Example 4: Implementing VPD Grouped Policies

Overview

There may be cases where you want to use different VPD policies on the same object. In such cases VPD offers a feature named grouped policies that can be used to assign policies to different groups and to trigger them depending on certain conditions. Enabling one policy or another will be decided by a driver context according to certain parameters declared at the application level. The following recipe will demonstrate how to use this VPD feature.

In this recipe we will create a table that will contain three different department groups.

We will create a new user STOBIAS in addition to the DOCONNEL and JWHALEN users created earlier, in order to have one user for each group of departments. For each group of departments a group policy will be defined. These grouped policies will isolate the role of each group based on user membership. Each user will see his department determined by a driver context.

Workflow

1. Connect as user HR and create the DEPARTMENT_CATEGORY table as follows:

```
CREATE TABLE HR.DEPARTMENT_CATEGORY
      (
        DEPID_CAT1 NUMBER,
        DEP_CAT1   VARCHAR2(100 BYTE),
        DEPID_CAT2 NUMBER,
        DEP_CAT2   VARCHAR2(100 BYTE),
        DEPID_CAT3 NUMBER,
        DEP_CAT3   VARCHAR2(100 BYTE)
      )
      SEGMENT CREATION IMMEDIATE PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING STORAGE
      (
        INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE
0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT
CELL_FLASH_CACHE DEFAULT
      )
      TABLESPACE USERS ;
```

2. Next, insert into department_category control data. The data will be used by the driving context:

```
Insert into DEPARTMENT_CATEGORY
(DEPID_CAT1,DEP_CAT1,DEPID_CAT2,DEP_CAT2,DEPID_CAT3,DEP_CAT3) values
(10,'Administration',20,'Marketing',30,'Purchasing');
Insert into DEPARTMENT_CATEGORY
(DEPID_CAT1,DEP_CAT1,DEPID_CAT2,DEP_CAT2,DEPID_CAT3,DEP_CAT3) values (40,'Human
Resources',50,'Shipping',60,'IT');
Insert into DEPARTMENT_CATEGORY
(DEPID_CAT1,DEP_CAT1,DEPID_CAT2,DEP_CAT2,DEPID_CAT3,DEP_CAT3) values (70,'Public
Relations',80,'Sales',90,'Executive');
Insert into DEPARTMENT_CATEGORY
(DEPID_CAT1,DEP_CAT1,DEPID_CAT2,DEP_CAT2,DEPID_CAT3,DEP_CAT3) values
(100,'Finance',110,'Accounting',120,'Treasury');
Insert into DEPARTMENT_CATEGORY
(DEPID_CAT1,DEP_CAT1,DEPID_CAT2,DEP_CAT2,DEPID_CAT3,DEP_CAT3) values
(130,'Corporate Tax',140,'Control And Credit',150,'Sha
reholder Services');
Insert into DEPARTMENT_CATEGORY
(DEPID_CAT1,DEP_CAT1,DEPID_CAT2,DEP_CAT2,DEPID_CAT3,DEP_CAT3) values
(160,'Benefits',170,'Manufacturing',180,'Construction'
);
Insert into DEPARTMENT_CATEGORY
(DEPID_CAT1,DEP_CAT1,DEPID_CAT2,DEP_CAT2,DEPID_CAT3,DEP_CAT3) values
(190,'Contracting',200,'Operations',210,'IT Support');
Insert into DEPARTMENT_CATEGORY
(DEPID_CAT1,DEP_CAT1,DEPID_CAT2,DEP_CAT2,DEPID_CAT3,DEP_CAT3) values
(220,'NOC',230,'IT Helpdesk',240,'Government Sales');
Insert into DEPARTMENT_CATEGORY
(DEPID_CAT1,DEP_CAT1,DEPID_CAT2,DEP_CAT2,DEPID_CAT3,DEP_CAT3) values (250,'Retail
Sales',260,'Recruiting',270,'Payroll');
commit;
```

3. Connect as user system and create a user STOBIAS, grant create session privilege to it:

```
conn system
Enter password:
create user STOBIAS identified by STOBIAS;
grant create session to STOBIAS;
```

4. Next, grant select on DEPARTMENT_CATEGORY to DOCONNEL, JWHALEN, and STOBIAS as follows:

```
grant select on hr.department_category to stobias,doconnel,jwhalen;
```

5. Connect as system and create the driving context dep_cat_context as follows:

```
conn system
Enter password:
CREATE OR REPLACE CONTEXT dep_cat_context USING department_cat_pkg;
```

6. From now on we will create one policy for each category. Create policy_group category_dept_one as follows:

```
BEGIN
 DBMS_RLS.CREATE_POLICY_GROUP( object_schema => 'HR', object_name =>
'department_category', policy_group => 'category_dept_one');
END;
```

7. Create policy group category_dept_two as follows:\

```
BEGIN
DBMS_RLS.CREATE_POLICY_GROUP( object_schema => 'HR', object_name =>
'department_category', policy_group => 'category_dept_two');
END;
```

8. Create policy group category_dept_three as follows:

```
BEGIN
 DBMS_RLS.CREATE_POLICY_GROUP( object_schema => 'HR', object_name =>
'department_category', policy_group => 'category_dept_three');
END;
```

9. Next, we will create three policy functions that will be assigned to each grouped policy. Create the policy function for category one named vpd_function_category_one as follows:

```
CREATE OR REPLACE
    FUNCTION VPD_FUNCTION_CATEGORY_ONE
    (
     V_SCHEMA IN VARCHAR2,
        V_TABLE  IN VARCHAR2
 )
      RETURN VARCHAR2
    AS
      PREDICATE VARCHAR2(8) DEFAULT NULL;
    BEGIN
      IF (SYS_CONTEXT('USERENV','SESSION_USER')) = 'JWHALEN' THEN
        predicate                               := '1=2';
      ELSE
        NULL;
      END IF;
      RETURN predicate;
 END;
```

10. Create the policy function vpd_function_category_two as follows:

```
CREATE OR REPLACE FUNCTION vpd_function_category_two
    (v_schema in varchar2, v_table in varchar2) return varchar2 as
     predicate varchar2(8) default NULL;
      BEGIN
      IF (SYS_CONTEXT('USERENV','SESSION_USER')) = 'DOCONNEL'
      THEN predicate := '1=2';
      ELSE NULL;
      END IF;
      RETURN predicate;
    END;
```

11. Create the policy function vpd_function_category_three as follows:

```
CREATE OR REPLACE
    FUNCTION vpd_function_category_three
      (
        v_schema IN VARCHAR2,
        v_table  IN VARCHAR2)
      RETURN VARCHAR2
    AS
      predicate VARCHAR2(8) DEFAULT NULL;
    BEGIN
      IF (SYS_CONTEXT('USERENV','SESSION_USER')) = 'STOBIAS' THEN
        predicate                               := '1=2';
      ELSE
        NULL;
      END IF;
      RETURN predicate;
    END;
```

12. Next, we will create the grouped policies for each department category. Create the grouped policy named vpd_function_category_one_plc for category one as follows:

```
BEGIN
 DBMS_RLS.ADD_GROUPED_POLICY
 (
  object_schema => 'HR',
  object_name => 'department_category',
  policy_group => 'category_dept_one',
  policy_name => 'vpd_function_category_one_plc',
  policy_function => 'vpd_function_category_one',
  statement_types => 'select',
  policy_type => DBMS_RLS.CONTEXT_SENSITIVE,
  sec_relevant_cols => 'depid_cat2,dep_cat2,depid_cat3,dep_cat3',
  sec_relevant_cols_opt => DBMS_RLS.ALL_ROWS);
END;
```

13. Next create a grouped policy named vpd_function_category_two_plc for category two as follows:

```
BEGIN
 DBMS_RLS.ADD_GROUPED_POLICY
 (
  object_schema => 'HR',
  object_name => 'department_category',
  policy_group => 'category_dept_two',
  policy_name => 'vpd_function_category_two_plc',
  policy_function => 'vpd_function_category_two',
  statement_types => 'select',
  policy_type => DBMS_RLS.CONTEXT_SENSITIVE,
  sec_relevant_cols=> 'depid_cat1,dep_cat1,depid_cat3,dep_cat3',
  sec_relevant_cols_opt => DBMS_RLS.ALL_ROWS
 );
END;
```

14. And finally create policy named vpd_function_cat_three_plc for the last department category as follows:

```
BEGIN
 DBMS_RLS.ADD_GROUPED_POLICY
 (
  object_schema => 'HR',
  object_name => 'department_category',
  policy_group => 'category_dept_three',
  policy_name => 'vpd_function_cat_three_plc',
  policy_function => 'vpd_function_category_three',
  statement_types => 'select',
  policy_type => DBMS_RLS.CONTEXT_SENSITIVE,
  sec_relevant_cols => 'depid_cat1,dep_cat1,depid_cat2,dep_cat2',
  sec_relevant_cols_opt => DBMS_RLS.ALL_ROWS
 );
END;
```

15. Next, create package and package body department_cat_pkg associated with context dep_cat_context:

```
CREATE OR REPLACE
  PACKAGE department_cat_pkg
  IS
  PROCEDURE set_dep_cat_context
    (
      plc_grp VARCHAR2 DEFAULT NULL);
  END;

CREATE OR REPLACE
  PACKAGE BODY department_cat_pkg
  AS
  PROCEDURE set_dep_cat_context
    (
      plc_grp VARCHAR2 DEFAULT NULL)
  IS
  BEGIN
    CASE (SYS_CONTEXT('USERENV', 'SESSION_USER'))
    WHEN 'JWHALEN' THEN
      DBMS_SESSION.SET_CONTEXT('dep_cat_context','plc_grp','CATEGORY_DEPT_ONE');
    WHEN 'DOCONNEL' THEN
      DBMS_SESSION.SET_CONTEXT('dep_cat_context','plc_grp','CATEGORY_DEPT_TWO');
    WHEN 'STOBIAS' THEN

DBMS_SESSION.SET_CONTEXT('dep_cat_context','plc_grp','CATEGORY_DEPT_THREE');
    ELSE
      NULL;
    END CASE;
  EXCEPTION
  WHEN NO_DATA_FOUND THEN
    NULL;
  END set_dep_cat_context;
END;
```

16. Next, assign dep_cat_context context to department_category as driving context:

```
BEGIN
 DBMS_RLS.ADD_POLICY_CONTEXT
 (
  object_schema =>'HR',
  object_name =>'department_category',
  namespace =>'dep_cat_context',
attribute =>'plc_grp'
);
END;
```

17. Next, create a new logon trigger to set the driving context after connect as follows:

```
CREATE OR REPLACE TRIGGER set_dep_cat_context_trg AFTER LOGON ON DATABASE
BEGIN
security_adm.department_cat_pkg.set_dep_cat_context;
END;
```

18. Next, connect as DOCONNEL, check the plc_grp value from the driving context, and select from department_category to check if the

grouped policy if enforced:

```
conn DOCONNEL
Enter password:
Connected.

select sys_context('dep_cat_context','plc_grp') as DRIVING_CONTEXT from dual;

select depid_cat1,dep_cat1,depid_cat2,dep_cat2,depid_cat3,dep_cat3 from
hr.department_category;
```

19. Connect as STOBIAS user, check the plc_grp value from the driving context, and select from department_category to check if the grouped policy if enforced:

```
conn STOBIAS/STOBIAS
Connected.

select sys_context('dep_cat_context','plc_grp') from dual;

select depid_cat1,dep_cat1,depid_cat2,dep_cat2,depid_cat3,dep_cat3 from
hr.department_category;
```

20. And finally connect as user JWHALEN, check the plc_grp value from the driving context, and select from department_category to check if the grouped policy if enforced:

```
conn JWHALEN/JWHALEN

Connected.

select sys_context('dep_cat_context','plc_grp') from dual;

select depid_cat1,dep_cat1,depid_cat2,dep_cat2,depid_cat3,dep_cat3 from
hr.department_category;
```

How it works

In grouped policies, the active policy is decided by using the driving context. In our example, the driving context is "dep_cat_context" defined with the ADD_POLICY_CONTEXT procedure from the DBMS_RLS package. Its attribute is modified depending on which user connects.

There's more

More information about grouped policies can be found in the ALL_POLICIES_GROUP, DBA_POLICIES_GROUPS, and DBA_POLICY_CONTEXTS dictionary views.