

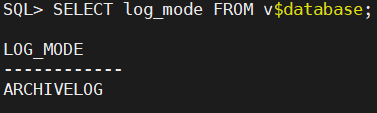
**ORACLE DATABASE VAULT**

**implement oracle vault in database:**

**Step 1-1: Verify Prerequisites:**

-Verify that your database is in ARCHIVELOG mode:

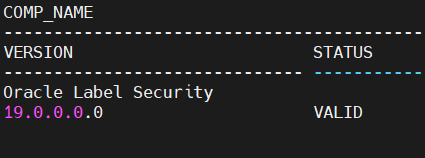
SQL> SELECT log\_mode FROM v$database;



-Check if Oracle Label Security (OLS) is enabled, as Database Vault requires it, If it is not installed, you must enable it.

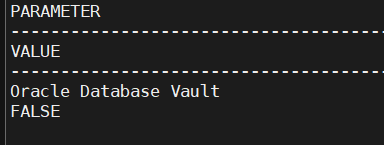
SQL>

SELECT COMP\_NAME, VERSION, STATUS FROM dba\_registry WHERE comp\_id='OLS';

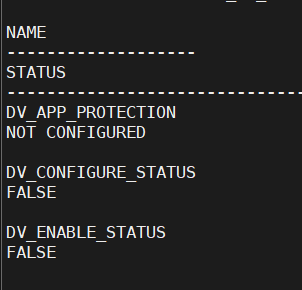


**Step 1-2: Check whether DB Vault is enabled:**

SQL> select \* from v$option where lower(PARAMETER) like '%vault%';



SQL> select \* from dba\_dv\_status;



**Step 2-1: Users to manage database vault:**

SQL>

create user c##dvowner identified by <password>;

create user c##dvactmgr identified by <password>;

to set configuration for users

BEGIN

 DVSYS.CONFIGURE\_DV (

 dvowner\_uname => 'c##dvowner',

 dvacctmgr\_uname => 'c##dvactmgr');

END;

/

**Step 2-2: to enable database vault:**

SQL> conn c##dvowner/dvowner

EXEC DBMS\_MACADM.ENABLE\_DV;

execute dvsys.dbms\_macadm.enable\_app\_protection(NULL);

EXEC DBMS\_MACADM.DISENABLE\_APP\_PROTECTION;

EXEC DBMS\_MACADM.ENABLE\_APP\_PROTECTION ('PDB\_NAME');

**Step 2-3: restart database to confige database vault:**

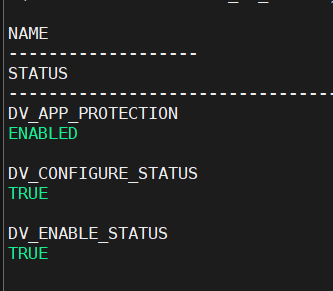
SQL> conn / as sysdba

SQL> shutdown immediate

SQL> startup

SQL> alter pluggable database all open;

SQL> select \* from dba\_dv\_status;



**Step 3-1: Define Realms, Command Rules, and Factors:**

Steps to Prevent Access Example: -

1. Create a Database Vault Realm

Create a realm specifically to protect these 3 tables.

BEGIN

   DBMS\_MACADM.CREATE\_REALM (

      realm\_name       => 'Protect\_bank\_Tables',

      description      => 'Protect table1, table2, and table3 in the bank schema',

      enabled          => DBMS\_MACUTL.G\_YES

   );

END;

/

2. Add the Specific Tables to the Realm

Add only the 3 tables to the realm

BEGIN

   -- Add table1

   DBMS\_MACADM.ADD\_OBJECT\_TO\_REALM(

      realm\_name       => 'Protect\_bank\_Tables',

      object\_schema    => 'BANK',

      object\_name      => 'TABLE1',

      object\_type      => 'TABLE'

   );

   -- Add table2

   DBMS\_MACADM.ADD\_OBJECT\_TO\_REALM(

      realm\_name       => 'Protect\_bank\_Tables',

      object\_schema    => 'BANK',

      object\_name      => 'TABLE2',

      object\_type      => 'TABLE'

   );

   -- Add table3

   DBMS\_MACADM.ADD\_OBJECT\_TO\_REALM (

      realm\_name       => 'Protect\_bank\_Tables',

      object\_schema    => 'BANK',

      object\_name      => 'TABLE3',

      object\_type      => 'TABLE'

   );

END;

/

3. Use Database Roles for Fine-Grained Privileges

Create a role for users who need access

CREATE ROLE bank\_read\_role;

GRANT SELECT ON bank. <table\_name> TO bank\_read\_role;

2.Assign this role only to specific users:

GRANT bank\_read\_role TO user\_allowed\_access;

4.audit

AUDIT SELECT, INSERT, UPDATE, DELETE ON bank. <table\_name> BY ACCESS;

**Step 3-2: Optionally Restrict SYS from Performing Administrative Actions:**

A) Create a Database Vault Realm for SYS

- Switch to secure\_admin (Database Vault Owner) and create a realm to restrict SYS from modifying system objects:

BEGIN

  DVSYS.DBMS\_MACADM.CREATE\_REALM(

    realm\_name    => 'Restrict SYS Realm',

    realm\_desc    => 'Prevents SYS from accessing sensitive data',

    enabled       => DBMS\_MACUTL.G\_YES

  );

END;

/

B) Add SYS-Related Objects to the Realm

- Protect important schemas (like SYS, SYSTEM):

- This prevents SYS from modifying any table in the SYS schema.

BEGIN

  DVSYS.DBMS\_MACADM.ADD\_REALM\_OBJECT(

    realm\_name   => 'Restrict SYS Realm',

    object\_owner => 'SYS',

    object\_name  => '%',

    object\_type  => 'TABLE'

  );

END;

/

Prevent SYS from Running Critical Commands

==================================================

- Create a Command Rule to block SYS from executing certain commands:

A) Block SYS from Creating Users Example

BEGIN

  DVSYS.DBMS\_MACADM.CREATE\_COMMAND\_RULE(

    command        => 'CREATE USER',

    rule\_name      => 'Block SYS Create User',

    object\_owner   => NULL,

    object\_name    => NULL,

    rule\_expr      => 'SYS\_CONTEXT (''USERENV'', ''SESSION\_USER'')!= ''SYS''',

    enabled        => DBMS\_MACUTL.G\_YES

  );

END;

/

B) Block SYS from Executing DDL Statements, This prevents SYS from executing ALTER SYSTEM commands.

BEGIN

  DVSYS.DBMS\_MACADM.CREATE\_COMMAND\_RULE (

    command        => 'ALTER SYSTEM',

    rule\_name      => 'Block SYS Alter System',

    object\_owner   => NULL,

    object\_name    => NULL,

    rule\_expr      => 'SYS\_CONTEXT(''USERENV'', ''SESSION\_USER'') != ''SYS''',

    enabled        => DBMS\_MACUTL.G\_YES

  );

END;

/