

Oracle Label Security For Privacy and Compliance

*An Oracle White Paper
June 2007*

Oracle Label Security For Privacy and Compliance

Introduction	3
Oracle Label Authorizations	3
Label Authorizations and Oracle Database Vault.....	4
Oracle Database Vault Rules and Rule Sets.....	4
Label Authorizations and Database Vault Use Cases.....	5
Restricting Database Connections Using Label Factors	5
Restricting Application Access Using Label Factors	6
Oracle Label Security with Secure application Roles	6
Oracle Label Security with Virtual Private Database	7
Oracle Label Security Policies	7
Sensitivity labels	8
Level Components.....	8
Compartment Components	9
Group Components	9
User Label Authorizations	9
Proxy with SET_ACCESS_PROFILE Command	10
Advanced Row Level Data Labeling	11
Enforcement Options	11
Labeling Functions	12
Predicates	12
Summary	12
Appendix A - Label Security with Oracle Database Vault.....	13
Appendix B - Label Security with VPD Policy Example	14
Appendix C - Label Security with Secure Application Roles.....	15
Appendix D - Policy Enforcement Options	16

Oracle Label Security For Privacy and Compliance

INTRODUCTION

Oracle Label Security helps organizations address security and compliance requirements using sensitivity labels such as *confidential* and *sensitive*. Sensitivity labels can be assigned to users in the form of label authorizations and associated with operations and objects inside the database using data labels. Label authorizations provide tremendous flexibility in making access control decisions and enforcing separation of duty. Oracle Label Security can be used to address numerous operational issues related to security, compliance and privacy.

Used with Oracle Database Vault, Oracle Label Security label authorizations are factors that control access to applications, databases and data. Label authorizations can be used in conjunction with virtual private database to mask out PII data. Secure application role policies can use label authorizations to control access to powerful privileges. Sensitivity labels can also be used for informational purposes, informing an application user the database contains privacy related data and the data should be handled with care.

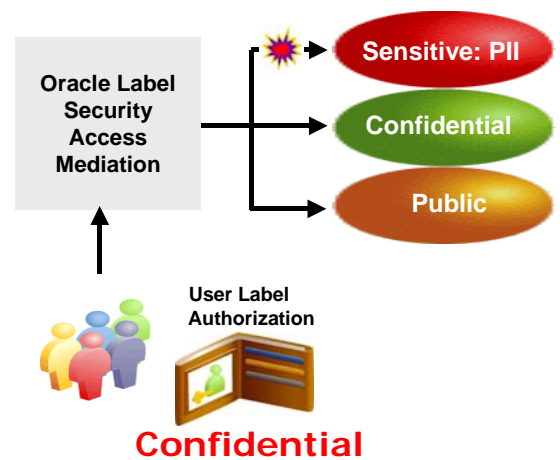


Fig 1. Oracle Label Security Overview

ORACLE LABEL AUTHORIZATIONS

Label authorizations can be used within numerous types of access control policies, ranging from controlling access to privacy related information to enforcing separation of duty. For example, Oracle Database Vault command rules can check whether a user has been authorized access to *Sensitive* data considered *Personally Identifiable Information (PII)*. Database administrators can be assigned different label authorizations, enforcing separation of duty within a consolidated application environment. Label authorizations can be assigned to database or application users. Label authorizations are managed using Oracle Enterprise Manager or the Oracle

Label Security command line interface. Label authorizations can optionally be managed for the entire enterprise using Oracle Identity Management.

ORACLE Enterprise Manager 11g Database Control

Database Instance: orcl.oracle.com > Label Security Policies in As LBACSYS

Authorization: ACME

Users Trusted Program Units

This table lists the users who are authorized for this policy. A user can be authorized under multiple policies.

Add Users

Edit View Create Like Delete

Select	User	Maximum Read Label	Maximum Write Label	Privileges
<input type="checkbox"/>	ISMITH	SENS:PII	SENS:PII	

Fig 2. Defining User Label Authorizations In Enterprise Manager

LABEL AUTHORIZATIONS AND ORACLE DATABASE VAULT

Label authorizations are powerful factors and can be used in a variety of ways within Oracle Database Vault, including within command rules and to enforce separation of duty requirements. For example, a *Select* command rule can check a user's label authorization before allowing access to an application table.

Oracle Database Vault Rules and Rule Sets

Oracle Database Vault provides numerous built-in factors, such as IP address, that enable command rules to control who, when, where and how applications, databases and data are accessed.

Rules Associated To The Rule Set

Create Add Existing Rules

Edit Remove

Select	Rule Name	Rule Expression
<input type="checkbox"/>	Enforce Local Access	sys_context('userenv','ip_address')='130.35.46.77'

Edit Remove

Fig 3. Oracle Database Vault Rule Using IP Address

Using Oracle Label Security, Oracle Database Vault command rules can reference label authorization factors using Oracle Label Security functions and verify the user has access to *Sensitive Personally Identifiable Information (SENS:PII)*

Select	Rule Name	Rule Expression
<input checked="" type="checkbox"/>	Check Label Authorization	dominates(sa_utl.numeric_label('ACME'), char_to_label('ACME','SENS:PII'))='1'

Fig 4. Oracle Database Vault Rule Using Label Authorizations

Label Authorizations and Database Vault Use Cases

Label authorizations combined with Oracle Database Vault enables powerful access control policies within the database.

- Limit connections to *databases* based on whether label authorizations include Sensitive Personally Identifiable Information (PII) access
- Limit access to *application tables* based on whether label authorizations include Sensitive Personally Identifiable Information (PII) access
- Limit DDL such as *Create Table* based on whether label authorizations include sensitive Personally Identifiable Information (PII) access

Restricting Database Connections Using Label Factors

ORACLE Database Vault Help Logout Database

Database Instance: orcl > Command > Edit Command Rule/SEEC

Edit Command Rule: CONNECT

Cancel OK

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:

Status: Enabled Disabled

Applicability

Object Owner:

Object Name:

Rule Set

Fig 5. Connect Command Rule With Label Authorization Rule

Restricting Application Access Using Label Factors

ORACLE Database Vault Help Logout
Database

Database Instance: orcl > Command > Edit Command Rule: SELECT

Edit Command Rule: SELECT

Cancel OK

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

Status Enabled Disabled

Applicability

Object Owner

Object Name

Rule Set

Fig 6. Select Command Rule With Label Authorization Rule

ORACLE LABEL SECURITY WITH SECURE APPLICATION ROLES

Using Oracle Label Security with secure application roles provides powerful controls over who can have access to powerful database privileges. Secure application roles are simply standard database roles associated with a PL/SQL package. After a user successfully authenticates to the database he or she must call the PL/SQL package to turn on the database role. The PL/SQL package can perform any number of security checks such as checking label authorizations before enabling the database role. Appendix C shows an example PL/SQL package that checks to make sure the user has access to at least *confidential* data before turning on the database role.

ORACLE LABEL SECURITY WITH VIRTUAL PRIVATE DATABASE

Using Oracle Label Security with Virtual Private Database (VPD) provides powerful controls over access to privacy related data. The new VPD column relevant feature introduced in Oracle Database 10g Release 1 enables a VPD policy to be associated with a column such as Social Security Number (SSN). Using Oracle Label Security functions inside the VPD policy function allows access to PII data to be easily controlled using label authorizations.

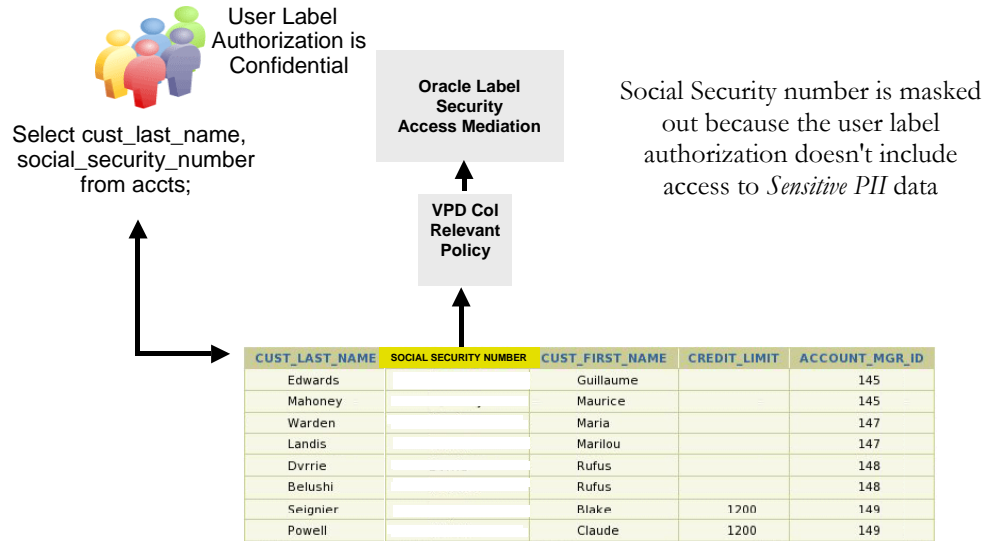


Fig 7. Oracle Label Security with Virtual Private Database

Please refer to appendix B for an example of VPD policy using Oracle Label Security.

ORACLE LABEL SECURITY POLICIES

Oracle Label Security policies are simply containers for valid sensitivity labels, data labels, label authorizations and optionally protected application tables. Oracle Label Security policies can be assigned descriptive names such as *HR*, *Finance*, *Legal* or *Privacy* and multiple policies can be provisioned in a single database. In addition, Oracle Label Security integrates with Oracle Identity Management, enabling centralized management of policy definitions.

Policies can be created by the Oracle Label Security administrator *LBACSYS* using Oracle Enterprise Manager or the Oracle Label Security command line interface. Please note that the LBACSYS account is locked by default after an Oracle installation. The database account manager can unlock and initialize the account password.

```
Execute SA_SYSDBA.CREATE_POLICY ( 'PRIVACY' , 'SEC_LABEL' , 'HIDE' );
```

When you create a policy you can optionally specify a column name to be used when policies are applied to application tables. You can designate the column as *hidden* using the HIDE option when the policy is applied, providing complete transparency to existing application SQL. The column name will be used only if you choose to apply the policy to an application table. Please refer to the section in this document on *Advanced Row Level Data Labeling*. Applying Oracle Label Security to an application table is not required to use label authorizations within Oracle Database Vault.

SENSITIVITY LABELS

Sensitivity labels are built using the various label components. Sensitivity labels are comprised of a single level component, plus zero or more compartments, plus zero or more groups. Label authorizations are also composed of these components. The syntax for a sensitivity label uses a colon to separate levels, compartments and groups. Commas are used to separate multiple compartments or groups within a given sensitivity label. The number associated with the label is known as the label tag and is used both internally and when applying an Oracle Label Security policy to an application table.

```
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL  
( 'PRIVACY' , 1000 , 'C' );  
EXECUTE SA_LABEL_ADMIN.CREATE_LABEL  
( 'PRIVACY' , 1100 , 'C:PII' );
```

Level Components

Basic sensitivity labels are comprised of a level component. Levels are hierarchical and denote the overall sensitivity. A typical organization might define three levels confidential, sensitive and highly sensitive. Each level must have a number associated with it that corresponds to its relative sensitivity. For example, sensitive (2000) is higher than confidential (1000). Levels can be defined using Oracle Enterprise Manager or the Oracle Label Security API.

```
EXECUTE SA_COMPONENTS.CREATE_LEVEL  
( 'PRIVACY' , 1000 , 'C' , 'CONFIDENTIAL' );  
EXECUTE SA_COMPONENTS.CREATE_LEVEL  
( 'PRIVACY' , 2000 , 'S' , 'SENSITIVE' );  
EXECUTE SA_COMPONENTS.CREATE_LEVEL  
( 'PRIVACY' , 3000 , 'HS' , 'HIGHLY_SENSITIVE' );
```

Compartment Components

More advanced sensitivity labels are comprised of a level and zero or more compartments and groups. Compartments and groups are additional label components that act as knowledge areas that can be part of a users label authorizations and a data label.

In order to pass a security check, the user's label authorization must have a level equal to or greater than the level associated with the object *and* all the compartments associated with the data label. In other words, the compartments the user has must be a superset of the compartments associated with a sensitivity label. The number used when defining a compartment is used for controlling the display order of the compartments.

```
EXECUTE SA_COMPONENTS.CREATE_COMPARTMENT
('PRIVACY',100,'PII','Personally Identifiable
Information');
```

Group Components

Groups are similar to compartments but can optionally have parent child relationships. In addition, during a security check, a user need only have a subset of the groups associated with a sensitivity label. If both groups and compartments are checked, then a user must have at least one of the groups and all of the compartments. The number associated with a group is used for controlling the display order of the groups.

```
EXECUTE SA_COMPONENTS.CREATE_GROUP
('PRIVACY',500,'EU','Europe');
```

USER LABEL AUTHORIZATIONS

Label authorizations are assigned after the label components have been defined. Oracle Label Security label authorizations can be assigned to database users or application users. Application users are users that don't have a physical account inside the database. Label authorizations can be assigned using Oracle Enterprise Manager or the Label Security command line interface.

```
EXECUTE SA_USER_ADMIN.SET_USER_LABELS
('PRIVACY','TRODGERS_US','S');
```

```
EXECUTE SA_USER_ADMIN.SET_USER_LABELS
('PRIVACY','JSMITH_US','S:PII');
```

PROXY WITH SET_ACCESS_PROFILE COMMAND

Oracle Label Security provides the ability for an authorized user to assume an Oracle Label Security label authorizations of another user using the SET_ACCESS_PROFILE procedure. The Oracle Label Security PROFILE_ACCESS authorization is required to execute the SET_ACCESS_PROFILE procedure. The SET_ACCESS_PROFILE procedure is provided so that Oracle Label Security can be used with application architectures that use one big user models, enterprise user security, or client identifiers. To accomplish this, Oracle Label Security does not enforce a mapping between a physical database account and the user name specified when establishing label authorizations. For example, label authorization could even be assigned to an IP address.

Applications can utilize one of the many Oracle SYS_CONTEXT variables to determine which Oracle Label Security label authorization profile should be specified in the SET_ACCESS_PROFILE command. For enterprise users the EXTERNAL_NAME SYS_CONTEXT value could be passed to the SET_ACCESS_PROFILE command.

```
SQL> execute sa_session.set_access_profile
      ('PRIVACY',SYS_CONTEXT('userenv','EXTERNAL_NAME'));
```

```
SQL> execute sa_session.set_access_profile
      ('PRIVACY',SYS_CONTEXT('userenv','PROXY_USER'));
```

```
SQL> execute sa_session.set_access_profile
      ('PRIVACY',SYS_CONTEXT('userenv','CLIENT_IDENTIFIER'));
```

ADVANCED ROW LEVEL DATA LABELING

Oracle Label Security policies can be applied to application tables using Oracle Enterprise Manager or the Oracle Label Security command line interface. Once applied, no data will be visible until existing data has been assigned a valid data label. New data inserted into the application table can be labeled automatically. Row level labeling is more advanced and requires application analysis. Oracle Label Security must be applied to an application table for row level data labeling to be enabled.

CUST_LAST_NAME	CUST_FIRST_NAME	Oracle Label Security Data Label
Edwards	Guillaume	Sensitive
Mahoney	Maurice	Sensitive
Warden	Maria	Sensitive
Landis	Marilou	Highly Sensitive
Dvrrie	Rufus	Sensitive
Belushi	Rufus	Sensitive
Seignier	Blake	Highly Sensitive
Powell	Claude	Highly Sensitive

Fig 8. Oracle Label Security Row Level Data Labeling

Oracle Enterprise Manager or the Oracle Label Security command line interface can be used to apply an Oracle Label Security policy to an application table.

```
sa_policy_admin.apply_table_policy (  
    POLICY_NAME => 'PRIVACY',  
    SCHEMA_NAME => 'APPADM',  
    TABLE_NAME => 'ACCTS',  
    TABLE_OPTIONS => 'READ_CONTROL, LABEL_DEFAULT',  
    LABEL_FUNCTION => NULL);
```

Please refer to the Oracle Label Security Best Practices For Government and Defense Applications white paper for more guidance on using sensitivity labels for row level data labeling, performance considerations, and labeling legacy data.

Enforcement Options

Oracle Label Security policy table enforcement options can be customized for each policy. For example, an HR and Finance policy can exist in the same Oracle database and provide different degrees of protection. The HR application might use the READ CONTROL option and the Finance policy might use the READ CONTROL and WRITE CONTROL options. Please refer to appendix D for a complete list of enforcement options.

Labeling Functions

Label functions are administrator defined PL/SQL functions that can be used to compute data labels for new data. During insert, the label function is called using an internal trigger within the database. Label functions will increase performance overhead. Please refer to Chapter 8 of the Oracle Label Security documentation for an example of a labeling function.

Predicates

Predicates or *where* clauses can optionally be added when applying a policy to an application table. This extensibility features allows simple VPD like policies to be incorporated into the label security policy enforcement. In the example below, the policy is checking the users label authorizations *and* verifying that the information is being requested during workdays.

```
sa_policy_admin.apply_table_policy (  
    POLICY_NAME => 'PRIVACY',  
    SCHEMA_NAME => 'APPADM',  
    TABLE_NAME => 'ACCTS',  
    TABLE_OPTIONS => 'READ_CONTROL, LABEL_DEFAULT',  
    LABEL_FUNCTION => NULL,  
    PREDICATE => 'to_char(sysdate,' || "" || 'd' || "" || ' ') in (2,3,4,5,6) );
```

SUMMARY

Oracle Label Security adds powerful access control capabilities to the Oracle Database. Used in combination with Oracle Database Vault, label authorizations are powerful factors that can be used for enforcing numerous security, compliance and privacy policies, including controlling access to applications, databases and data. Embedding Oracle Label Security within virtual private database policies provides highly granular and efficient controls over access to PII data. Used in conjunction with secure application roles, label authorizations help determine who should have access to powerful database privileges. The Oracle Label Security best practices paper for government and defense provides guidance on using Oracle Label Security for advanced row level data labeling. Please refer to the September/October 2006 Oracle Magazine issue for a feature article on how Artear, a media company, is using Oracle Label Security.

APPENDIX A - LABEL SECURITY WITH ORACLE DATABASE VAULT

```
Begin
-- Create Rule "Check Label Authorization"
  dvsys.dbms_macadm.CREATE_RULE(
    rule_name => 'Check Label Authorization',
    rule_expr => 'dominates(sa_utl.numeric_label(PRIVACY),
                  char_to_label('PRIVACY','S:PII') = '1')');

End;

Begin
  dvsys.dbms_macadm.CREATE_RULE_SET(
    rule_set_name => 'Check OLS Factors',
    description => 'Authorize action based on label
authorization',
    enabled => 'Y',
    eval_options => 2,
    audit_options => 1,
    fail_options => 1,
    fail_message => '',
    fail_code => NULL,
    handler_options => 0,
    handler => '');

End;

Begin
  dvsys.dbms_macadm.ADD_RULE_TO_RULE_SET(
    rule_set_name => 'Check OLS Factors',
    rule_name => 'Check Label Authorization');

End;

Begin
-- Create SELECT command rule
  dvsys.dbms_macadm.CREATE_COMMAND_RULE(
    command => 'SELECT'
    ,rule_set_name => 'Check Label Authorization'
    ,object_owner => 'APPDBA'
    ,object_name => '%'
    ,enabled => 'Y');
  commit;

End;

Begin
  dvsys.dbms_macadm.SYNC_RULES;

End;
```

APPENDIX B - LABEL SECURITY WITH VPD POLICY EXAMPLE

```
CREATE OR REPLACE PACKAGE SECURITY_PACKAGE AS
FUNCTION mask_pii(owner varchar2, objname varchar2)
    return varchar2;
END SECURITY_PACKAGE;
/

CREATE OR REPLACE PACKAGE BODY SECURITY_PACKAGE IS

    FUNCTION MASK_PII (owner varchar2, objname varchar2)
        return varchar2 is
            predicate varchar2(2000);
    begin

        predicate := '1=2';

        if dominates(sa_utl.numeric_label(PRIVACY),
            char_to_label('PRIVACY','S:PII')) = 1 then

            predicate := '1=1';

        else

            predicate := '1=2'

        end if

        return predicate;

    END MASK_PII;
END SECURITY_PACKAGE;
/
```

APPENDIX C - LABEL SECURITY WITH SECURE APPLICATION ROLES

```
CREATE ROLE fin_admin IDENTIFIED USING SECADM.HR_ADMIN;

CREATE OR REPLACE PACKAGE secadm.hr_admin IS
PROCEDURE hr_admin_check;
END;
/
CREATE OR REPLACE PACKAGE BODY hr_admin_check authid
current_user IS
PROCEDURE hr_admin_check IS
BEGIN

    /* Simple check to see if current session has a label
       authorization of at least Confidential:PII */

    if dominates(sa_utl.numeric_label(PRIVACY),
                char_to_label('PRIVACY','S:PII')) = 1
    then

        dbms_session.set_role('admin_role');

    else

        null;

    end if;

END;
```

APPENDIX D - POLICY ENFORCEMENT OPTIONS

READ CONTROL — Applies policy enforcement to all queries; only authorized rows are accessible for SELECT, UPDATE, and DELETE operations.

INSERT CONTROL — Applies policy enforcement to INSERT operations, according to the Oracle Label Security algorithm for write access.

UPDATE CONTROL — Applies policy enforcement to UPDATE operations on the data columns within a row, according to the Oracle Label Security algorithm for write access.

DELETE CONTROL — Applies policy enforcement to DELETE operations, according to the Oracle Label Security algorithm for write access.

WRITE CONTROL — Determines the ability to INSERT, UPDATE, and DELETE data in a row. 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 label authorization values specified for the user. A user can set the row label independently, but only to:

A level that 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 an INSERT or UPDATE statement.

NO CONTROL — Applies no enforcement options. A labeling function or a SQL predicate can nonetheless be applied.



Oracle Label Security For Privacy and Compliance

June 2007

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

Copyright © 2007, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.