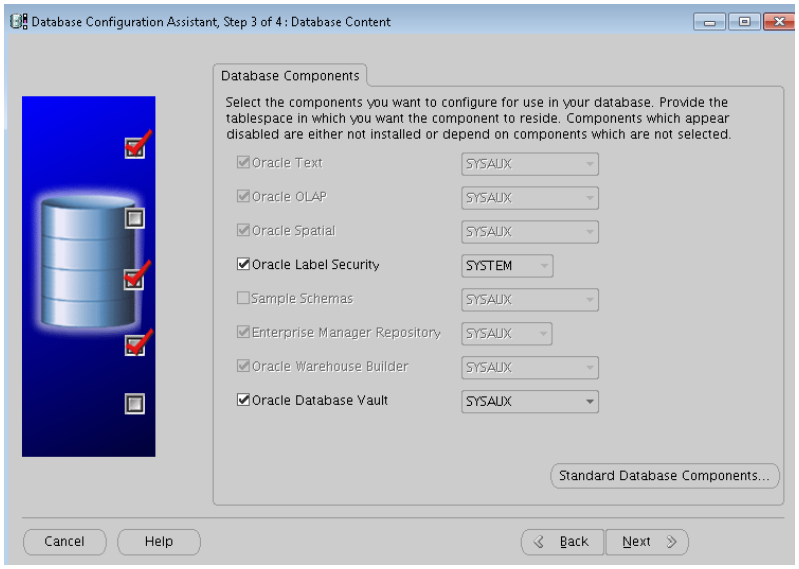
Oracle Advance Security option is needed for the database to implement these security solutions in the database. Oracle Database Vault is an option to install when performing the Oracle base installation, and then enabling the option in the database. Start database configuration assistant (dbca) and choose Configure Database Options.



Step two is to choose a database instance, and provide the user name and password with at least DBA role.



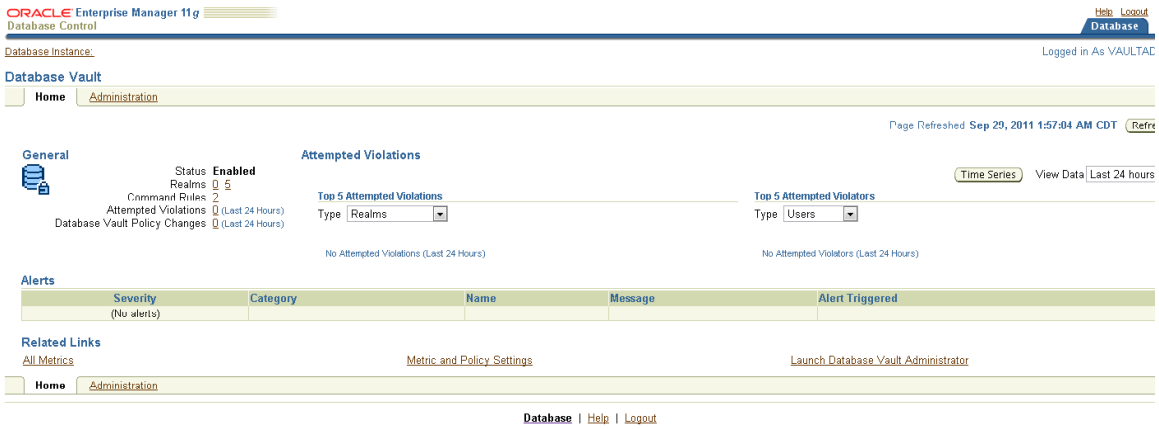
Choose the Database Components to install.

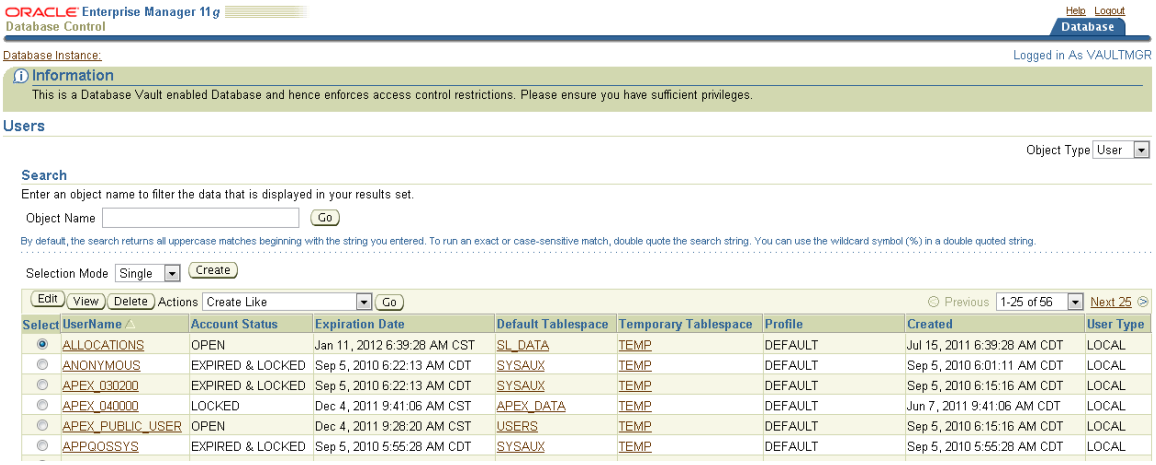


This will install the component, and to check that it is enabled in the database, run the following command:

```
SQLPLUS> select * from v$option where parameter = 'Oracle
Database Vault';
```

There is an option to choose an administrator for the vault and a security manager for the vault. This will depend on the requirements of separation of duties and if the person creating the users will also manage the vault realms for permissions. The other piece of this, the vault administrator, the one who manages the realms and creating user, is no longer the DBA. Having the DBA perform this task, even though audited, would mean that they could grant the permissions to see the data in the realm that was supposed to be protected from the DBA seeing. This might have been part of the discussion in the initial planning of the security requirements, but the owner of the data might now be in charge of creating the realms and granting these permissions. Database Vault Administrator is part of Enterprise Manager, and can be used to manage the realms and add users.





The users and schemas along with objects are created. The realm is created and considered a grouping of the objects or schemas and the users that are allowed to access the data and objects. Any users that is not in the realm is prohibited from accessing what is in the realm, and this includes all of the administrative accounts.
The realm can have all of the objects for the application, or it can just contain the sensitive objects. Having it contain just the objects that are of sensitive in nature at least allows for easier access to the rest of the data especially if used for other applications.

Database Vault not Enough
It was already discussed that more than one security option might be needed to secure the environment. This is also the case for the database vault. It works well with the other options such as transparent data encryption. Implementing the Advanced Security Options brings along the transparent data encryption, TDE. This provides an ability to encrypt the stored data. Just having the database vault installed only prevents the access when the database is being accessed as it should be either through the application or database access tools. Not that this is recommended, but reading through the data file will show the data in plain text and if parsed the information can be deciphered.
TDE needs a wallet location. The wallet holds the encryption and decryption keys and is used when the database is open. Losing the wallet password and not having the wallet open to open the database will not decrypt the data when the database is opened. An ecrypted tablespace can now be created, and the table with the sensitive data can either be created in this tablespace, or moved to this tablespace.

```
CREATE TABLESPACE DATA_ENCRYPT01 DATAFILE
'/u01/app/oracle/oradata/mmtest/data_encrypt01.dbf' SIZE 100M
   ENCRYPTION DEFAULT STORAGE (ENCRYPT);
```

The advantage here, is that the application does not need to script for encrypting and decrypting the data. Also there is minimal performance impact for this type of encryption.

Another Option
I am sure that it isn't going to come as a surprise that the vault and encryption is not enough. The data needs to be accessed by the application and then there are additional queries outside of the application that can run on an ad-hoc basis. The data access is based on a role and a category that is part of the table columns.

This is where creating policies and a virtual private database can come into play. Setting up the virtual private database involves a policy, database trigger on login and procedure to set the context.

```
BEGIN
  SYS.DBMS_RLS.ADD_POLICY    (
    object_schema         => 'HR'
    ,object_name          => 'EMP_DETAILS'
    ,policy_name          => 'EMP_IU'
    ,function_schema      => 'HR'
    ,policy_function      => 'MANAGER_ROLE_ONLY'
    ,statement_types      => 'SELECT'
    ,policy_type          => dbms_rls.dynamic
    ,long_predicate       => FALSE
    ,update_check         => TRUE
    ,static_policy        => FALSE
    ,enable               => TRUE );
END;
/
CREATE OR REPLACE PROCEDURE HR.set_role_mgr
as
var_role varchar2(30);
begin
select rolename into var_role
from HR_ROLES
where upper(username)=upper(sys_context ('userenv','session_user'));
dbms_session.set_context (namespace => 'realm_role_ctx', attribute => 'rolename', value =>
var_role);
end;
/
CREATE OR REPLACE TRIGGER SYS.set_user_role
after logon on database
begin
hr. set_role_mgr;
exception
when no_data_found
then
null;
end;
/
```

Testing

With the virtual private database not all rows are going to be return if the user doesn't have the full permissions, there are going to be different counts. A simple test can be setup to validate counts and differences to provide a check against the database to see the permissions at work. These types of tests are important to setup and continue to check. The initial test to just make sure that the security permission is setup properly, and the continuous tests to validate that as changes happen, there is are no changes to permissions.

Auditing of the access can be also validated by turning on auditing for the table and reviewing the logs. The audit logs in the database vault administrator will provide details on changes or access that has been granted.

Summary
The title of this presentation is called in and outside of the database vault, because it isn't just the security of the database vault that needs to be configured and setup to secure the environment. It is normally a few options working together to secure the sensitive data and environment. The initial step is planning for the security. What needs to be secured? How sensitive is the data? And understanding the different options available.
Implementation of the security options is a straight forward process. However, knowing the security options to what options to pair together and validate the implementation is where the value is at. Finally, it is educating others about the access, and how to maintain the access through the different tools such as database vault administration. Understanding what security can be completed in the database, can make it easier on the application side.

Contact Address:
540 W. Madison
Chicago, IL 60661
312-542-8909

Michelle Malcher
Email: michelle_malcher@ioug.org