ORACLE DATABASE 12c

Oracle Database Vault

Best Practices

ORACLE WHITE PAPER | MAY 2015

ORACLE

Table of Contents

Executive Overview

Oracle Database Vault provides powerful security controls for protecting applications and sensitive data. Oracle Database Vault prevents privileged users from accessing application data, restricts ad hoc database changes and enforces controls over who, when, where, and how application data can be accessed. Oracle Database Vault secures existing database environments transparently, eliminating costly and time consuming application changes.

This paper provides best practices for rapid deployment of Oracle Database Vault protections to secure sensitive application data inside the database. It covers the following main topics:

- Installation
- Separation of Duty
- Database Administration
- Defining Oracle Database Vault Protections
- Post Installation
- Maintenance Considerations

Installation

Starting with Oracle Database 12c, Oracle Database Vault is installed by default but not enabled. Customers can enable it using DBCA or from the command line using SQL*Plus in a matter of minutes.

Oracle Database Vault can be enabled in existing environments where Oracle and third party applications are already installed. Subsequent installation of new applications or patching require Oracle Database Vault DV_PATCH_ADMIN role to be granted to the user doing the installation or patching.

Pre-Installation Notes

During the enablement process, DBCA provides the ability to create an account management responsibility. Oracle recommends creating this responsibility to provide enhanced separation of duties between Oracle Database Vault administration, database account management, and the DBA responsibilities. Customers can use Oracle Enterprise Manager Cloud Control 12c to manage Oracle Database Vault.

Separation of Duty

Separation of duty has taken on increased importance over the past 10 years. For many organizations separation of duty is a new concept that continues to evolve. Database consolidation, regulatory compliance and outsourcing are just a few of the drivers for increased separation of duty. Database Vault separation of duty strengthens security by separating out security related administration from day to day DBA operations. Database Vault allows organizations to tailor their Database Vault separation of duty implementation to easily adapt to current and future business requirements. Small organizations, in particular, need flexibility as they attempt to increase their security profile with limited resources.

Before separation of duty can be successful, it is important to understand who performs basic administration tasks in your environment and what those administration tasks are. Even if a single DBA is responsible for managing both new database account provisioning and application patching, these individual tasks are important to document and plan for. Using separate administration accounts for these types of tasks provides increased accountability and reduces associated risks. In midsize to large organizations database administrators typically need to perform common administration tasks but they don't need access to business data managed by the application. Creating a matrix for your separation of duty can be a helpful exercise when planning your Database Vault deployment. Additional tasks and associated users can be added to this list. This information should become part of the overall enterprise security documentation for your organization.

Separation of Duty Matrix

User, Process or Application	Account Creation	Database Administration					Security Administration
		SYSDBA	Backup	Tuning	Patching	Monitoring	
MarySmith	X					X	
HirotoSato							X
EmmaSchmidt			X				
MiguelSilva					X		
FatmaYilmaz				X			
SYSTEM							
RMAN		X	X				
...							

Table 1 Example Separation of Duty Matrix

In some cases, some of the system management tasks might require temporary access to data through specific tools and programs. Provisions for this temporary or emergency access need to be built into the Database Vault application protection rules.

Oracle Database Administration

Oracle recommends that customers who use the Oracle SYSTEM account for general DBA purposes create named DBA accounts for their database administrators. Doing so will provide increased accountability for administrative actions in the database.

Oracle SYSTEM User

Many applications developed in the past have used the Oracle user account SYSTEM for holding some application tables. It may be necessary to add the SYSTEM account to your Realm authorizations for some applications to continue working normally. Note that restrictions can be placed on the SYSTEM account to increase security in these scenarios. For example, an Oracle Database Vault rule set could be used to restrict connections as the SYSTEM user to specific IP addresses.

Oracle SYSDBA Access

Oracle recommends strictly limiting connections using the SYSDBA role. Only connect to the database using the SYSDBA role when absolutely necessary and for those applications that still require SYSDBA connections, such as Oracle RMAN and mandatory patching processes. For all other cases, create named database accounts to perform daily database administration. In the future, Oracle will be eliminating the requirement to connect as SYSDBA for any activity.

ROOT and other Operating System Access

As noted in the Oracle Database Vault administrator's guide, Oracle Database Vault doesn't prevent highly privileged operating system users from directly accessing database files. For this kind of protection, Oracle's Transparent Data Encryption is recommended. Oracle recommends carefully reviewing and restricting direct access to the operating systems.

Oracle recommends having personalized accounts to access the operating system. These personalized accounts should, in the Linux or UNIX environments, do sudo to the oracle software owner when needed. And with sudo you control which specific command each personalized user can execute.

Naming Conventions

Using a consistent and good naming convention when creating Database Vault security policies helps security administrators, auditors, and business users better understand what is being protected and how the different security elements relate to each other. Use the following naming convention when creating Database Vault security policies:

Realms	» *Use the protected application's name as the realm name* » *In the realm description, describe the business objective of the given application protection and document all other security policies the compliment the realm's protection. You also need to document who is authorized to the realm, for what purpose, and any possible emergency authorizations.*
Rule Sets	» *Start the name with a noun and complete it with the realm or command rule name the rule set will be attached to* » *Document the business requirement of the rule set in the description field*
Rules	» *Start the name with a verb and complete the name with the purpose of the rule* » *Rules don't have a description field, so make the name explicit but be sure to go over 90 characters*
Factors	» *Start the name with a noun and complete the name with a description of the derived value.*

Table 2 Naming Conventions

Defining Oracle Database Vault Realms

Upon installation, Oracle Database Vault creates four realms out-of-the-box. One of the default realms is called the data dictionary realm. Named administrators will need to be added to the Data Dictionary realm as owners or participants. While the Oracle SYSTEM account can be added as an authorized user to the Data Dictionary realm, Oracle discourages customers from using this generic database account.

Oracle Database Vault realms can protect a single object or an entire application schema. In most cases protecting the entire application provides a simplified yet robust protection model. Once a realm has been created, multiple users can be authorized to access the realm. Database objects (accounts, roles....) can be authorized in multiple realms.

Assigning Roles Realm Authorization

Be mindful of the privileges currently allowed to a role that you plan to add as a realm authorization. Realm authorization of a role can be accidentally granted and not readily apparent because the creator of a database role is implicitly granted the role when the role is created. As a result, if an account such SYSTEM creates a role and the Oracle Database Vault administrator subsequently adds this role as a realm authorization, then the SYSTEM user

would implicitly be given access to the realm. This is because the account that creates a role is implicitly granted the role when it is created. As a best practice, always create realm specific roles as the realm owner.

Realm Authorizations

1. **Application Owner** - The application owner typically corresponds to the schema containing the objects associated with the application. This user can be designated as the realm owner. Application servers typically connect to the application using the application owner account. In addition, server based batch jobs may connect to the application owner either directly or via a proxy connection.
2. **Application Users** - Application users usually authenticate to the middle tier and communicate with the backend database through a one-big user model. The one big user connection usually authenticates to the application owner. You can limit the application owner account access to the database through the middle tier processes and restrict the access to the middle tier servers IP addresses or host names. To accomplish this, the customer would need to create a Rule Set. In this Rule Set the customer would need to create rules that specify the application user, the middle tier processes he can connect through, and the IP address/s of the middle tier or the middle tier's computer name/s. After that, another rule need to be added to this Rule Set that evaluates to true if the connecting user is not the application user. The Rule Set Evaluation Options must be set to Any True. Once this is done, a Command Rule can be created for the CONNECT command and the Rule Set can be associated with it. You can refer to the Oracle Database Vault security policies published for PeopleSoft for an example of this.
3. **Application DBA** - this user can be added to the application Realm as a participant and associated with a Rule Set that allows him / her to perform all required patching and maintenance of the application while prohibiting him / her form doing SELECT on application data. This user can be further restricted by limiting his / her access to specific day of the week, time of the day, and or computer or subnet he uses to access the Database. Additional customer-specific restrictions can be added based on the customer's security requirements.

Defining Oracle Database Vault Rules Sets

Rule sets can be created that restrict access based on time, specific hosts, subnets or any other Database Vault factors supplied out-of-the-box. In addition, custom factors can be created using the Oracle Application Context.

- Each authorized user can be associated with a different Database Vault Rule Set.
- Each authorized user can be associated with a different Rule Set that specifies conditions and restrictions on access to the objects protected by the realm.

Command Rules

Oracle Database Vault Command Rules can be used to protect application objects from modification. For example, command rules can be used to place restrictions on the drop table command. Once created, the command rule can be associated with a Database Vault rule set that is called Disabled. For patching or maintenance operations the command rule can be edited and associated with a rule set called Enabled.

A separate white paper has been published with a suggested list of such Command Rules that need to be created and is downloadable from OTN (Oracle Technology Network web site http://otn.oracle.com).

Rule sets provide an easy way to group individual rules together into a meaningful set. You can share rules among multiple rule sets. This lets you develop a library of reusable rule expressions. Oracle recommends that you design such rules to be discrete, single-purpose expressions. As a naming convention, name your rule starting with a verb and complete the name with the purpose of the rule. For example, to create a rule that allows connections coming from certain IP addresses, name the rule: "Allow Connect from Middle Tier IP Addresses". Name Rule Sets starting with a noun and complete the name with the name of the Command Rule, Factor, or Realm authorization that it will be associated with. For example, the name for the rule set that will be associated with the SADM user's access to the Siebel Realm will be: "Siebel SADM Realm Access". In the Rule Set Description field, document the business requirements that are accomplished by this Rule Set.

Oracle Database Vault factors can be leveraged in your rule expressions to provide powerful checks and also to increase overall security by eliminating the requirement to manually define context values inside Oracle. Quite simply, factors provide contextual information to use in your security rules expressions.

You can use custom event handlers to extend Oracle Database Vault security policies to integrate external systems for error handling or alerting. The Oracle Database Vault Administrator's Guide shows how utility packages such as UTL_TCP, UTL_HTTP, UTL_MAIL, UTL_SMTP, or DBMS_AQ can be used to achieve this type of integration and do things like sending an email alert. An example of sending an email alert is documented in the Database Vault Administration Guide.

It's important to test rule sets thoroughly. When testing some rule sets, it's especially important to have a separate simultaneous connection as the Database Vault security administrator. For example, if you create a rule set for the CONNECT operation, you may need to disable the rule set so that you can change or fix an issue. By having the Database Vault administrator logged in simultaneously you are still able to disable the rule set. Otherwise you might be locked out of the database by a faulty rule set.

It's also important to test individual rules and rule sets in non-production or test environments before applying the rules to protect sensitive data. You can test rule expressions directly with the following SQL statement:

```
SQL> SELECT SYSDATE from DUAL where [rule expression goes here];
```

You can nest rule expressions inside a single rule. This helps satisfy more complex situations where you would need a logical AND for a subset of rules and a logical OR with the rest of the rules. See section 5-8 in the Oracle Database Vault administrator's guide for an example.

Planning Your Oracle Database Vault Protections

Planning your protection is an important part of the Oracle Database Vault deployment. Oracle Database Vault Realms, Command Rules, Rule Sets, and Factors can be used with a high degree of granularity. However, knowing the middle tier connections, batch jobs and processes that interact with the application is important before moving forward.

Understanding Your Application Architecture

First, it's important to understand the basic architecture of the application you wish to protect. For example, are the objects associated with the application spread across multiple database schemas or are they contained in a single database schema? This analysis should include all objects related to application data including tables, views, materialized views, and stored procedures. Identify the programs, processes, middle tier connections, database users, and application administrators that interact with the application objects. Once this information is obtained, the Oracle Database Realm definitions can be created and you can authorize who should be able to access application data. Application end users typically access application data through the middle tier. Some legacy applications may still use the client server architecture where end users have their own account in the database. More sophisticated applications may have application specific processes that run on the server hosting the Oracle Database.

Application Protection Matrix

Creating an application protection matrix will reduce the chances of overlooking authorization requirements during Database Vault deployment. Figure 3.0 below shows the PeopleSoft protection matrix. The y-axis shows the protection type (Realm, Command Rule) and the x-axis shows the authorizations and their associated rule sets. This matrix can serve as example on how to create one for your own custom applications.

Protection Type		Authorized with Rule Set		
PeopleSoft Realm	Owner	Owner	No Access	No Access
Select Command Rule		Limit PSFTDB Rule Set	No Access	No Access
Connect Command Rule	PeopleSoft Access Rule Set		No Access	No Access
Drop Tablespace Command Rule	Disabled Rule Set	Disabled Rule Set	Disabled Rule Set	Disabled Rule Set

Table 3 Example PeopleSoft Protection Matrix

Post Installation Tasks

Document

Documenting your security policies is important for demonstrating control processes to both internal and external auditors as well as providing operational continuity. Consideration should be given to documenting the following:

Processes and procedures	» *Backup* » *Patching* » *Tuning and monitoring*
Database accounts	» *Purpose* » *Production status* » *SYSDBA access*
SYSTEM access	» *When should SYSTEM be used*
SYSDBA	» *When should SYSDBA be used*
Reporting	» *Report names* » *Report frequency* » *Report distribution*
Emergency procedures	» *When should security policies be disabled*

Table 4 Security Documentation Matrix

Emergency Access

In some cases it may be necessary to temporarily relax realm protections for an administrative task. Oracle recommends having the Security Manager (DV_ADMIN or DV_OWNER) log in, add the named account to the authorized accounts for the realm, and set the authorization rule set to Enabled. This approach is better than temporarily disabling the Realm because protections are still in place and the new authorizations have been limited. Then in the enabled rule set, turn on all auditing for the rule set. You can remove the realm authorization when the administrative task is complete. This case also applies to emergency situations and sometimes commonly called “Break the Glass” scenario. Customers can also use Mandatory Realms to setup protection around very sensitive tables that should be protected even during maintenance. A Mandatory Realm can be set and enabled around these tables and can be disabled once the maintenance and emergency access is finished.

Appendix A – Command Rule Tips

Following these guidelines for configuring command rules:

- Create finer-grained command rules, because they are far easier to maintain. For example, if you want to prevent SELECT statements from occurring on specific schemas, design the command rule to stop the SELECT statement on the specific schema or table versus blocking SELECT statements in all cases.
- When designing rules for the CONNECT event, be careful to include logic that does not inadvertently lock out the Oracle Database Vault Owner or Administrator. If the rule set associated with the CONNECT command rule blocks all sessions you will need to disable Oracle Database Vault and disable the associated rule set then enable Database Vault again, before being able to work again on designing the right rule set. See "Enabling and Disabling Oracle Database Vault" in Appendix B of the Oracle Database Vault administrator's guide for more information.
- Sometimes you need to temporarily relax an enabled command rule for an administrative task. Rather than disabling the command rule, have the Security Manager (the account with the DV_ADMIN or DV_OWNER role) log in, set the rule set to Enabled, turn on Auditing on Success or Failure for the Enabled rule set, and then set the command rule back to its original rule set when the task is complete.
- When designing Command Rules, be careful to consider automated processes such as backup where these procedures may be inadvertently disabled. You can account for these tasks by creating rules that allow the command when a series of Oracle Database Vault factors is known to be true, for example, the program being used, and the account being used or the computer or network on which the client program is running.

Appendix B – Factor Tips

Follow these guidelines for configuring factors:

- Do not specify a retrieval method if the factor identification is set to Identified By Factors.
- Retrieval methods are only needed if you set the factor to By Method or By Constant.
- Consider using a validation method if a factor has an assignment rule set. Doing so helps to verify that invalid identities are not submitted.
- Only specify an evaluation option of By Access if the value returned by the retrieval method could change from one invocation to the next in the same session, for example, time-based factors.
- Optimize the internal logic of a function used for the factor retrieval method using traditional SQL and PL/SQL optimization techniques. For more information about performance and optimization, see Oracle Database Performance Tuning Guide.
- If the discrete values returned by the retrieval method are known, be sure to define identities for each value so that you can assign trust levels for them. Trust levels add value to factors as you also can use the trust level in application logic based on factors.
- A security policy based on more than one factor stronger than one based on fewer factors. You can create a new factor that is identified by other factors to store combinations of factors into logical grouping using identity maps. Client-supplied factors can only be trusted when the client software is trusted and the communications channel from the client software is known to be secure. Using multi-factor authorization dramatically increases the level of security.
- You can design a database client application to pass one or more security, end-user, or environmental attributes so that they are available to an associated database session. To do this, create a single factor for each attribute and then use an assignment rule set to control when these attributes can be assigned, for example only when using a specific Web application on specified named application server computers. Oracle Database Vault factors when used in this fashion are very much like the Oracle procedure DBMS_SESSION.SET_IDENTIFIER but also include a capability to control when they can be set. For more information about the DBMS_SESSION package, see Oracle Database PL/SQL Packages and Types Reference.

ORACLE

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

Oracle Database Vault Best Practices
May 2015