Analyst Program

SANS

Sponsored by Oracle

SANS Institute Product Review: Oracle Database Vault

August 2011

A SANS Whitepaper

Written by: Tanya Baccam

Introduction

In today's world, despite layered protections, intruders, insiders and financially-motivated attackers will try to exploit privileged accounts to access sensitive application data. When it comes to the database, this includes abuse of privileged user accounts that have the powerful Database Administrator (DBA) role. Because of the extensive access given to such accounts, damage done by attackers using privileged accounts often is the hardest to detect and the most extensive. This is why controlling the use of administrative access is number eight on the list of SANS 20 Critical Security Controls V3.0, updated in August, 2011.[1]

One way to minimize the risk of privileged user access to sensitive application data in the database is to establish protection zones that block powerful DBA privileges from being misused by insiders, external hackers or malware. This is especially important given initiatives (such as outsourcing), and the use of modern IT infrastructures (such as cloud computing) that provide efficiencies by automatically provisioning and consolidating databases. In these environments, privileged accounts have even greater access to sensitive and regulated application data.

The ability to establish critical protection zones within the database, whether they're operating in or out of the cloud, is the purpose of Oracle Database Vault. Oracle Database Vault enforces powerful operational controls inside the Oracle database by introducing new technologies including realms, command rules, and factors. Combined, these new technologies give database administrators the ability to zero out the collateral damage resulting from attacks that target privileged accounts. Oracle Database Vault provides the ability to enforce controls over who, when, where and how various operations can be performed inside the database. This level of enforcement eliminates configuration drift and blocks unauthorized changes to the database, such as adding new database accounts and copying application tables.

This paper is a review of Oracle Database Vault with Oracle Database Enterprise Edition 11g Release 2. Overall, Oracle Database Vault performed well, while making it easy to add, change and modify rules and groups. It was also easy to gain visibility into user activity through a variety of audit and compliance reports available through the Oracle Database Vault application.

[1] https://www.sans.org/press/20-critical-controls.php

Overview and Setup

Oracle Database Vault protects against abuse of privileged accounts to access sensitive data. It also helps meet regulatory compliance requirements by enforcing strong operational controls around administrator and other privileged user access. To this end, Oracle Database Vault contains several capabilities, including access control enforcement, reporting on violations, and Oracle Database security configuration management. This review covers the functional aspects of Oracle Database Vault to manage access for database applications and protect the critical data contained in them.

Unlike many security controls that are loosely coupled with the database, Oracle Database Vault provides strong controls by enforcing security inside the kernel of the Oracle database. As a result of this tight integration, the performance overhead is typically less than two percent, according to Oracle user FAQs.[2] Integrating security into the database also tackles the challenge of control bypass by attackers and authorized users.

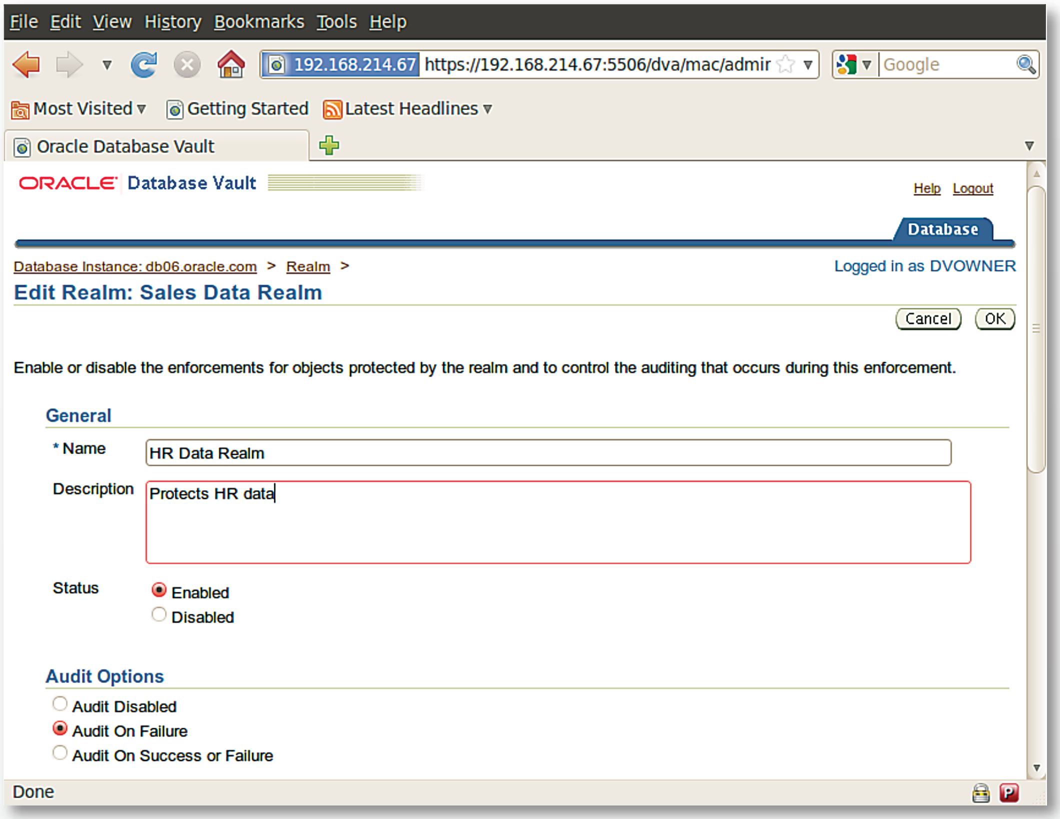
Oracle Database Vault technology works to protect applications using separate realms. Realms provide protection for each application separately, while enabling the visibility application administrators need to do their jobs. So, administrators can access only their applications (rather than all the applications in the database); and even with their approved access, they can't view, copy or move the sensitive data within those applications. This level of granularity is especially useful when multiple applications are consolidated into a single database or in cloud environments where multiple instances of the database may move around. Using separate realms also adds a layer of protection for users themselves. They may not want access to sensitive data to protect themselves from liability, avoid mistakes and improve work efficiency.

Realms can be set to protect a set of database schemas, objects and/or roles within the database, based on user preference. For example, if a database stores human resource (HR) department data that needs to be protected, Oracle Database Vault could place all of the data in a realm, and then control access to this data.

[2] www.oracle.com/technetwork/database/security/dbv-faq-083210.html

Creating and Testing Realms

Throughout this review, we used the Database Vault Administrator (DVA) console to administer Oracle Database Vault. Using DVA, we created an HR Data Realm to protect human resources data. Setting up this realm with DVA involved clicking *Realms*, clicking *Create*, and then naming and defining the realm *HR Data Realm* (See Figure 1).



Database Instance: db06.oracle.com > Realm >

Logged in as DVOWNER

Edit Realm: Sales Data Realm

Enable or disable the enforcements for objects protected by the realm and to control the auditing that occurs during this enforcement.

General

* Name: HR Data Realm

Description: Protects HR data

Status: ● Enabled ○ Disabled

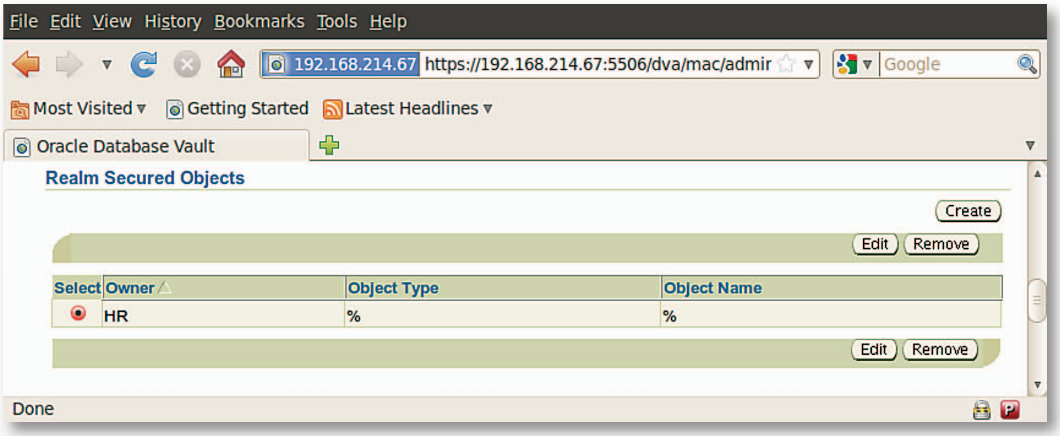
Audit Options

○ Audit Disabled
● Audit On Failure
○ Audit On Success or Failure

Figure 1. Setting Up a Realm

In setting up this realm, the objective was to ensure that highly privileged users did not have access to the HR data but could still administer the database containing the HR Data Realm. Once the realm was named and enabled, we selected *Audit on failure* to send a notification when rules are violated.

The next step was to identify what objects need to be protected. These are referred to as *Realm Secured Objects*. The object owner, object type and object name need to be specified for each object included in the realm. In this case, we leveraged the wildcard (%) option to identify all objects owned by the HR user (See Figure 2).

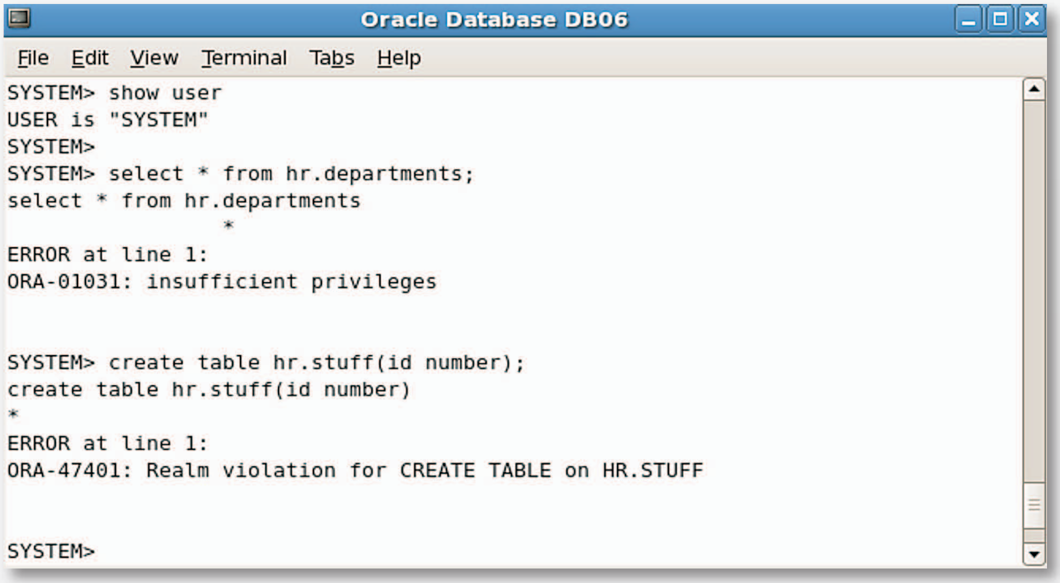


Realm Secured Objects

Select	Owner	Object Type	Object Name
●	HR	%	%

Figure 2. Securing Objects in the Realm

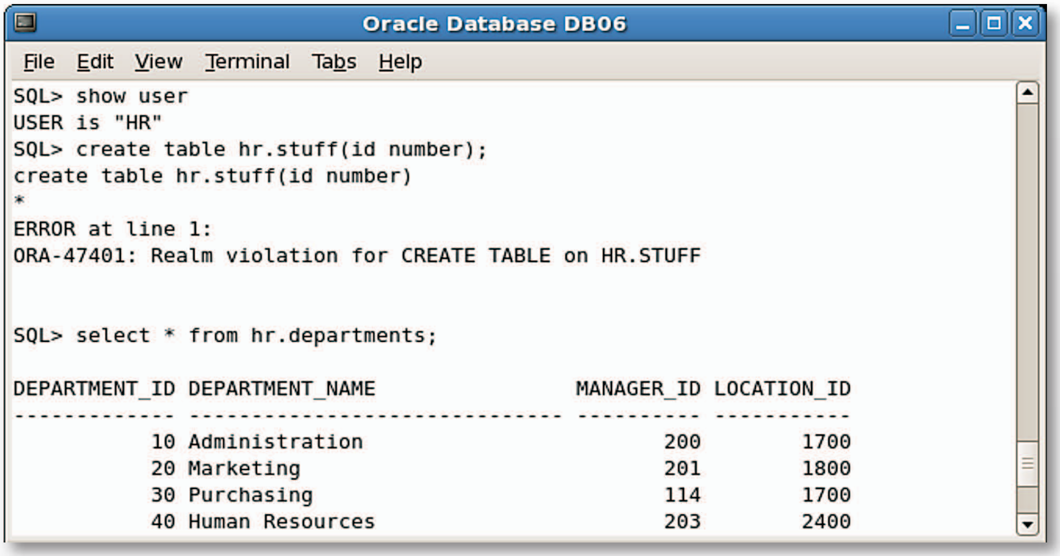
With the HR Data Realm in place, the next step was to determine how to control the access of a privileged user, such as SYSTEM, when the user accesses objects in the realm. In this case, an attempt to query an HR object (specifically the departments table) resulted in a message that SYSTEM had insufficient privileges. Similarly, SYSTEM could not create objects in the HR Data Realm, and Oracle Database Vault returned a violation notification. Figure 3 illustrates these attempts.



```
SYSTEM> show user
USER is "SYSTEM"
SYSTEM>
SYSTEM> select * from hr.departments;
select * from hr.departments
                 *
ERROR at line 1:
ORA-01031: insufficient privileges


SYSTEM> create table hr.stuff(id number);
create table hr.stuff(id number)
*
ERROR at line 1:
ORA-47401: Realm violation for CREATE TABLE on HR.STUFF


SYSTEM>
```

Figure 3. Violation Messages Returned after Unsuccessful Access and Modification Attempts in the Protected Realm

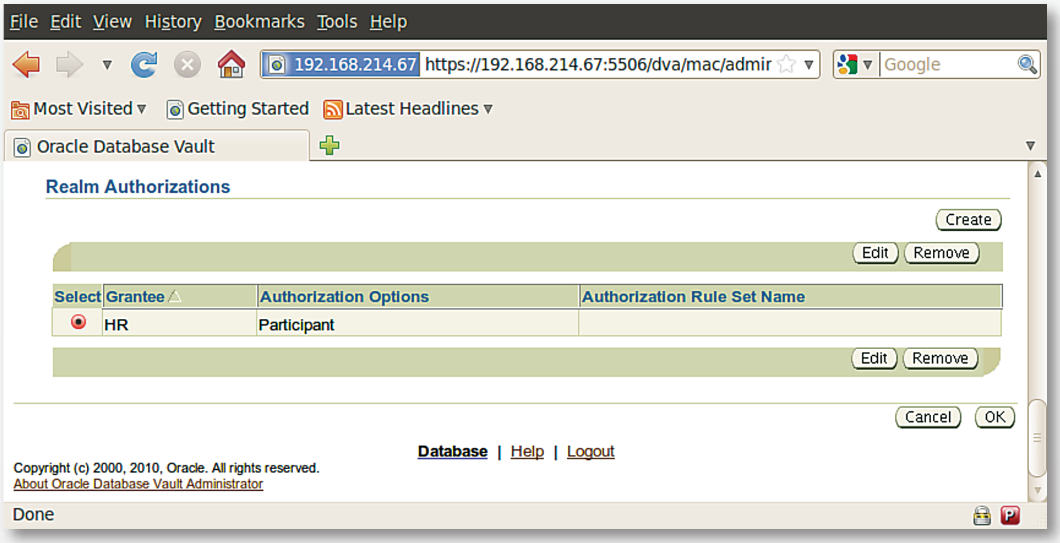
We also ran queries as the HR user to test what the owner of the objects could do when a Secured Realm existed for the objects they owned. No specific privileges had been granted within Oracle Database Vault to HR at this point. By default, the data could be queried by the owner of the object, but only Data Manipulation Language (DML) statements could be issued, not Data Definition Language (DDL), as illustrated in Figure 4.



```
SQL> show user
USER is "HR"
SQL> create table hr.stuff(id number);
create table hr.stuff(id number)
*
ERROR at line 1:
ORA-47401: Realm violation for CREATE TABLE on HR.STUFF


SQL> select * from hr.departments;

DEPARTMENT_ID DEPARTMENT_NAME                MANAGER_ID LOCATION_ID
------------- ------------------------------ ---------- -----------
           10 Administration                        200        1700
           20 Marketing                             201        1800
           30 Purchasing                            114        1700
           40 Human Resources                       203        2400
```

Figure 4. Default Settings So Realm Owners Can Access Their Objects

Some employees will need authorization to modify the database as business needs dictate. After running the test above, the user, HR, was added to HR Data Realm using realm authorizations, as shown in Figure 5.



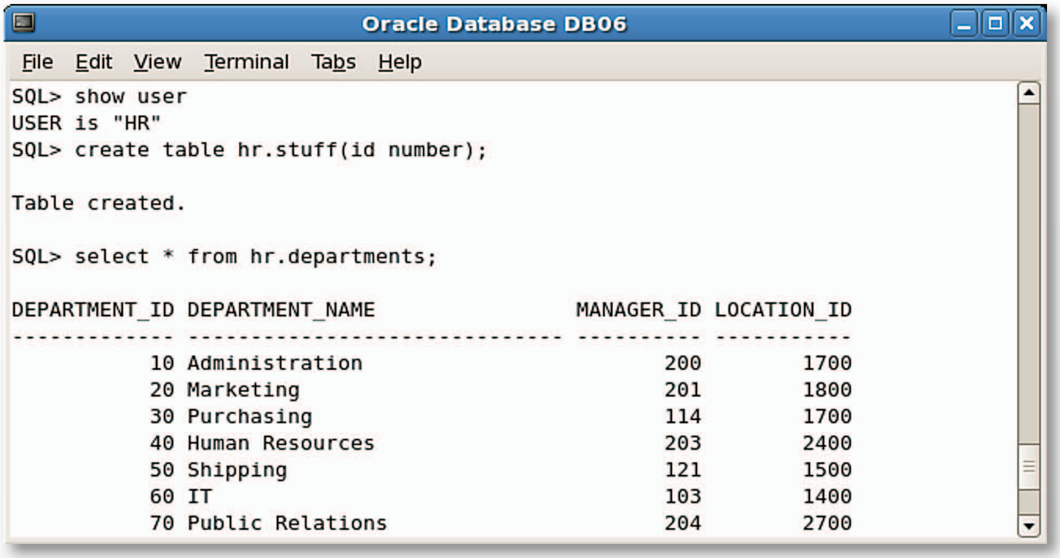
Realm Authorizations

Select	Grantee	Authorization Options	Authorization Rule Set Name
●	HR	Participant	

Copyright (c) 2000, 2010, Oracle. All rights reserved.
About Oracle Database Vault Administrator

Figure 5. Granting Realm Authorizations

Once authorized, this user could issue any statement chosen, including DDL and DML statements, as shown in Figure 6.



```
SQL> show user
USER is "HR"
SQL> create table hr.stuff(id number);

Table created.

SQL> select * from hr.departments;

DEPARTMENT_ID DEPARTMENT_NAME                MANAGER_ID LOCATION_ID
------------- ------------------------------ ---------- -----------
           10 Administration                        200        1700
           20 Marketing                             201        1800
           30 Purchasing                            114        1700
           40 Human Resources                       203        2400
           50 Shipping                              121        1500
           60 IT                                    103        1400
           70 Public Relations                      204        2700
```

Figure 6. Authorization to Create Tables

Rules, Roles and Factors for Granular Control

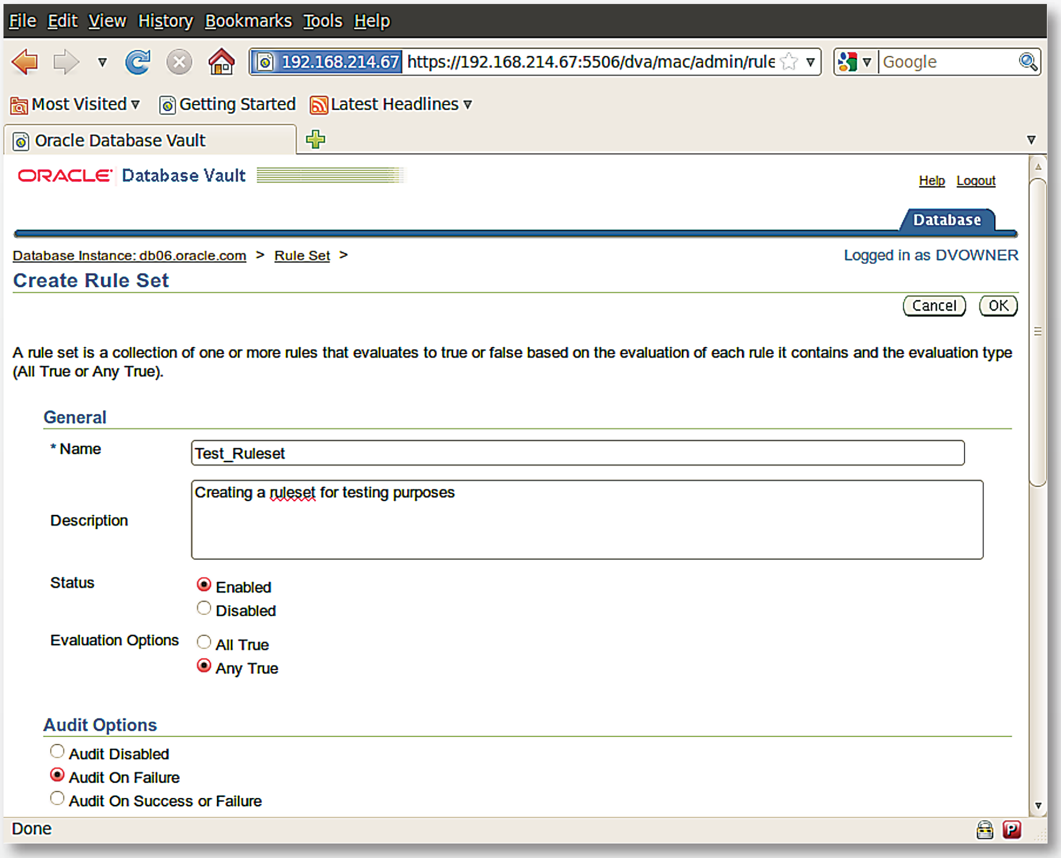
In addition to the options discussed previously, there are a number of additional options in how a realm can be used to protect data. Again, a realm is a collection of the objects being protected whereas the rules and factors are essentially the definition of how the data is being protected.

Overall, the process of setting up realms and then creating rule sets, rules, command rules and factors was relatively easy. There is a bit of complexity and a learning curve involved when it comes to understanding what each of these capabilities provides; but, once you have a good understanding of the possibilities and how you can take advantage of them, the implementation process is relatively simple.

Rule Sets

First, we'll look at rule sets because they can be used by factors, realms, command rules and secure application roles. Essentially, a rule set is a container for other rules and must exist so rules can be added. Creating a rule set involves going to the Oracle Database Vault Administration screen and clicking *Rule Sets*. For each rule set, we can provide a name, a description, the status (enabled or disabled) and the evaluation options (All True or Any True).

In this case, we named the container Test_Ruleset and set the evaluation option to *Any True*. With this option, once rules are added, if any of the rules are evaluated to *True*, the rule set as a whole would return *True*. Additionally, we set the Audit Options to *Audit on Failure*, as shown in Figure 7.


File Edit View History Bookmarks Tools Help
192.168.214.67 https://192.168.214.67:5506/dva/mac/admin/rule
Google
Most Visited Getting Started Latest Headlines
Oracle Database Vault
ORACLE Database Vault
Help Logout
Database
Done

Database Instance: db06.oracle.com > Rule Set >

Logged in as DVOWNER

Create Rule Set

Cancel OK

A rule set is a collection of one or more rules that evaluates to true or false based on the evaluation of each rule it contains and the evaluation type (All True or Any True).

General

* Name: Test_Ruleset

Description: Creating a ruleset for testing purposes

Status: ● Enabled ○ Disabled

Evaluation Options: ○ All True ● Any True

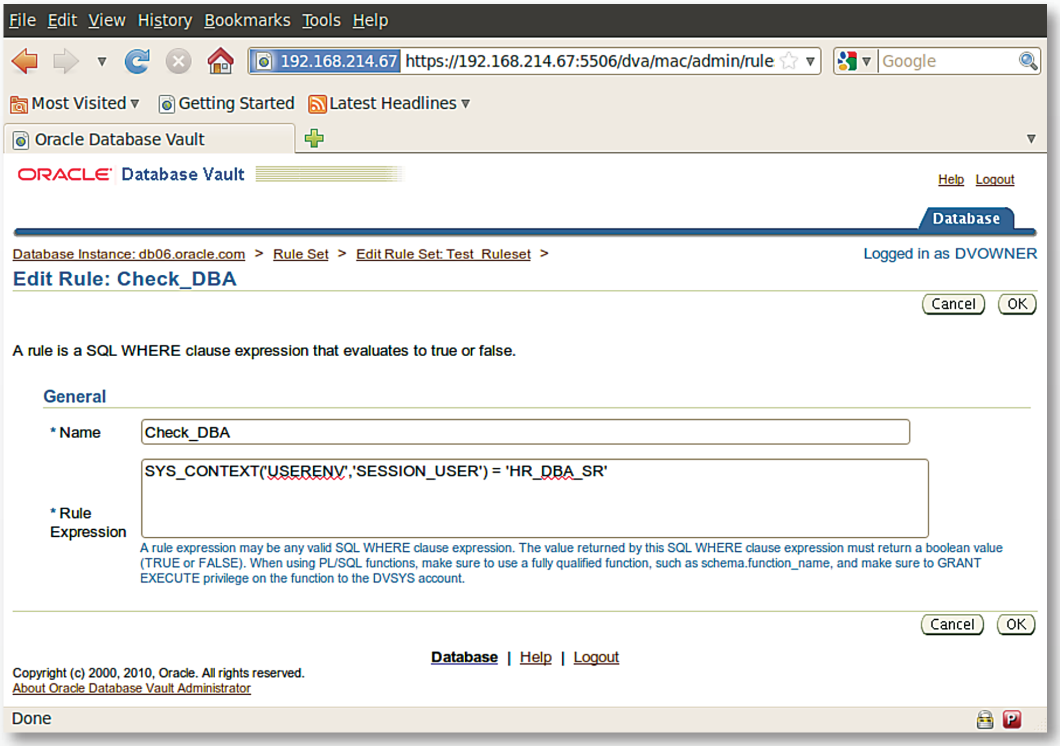
Audit Options

○ Audit Disabled
● Audit On Failure
○ Audit On Success or Failure

Figure 7. Creating Rule Sets

Now that we have our rule set, we can add rules. Rules are easy to set up within Oracle Database Vault, as long as you know the basic database syntax required. A rule is simply in the form of a WHERE clause, so any valid WHERE clause could be used to create rules.

As shown in Figure 8, we added a rule called *Check_DBA*, which checks whether the user is *HR_DBA_SR*.



Database Instance: db06.oracle.com > Rule Set > Edit Rule Set: Test_Ruleset >

Logged in as DVOWNER

Edit Rule: Check_DBA

A rule is a SQL WHERE clause expression that evaluates to true or false.

General

* Name: Check_DBA

* Rule Expression: SYS_CONTEXT('USERENV','SESSION_USER') = 'HR_DBA_SR'

A rule expression may be any valid SQL WHERE clause expression. The value returned by this SQL WHERE clause expression must return a boolean value (TRUE or FALSE). When using PL/SQL functions, make sure to use a fully qualified function, such as schema.function_name, and make sure to GRANT EXECUTE privilege on the function to the DVSYS account.

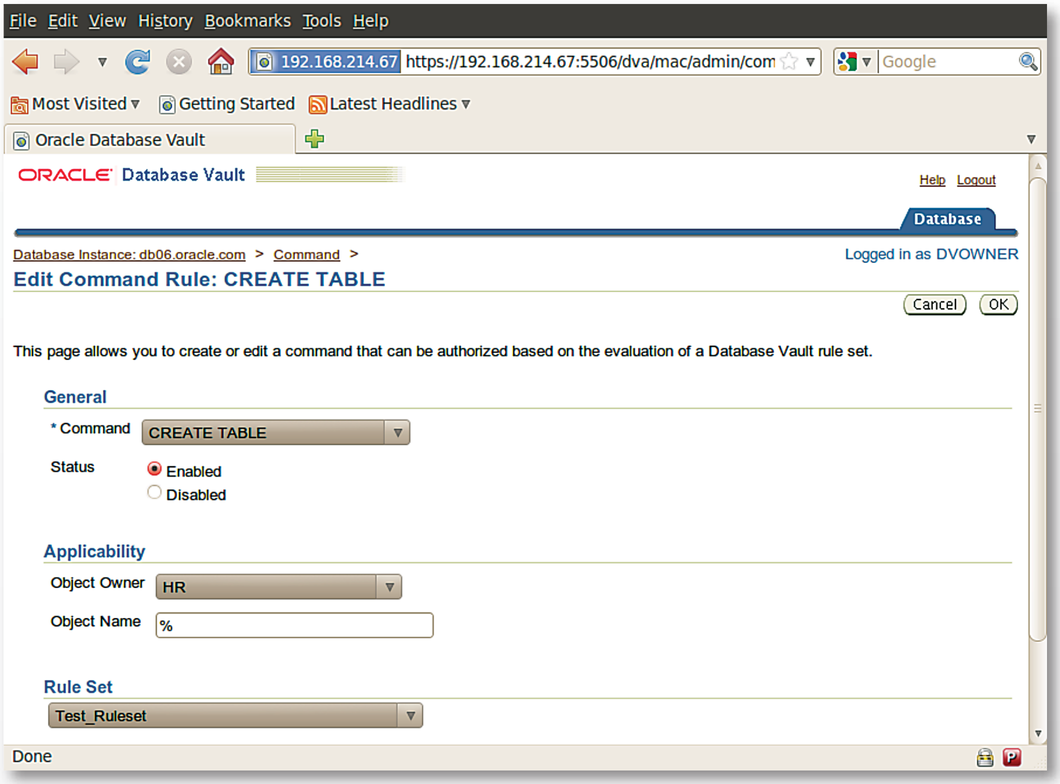
Copyright (c) 2000, 2010, Oracle. All rights reserved.
About Oracle Database Vault Administrator

Figure 8. Adding Rules

Command Rules

Once rules are set up and added, they can be integrated with command rules. Command rules give the capability to set up more minute controls around how a database is secured and can be used to attach security policies, provide additional specifics on how the realm or the database objects can be accessed, and control the capability to issue DDL commands and complete specific database operations.

For testing, we set up a command rule for CREATE TABLE that blocks users from creating new tables within HR. This command rule was associated with the Test_Ruleset rule set, as illustrated in Figure 9.


File Edit View History Bookmarks Tools Help
192.168.214.67 https://192.168.214.67:5506/dva/mac/admin/com
Google
Most Visited
Getting Started
Latest Headlines
Oracle Database Vault
ORACLE Database Vault
Help Logout
Database
Database Instance: db06.oracle.com > Command >
Logged in as DVOWNER
Cancel
OK
Done

Edit Command Rule: CREATE TABLE

This page allows you to create or edit a command that can be authorized based on the evaluation of a Database Vault rule set.

General

* Command: CREATE TABLE

Status: ● Enabled ○ Disabled

Applicability

Object Owner: HR

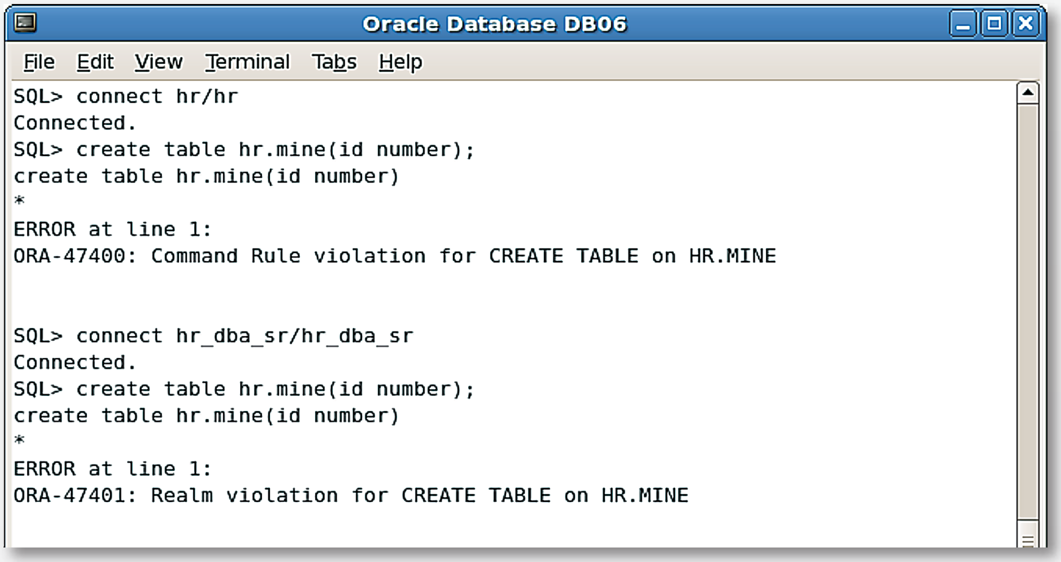
Object Name: %

Rule Set

Test_Ruleset

Figure 9. Setting Up a Command Rule to Block Users from Creating Tables in HR

Once the Test_Ruleset was associated with the CREATE TABLE command rule for the HR objects, it was time to test queries against the database. The Test_Ruleset would also now be used to validate if a user can create a table (See Figure 10).

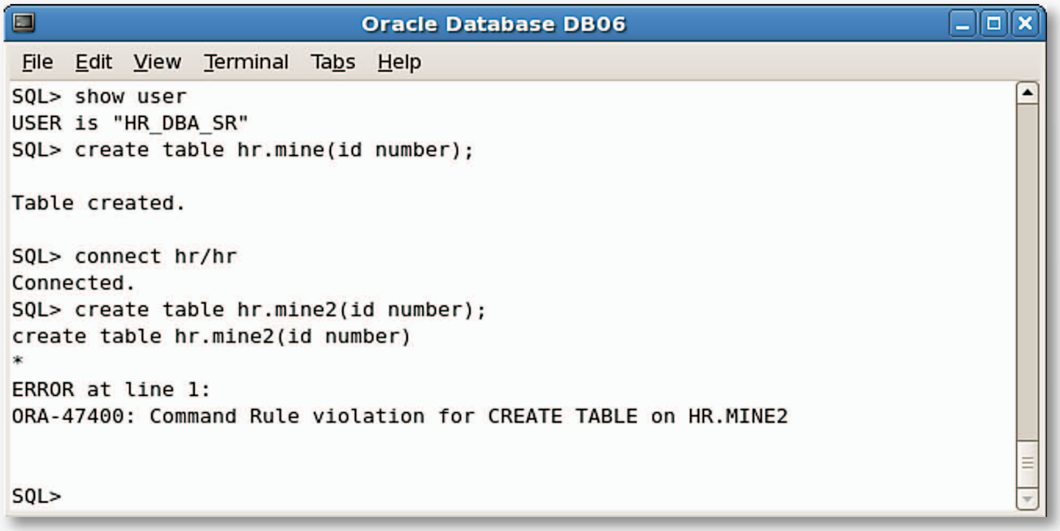


```
SQL> connect hr/hr
Connected.
SQL> create table hr.mine(id number);
create table hr.mine(id number)
*
ERROR at line 1:
ORA-47400: Command Rule violation for CREATE TABLE on HR.MINE


SQL> connect hr_dba_sr/hr_dba_sr
Connected.
SQL> create table hr.mine(id number);
create table hr.mine(id number)
*
ERROR at line 1:
ORA-47401: Realm violation for CREATE TABLE on HR.MINE
```

Figure 10. User's Attempt to Create Tables Blocked

Based on the results above, neither the HR user nor the privileged HR_DBA_SR user could create a table; however, their authorization issues are different. The HR user cannot create the table because of the command rule that was just created. The privileged user, HR_DBA_SR, passed the command rule test, but could not create the table because of a realm violation. This realm violation was due to the initial setup of the HR realm, in which it was specified that all HR objects are being protected by the realm; and HR_DBA_SR has not been authorized in that setup.

The next step was to change the rule and allow the HR_DBA_SR user to create a new table. In order to allow the HR_DBA_SR user to create tables, the DBA had to be added to realm authorizations. This enables the system to distinguish among separate privileged users. When the tests were run and attempts made to create tables, the privileged user, 'HR_DBA_SR,' could issue a **CREATE TABLE** statement, as shown in Figure 11.



```
SQL> show user
USER is "HR_DBA_SR"
SQL> create table hr.mine(id number);

Table created.

SQL> connect hr/hr
Connected.
SQL> create table hr.mine2(id number);
create table hr.mine2(id number)
*
ERROR at line 1:
ORA-47400: Command Rule violation for CREATE TABLE on HR.MINE2


SQL>
```

Figure 11. DBA Allowed To Create Table; User Denied

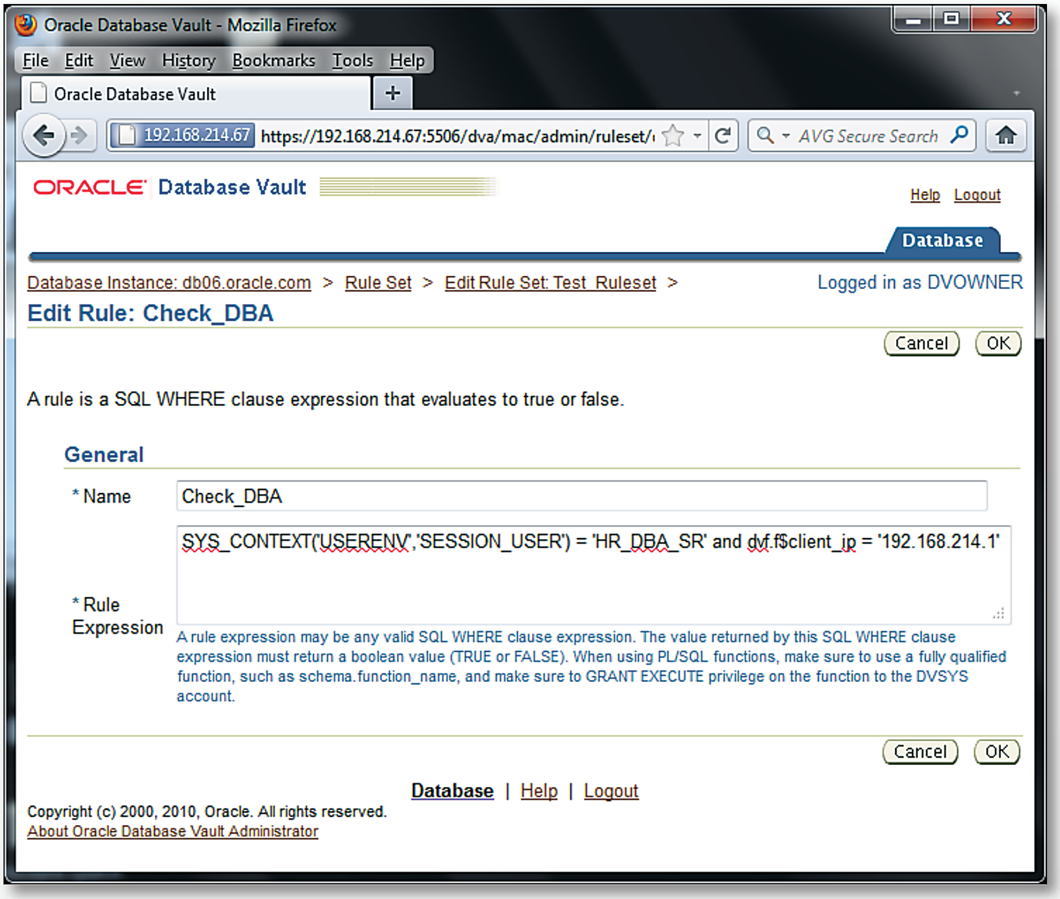
Because of the command rule, the HR user still could not create a table, which was as intended. Therefore, the user has to be authorized within the realm and able to pass the command rules that have been set up in order to achieve that permission level. Because the command rules are checking for the HR_DBA_SR user, HR cannot pass the command rule test within that protected realm.

Factors

Finally, we tested factors, which are used to set up rules that can evaluate information such as the user's IP address, the source application, the database domain and host name, the network protocol, the machine and the proxy to name a few. Factors give rules more flexibility and make it easier to check the characteristics of the current user's session. You can query these factors by issuing commands such as **SELECT DVF.F$<<factor_name>> FROM DUAL**, where **<<factor_name>>** is the name of the factor to be leveraged.

Oracle Database Vault has a number of default factors (such as those mentioned above) that you can leverage.[3] Additionally, you can create custom factors with any valid PL/SQL.

For this review, we added a factor to look at the Check_DBA rule (part of the Test_Ruleset). By adding **dvf.f$client_ip** and specifying the IP address, we modified the Check_DBA rule to allow access only to the HR_DBA_SR user accessing the database from the IP address 192.168.214.1, as illustrated in Figure 12.


Oracle Database Vault - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Oracle Database Vault
+
192.168.214.67 https://192.168.214.67:5506/dva/mac/admin/ruleset/
AVG Secure Search
ORACLE Database Vault
Help Logout
Database

Database Instance: db06.oracle.com > Rule Set > Edit Rule Set: Test_Ruleset >

Logged in as DVOWNER

Edit Rule: Check_DBA

Cancel OK

A rule is a SQL WHERE clause expression that evaluates to true or false.

General

* Name: Check_DBA

* Rule Expression: SYS_CONTEXT('USERENV','SESSION_USER') = 'HR_DBA_SR' and dvf.f$client_ip = '192.168.214.1'

A rule expression may be any valid SQL WHERE clause expression. The value returned by this SQL WHERE clause expression must return a boolean value (TRUE or FALSE). When using PL/SQL functions, make sure to use a fully qualified function, such as schema.function_name, and make sure to GRANT EXECUTE privilege on the function to the DVSYS account.

Cancel OK

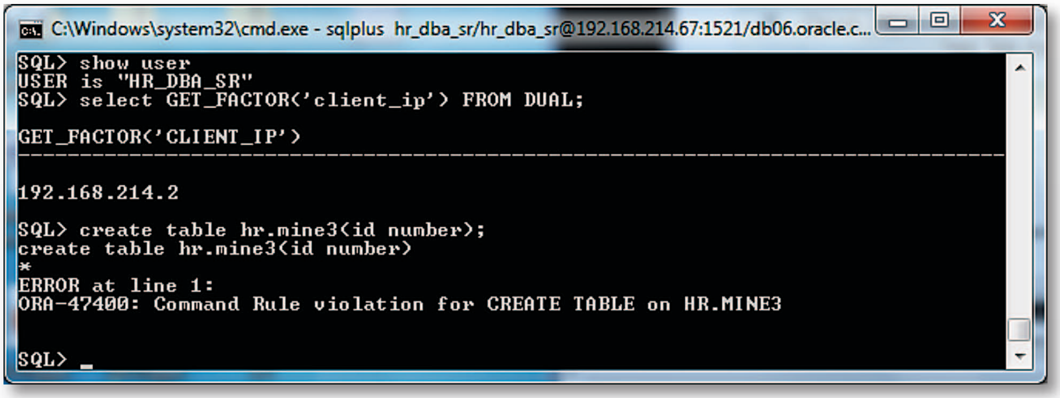
Database | Help | Logout

Copyright (c) 2000, 2010, Oracle. All rights reserved.
About Oracle Database Vault Administrator

Figure 12. Rule to Limit Access to Specific IP Address

[3] See http://download.oracle.com/docs/cd/B28359_01/server.111/b31222/cfgfact.htm#BABIEIJB

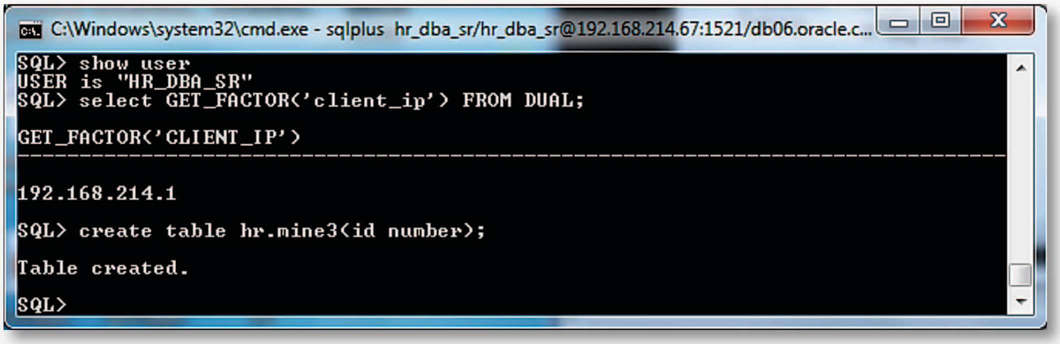
To test this, we first attempted to query the table coming from a different IP address than the one specified in the rule. Oracle Database Vault blocked the action and the HR_DBA_SR could not create a table due to the initiating IP address (See Figure 13).


C:\Windows\system32\cmd.exe - sqlplus hr_dba_sr/hr_dba_sr@192.168.214.67:1521/db06.oracle.c...

```
SQL> show user
USER is "HR_DBA_SR"
SQL> select GET_FACTOR('client_ip') FROM DUAL;

GET_FACTOR('CLIENT_IP')
--------------------------------------------------------------------------------

192.168.214.2

SQL> create table hr.mine3(id number);
create table hr.mine3(id number)
*
ERROR at line 1:
ORA-47400: Command Rule violation for CREATE TABLE on HR.MINE3


SQL> _
```

Figure 13. Limiting Access by IP Address

However, as shown in Figure 14, when initiating the request from the IP address 192.168.214.1, HR_DBA_SR could create a table.


C:\Windows\system32\cmd.exe - sqlplus hr_dba_sr/hr_dba_sr@192.168.214.67:1521/db06.oracle.c...

```
SQL> show user
USER is "HR_DBA_SR"
SQL> select GET_FACTOR('client_ip') FROM DUAL;

GET_FACTOR('CLIENT_IP')
--------------------------------------------------------------------------------

192.168.214.1

SQL> create table hr.mine3(id number);

Table created.

SQL>
```

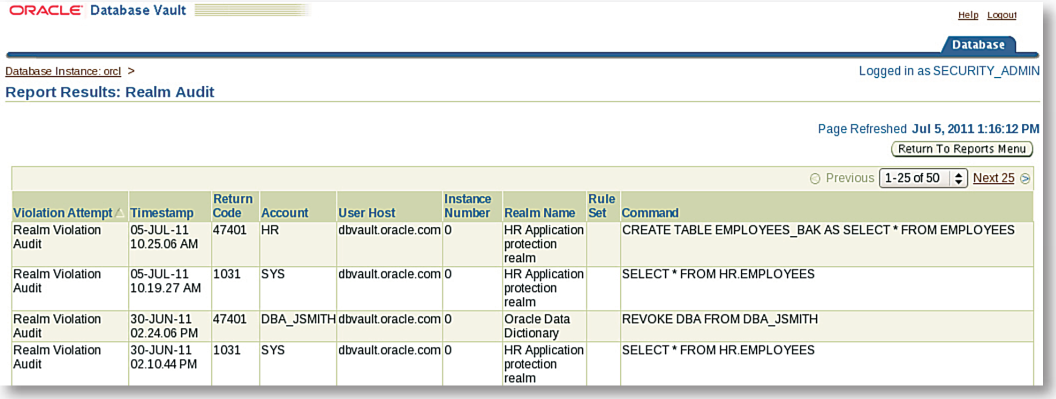
Figure 14. Allowing Access by IP Address

By using the client_ip factor we could more precisely create rules that controlled what activity could occur, and where that activity could be initiated from, as needs dictate.

Compliance Reports

The other component to Oracle Database Vault is its compliance reporting. There are a number of reports available in Oracle Database Vault that are important to specific vertical industries and industries that face multiple compliance rules such as HIPAA and PCI DSS.

Oracle Database Vault's reports fall into two categories: Oracle Database Vault Reports and General Security Reports. Oracle Database Vault Reports give the capability to check configuration issues with realms, command rules, factors, rule sets, and secure application roles. The reports can identify if there have been realm violations and report on activity that has occurred. Figure 15 is an example of auditing for realm activity.



Logged in as SECURITY_ADMIN

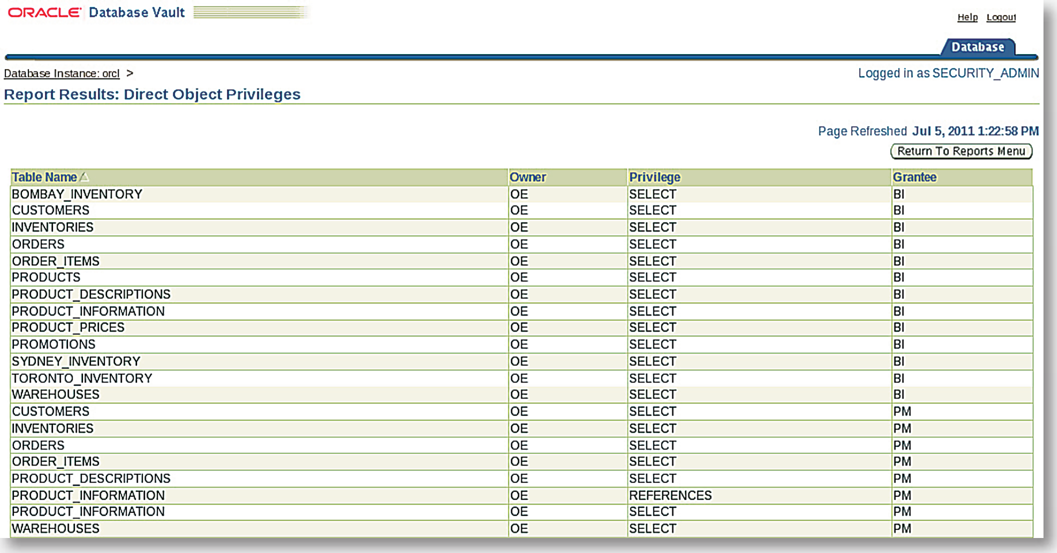
Report Results: Realm Audit

Page Refreshed Jul 5, 2011 1:16:12 PM

Violation Attempt	Timestamp	Return Code	Account	User Host	Instance Number	Realm Name	Rule Set	Command
Realm Violation Audit	05-JUL-11 10.25.06 AM	47401	HR	dbvault.oracle.com	0	HR Application protection realm		CREATE TABLE EMPLOYEES_BAK AS SELECT * FROM EMPLOYEES
Realm Violation Audit	05-JUL-11 10.19.27 AM	1031	SYS	dbvault.oracle.com	0	HR Application protection realm		SELECT * FROM HR.EMPLOYEES
Realm Violation Audit	30-JUN-11 02.24.06 PM	47401	DBA_JSMITH	dbvault.oracle.com	0	Oracle Data Dictionary		REVOKE DBA FROM DBA_JSMITH
Realm Violation Audit	30-JUN-11 02.10.44 PM	1031	SYS	dbvault.oracle.com	0	HR Application protection realm		SELECT * FROM HR.EMPLOYEES

Figure 15: Database Vault Report: Realm Audit

General Security Reports provide the capability to check the status of object privileges, system privileges, database objects, privilege management, administrative accounts and roles, initialization parameters, profiles, account passwords, security audits, and other security vulnerability reports.[4] Figure 16 is an example of a report that provides information about direct object privileges.



Logged in as SECURITY_ADMIN

Report Results: Direct Object Privileges

Page Refreshed Jul 5, 2011 1:22:58 PM

Table Name	Owner	Privilege	Grantee
BOMBAY_INVENTORY	OE	SELECT	BI
CUSTOMERS	OE	SELECT	BI
INVENTORIES	OE	SELECT	BI
ORDERS	OE	SELECT	BI
ORDER_ITEMS	OE	SELECT	BI
PRODUCTS	OE	SELECT	BI
PRODUCT_DESCRIPTIONS	OE	SELECT	BI
PRODUCT_INFORMATION	OE	SELECT	BI
PRODUCT_PRICES	OE	SELECT	BI
PROMOTIONS	OE	SELECT	BI
SYDNEY_INVENTORY	OE	SELECT	BI
TORONTO_INVENTORY	OE	SELECT	BI
WAREHOUSES	OE	SELECT	BI
CUSTOMERS	OE	SELECT	PM
INVENTORIES	OE	SELECT	PM
ORDERS	OE	SELECT	PM
ORDER_ITEMS	OE	SELECT	PM
PRODUCT_DESCRIPTIONS	OE	SELECT	PM
PRODUCT_INFORMATION	OE	REFERENCES	PM
PRODUCT_INFORMATION	OE	SELECT	PM
WAREHOUSES	OE	SELECT	PM

Figure 16: General Security Report

[4] http://download.oracle.com/docs/cd/B28359_01/server.111/b31222/reports.htm

Conclusion

When administering the database, privileged users are often granted the keys to the kingdom, giving them access to critical data within the database, even if they do not need to see, copy or touch that data in order to do their jobs. Regulations require that only those who need access to sensitive data should be given access. All other users should not be given access, including privileged users and administrators.

Until now, getting granular controls around DBA access to the database contents has been problematic because databases don't have the capability to limit what a database administrator or other privileged user can do in the database. Oracle Database Vault operates within the database, giving the capability to limit access of privileged users and DBAs to critical data and actions by separating critical data and objects into realms. With these controls, administrators can be granted privileges for their daily functions but not the data residing in the database, while organizations can leverage multiple granular access control rules on an as-needed basis.

Oracle Database Vault, part of the larger Oracle Database Security family of security and compliance solutions, is certified for Oracle E-Business Suite, PeopleSoft, Siebel, SAP, and JD Edwards EnterpriseOne database applications. By using the Oracle Database Vault out-of-the-box policies for these applications, organizations can get value quickly.

During this review, Oracle Database Vault was easy to set up and administer through the Oracle Database Vault Administrator (DVA). Rules were easy to create, manage and change on the fly through the Oracle DVA. Attempts to circumvent Oracle controls were met with additional layers of protection based on location and other levels of granularity not otherwise found in database access controls. Plus, with more than three dozen out-of-box reports, Oracle Database Vault can identify who has access to what, helping to demonstrate proof of compliance.

Oracle Database Vault does not replace existing database privileges. It simply provides an additional level of security to control for better compliance and controls around sensitive, regulated data within Oracle databases.

About the Author

Tanya Baccam is a SANS senior instructor as well as a SANS courseware author. She is the current author for the SANS Security 509: Securing Oracle Databases course.[5] Tanya works for Baccam Consulting, where she provides many security consulting services for clients, including system audits, vulnerability and risk assessments, database audits, and web application audits. Today much of her time is spent on the security of databases and applications within organizations. Tanya has also played an integral role in developing multiple business applications. She currently holds the CPA, GCFW, GCIH, CISSP, CISM, CISA, and OCP DBA certifications.

SANS would like to thank its sponsor:

ORACLE®

[5] www.sans.org/security-training/securing-oracle-74-mid