An Oracle White Paper
March 2009

Oracle Label Security in Government and Defense Environments

Protecting Sensitive Information 2

Oracle Label Security Overview... 2

Getting Started with Oracle Label Security 3

Separation of Duty 3

Policy Creation 4

Label Components Definition 5

Security Clearances 7

Security Clearances for Application Users...................................... 8

User Privileges 9

Enforcement Options 10

Conditional Where Clauses.. 11

Identity Management Integration... 12

Creating and Defining a Label Strategy 13

Labeling Legacy Data 13

Performance Considerations... 15

Partitioning Based On Data Classification 16

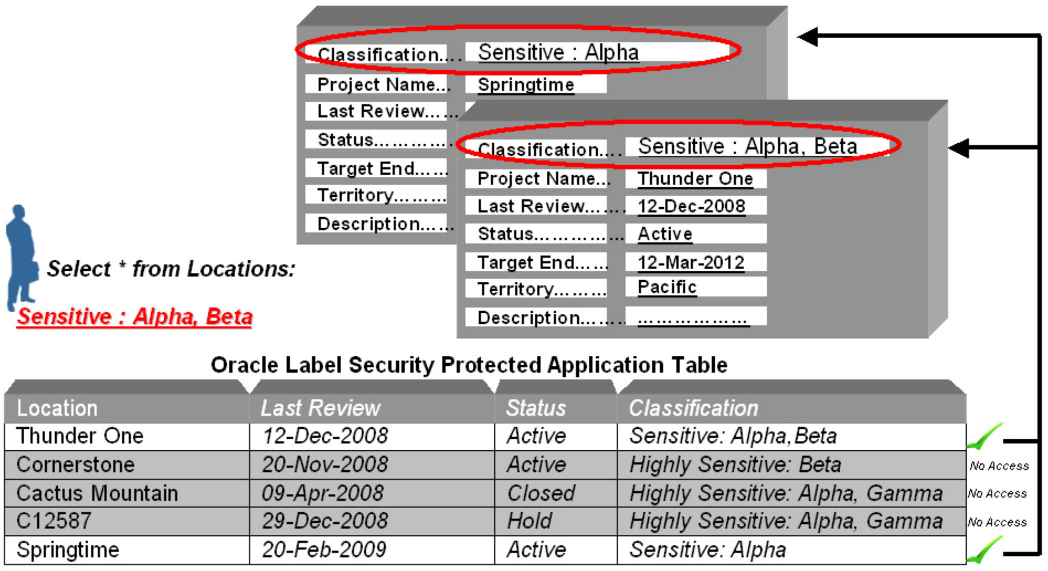
Conclusion 17

Protecting Sensitive Information

Over the past 30 years Oracle has been the industry leader in building advanced data security solutions that make it possible to protect sensitive information. Oracle Label Security (OLS) is part of Oracle's defense-in-depth approach to security and is the industry's most advanced solution for controlling access based on data classification. The ability to control access based on data classification is important for enforcing the principle of "need-to-know" as well as data consolidation. Data consolidation not only reduces cost, but also enables increased efficiencies in data analysis and decision making.

Oracle Label Security Overview

OLS enforces access controls by comparing a data classification label with a user's security clearance. A security clearance can be thought of as an extension to standard database privileges and roles. For example, a very common database operation is to *grant select* on an application table to a user or a role. However, how do you *restrict access to highly sensitive* data? For that to happen two things have to take place: First the database has to know what data is considered *highly sensitive* and secondly the database has to know the security clearance of the user. Oracle Label Security solves this problem by providing the ability to define data classification labels, assign security clearances to users, assign data classification labels to data, and enforce access control. Historically the design approach used to achieve this type of functionality was based on database views, triggers and lookup tables. However, that approach required extensive application changes and resulted in inconsistent implementations across applications. Oracle Label Security is enforced within the database, below the application layer, providing stronger security and eliminating the need for application views and triggers. For organizations that require the use of independently evaluated products, Oracle Label Security has been evaluated under the international common criteria at EAL4+.



Oracle Label Security Protected Application Table

Location	Last Review	Status	Classification	
Thunder One	12-Dec-2008	Active	Sensitive: Alpha,Beta	✓
Cornerstone	20-Nov-2008	Active	Highly Sensitive: Beta	No Access
Cactus Mountain	09-Apr-2008	Closed	Highly Sensitive: Alpha, Gamma	No Access
C12587	29-Dec-2008	Hold	Highly Sensitive: Alpha, Gamma	No Access
Springtime	20-Feb-2009	Active	Sensitive: Alpha	✓

FIGURE 1. ORACLE LABEL SECURITY OVERVIEW

Getting Started with Oracle Label Security

Oracle Label Security is not installed by default with typical database installations. In order to install Oracle Label Security the *custom* installation option must be selected within the Oracle installer and Oracle Label Security option must be specifically checked from the list of available software components.

Separation of Duty

When the Oracle Database Configuration Assistant (DBCA) is run, a new security related account called *LBACSYS* will be created. The *LBACSYS* account is the primary OLS administrator and stores OLS policies, data labels, and security clearances. LBACSYS stands for *Label Based Access Control SYS*. This user is separate from the Oracle Database *SYS* and SYSTEM accounts and is the primary account used for administering OLS. After installation of OLS the *LBACSYS* account will be locked. The designated Oracle database security administrator will need to unlock the *LBACSYS* account before it can be used. The installation of OLS will also move the database audit table *AUD$* from the SYS schema to the SYSTEM schema. This is done to enable additional audit columns to be appended to the AUD$ table specific to OLS.

Access to information stored in *LBACSYS* is controlled through policy specific roles and database views. Management of specific policies can be delegated to authorized individuals using OLS specific database roles and by granting privileges on specific administrative packages.

In addition to holding the meta data associated with OLS, the *LBACSYS* account will also hold several dozen procedures and functions. Examples of OLS specific functions installed include *char_to_label* and *label_to_char.* These two functions provide translation between an external human readable *label* and the internal label representation stored within the database. Another important function is the *dominates* function. This function enables a program module to compare two labels and determine whether one label dominates another label. For example, a program module might want to determine whether a user can perform a specific action by comparing a user's active session label with a fixed label. This function can also be used within Oracle Database Vault command rules to determine whether a user should be able to perform a specific operational task within the database. This is an alternative use case for security clearances outside of pure data classification and provides for a finer grained separation of duty capability.

Policy Creation

The first step in setting up OLS is defining a *policy*. OLS policies are *named* containers for a collection of data labels, user security clearances, and protected objects. Multiple policies can be defined within a single database. Each OLS policy can have a default set of protective enforcement options, such as READ CONTROL and WRITE CONTROL. The default enforcement options are used when a policy is applied to an application table. Enforcement options can also be customized on a per table basis. When defining an OLS policy, a column name must be provided to store the data classification label. When a policy is applied to an existing application table, the additional column can be appended as a *hidden* column, thus enabling existing SQL statements to continue working without any changes.

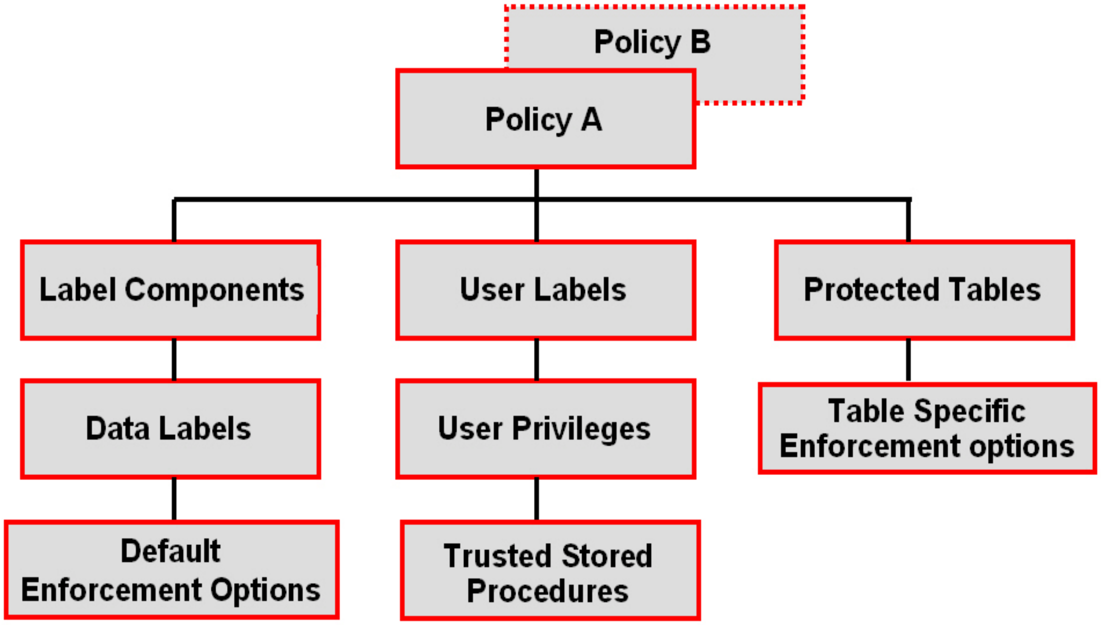

Policy B
Policy A
Label Components
User Labels
Protected Tables
Data Labels
User Privileges
Table Specific Enforcement options
Default Enforcement Options
Trusted Stored Procedures

FIGURE 2. ORACLE LABEL SECURITY POLICY MODEL

Label Components Definition

The second step in setting up an OLS policy is defining the label components. Label components consist of levels, compartments and groups. These label components are used to create data labels as well as to assign security clearances to database or application type users. *Levels* are hierarchical in nature and are used to assign the degree of sensitivity. *Compartments* are used to segregate data within a given *Level* and *Groups* are used to segregate data organizationally within a given *Level*. A given data label can have one *level*, zero or more *compartments* and zero or more *groups* associated with it.

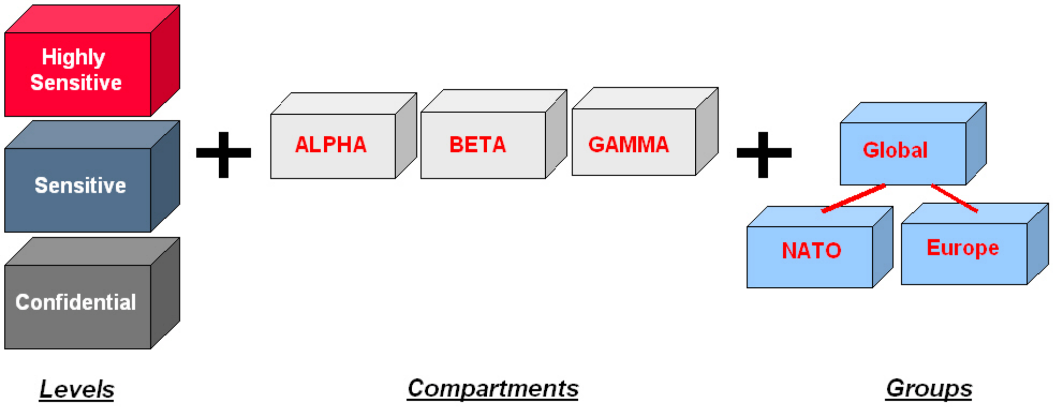


Figure 3. Data Label Components

TABLE 1. DATA LABEL COMPONENTS

COMPONENT	DESCRIPTION
Level	The *level* is a hierarchical component that denotes the sensitivity of the data. Each and every data label *must* have a level. A typical organization might define levels such as *Confidential*, *Sensitive* and *Highly Sensitive*.
Compartment	The *compartment* component is optional and is sometimes referred to as a category and is non hierarchical. Typically one or more compartments are defined to compartmentalize data. Compartments might be defined for a specific type of data, knowledge area or project that requires special approval.
Group	The *group* component is optional and is very similar to a compartment with a few exceptions. Each group can have a parent child relationship. Groups are most often used to segregate data by organization.

OLS provides the ability to define data classification labels to match specific business and organizational requirements. For example, a government organization might have levels such as *Secret* and *Confidential* while a commercial organization might have levels such as *Confidential* and *Public.*

TABLE 2. INDUSTRY SPECIFIC POLICIES

INDUSTRY	POLICY LABEL	LEVEL	COMPARTMENT	GROUP
Government and Defense		Confidential Secret Top Secret	Desert Storm Border Protection	NATO Homeland Security
Law Enforcement		Level 1 Level 2 Level 3	Internal Affairs Drug Enforcement	Local Jurisdiction FBI Justice Department
Human Resources		Confidential Sensitive Highly Sensitive	PII Data Investigation	WORLD US EMEA
Health Care		Confidential Public	Patient Doctor	Lab_Technician Medical_Assistant

Security Clearances

The third step in setting up an OLS policy is to assign security clearances. For example, a user can be assigned a maximum level of *Sensitive* and a minimum level of *Public*. Database users also have a default label that is initialized when the user connects to the database. This is sometimes referred to as the *active session label*. The *session label* is simply the user's *current level* combined with *compartments and groups.*

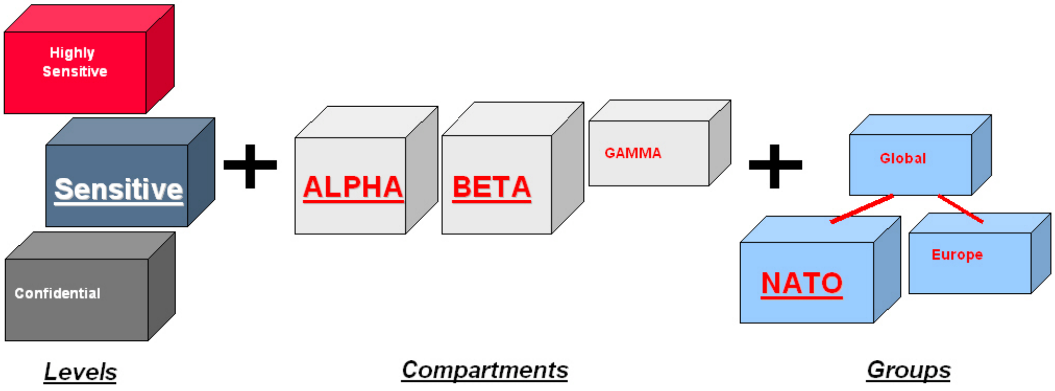


FIGURE 4. USER SECURITY CLEARANCE

User security clearances, sometimes referred to as label authorizations, are comprised of a minimum and maximum level, a default level and a row level. In addition, a security clearance can have compartments and groups.

TABLE 3. ORACLE LABEL SECURITY – SECURITY CLEARANCE COMPONENTS

CLEARANCE COMPONENT:	DESCRIPTION:
Maximum Level	The maximum sensitivity level a user is authorized to access. For example this might be Sensitive or Highly Sensitive.
Minimum Level	The minimum sensitivity level a user is authorized to write data. For example, an administrator can prevent users from labeling data as Confidential by assigning a minimum level of Sensitive.
Default Level	The level used by default when a user connects to the database. For example, a user can set his or her default level to Sensitive. When he or she connects to the system, the default level will be initialized to Sensitive.
Row Level	The default level used to label data inserted into the database by the user through the application or directly through a tool such as SQL*Plus.
Read Compartments	The set of compartments assigned to the user and used during READ access mediation. For example, if a user has compartments A, B and C, he could view data which has compartments A and B but not data which has compartments A, B, C and D.
Write Compartments	The set of compartments assigned to the user and used during WRITE access mediation. For example, a user could be given READ and WRITE access to compartments A and B but READ-ONLY access to compartment C. If an application record was labeled with compartments A, B and C, the user would not be allowed to update the record because he or she does not have WRITE access on compartment C.
Read Groups	The set of groups assigned to the user and used during READ access mediation. For example, if a user had the group Manager, he could view data that has the Manager group but not data that had only the Senior VP group.
Write Groups	The set of groups assigned to the user and used during WRITE access mediation. For example, a user could be given READ and WRITE access to group *Senior VP* but READ-ONLY access to group *Manager*. If an application record was labeled with a single group, *Manager*, the user would not be allowed to update the record because he or she does not have WRITE access on the *Manager* group.

Security Clearances for Application Users

OLS supports common application architectures including situations where the middle-tier connects to the database using a single database account. OLS does not enforce a relationship between physical database users and security clearances. For example, a security clearance as well as OLS specific privileges could be assigned to a database user named Scott who happens to have a database account or an application user such as JSMITH who is only known to the application layer and doesn’t have a real account in the database. The only difference is that when the user SCOTT logs into the database

OLS will automatically establish an active session label based on levels, compartments and groups assigned to SCOTT. In order for the active session label to be established for application user JSMITH a call to the OLS function set_access_profile is required. This function acts as a proxy for OLS and accepts an OLS policy name along with an application user name. Since multiple OLS policies can reside in any database, the OLS policy name must be provided when calling the function. The OLS PROFILE_ACCESS privilege is required to execute the SET_ACCESS_PROFILE procedure.

Applications can use one of the many Oracle SYS_CONTEXT variables in combination with the SET_ACCESS_PROFILE command. Applications using Oracle Enterprise User Security can pass the EXTERNAL_NAME SYS_CONTEXT value to the SET_ACCESS_PROFILE command:

```
SQL> execute sa_session.set_access_profile
       (‘DEFENSE’,SYS_CONTEXT('userenv','EXTERNAL_NAME');
```

Applications can also pass the PROXY_USER or CLIENT_IDENTIFIER as follows:

```
SQL> execute sa_session.set_access_profile
       (‘DEFENSE’,SYS_CONTEXT('userenv','PROXY_USER');
```

```
SQL> execute sa_session.set_access_profile
       (‘DEFENSE’,SYS_CONTEXT('userenv','CLIENT_IDENTIFIER');
```

Note that OLS security clearances can be assigned to Database Vault factors such as *IP addresses*. This capability provides powerful and interesting security deployments for organizations with complex security requirements.

User Privileges

For added flexibility, users can also be assigned OLS specific privileges. Examples of OLS specific privileges include *READ* and FULL. The *READ* privilege simply allows a user to view all data regardless of its data classification. These privileges can also be granted to stored procedures, enabling access all data within the execution context of stored procedure but not directly by the user calling the stored procedure or function.

TABLE 4. ORACLE LABEL SECURITY SPECIAL USER PRIVILEGES

PRIVILEGE NAME	DESCRIPTION
READ	The READ authorization enforces no additional read access control. Access mediation is still enforced on UPDATE, INSERT and DELETE operations. Oracle Label Security makes no mediation check on SELECT
FULL	The FULL authorization turns off all Oracle Label Security access mediation. A user with the FULL authorization can perform SELECT, UPATE, INSERT and DELETE operations with no label authorizations. Note that Oracle SYSTEM and OBJECT authorizations are still enforced. For example, a user must still have SELECT on the application table. The FULL authorization turns off the access mediation check at the individual row level.
WRITEDOWN	The WRITEDOWN authorization allows a user to modify the level component of a label and lower the sensitivity of the label. For example, application data which is labeled *Highly Sensitive: Alpha, Beta* could be changed to *Sensitive: Alpha, Beta*. This authorization is only applicable to policies that use the *label update* enforcement option.
WRITEUP	The WRITEUP authorization allows a user to modify the level component of a label and raise the sensitivity of the label. For example, application data which is labeled *Sensitive: Alpha, Beta* could be changed to *Highly Sensitive: Alpha, Beta*. Note that the Maximum Level label authorization assigned to the user would limit modification. This authorization is only applicable to policies that use the *label update* enforcement option.
WRITEACROSS	The WRITEACROSS authorization allows a user to modify the compartments and groups in a label to any valid compartment and group defined in Oracle Label Security for the policy. For example, data labeled *Sensitive: Alpha* could be modified to *Sensitive: Alpha, Beta* even though the user was not authorized for the Delta compartment. This authorization is only applicable to policies that use the label update enforcement option.
PROFILEACCESS	The PROFILE ACCESS authorization allows a user to assume the Oracle Label Security authorizations of another user. For example, user Scott who has access to compartments A, B, and C could assume the profile of user Joe who has access to compartments A, B, C and D. This functionality might be useful in an environment where an application uses a single application account for all application users. Note that the PROFILEACCESS privilege cannot be granted to a stored procedure.

Enforcement Options

Oracle Label Security policies by definition have a default set of enforcement options. The enforcement options SELECT, INSERT, UPDATE, UPDATE, LABEL UPDATE, and CHECK CONTROL. Once a policy is applied to an application table, the enforcement options can be customized.

TABLE 5. ORACLE LABEL SECURITY POLICY ENFORCEMENT OPTIONS

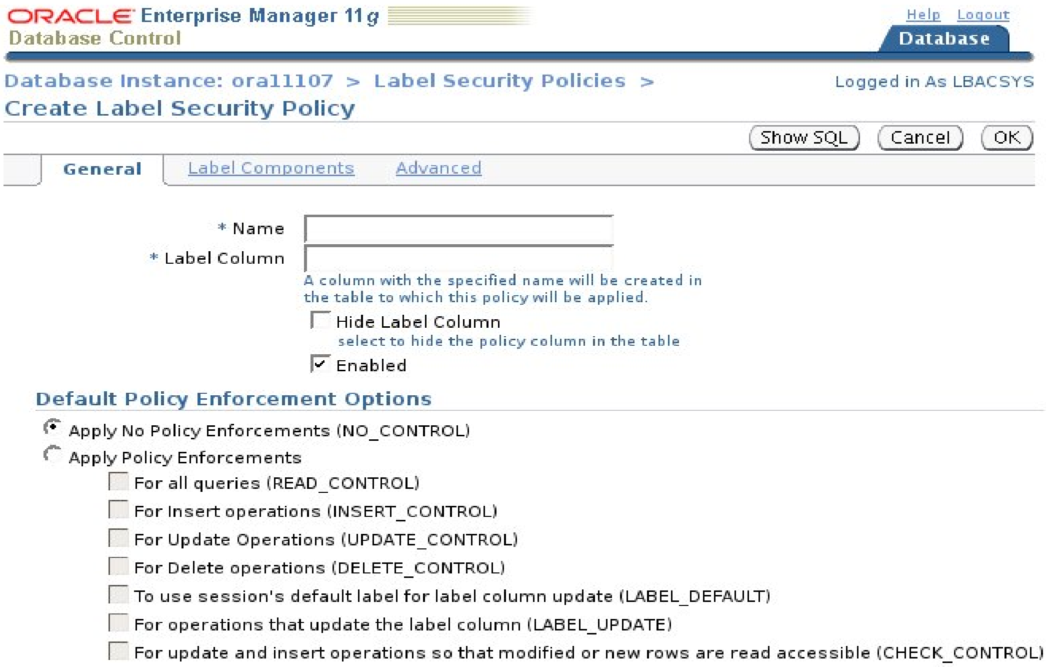
ENFORCEMENT OPTION	DESCRIPTION
READ CONTROL	Applies policy enforcement to SELECT operations using the Oracle Label Security algorithm for read access.
INSERT CONTROL	Applies policy enforcement to INSERT operations using the Oracle Label Security algorithm for write access.
UPDATE CONTROL	Applies policy enforcement to UPDATE operations using the Oracle Label Security algorithm for write access.
DELETE CONTROL	Applies policy enforcement to DELETE operations using the Oracle Label Security algorithm for write access.
WRITE CONTROL	Applies policy enforcement on INSERT, UPDATE, and DELETE operations. If this option is set, it enforces INSERT_CONTROL, UPDATE_CONTROL, and DELETE_CONTROL.
LABEL DEFAULT	If the user does not explicitly specify a label on INSERT, the user's default row label value is used. By default, the row label value is computed internally by Oracle Label Security using the user's label. The default value would be comprised of the default ROW LEVEL combined with the WRITE COMPARTMENTS and WRITE GROUPS. A user can set the row label independently, but only to: A level which is less than or equal to the level of the session label, and greater than or equal to the user's minimum level. Include a subset of the compartments and groups from the session label, for which the user is authorized to have write access.
LABEL UPDATE	Applies policy enforcement to UPDATE operations that set or change the value of a label attached to a row. The WRITEUP, WRITEDOWN, and WRITEACROSS privileges are only enforced if the LABEL_UPDATE option is set.
LABEL CHECK	Applies READ_CONTROL policy enforcement to INSERT and UPDATE statements to assure that the new row label is read-accessible by the user after and INSERT or UPDATE statement.
NO CONTROL	Applies no enforcement options. A labeling function or a SQL predicate can nonetheless be applied.

Conditional Where Clauses

OLS also provides the ability to add an adhoc restrictive *'where*' clause or *'condition'* when a policy is applied to an application table. This '*where'* clause is used in conjunction with data labels to determine access and provides an easy to use, simple capability similar to creating an Oracle Virtual Private Database (VPD) policy. The *'where'* clause is attached to the OLS policy, thus there is no need to create a separate

PL/SQL package as is the case with pure VPD type implementations.OLS and Enterprise Manager

Starting with Oracle Database 11g, OLS can be managed from Oracle Enterprise Manager. The OLS management link can be found in the *Security* section under the *Server* Tab. The new EM interface replaces the OLS management tool called Oracle Policy Manager that was shipped with releases starting with Oracle9i. To begin managing OLS from Oracle Enterprise Manager, the LBACSYS account must first be granted an additional privilege known as the *select any dictionary* privilege. The database administrator can grant the LBACSYS account this privilege.



Create Label Security Policy

* Name

* Label Column

A column with the specified name will be created in the table to which this policy will be applied.

□ Hide Label Column
select to hide the policy column in the table

☑ Enabled

Default Policy Enforcement Options

● Apply No Policy Enforcements (NO_CONTROL)
○ Apply Policy Enforcements
- □ For all queries (READ_CONTROL)
- □ For Insert operations (INSERT_CONTROL)
- □ For Update Operations (UPDATE_CONTROL)
- □ For Delete operations (DELETE_CONTROL)
- □ To use session's default label for label column update (LABEL_DEFAULT)
- □ For operations that update the label column (LABEL_UPDATE)
- □ For update and insert operations so that modified or new rows are read accessible (CHECK_CONTROL)

FIGURE 5. ORACLE LABEL SECURITY POLICY MODEL

Identity Management Integration

Starting with Oracle Database 10g, Oracle Label Security provides integration with Oracle Identity Management. This feature enables centralized management of policy definitions, data labels and user label authorizations. Oracle Identity Management must be licensed separately for this capability.

Creating and Defining a Label Strategy

Restricting access based on data classification requires a firm understanding of the various roles and responsibilities that exist among application users accessing data. The first step that needs to be performed is a comparison between the defined data labels and user security clearances. The reason this step is important is to make sure data is accessible to users who should have access based on their job responsibility. In other words, the information required to perform a specific job responsibility might be out of reach to the application user based on their security clearance. In the worst case, data might be assigned a data label that no user can access, effectively hiding the data.

Table	User Data	C	S	S:A:US	S:A,B:US,UK
Assets	C::UK	No Access	No Access	No Access	Access
	C::US	No Access	No Access	Access	Access
Projects	C	Access	Access	Access	Access
	S	No Access	Access	Access	Access
	S:A:US	No Access	No Access	Access	Access
	S:B:UK	No Access	No Access	No Access	Access
	S:A,B:US	No Access	No Access	No Access	Access

TABLE 6. ORACLE LABEL SECURITY ANALYSIS MATRIX

Labeling Legacy Data

Once an OLS policy is applied to an application table with READ CONTROL no rows will be visible until valid data labels have been assigned to each data row. This due to the fact that the policy label column is NULL. This situation is most commonly encountered when a policy is first applied to an application table. A common way this problem is addressed is to grant the security administrator responsible for labeling the initial data the Label Security authorization *FULL*. This will allow the administrator to see all rows regardless of the data label and ensure that all legacy data rows are property labeled. Note that data loaded subsequent to a policy being applied can have a label automatically assigned based on the users *active session label* or a labeling function.

Several methods exist for labeling legacy data. The first method for labeling legacy data simply uses an update statement against the base table.

```
SQL> UPDATE LOCATIONS SET SECLAB = char_to_label(‘Defense’,’S’) WHERE
REGION_ID = 104;
```

This statement updates the *locations* table and sets the policy label column *seclab* equal to the internal label tag defined for *sensitive* data in the *defense* policy and where the *locations* table column *region_id* is equal to 104.

The second method for labeling legacy data is to perform multiple data loads and switch database connections. This method works very well if you are moving data from distributed databases with known data classifications. If the policy applied to the *locations* table includes the LABEL_DEFAULT option, the users default ROWLABEL value will be used to initialize the label tag column.

```
SQL> CONNECT LOCATIONS_MANAGER1 / PWD
      INSERT INTO SALES (Col1, Col2, Col3) VALUES (‘ACME’,......);

SQL> CONNECT LOCATIONS_MANAGER2 / PWD
      INSERT INTO SALES (Col1, Col2, Col3) VALUES (‘WIDGET’,......);
```

Oracle Data Pump could also be used in a similar method using data exports from the distributed databases. The user specified on the Oracle Data Pump command line would have a default *active session label* equal to the desired data classification value.

Performance Considerations

Performance is important to all applications. Adding new functionality to existing applications requires due diligence up front to minimize the performance impact. Oracle Label Security provides row level security, basically turning on a security check at each row prior to allowing access. OLS will add a delay during login authentication to initalize additional security contexts in Oracle memory. For common application type models, the same delay will be encountered when calling the *set_access_profile* function. The amount of delay will vary depending on the number of Oracle policies and the number of label components defined. Runtime performance overhead will depend on a variety of factors including:

- Number of tables protected by Label Security
- Label Security enforcement options used
- Complexity of existing application SQL logic

Identifying the tables that require a Label Security policy is an important part of the upfront analysis. Careful consideration of where to apply Label Security will result in an efficient use of the technology. Carefully consider the enforcement options you apply to an application table and use only those that are necessary to meet your security requirements. Each additional security check performed by Oracle Label Security will add additional performance overhead.

Oracle also recommends defining the associated label tags so that they fall within the range associated with the level of the data label. Label tags are used in the *hidden* column appended to OLS protected tables and map to the external or human readable form of the data label. For example, suppose the levels *confidential* and *sensitive* have been defined along with two compartments, *alpha* and *beta*. The number associated with Confidential is 5000 and the number associated with Sensitive is 10000. When the valid data labels are defined, the associated label tags associated with confidential should be between 5000 and 10000. For example the data label *confidential: alpha* could have a label tag of 5050 and the data label *sensitive: alpha, beta* could have a label tag of 10055.

Existing composite indexes can be modified to include the policy column added by Label Security. This can substantially improve performance for complex queries.

Should any user or stored procedure need access to all data it is recommended that the user or stored procedure be given the Oracle Label Security specific privilege READ or FULL. This will help reduce overhead and increase performance.

When labeling new data, Label Security label functions will have the most performance overhead as they will invoke an internal database trigger. Using the *LABEL DEFAULT* enforcement policy enforcement option will have the least performance overhead.

Depending on the application usage, consideration should be given to creating bitmap indexes on the column added by Oracle Label Security to the application table. For large tables, the percentage of unique labels compared to the number of data rows is usually low. Bitmap indexes will slow down data loads but increase performance on *select* statements.

Partitioning Based On Data Classification

Oracle partitioning can be used with Oracle Label Security to physically partitioned data based on data classification. For example, data with a classification of *Highly Sensitive* can be located in a separate partition from data with a classification of *Sensitive*. Partitioning can also provide performance benefits through partition elimination, enabling OLS to quickly skip data that resides outside of the users security clearance.

Conclusion

Data classification plays a vital role not only in enforcing the principle of *need-to-know* but also in securely consolidating highly sensitive data. Historically highly sensitive data has been stored in physically separate systems. However, this approach has limited the ability to perform advanced analysis and business intelligence. OLS provides the industries most advanced and flexible data classification solution. Using a policy based architecture, OLS provides the ability to define data labels, assign security clearances and protect application tables within the Oracle database. First introduced in Oracle8i, OLS has been independently evaluated under the international common criteria at EAL4+.

Oracle Label Security in Goernment and
Defense Environments
March 2009
Author: Paul Needham
Contributing Authors: Peter Wahl

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Oracle is committed to developing practices and products that help protect the environment

0109