

**SECURE DATA AT THE SOURCE
SAVE TIME AND MONEY**

Security Inside Out

Oracle Database Security

Oracle Powers Innovation



Oracle 11g Database Security Workshop

**Providing Defense-in-Depth Solutions to Secure Data at
the Source, Reduce Risk and Simplify Compliance**

Workshops2Go Series
V5.12 REV: April 24, 2012

ORACLE®

Author(s)

Mark Waldron
Ken Zeng
Barbara Gingrande

**Technical Contributors
and Reviewers**

Todd Bottger
Peter Wahl
Kamal Tbeileh
Tammy Bednar
Stuart Sharp
Jagan Athreya
Gary Fisk
Donald Shepherd

Copyright © 2012, Oracle. All rights reserved.

This documentation contains proprietary information of Oracle Corporation. It is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited. If this documentation is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

Restricted Rights Legend

Use, duplication or disclosure by the Government is subject to restrictions for commercial computer software and shall be deemed to be Restricted Rights software under Federal law, as set forth in subparagraph (c)(1)(ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

This material or any portion of it may not be copied in any form or by any means without the express prior written permission of Oracle Corporation. Any other copying is a violation of copyright law and may result in civil and/or criminal penalties.

If this documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with "Restricted Rights," as defined in FAR 52.227-14, Rights in Data-General, including Alternate III (June 1987).

The information in this document is subject to change without notice. If you find any problems in the documentation, please report them in writing to Education Products, Oracle Corporation, 500 Oracle Parkway, Redwood Shores, CA 94065. Oracle Corporation does not warrant that this document is error-free.

Oracle and all references to Oracle Products are trademarks or registered trademarks of Oracle Corporation.

All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.

Contact For This Document

Please direct any questions or comments regarding the contents of this document to Mark Waldron (mark.d.waldron@oracle.com) or Ken Zeng (ken.zeng@oracle.com).

TABLE OF CONTENTS

TABLE OF CONTENTS	3
ORACLE SOLUTION CENTER - Summary of Accounts and Passwords	5
ORACLE VIRTUAL MACHINE IMAGE - Summary of Accounts and Passwords	6
Important Aliases And URLs	7

ORACLE ADVANCED SECURITY

LAB CONFIGURATION – ADVANCED SECURITY OPTION.....	8
LAB EXERCISE 00 – DATABASE - ADVANCED SECURITY OVERVIEW	12
LAB EXERCISE 01 – CONFIGURING AND DEMONSTRATING NETWORK ENCRYPTION.....	13
LAB EXERCISE 02 – CREATION OF THE ENCRYPTION WALLET	21
LAB EXERCISE 03 – PROTECTING SENSITIVE DATA – TABLESPACE LEVEL ENCRYPTION.....	28
LAB EXERCISE 04 – Wallet Management Operations.....	38
LAB EXERCISE 05 – PROTECTING SENSITIVE DATA AND OPTIMIZING STORAGE WITH ADVANCED COMPRESSION ON DISK & BACKUPS (OPTIONAL)	48

ORACLE DATABASE VAULT

LAB CONFIGURATION – ORACLE DATABASE VAULT.....	81
LAB EXERCISE 00 – ORACLE DATABASE VAULT OVERVIEW	85
LAB EXERCISE 01 – PROTECTING SENSITIVE DATA FROM PRIVILEGED USER ACCESS USING ORACLE DATABASE VAULT REALMS.....	86
LAB EXERCISE 02 – USING ORACLE DATABASE VAULT REALMS TO ENABLE DATABASE CONSOLIDATION.....	99
LAB EXERCISE 03 – ENFORCING OPERATIONAL CONTROLS USING ORACLE DATABASE VAULT MULTI-FACTOR AUTHORIZATION	115
LAB EXERCISE 04 – INCREASE VISIBILITY OF DATABASE ACCESS CONTROLS USING ORACLE DATABASE VAULT MONITORING & REPORTING	127

ORACLE AUDIT VAULT

LAB CONFIGURATION – SETUP OF THE AUDIT VAULT ENVIRONMENT.....	133
LAB EXERCISE 00 – AUDIT VAULT OVERVIEW.....	137
LAB EXERCISE 01 – EFFECTIVELY MANAGING DATABASE AUDIT POLICY	139
LAB EXERCISE 02 – REDUCE TIME TO COMPLIANCE USING ORACLE AUDIT VAULT REPORTING	156
LAB EXERCISE 03 – GAIN REAL-TIME DATABASE ACTIVITY MONITORING USING AUDIT VAULT ALERTING	176

ORACLE DATABASE FIREWALL

LAB CONFIGURATION – ORACLE DATABASE FIREWALL	192
LAB EXERCISE 00 – ORACLE DATABASE FIREWALL OVERVIEW	194
LAB EXERCISE 01 – ORACLE DATABASE FIREWALL ENFORCEMENT POINTS TO MONITOR AND PROTECT DATABASES.....	196
LAB EXERCISE 02 – ORACLE DATABASE FIREWALL – USE THE TRAFFIC ANALYZER TO CONFIGURE POLICIES AND BLOCK UNAUTHORIZED TRAFFIC.....	221
LAB EXERCISE 03 – ORACLE DATABASE FIREWALL – GAIN VISIBILITY AND SATISFY REQUIREMENTS THROUGH REPORTING	252
LAB EXERCISE 04 – ORACLE DATABASE FIREWALL – USING WHITELISTS TO PREVENT SQL INJECTION ATTACKS	266

ORACLE ENTERPRISE MANAGER - DATA MASKING

LAB CONFIGURATION – ENTERPRISE MANAGER 12c DATA MASKING	288
LAB EXERCISE 00 - ENTERPRISE MANAGER DATA MASKING PACK OVERVIEW.....	291
LAB EXERCISE 01 – CREATING A DATA MODEL	295
LAB EXERCISE 02 – IDENTIFYING SENSITIVE DATA	306
LAB EXERCISE 03 – CREATING, EXPORTING & IMPORTING DATA MASKING FORMATS	323
LAB EXERCISE 04 – MASKING SENSITIVE APPLICATION DATA.....	330
 PLEASE COMPLETE THE FEEDBACK SURVEY.....	 347

ORACLE SOLUTION CENTER - Summary of Accounts and Passwords

IMAGE NAME AND IP ADDRESS:

cloud.oracle.com - Ask Instructor
database firewall (DBFW) - Ask Instructor
windows management server - Ask Instructor

LINUX IMAGE OPERATING SYSTEM ACCOUNTS:

oracle/oracle
root/oracle

DATABASE FIREWALL IMAGE OPERATING SYSTEM ACCOUNTS:

root/oscdbfw
support/oscdbfw

WINDOWS IMAGE OPERATING SYSTEM ACCOUNTS:

mgmtserv1/oracle1

11g DATABASE ACCOUNTS (DB06) :

sysman/oracle1
sys/oracle1
system/oracle1

(where applicable)

dvowner/oracle12#
dvacctmgr/oracle12#

MASKING_ADMIN/oracle12#
INFOSEC_ISABEL/Manager_1
SEC_ADMIN_OWEN/Manager_1
ACCTS_ADMIN_ACE/Manager_1
DBA_DEBRA/Manager_1
DBA_NICOLE/Manager_1
APPS_DBA_HARVEY/Manager_1
APPS_DBA_SAM/Manager_1
APPS_DBA_OLIVER/Manager_1
MALICIOUS_MALFOY/Manager_1
USER_BARACK/Manager_1
SEC_ANALYST_ALLEN/Manager_1

AUDIT VAULT ACCOUNTS:

avadmin/oracle12#
avauditor/oracle12#
avdvo/oracle12#
advam/oracle12#
sys/oracle1
system/oracle1

DATABASE FIREWALL ACCOUNTS:

admin/tdsdbfw01

GRID CONTROL ACCOUNTS:

sysman/oracle123

ORACLE VIRTUAL MACHINE IMAGE - Summary of Accounts and Passwords

IMAGE NAME AND IP ADDRESS:

cloud.oracle.com - 192.168.214.67
database firewall (DBFW) - 192.168.56.30
windows management server - 192.168.214.10, 10.0.6.10

LINUX IMAGE OPERATING SYSTEM ACCOUNTS:

oracle/oracle1
root/oracle1

DATABASE FIREWALL IMAGE OPERATING SYSTEM ACCOUNTS:

root/oracle1

WINDOWS IMAGE OPERATING SYSTEM ACCOUNTS:

mgmtserv/g0Oracle12#

11g DATABASE ACCOUNTS (DB06):

sysman/oracle1
sys/oracle1
system/oracle1

(where applicable)

dvowner/oracle12#
dvacctmgr/oracle12#

MASKING_ADMIN/oracle12#
INFOSEC_ISABEL/Manager_1
SEC_ADMIN_OWEN/Manager_1
ACCTS_ADMIN_ACE/Manager_1
DBA_DEBRA/Manager_1
DBA_NICOLE/Manager_1
APPS_DBA_HARVEY/Manager_1
APPS_DBA_SAM/Manager_1
APPS_DBA_OLIVER/Manager_1
MALICIOUS_MALFOY/Manager_1
USER_BARACK/Manager_1
SEC_ANALYST_ALLEN/Manager_1

AUDIT VAULT ACCOUNTS:

avadmin/oracle12#
avauditor/oracle12#
avdvo/oracle12#
advam/oracle12#
sys/oracle1
system/oracle1

DATABASE FIREWALL ACCOUNTS:

admin/Oracle1

GRID CONTROL ACCOUNTS:

sysman/oracle123

Important Aliases And URLs

Aliases:

Alias	Execution Path	Description
agent	' . /home/oracle/agent.sh'	Sets environment for Grid control agent
av	' . /home/oracle/av.sh'	Sets environment for Audit Vault server
avagent	' . /home/oracle/avagent.sh'	Sets environment for Audit Vault agent
db	' . /home/oracle/db.sh'	Sets environment for
db06	' . /home/oracle/db06.sh'	Sets environment for DB06 database instance
emrep	' . /home/oracle/emrep.sh'	Sets environment for Grid Control EM repository database
oms	' . /home/oracle/oms.sh'	Sets environment for Oracle Management Server (OMS)
ora	'env grep ORA'	Shows current environment settings for session

URLs:

URL	Description
http://cloud.oracle.com:7799/em	EM 12c Grid Control Console
https://cloud.oracle.com:1158/dva	DB06 Database Console DVA
http://cloud.oracle.com:5500/av	Audit Vault Console

LAB CONFIGURATION – ADVANCED SECURITY OPTION

OVERVIEW

For these lab exercises, the following infrastructure components need to be started and available for your use.

- **Database: Database DB06**
- **Here is a summary of the users.**
 - DBA_DEBRA – Database Administrator
 - APPS_DBA_HARVEY – HR (Human Resources) Applications Specific DBA
 - INFOSEC_ISABEL – DBA Responsible for TDE Wallet Operations
 - USER_BARACK – Non-DBA User Account

Let's get started.

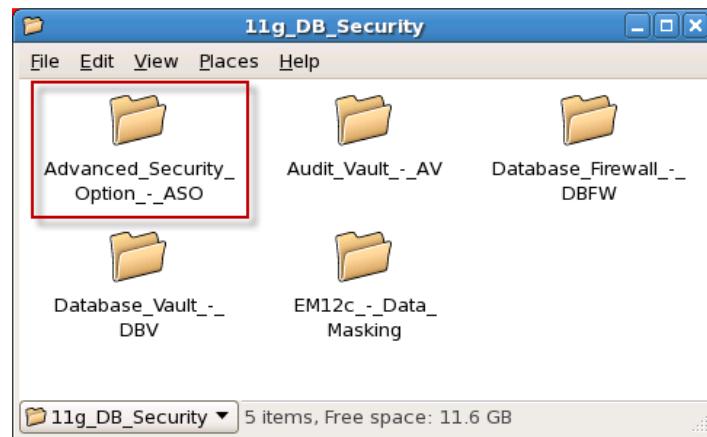
1. On the desktop, navigate to the **Labs** folder, double-click and open the contents.



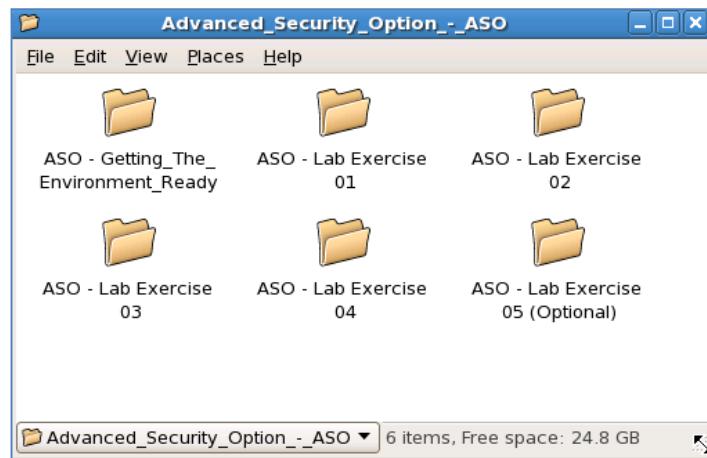
2. Select the folder, **11g_DB_Security**.



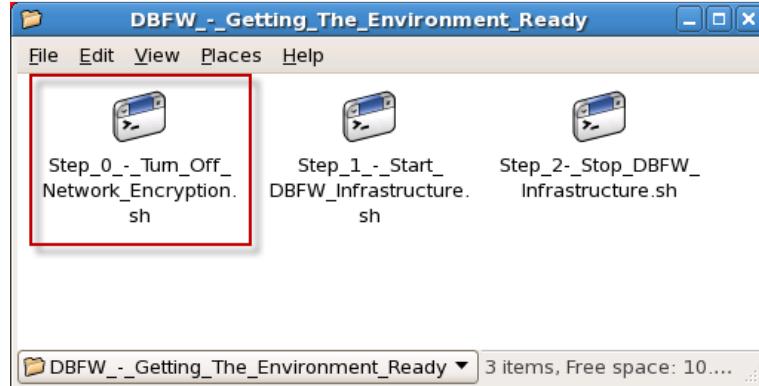
3. Select the folder, **Advanced_Security_Option_-_ASO**.



4. Within the **Advanced_Security_Option_-_ASO**, you can access all of the Lab folders. Select 'ASO – Getting The Environment Ready'.



5. Select '**Step_0_Turn_Off_Network_Encryption**'. This script turns off network encryption that you may have turned on in a previous lab.



When you double click on the script, the OS will prompt you to specify what you want to do with the shell script.

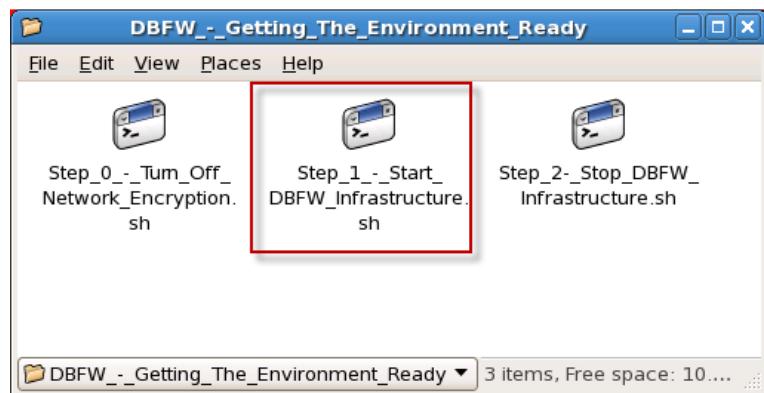


You can select either '**Run in Terminal**' to execute the script, or '**Display**' if you want to see more detail on what is actually being executed. You obviously must run each script to execute the labs, but feel free to display them if you wish to see what is being executed.

You will notice that output files will be saved in the folder after the script executes. You may review this output, as well.

Unless otherwise indicated, the windows will close when the script has completed. Please wait for each script to complete before executing the next script.

6. Select '**Step_1_Start_Infrastructure.sh**'. This script will start the database and initialize the environment used in this lab.



7. You are ready move forward with the ASO labs. Enjoy!!

LAB EXERCISE 00 – DATABASE - ADVANCED SECURITY OVERVIEW

INTRODUCTION

Oracle Advanced Security, part of Oracle's comprehensive portfolio of database security solutions, helps organizations comply with privacy and regulatory mandates such as Sarbanes-Oxley, Payment Card Industry (PCI) Data Security Standard (DSS), Health Insurance Portability and Accountability Act (HIPAA), as well as numerous breach notification laws. With Oracle Advanced Security, customers can transparently encrypt all application data or specific sensitive columns, such as credit cards, social security numbers, or personally identifiable information (PII). By encrypting data at rest in the database as well as whenever it leaves the database over the network or via backups, Oracle Advanced Security provides the most cost-effective solution for comprehensive data protection.

A. Lab Scenarios and Objectives

In our fictitious company, CashBankTrust is evaluating encryption technologies for their database environment. In addition to the identified challenges below, they are working towards meeting the requirements for the Payment Card Industry (PCI) standards. This involves encrypting certain of the data at rest AND encrypting data as it passes over the network-- specifically, under sections 3 & 4 of the PCI requirements. The Advanced Security Option (ASO) labs that you will complete will demonstrate solutions specifically to the identified challenges below.

Product	Identified Challenges
Advanced Security Option	Protecting Data in Transit Across Networks. Sensitive Information and Data is travelling over the network in clear text and is vulnerable to potential breach and potential exposure to global disclosure regulations
	At Rest Data Encryption. Data at rest (on disk and backup) is vulnerable to potential breech and potential exposure to global disclosure regulation.
	All locations of sensitive information (i.e. Credit Card Information, PII) within database systems have not been fully identified and documented.
	Reduce the management and performance overhead of deploying at rest data encryption in production.
	Proper Management of Encryption Keys and Wallets to Ensure Data Protection. Limited reporting and visibility to demonstrate that encryption controls are being implemented and enforced to identified sensitive data. Need to document and implement all key management processes and procedures used for encryption.

LAB EXERCISE 01 – CONFIGURING AND DEMONSTRATING NETWORK ENCRYPTION

Identified Challenge – Protecting Data in Transit Across Networks

Sensitive data is travelling over the network in clear text and is vulnerable to potential breach and potential exposure to global disclosure regulations.

INTRODUCTION

Oracle Advanced Security has included encryption of traffic between database servers and clients for many years. In modern, tiered infrastructures, SSL encryption between endusers and web clients is standard practice, but it is also important to consider encryption requirements between endusers and database servers directly (through reporting tools, command line tools, etc.), and between database servers (in DR environments, data distribution, etc.).

Oracle Advanced Security's network encryption is an extremely easy, efficient, non-intrusive technology to secure those crucial data flows.

A. Overview

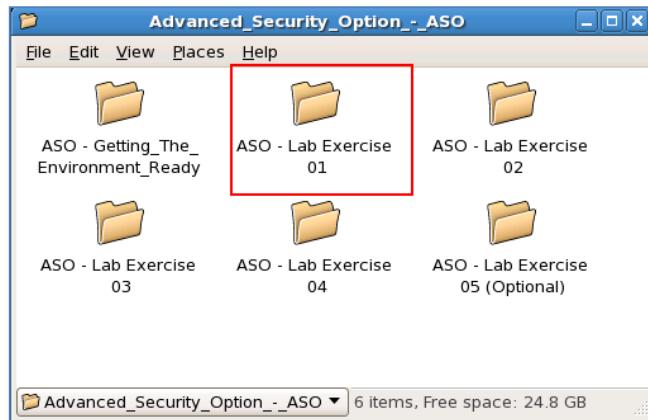
In this lab exercise, you will accomplish the following:

1. *Configure sqlnet.ora for supporting network encryption*
2. *Use tcpdump to demonstrate the before and after effect when the Advanced Networking Option (network encryption) is being used.*

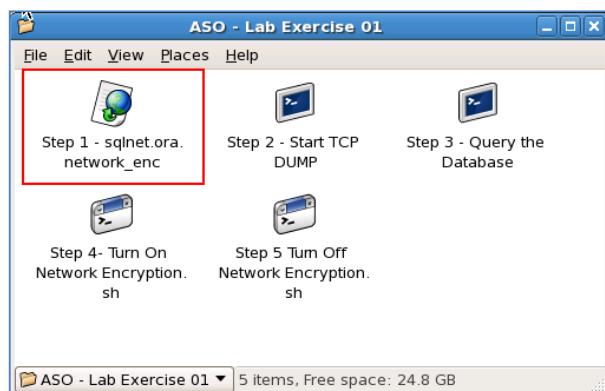
In any network connection, it is possible for both the client and server to support more than one encryption algorithm and more than one integrity algorithm. When a connection is made, the server selects which algorithm to use, if any, from those algorithms specified in the **sqlnet.ora** files.

B. Setup & Preparation

1. Navigate to the folder, **ASO – Lab Exercise 01**.



2. Initially, our environment is not setup to use ASO network encryption. To set up ASO network encryption, we will need to make changes to the SQLNET.ORA file in the \$ORACLE_HOME/network/admin directory. Click on the icon **Step 1 – sqlnet.ora.network_enc** to review the changes that will be added to the SQLNET.ORA to set up client side and server side network encryption.



Each parameter is explained below. This is all you need to do to implement Network Security.

```
SQLNET.CRYPTO_CHECKSUM_SERVER = REQUIRED  
SQLNET.ENCRYPTION_SERVER = REQUIRED
```

- To negotiate whether to turn on integrity (CHECKSUM) or encryption (ENCRYPTION), you can specify four possible values for the Oracle Advanced Security integrity and encryption configuration parameters – REJECTED, ACCEPTED, REQUESTED or REQUIRED. The four values are listed in order of increasing security. The value REJECTED provides the minimum amount

of security between client and server communications, and the value REQUIRED provides the maximum amount of network security. In this scenario, this side of the connection specifies that the security service must be enabled. The connection fails if the other side specifies REJECTED or if there is no compatible algorithm supported by the other side.

	Rejected	Accepted	Requested	Required
Rejected	OFF	OFF	OFF	ORA-12660
Accepted	OFF	OFF	ON	ON
Requested	OFF	ON	ON	ON
Required	ORA-12660	ON	ON	ON

`SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER = (MD5)`

- MD5 and SHA1 are the two integrity algorithms supported by Oracle ASO.

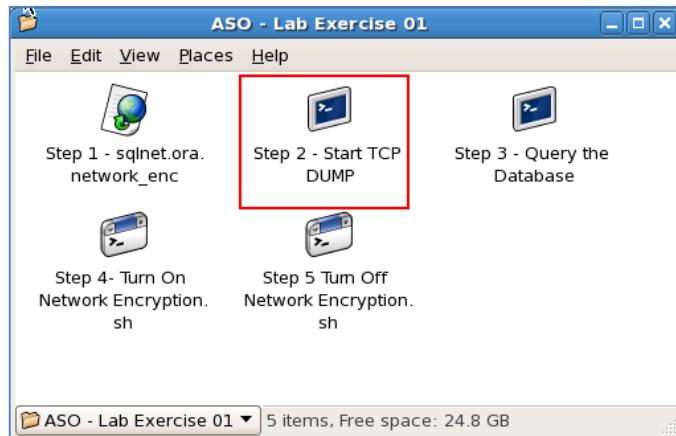
`SQLNET.ENCRYPTION_TYPES_SERVER = (DES40, RC4_40)`

- This parameter enumerates some subset of the encryption algorithms supported by ASO.

`SQLNET.CRYPTO_SEED="Between Ten and Seventy Random Characters"`

- Several seeds are used to generate a random number on the client and on the server. One of the seeds that can be used is a user-defined encryption seed. It can be 10 to 70 characters in length and changed at any time. The longer the string, the more secure the environment.
- Any client connecting to this server would need to have parallel settings in their local sqlnet.ora file. Otherwise their connections will be rejected.
- This change will take effect for all new connections to the database, since the parameters within sqlnet.ora are read during the establishment of every Oracle Net session. Note that existing connections i.e. those in place prior to the changes made to the sqlnet.ora files, will remain un-affected by these encryption settings. This would have implications for how organizations would enforce these new settings in a Production environment across, for example, an application server farm, where the use of pooled database connections implies the need to force re-connects from the mid-tier in order to pick up the new settings. In a 24x7 environment, this might be achieved via the use of ONS (Oracle Notification Service) to denote all such pooled connections as stale, thus forcing new connections to be established.

- To demonstrate that traffic is being encrypted over the network, we will be using the TCPDUMP utility as a sniffer-like device attached to the network. We will first view the data over the network before turning on network encryption and then view the network data after turning on network encryption. We will be monitoring network traffic on the Loopback (`lo`) port. Click on the icon **Step 2 – Start TCP DUMP** to open up a terminal window.



- Login in as root by typing the command '`su -`' (without the quotes) providing the provided password and then executing the script **`tcpdump.sh`**

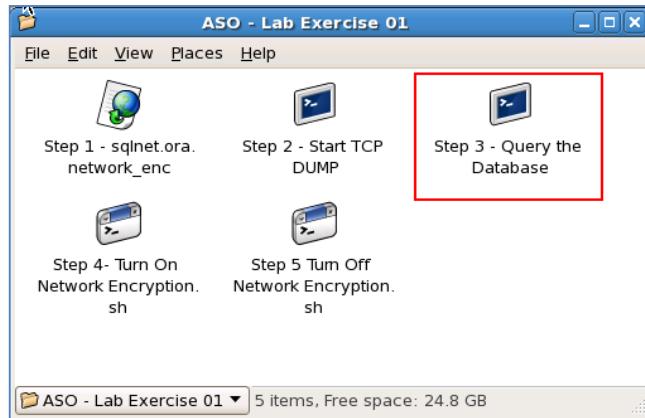
** Note that to execute this script, you must enter `' ./tcpdump.sh'` (again, no quotes) - that is 'dot, space, dot slash tcpdump.sh' **

This shell script runs the command `/usr/sbin/tcpdump -Xs 1518 -i lo port 1522`. This will allow you to see the network traffic on adapter `lo` from tcp port 1522.

```
[oracle@dbsecurity ~]$ su -
Password: <enter password>

-bash-3.1# /usr/sbin/tcpdump -Xs 1518 -i lo port 1522
--or--
-bash-3.1# . tcpdump.sh
tcpdump: verbose output suppressed, use -v or -vv for full protocol
decode
listening on lo, link-type EN10MB (Ethernet), capture size 1518 bytes
```

5. Click on the icon **Step2b – Query the Database**. In the opened window, set the alias to db01, connect to SQL Plus as **APPS_DBA_HARVEY/Manager_1@db06**. Query all data in the table CUSTOMER.



```
[oracle@dbsecurity ~]$ db06
ORACLE_SID=db06
ORACLE_HOSTNAME=cloud.oracle.com
ORACLE_BASE=/u01/oracle
ORACLE_HOME=/u01/oracle/product/11.2.0/dbhome_1
OH=/u01/oracle/product/11.2.0/dbhome_1
oracle@cloud.oracle.com:[/home/oracle]:DB06
$ sqlplus APPS_DBA_HARVEY/Manager_1@db06

SQL*Plus: Release 11.2.0.2.0 - Production on Wed Sep 21 22:26:30 2011

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - Production
With the Partitioning, Oracle Label Security, OLAP, Data Mining,
Oracle Database Vault and Real Application Testing options

SQL> select * from customer;

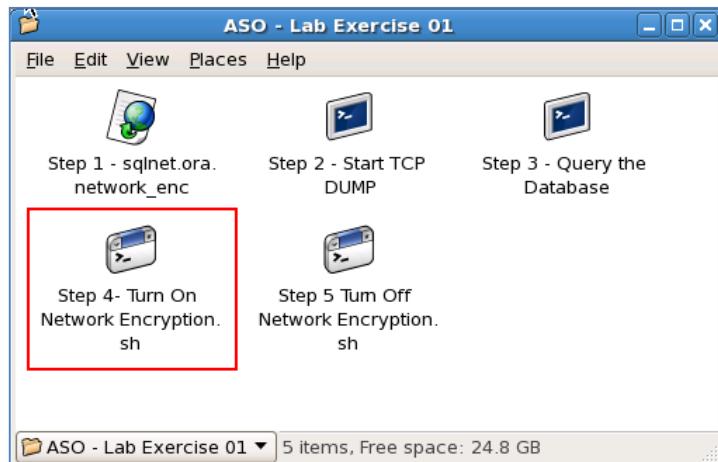
CUSTOMER_ID CUSTOMER_NAME          CUS CUSTOMER_CITY      CU
-----  -----
    101 HERTZ CORPORATION          LO   BERLIN           DE
    102 SUNGARD DATA SYSTEMS       GL   NEW YORK          US
    103 TEMASEK HOLDINGS          GL   SINGAPORE         SG
    104 NORDIC TELEPHONE          GL   STOCKHOLM        SE
    105 ORACLE CORPORATION         GL   REDWOOD SHORES  US
    106 QWEST COMMUNICATIONS      GL   DENVER            US
    107 OLD MUTUAL PRC             GL   LONDON            UK
    108 FRESENIUS MED CARE         GL   LONDON            UK
    109 EMI GERMANY CORPORATION    LO   FRANKFURT         DE
    110 DAIMLER                      GL   STUTTGART         DE

10 rows selected.
```

- Watch the window spooling the network traffic. Observe the unencrypted data for **CUSTOMER** being passed across the network. If it passes too quickly, type the '/' command in SQLPlus to repeat the last query. Your output will look something similar to the following data. Leave this window open. You will use this in a following step shortly.

```
14:54:48.913450 IP cloud.oracle.com.32800 > cloud.oracle.com.1522: . ack 4556
win 1134 <nop,nop,timestamp 505241 505201>          0x0050: 0000 0000 0000 0000
0000 0000 0000 0000 .....
0x0060: 072b 2c01 0503 c202 0314 5355 4e47 4152 .+.....SUNGAR
0x0070: 4420 4441 5441 2053 5953 5445 4d53 0247 D.DATA.SYSTEMS.G
0x0080: 4c08 4e45 5720 594f 524b 0255 5307 282c L.NEW.YORK.US.(
0x0090: 0105 03c2 0204 1054 454d 4153 454b 2048 .....TEMASEK.H
0x00a0: 4f4c 4449 4e47 5302 474c 0953 494e 4741 OLDINGS.GL.SINGA
0x00b0: 504f 5245 0253 4707 282c 0105 03c2 0205 PORE.SG.(,.....
0x00c0: 104e 4f52 4449 4320 5445 4c45 5048 4f4e .NORDIC.TELEPHON
0x00d0: 4502 474c 0953 544f 434b 484f 4c4d 0253 E.GL STOCKHOLM.S
0x00e0: 4507 2f2c 0105 03c2 0206 124f 5241 434c E./.....ORACL
0x00f0: 4520 434f 5250 4f52 4154 494f 4e02 474c E.CORPORATION.GL
0x0100: 0e52 4544 574f 4f44 2053 484f 5245 5302 .REDWOOD.SHORES.
0x0110: 5553 0729 2c01 0503 c202 0714 5157 4553 US.),.....QWES
0x0120: 5420 434f 4d4d 554e 4943 4154 494f 4e53 T.COMMUNICATIONS
0x0130: 0247 4c06 4445 4e56 4552 0255 5307 232c .GL.DENVER.US.#,
0x0140: 0105 03c2 0208 0e4f 4c44 204d 5554 5541 .....OLD.MUTUA
0x0150: 4c20 5052 4302 474c 064c 4f4e 444f 4e02 L.PRC.GL.LONDON.
0x0160: 554b 0727 2c01 0503 c202 0912 4652 4553 UK.,.....FRES
0x0170: 454e 4955 5320 4d45 4420 4341 5245 0247 ENIUS.MED.CARE.G
0x0180: 4c06 4c4f 4e44 4f4e 0255 4b07 2f2c 0105 L.LONDON.UK./...
0x0190: 03c2 020a 1745 4d49 2047 4552 4d41 4e59 ....EMI.GERMANY
0x01a0: 2043 4f52 504f 5241 5449 4f4e 024c 4f09 .CORPORATION.LO.
0x01b0: 4652 414e 4b46 5552 5402 4445 071f 2c01 FRANKFURT.DE...
0x01c0: 0503 c202 0b07 4441 494d 4c45 5202 474c .....DAIMLER.GL
0x01d0: 0953 5455 5454 4741 5254 0244 4504 0100 .STUTTGART.DE...
```

- Our next step will enable network encryption. We will use the pre-configured file that you previously reviewed in an earlier step. Click on the icon **Step 3 – Turn On Network Encryption.sh** to Run in Terminal. This script copies the file **sqlnet.ora.network_enc** to **\$ORACLE_HOME/network/admin** and renames the file as **sqlnet.ora**



8. Go back to the window you opened in Step 3 and exit out of SQLPlus.
We will start a new SQLPlus session, establishing an encrypted network session between SQLPlus and the database server and repeat the same query.

```
[oracle@dbsecurity ~]$ db06
ORACLE_SID=db06
ORACLE_HOSTNAME=cloud.oracle.com
ORACLE_BASE=/u01/oracle
ORACLE_HOME=/u01/oracle/product/11.2.0/dbhome_1
OH=/u01/oracle/product/11.2.0/dbhome_1
oracle@cloud.oracle.com:[/home/oracle]:DB06
$ sqlplus APPS_DBA_HARVEY/Manager_1@db06

SQL*Plus: Release 11.2.0.2.0 - Production on Wed Sep 21 22:26:30 2011

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - Production
With the Partitioning, Oracle Label Security, OLAP, Data Mining,
Oracle Database Vault and Real Application Testing options

SQL> select * from customer;
```

9. This time, observe that the query data is now being passed across the network in encrypted form. Here is some of the sample output. Enter CTRL-C to exit out of `tcpdump` when finished. Again, if the data passes too quickly, type the '`/`' command in SQLPlus to repeat the last query.

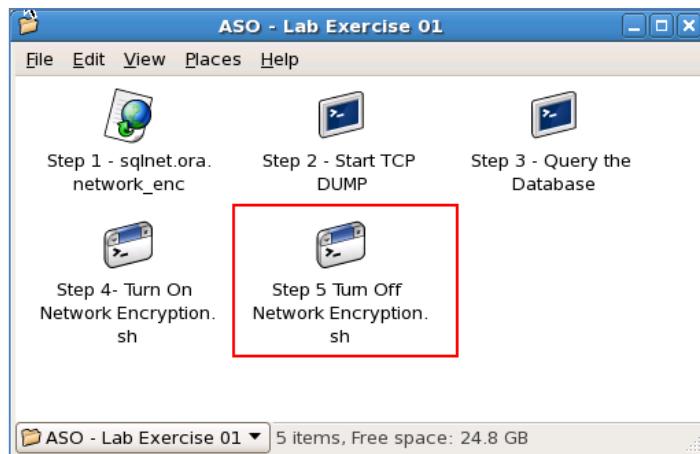
```
14:54:48.913450 IP cloud.oracle.com.32800 > cloud.oracle.com.1522: . ack 4556
win 1134 <nop,nop,timestamp 505241 505201>
0x0030: 00fc 90b6 023c 0000 0600 0000 0000 de8a ....<.....
0x0040: 94d5 88b4 3865 28f5 fb19 853b 4da4 5082 ....8e(...;M.P.
0x0050: bbe9 359a 9605 cc91 7ea6 d891 9163 63db ..5.....~....cc.
0x0060: 4582 2aed 2757 5c21 78ed b206 f1f7 1ed3 E.*.'W\!x.....
0x0070: ff49 9dd0 808e 8501 b8f0 ffad 8804 43d4 .I.....C.
0x0080: 60c0 8edb f105 21c3 b589 5e6c 7944 4dd9 `.....!...^lyDM.
0x0090: 5d6f 89d6 e4e1 51d0 c1be 3285 f506 797b lo....Q...2...y{
0x00a0: a9bb 3ee2 ab8b 7897 779e 3e4e 5ea5 c6fa ..>...x.w.>N^...
0x00b0: 933d 309e df48 0e68 2296 cd07 744d efe1 .=0..H.h"....tm..
0x00c0: 37cf 115e 894b 0755 fb57 3da8 c0db 0d94 7..^K.U.W=.....
0x00d0: 3da3 c5f3 2537 3e9b d79d 51b4 8971 baaf =....%7>...Q..q..
0x00e0: 4676 ee56 b6f3 e231 ce60 0530 7760 Fv.V..[...`0w`...
0x00f0: f511 a39d b470 08d2 1123 5026 3ebd c82e .....p...#P>...
0x0100: d97a 8eb8 63b9 9be6 6a7f 1a0d 87f7 3474 .z.c...j....4t
0x0110: 5ba3 8e25 fff9 b829 012c b15b 8b53 4536 [...%...),,[.SE6
0x0120: 7515 e468 62b2 b9c7 b18a a9a6 0faf 46cf u..hb.....F.
0x0130: c211 6185 5bfa 223a 2daf 3beb 4b19 c343 ..a.[.":-;K..C
0x0140: 226d a9c3 d216 0158 c6d3 a655 e736 430f "m.....X...U.6C.
0x0150: cb96 1b1d 1e56 ca92 d296 5e4b 264c 0c39 .....V....^K&L.9
0x0160: 444a 3743 b9ff 3107 a009 e2a0 1dea 33c7 DJ7C..1.....3.
0x0170: 3200 a9b7 155e 4f29 9ef5 941b 9b84 9c4a 2....^O).....J
0x0180: 4a90 6c7a ff3c 8efb 523d cd00 b871 5dd6 J..l.z.<..R=...q].
0x0190: 5ca2 8d1f 190e 0262 8e4e 4e52 a0ab cba4 \.....b.NNR...
0x01a0: dfd9 a20c a59c 7bal f82a 0c88 4c8b ebe6 .....{..*..L...
0x01b0: 94f6 c71a 593a cc35 be39 8714 324a 12f5 ....Y:.5.9..2J..
0x01c0: 40aa 1d66 50c8 474d 7a8d 1ff0 bd44 218b @..fp.GMz....D!.
0x01d0: fd60 cab3 e551 6763 ae56 583b 0215 d14e .`....Qgc.VX;...N
0x01e0: 25d5 6583 1a4a d6cc 578f 57cf cc16 e7fa %..e..J..W.W....
0x01f0: cd52 f794 1d8d 5449 2724 0e7a 27d0 3178 .R....TI'$..z'.1x
0x0200: 6a14 12a2 f461 f561 fa83 9604 8838 39eb j....a.a.....89.
0x0210: bb25 fad3 c61d 52d3 d6ef d790 65e2 9ed2 .%....R.....e...
```

```

0x0220: 0fe0 1279 6db6 63bd 71c9 7aa2 222d 7634 ...ym.c.q.z."-v4
0x0230: cf41 9b1b 8c5c 5bb0 10b0 988d ce22 63e2 .A...\[....."c.
0x0240: 51d8 ef3a dff9 d5db ebcd 1401 0339 c951 Q.:.....9.Q
0x0250: 1c4a 72d7 46b0 6774 85ee ff0d b9c6 8ebc .Jr.F.gt......
0x0260: c17a a4eb 160d 4f74 fe13 6115 6aef 0801 .z....Ot..a.j...

```

10. If you want to repeat the steps to turn off network encryption and see the unencrypted data again, click on the icon named **Step 3b – Turn Off Network Encryption.sh**. This sets the SQLNET.ORA file to its original state. Note that the encryption requirements of a connection are determined at connect time for that session – this is to say that you need to log out of SQLPlus and re-login to SQLPlus for the changes to take effect.



C. Additional Steps

1. As an additional step when running TCPDUMP, you can set the **-w** flag to capture the data to file. Here is an example. Alternatively, you can edit the provided **tcpdump.sh** file and uncomment/comment the appropriate commands you wish to use.
 - i. `/usr/sbin/tcpdump -Xs 1518 -i lo port 1522 -w /home/oracle/aso_scripts/tcpdump-aso-lab1.out`

D. Summary

You accomplished the following in this lab exercise:

1. Configured sqlnet.ora for supporting network encryption
2. Used tcpdump to demonstrate the before and after effect when the Advanced Networking Option (network encryption) is being used.

LAB EXERCISE 02 – CREATION OF THE ENCRYPTION WALLET

Identified Challenge – At Rest Data Encryption

Reduce the operational and performance overhead of deploying at rest data encryption in production.

Securely store and manage encryption keys for at rest data encryption.

INTRODUCTION

Key management is a critical part of any encryption scheme. Oracle Advanced Security uses a tiered key management infrastructure to provide maximum flexibility and security. These keys can be stored in a centralized Hardware Security Module (HSM) or stored in a secure wallet stored on a file system. Key management Oracle Advanced Security Transparent Data Encryption (TDE), first introduced in Oracle Database 10g Release 2, is the industry's most advanced encryption solution. TDE provides built-in key management and complete transparency for encryption of sensitive application data. The database encryption process is turned on using DDL commands, completely eliminating the need for application changes, programmatic key management, database triggers, and views.

Important Concepts

- **Master Key** – The encryption key used to encrypt secondary keys used for column encryption and tablespace encryption. Master keys are part of the Oracle Advanced Security two-tier key architecture.
- **Table Key** – Sometimes referred to as a Column Key, this key is used to encrypt one or more specific columns in a given table. Table keys were introduced in Oracle Database 10g Release 2. These keys are stored in the Oracle data dictionary, encrypted with the master key.
- **Tablespace Key** – The key used to encrypt a tablespace. These keys are encrypted using the master key and are stored in the tablespace header of the encrypted tablespace files.
- **Wallet** – A PKCS#12 formatted file outside of the database, encrypted using an administratively defined password.
- **Advanced Encryption Standard (AES)** – A symmetric cipher algorithm defined in the Federal Information Processing (FIPS) standard no. 197. AES provides 3 approved key lengths 128, 192 and 256 bits.
- **PKCS#12** – A file format standard published by RSA, used for storing cryptographic keys.

A. Overview

In this lab exercise, you will accomplish the following:

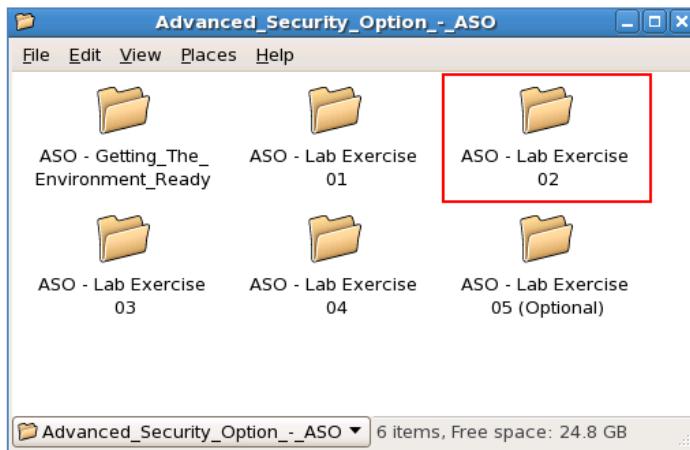
1. *Modify the SQLNET.ORA file to specify the wallet location*
2. *Create the TDE encryption wallet to be used.*

B. Setup & Preparation

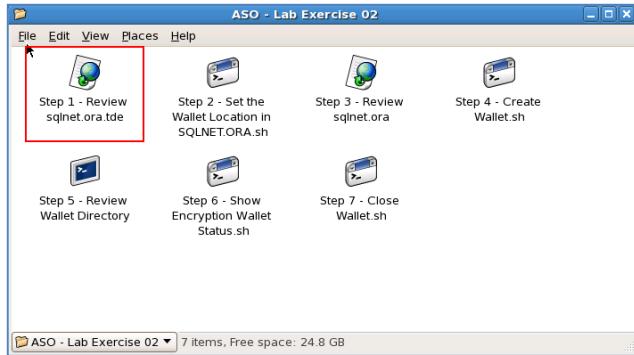
- All scripts used in this lab exercise can be found in the directory /home/oracle/aso_scripts.
- A directory named **/home/oracle/wallet** has already been created for you. This directory will be used to store the master encryption wallet file.
- Wallet definition and configuration is done by modifying the SQLNET.ORA file – we have made these modifications for you for these labs.

C. Creation and Configuration of the Encryption Wallet

1. Return to the **Advanced Security Option – ASO** directory and open the **ASO – Lab Exercise 02** folder.



2. Click on the icon Step 1 – Review **sqlnet.ora.tde**. In this file, notice the changes that we will be making to the **SQLNET.ORA** file.



Specifically, note the **ENCRYPTION_WALLET_LOCATION**. You will be implementing these changes in the next step.

3. Click on the icon **Step 2 – Set the Wallet Location in SQLNET.ORA.sh**. In this script, we are updating the SQLNET.ORA file with the changes that you just reviewed.



- Click on the icon **Step3 – Review sqtnet.ora** to review the updated file.
- Once the correct configuration parameter,
ENCRYPTION_WALLET_LOCATION is set, we are ready to proceed.



The following changes were added to the sqtnet.ora file:

```
ENCRYPTION_WALLET_LOCATION=
  (SOURCE= (METHOD=FILE) (METHOD_DATA=
    (DIRECTORY=/home/oracle/wallet)))
```

The **/home/oracle/wallet** directory has been already created for you on the file system.

The encrypted wallet ('ewallet.p12') offers strong protection of the master key, by encrypting the wallet with the wallet password.

The wallet is a container that is used to store authentication and signing credentials, including the TDE master key, PKI private keys, certificates, and trusted certificates needed by SSL. With TDE, wallets are used on the server to protect the TDE master key.

Oracle provides different types of encryption wallets: standard wallet and auto-open wallet. The standard wallet (filename ewallet.p12) is the one recommended for TDE and the one we will be using in this lab. It needs to be opened manually after database startup and prior to TDE encrypted data being accessed. If the Wallet is not opened, the database will return an error when TDE protected data is queried. The auto-open wallet (filename cwallet.sso) opens automatically when a database is started; hence it can be used for unattended Data Guard (Physical Standby only) environments where encrypted columns are shipped to secondary sites. The auto-wallet typically is copied to these secondary sites. A third type of wallet is “local” auto-open wallet. The local auto-open wallet is similar to auto-open wallet, except that it is tightly bound to the database and machine where it was created, preventing it from being copied and used elsewhere.

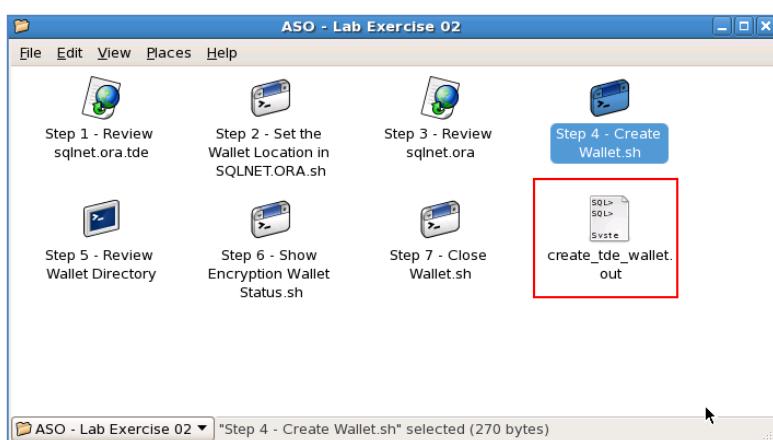
If this is the first time running through these labs, no encryption wallet exists. It is recommended that the wallet storage location on the server is distinct from any other files/directories that may be backed up as part of a backup regime. It is very important that the wallet is backed up, but it is best not to include it with any Oracle RDBMS backup.

Note that in large environments, key management can be centralized in a Hardware Security Module, or HSM. This is beyond the scope of this workshop, but the dynamics of key management, creation, and storage are similar.

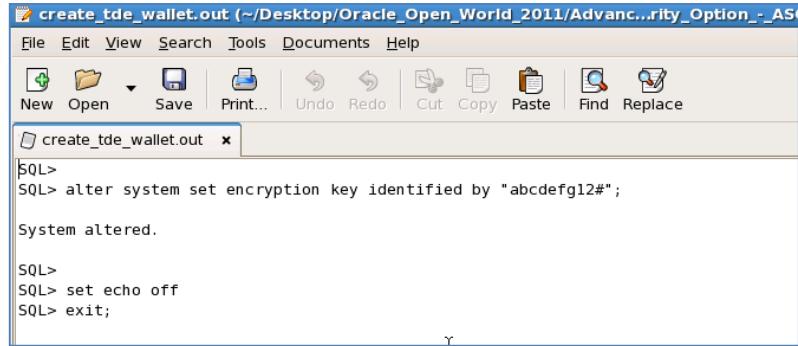
5. Click on the icon **Step 4 – Create Wallet.sh**. In this step we create the Master key so TDE can be used.



6. Click on the file **create_tde_wallet.out** to view the output of the scripts.



You will see that we have created a wallet secured by a passphrase.

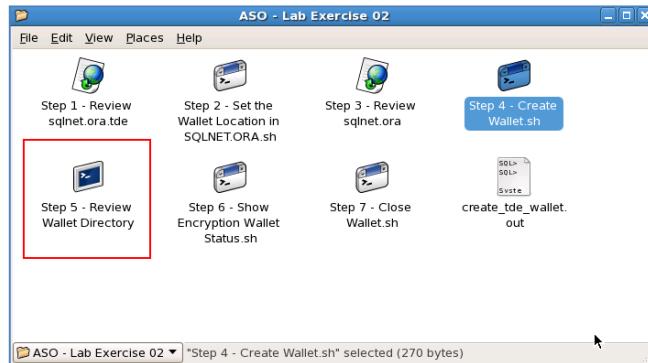


```
create_tde_wallet.out (~/Desktop/Oracle_Open_World_2011/Advanced_Option_ASO)
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
create_tde_wallet.out x
SQL>
SQL> alter system set encryption key identified by "abcdefg12#";
System altered.

SQL>
SQL> set echo off
SQL> exit;
```

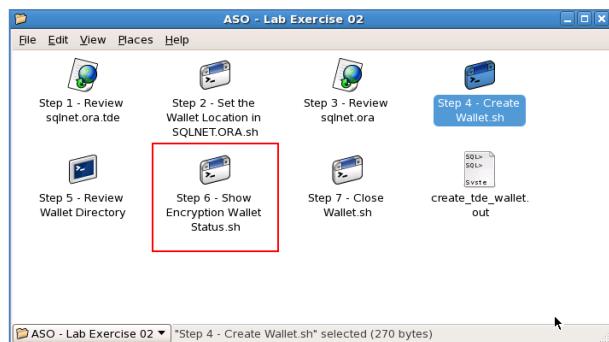
7. Click on the icon **Step 5 – Review Wallet Directory** to open up a terminal window so you can see how the wallet is stored in the file system.

The window will open in the **/HOME/ORACLE/WALLET** directory. Type the command **ls -l** to see the newly created wallet, named **ewallet.p12**



```
[oracle@dbsecurity wallet] ls -l
total 4
-rw-r--r-- 1 oracle dba 1837 Sep 21 05:20 ewallet.p12
[oracle@dbsecurity wallet]
```

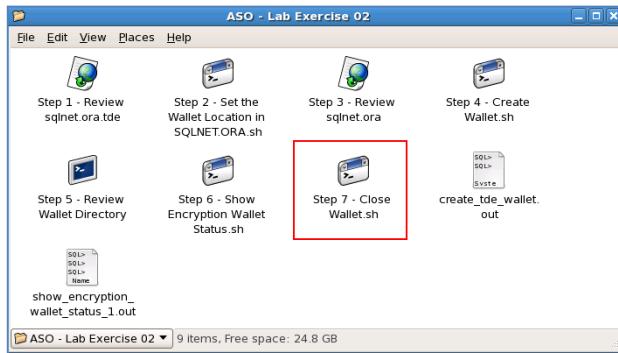
8. Click on **Step 6 0- Show Encryption Wallet Status'**



9. Open the file **show_encryption_wallet_status_1.out** to review the current status of the encryption wallet..



10. We will now close the wallet by executing '**Step 7 – Close Wallet.sh**'.



D. Summary

You accomplished the following in this lab exercise:

1. Modified the SQLNET.ORA file to specify the wallet location
2. Created the TDE encryption wallet to be used

LAB EXERCISE 03 – PROTECTING SENSITIVE DATA – TABLESPACE LEVEL ENCRYPTION

Identified Challenge – At Rest Data Encryption

All locations of sensitive information (i.e. Credit Card Information, PII) within database systems have not been fully identified and documented.

Reduce the management and performance overhead of deploying at rest data encryption in production.

INTRODUCTION

Column-level TDE has been available since Database 10g Release 2. This is still a very useful feature for applications in which the amount of data requiring encryption is small and well defined, but with Oracle Database 11g, new tablespaces can be defined as encrypted. Defining a tablespace as encrypted means the physical data files created on the operating system will be encrypted. Any tables, indexes and other objects defined in the new tablespace will be encrypted by default with no additional storage space requirements. During data reads, the Oracle database will automatically decrypt data before it arrives in database memory (SGA). Data that is moved out of the SGA and written to the file system will be encrypted. TDE tablespace encryption provides optimal performance by enabling existing indexes and foreign keys to continue working as they were before encryption was turned on. Execution plans remain the same and the requirement to identify individual columns to encrypt is completely eliminated.

Tablespace encryption in 11g is an attractive option for several reasons:

- The identification of all relevant columns with sensitive data has been difficult to evaluate for protecting PII (Personally Identifiable Information) and complying with the numerous regulations such as PCI and HIPAA to protect data.
- Less upfront analysis needs to be done to identify candidates (columns) for encryption. In 11g and tablespace encryption, only entire tablespaces need to be identified for encryption.
- Unlike with column-level TDE, no impact assessment needs to be made around columns used as indexes.
- Transparent encryption/decryption takes place during disk I/O and not for every logical access to the data. This leads to improved performance.

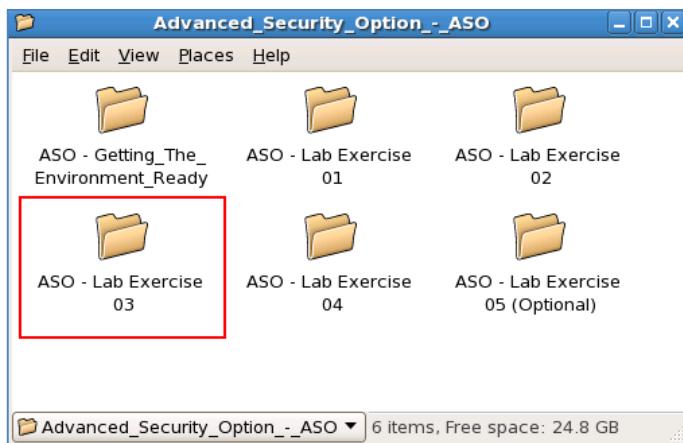
A. Overview

In this lab exercise, you will accomplish the following:

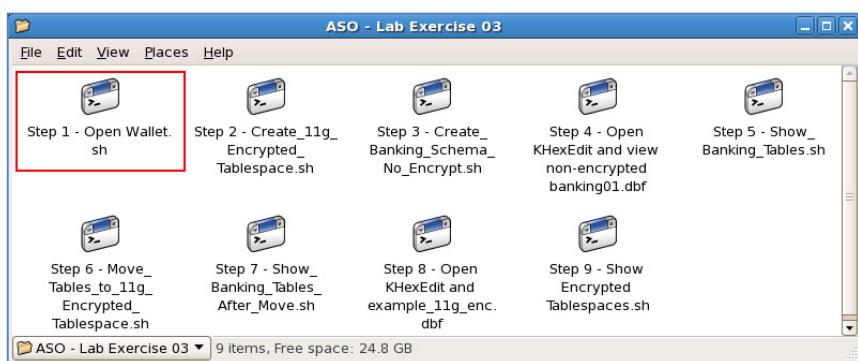
1. *Create a new encrypted tablespace.*
2. *Store application data in the encrypted tablespace.*
3. *Review the contents on disk in the table space and encrypted tablespace on disk.*

B. Setup & Preparation

- All scripts used in this lab exercise can be found in the directory `/home/oracle/aso_scripts`.
 - A directory named `/home/oracle/wallet` has already been created for you. This directory will be used to store the master encryption wallet.
1. Return to the **Advanced Security Option – ASO** directory and open the **ASO – Lab Exercise 03** folder.

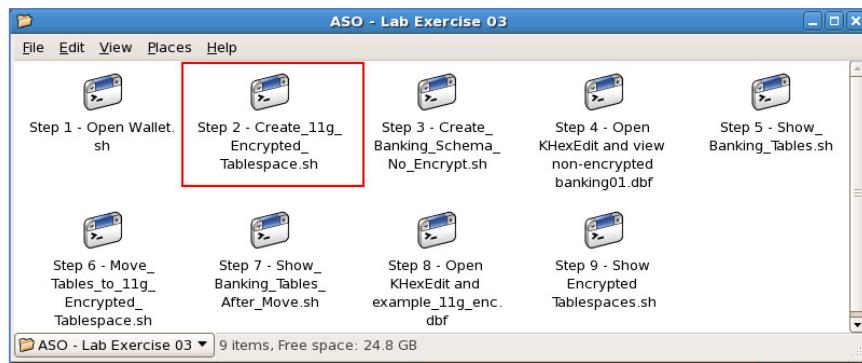


2. Click on '**Step 1 – Open Wallet.sh**' to open the wallet and enable transparent data encryption.

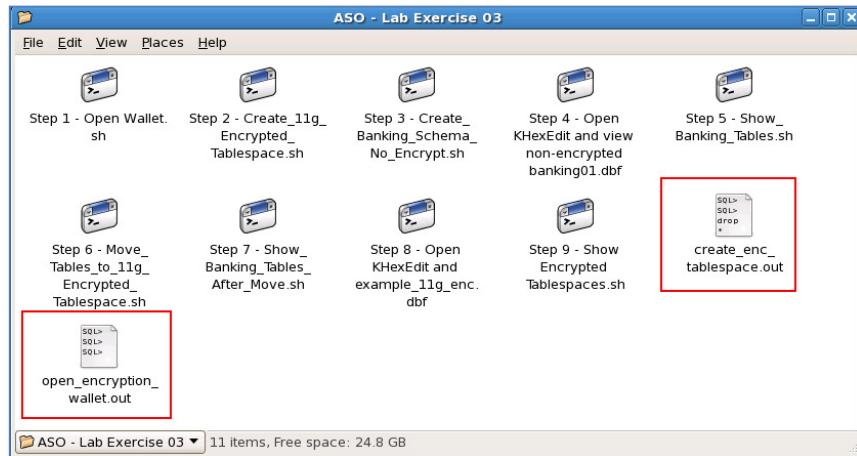


- To setup the encrypted tablespace and tables for the exercise, click on the icon, **Step 2 – Create_11g_Encrypted_Tablespace.sh**. When you review the script, you will notice that we are creating the necessary wallet file and creating a new encrypted tablespace named **example_11g_enc_tablespace** which we will move data into. Oracle Database 11g supports encrypting new tablespaces only.

```
create tablespace example_11g_enc_tablespace
datafile '/u01/oracle/oradata/db06/example_11g_enc.dbf'
size 50m
encryption using 'AES192'
default storage(encrypt)
```



- Review the **open_encryption_wallet.out** and **create_enc_tablespace.out** file for the output. You should see that the result of creating the wallet and the result of 'Tablespace created.'



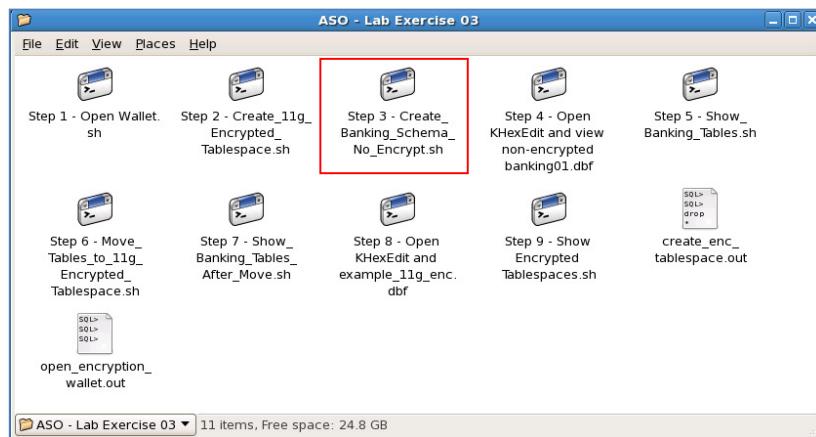
The following changes were added to the sqlnet.ora file:

```
ENCRYPTION_WALLET_LOCATION=
(SOURCE=(METHOD=FILE) (METHOD_DATA=
(DIRECTORY=/home/oracle/wallet) ))
```

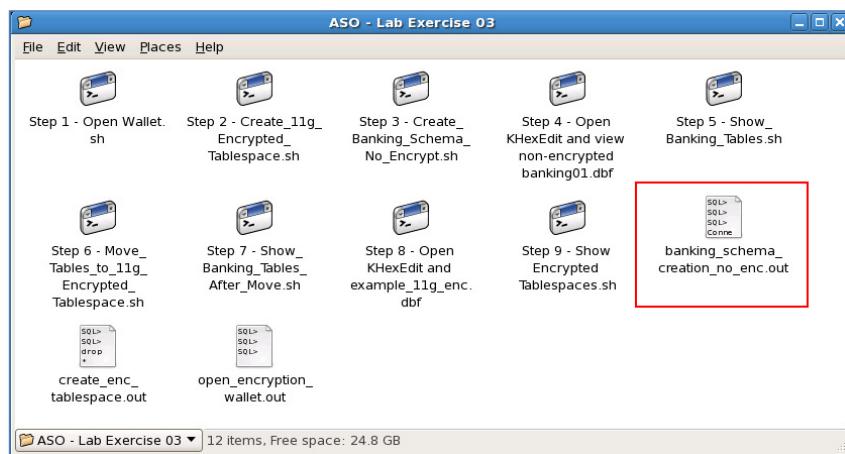
The `/home/oracle/wallet` directory has been already created for you on the file system.

The encrypted wallet ('ewallet.p12') offers strong protection of the master key, by encrypting the wallet with the wallet password. The wallet is a container that is used to store authentication and signing credentials, including the TDE master key, PKI private keys, certificates, and trusted certificates needed by SSL. With TDE, wallets are used on the server to protect the TDE master key.

5. To setup the proper unencrypted tablespaces and tables that we will copy into the encrypted tablespace, click on the icon, **Step 3 – Create_Banking_Schema_No_Encrypt.sh**. When you review the script, you will notice that we are creating two tablespaces, **banking01** and **banking02** and sample data to be used throughout the exercise.



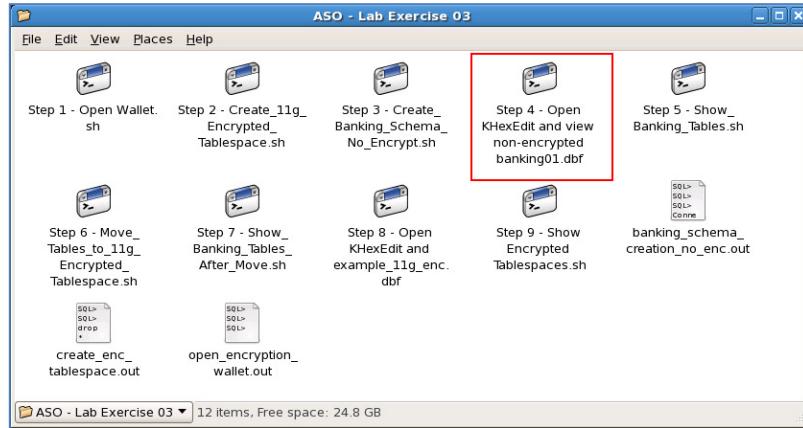
6. Review the **create_banking_schema.out** file for the output.



You have now completed the setup and configuration of the labs.

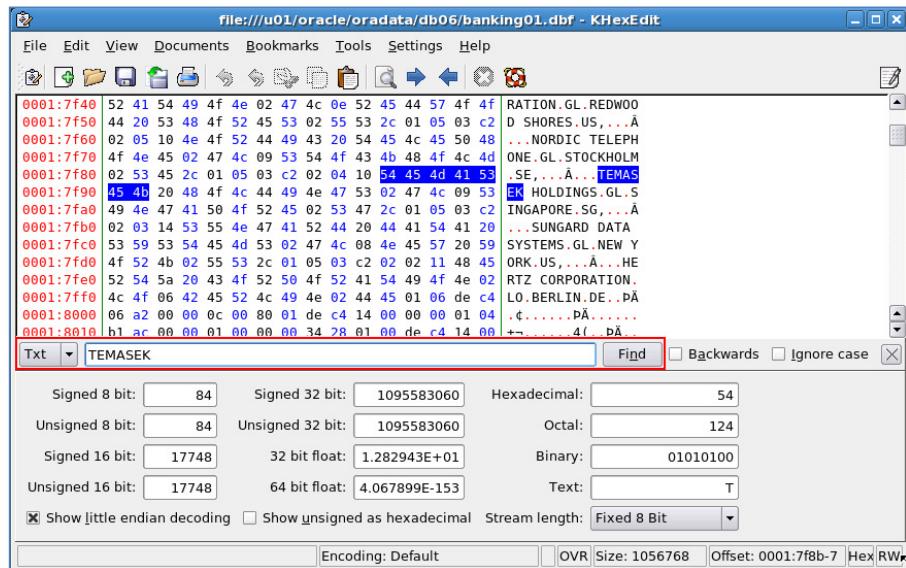
C. Masking Sensitive Data – Tablespace Level Encryption

- The next step in this exercise will be to view the banking01.dbf file using KHexEdit. Click on **Step 3 – Open KHexEdit and view non-encrypted banking01.dbf**. We are going to look at the .dbf file on disk before we proceed to the step of moving data into the encrypted tablespace.



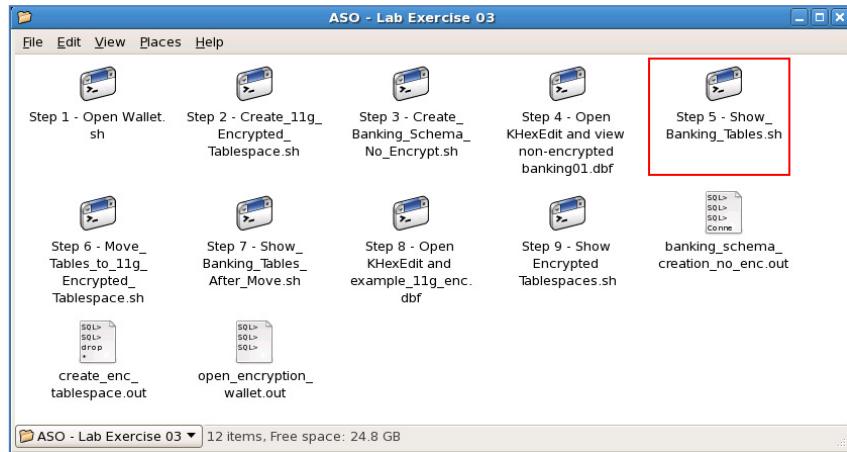
- In KHexEdit, we will search for a value in the **BANKING.CUSTOMER** table that we know is part of the table data being stored in the database. Select the **Txt** option in the drop-down list box, type in the value **TEMASEK** (stored in the **CUSTOMER_NAME** column) in the search entry field and click on the **Find** button.

Note that this is case sensitive, and make sure to select the **Txt** option.

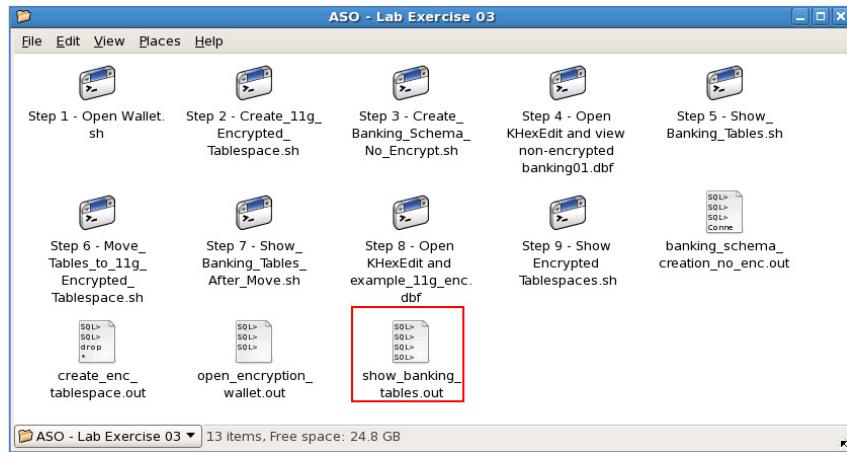


**** Notice that you were able to see all the data written to disk.**

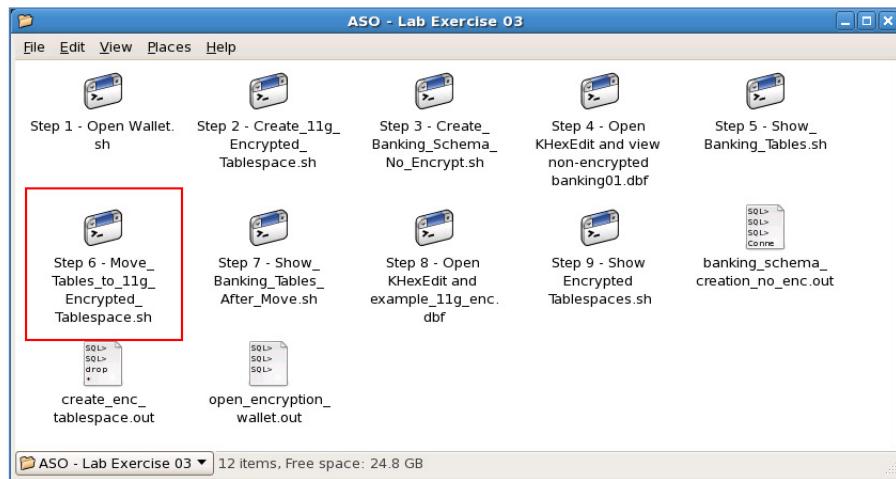
3. Let's review some of the details associated with the **BANKING** tables by querying the **DBA_TABLES** and **DBA_TABLESPACES** tables. Click on the icon **Step 5 – Show_Banking_Tables.sh**, review and run the provided script.



4. Review the **show_banking_tables.out** file for the output. Since we're reviewing the **BANKING01** tablespace, there are no encrypted tablespaces—as expected.

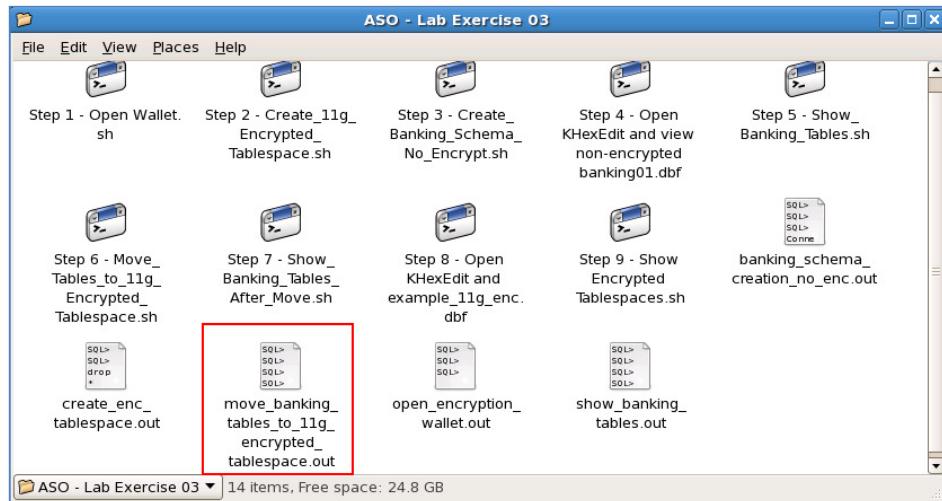


5. We will now copy the tables from the unencrypted **BANKING01** tablespace to the encrypted tablespace we created earlier named **example_11g_enc_tablespace**. Click on the icon **Step 6 – Move_Tables_to_11g_Encrypted_Tablespace.sh**, review and run the provided script. We are using the **ALTER TABLE... MOVE** command to complete this operation in this example. Alternatively, the powerful feature of Online Table Redefinition could have been used to allow all read and write operations during this process.

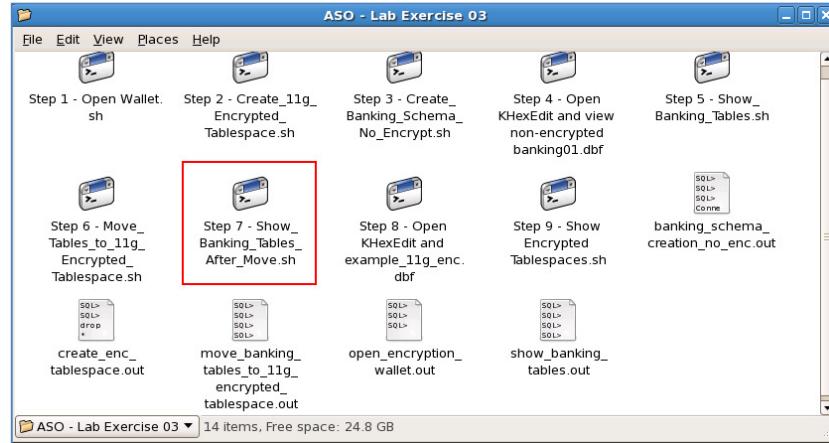


6. Review the file

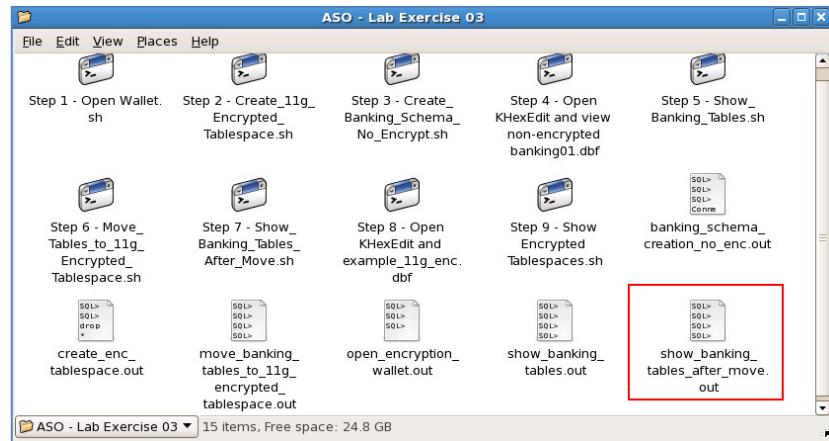
Move_banking_tables_to_11g_encrypted_tablespace.out and review the output.



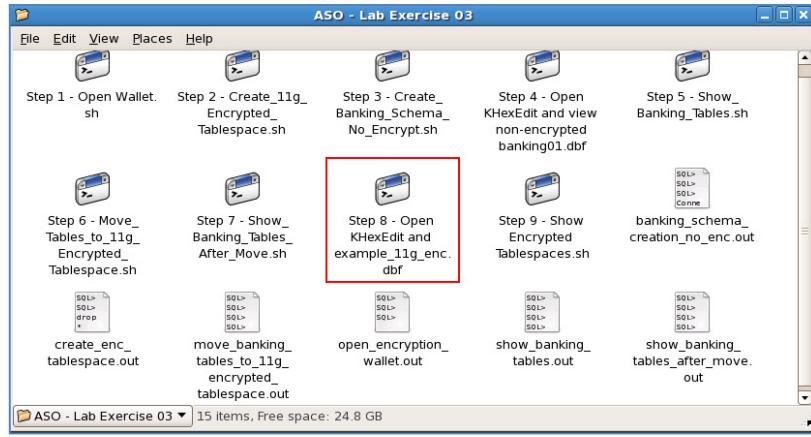
Let's again review some of the details associated with the **BANKING** tables by querying the **DBA_TABLES** and **DBA_TABLESPACES** tables after we have moved the tables from the unencrypted tablespace to the encrypted tablespace. Click on the icon **Step 7 – Show_Banking_Tables_After_Move.sh**, review and run the provided script.



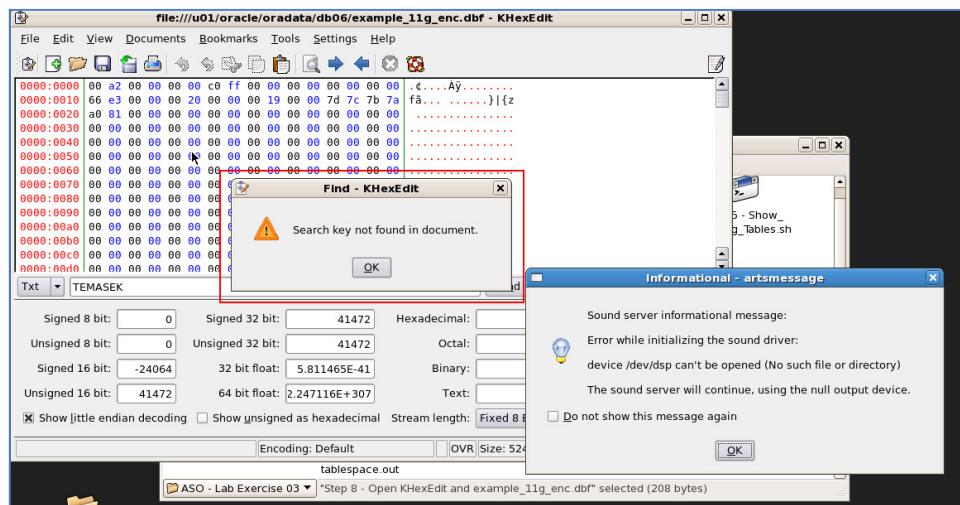
7. Review the **show_banking_tables_after_move.out** file for the output. Notice that the BANKING tables are now all showing as encrypted.



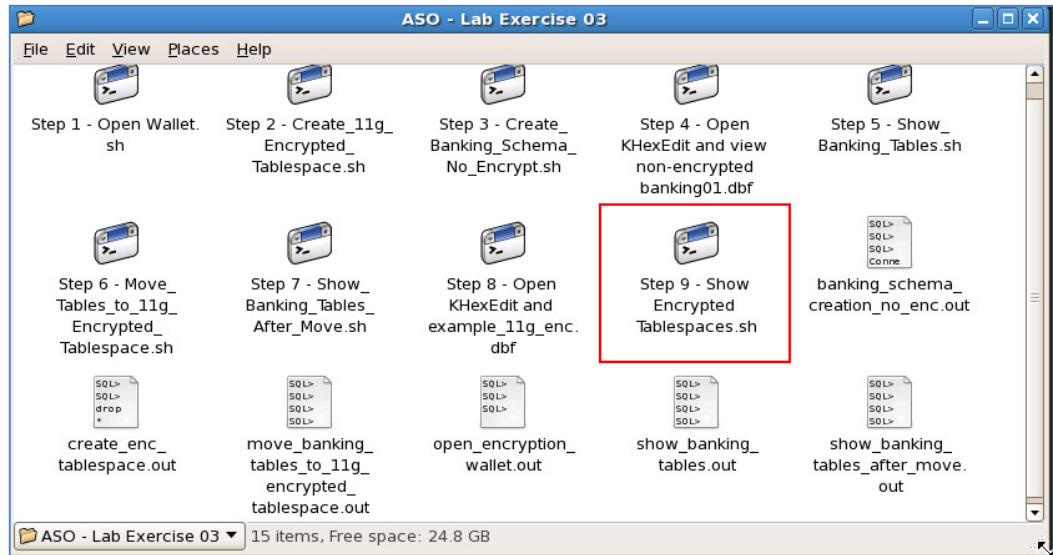
8. As the final step in the exercise, we will view the **example_11g_enc.dbf** file on disk using KHexEdit. Click on **Step 7 – Open KHexEdit and example_11g_enc.dbf**. We are going to look at the contents of the encrypted .dbf file on disk.



9. In KHexEdit Click on **Step 7 – Open KHexEdit and example_11g_enc.dbf**. We are going to look at the contents of the encrypted .dbf file on disk. Search again for the name **TEMASEK** in the **CUSTOMER_NAME** field and other values. You will notice that no search key (value) will be found in this encrypted file. Note that you may get a sound driver error due to “missing’ hardware on this VM. This can be ignored.



10. As a final step, we will execute some queries against the **V\$ENCRYPTED_TABLESPACES** and **DBA_TABLES** to review some information regarding the encrypted tablespaces that we created in this lab. Click on the icon, **Step 9– Show Encrypted Tablespaces.sh**, review and execute.



11. Open up the output file **show_encrypted_tablespaces.out**. This output will show the description of the **V\$ENCRYPTED_TABLESPACES** and the results of the queries executed. These queries can be useful to monitor and validate the encryption controls that have been established and implemented. The first query provides the name (name) and the file location (file_name) of the encrypted tablespace and the Algorithm (encryptionalg) used. The second query provides the tablespaces' owner (owner), table name (table_name) and Algorithm (encryptionalg) used



D. Summary

You accomplished the following in this lab exercise:

1. Created a new encrypted tablespace
2. Stored application data in the encrypted tablespace.
3. Reviewed the contents on disk in the table space and encrypted tablespace on disk.

LAB EXERCISE 04 – Wallet Management Operations

Identified Challenge – Proper Management of Encryption Keys and Wallets to Ensure Data Protection

Limited reporting and visibility to demonstrate that encryption controls are being implemented and enforced to identify sensitive data. Need to document and implement all key management processes and procedures used for encryption.

INTRODUCTION

Keys management is the heart of any encryption scheme. Encryption keys are values used in combination with an encryption algorithm to encrypt data. Oracle Advanced Security TDE uses a two-tiered encryption key architecture, consisting of a master key and one or more table and/or tablespace keys. The table and tablespace keys are encrypted using the master key. The master key is stored in an external security module (ESM) with the following options: An Oracle Wallet, Hardware Security Module (HSM), or external PKCS#11 compatible key management system.

Important Concepts

- **Master Key** – The encryption key used to encrypt secondary keys used for column encryption and tablespace encryption. Master keys are part of the Oracle Advanced Security two-tier key architecture.
- **Table Key** – Sometimes referred to as a Column Key, this key is used to encrypt one or more specific columns in a given table. Table keys were introduced in Oracle Database 10g Release 2. These keys are stored in the Oracle data dictionary, encrypted with the master key.
- **Tablespace Key** – The key used to encrypt a tablespace. These keys are encrypted using the master key and are stored in the tablespace header of the encrypted tablespace files.
- **Wallet** – A PKCS#12 formatted file outside of the database, encrypted using an administratively defined password.
- **Advanced Encryption Standard (AES)** – A symmetric cipher algorithm defined in the Federal Information Processing (FIPS) standard no. 197. AES provides 3 approved key lengths 128, 192 and 256 bits.
- **PKCS#12** – A file format standard published by RSA, used for storing cryptographic keys.

Key Generation and Backup

If the TDE master key is stored in an Oracle wallet, it is generated by Oracle during the initial configuration of TDE. You performed this step in Lab Exercise 02. The master key is generated using a pseudo-random number generator inside the Oracle database. If an HSM device is used to store the master key, the HSM device itself creates it.

From a best practices perspective, you want to backup the wallet associated with the master key immediately after it is initially created and whenever the master key is changed. The wallet is a critical component and should be backed up in a secure location, on-site and offsite. If you are using an HSM device, follow the manufacturer's instructions for insuring the recoverability of keys in case the HSM fails. This may involve the secure export of keys out of the HSM and/or authentication of multiple administrators for key recovery.

Key Storage

The TDE master key is stored in an security module external to the database: Either an Oracle wallet, HSM device, or external PKCS#11 compatible key management system. External security module support is dependent on your version of Oracle.

A. Overview

In this lab exercise, you will perform the following:

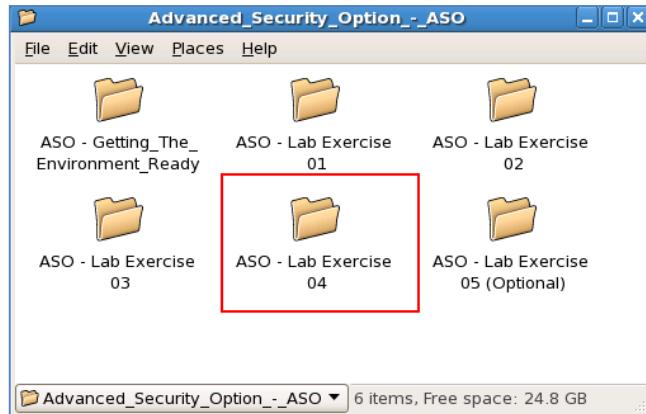
1. *Open Wallet and show the encryption wallet status*
2. *Perform a rekey of the master key*
3. *Review the Wallet Keys*

B. Setup & Preparation

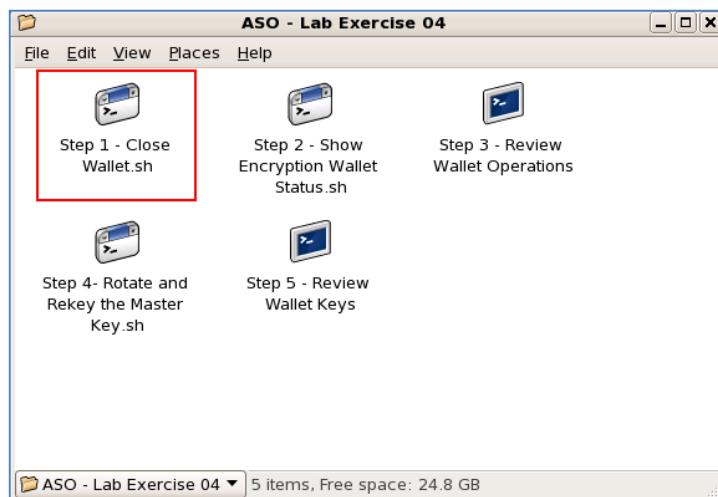
- All scripts used in this lab exercise can be found in the directory **/home/oracle/aso_scripts**.
- A directory named **/home/oracle/wallet** has already been created for you. This directory will be used to store the master encryption wallet file. If there is already a **ewallet.p12** file, *do not* delete this file.

C. REVIEW OF WALLET OPERATIONS AND ENCRYPTION CONTROLS

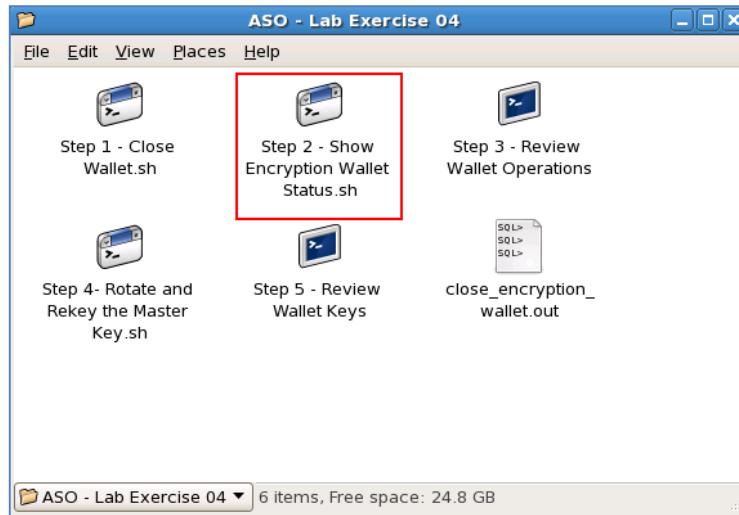
1. Return to the **Advanced Security Option – ASO** folder, and open the **ASO Lab Exercise 04** folder.



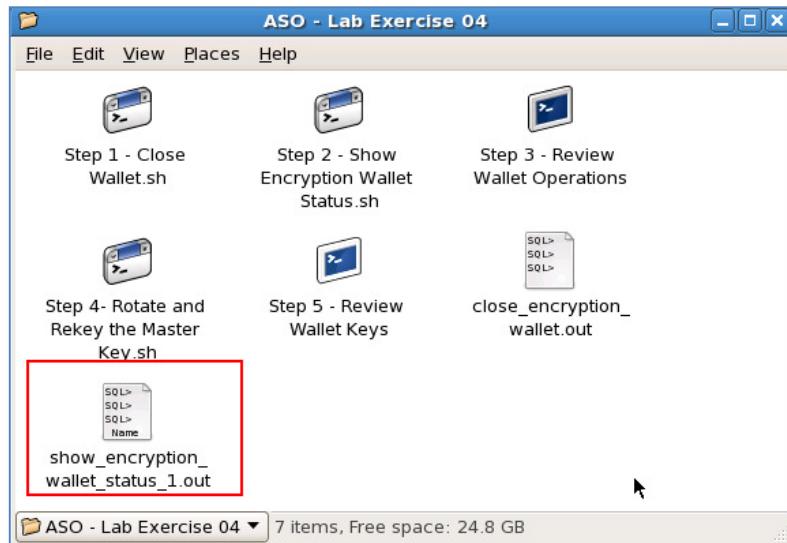
2. Click on the **Step 1 – Close Wallet.sh** to close the encryption wallet.



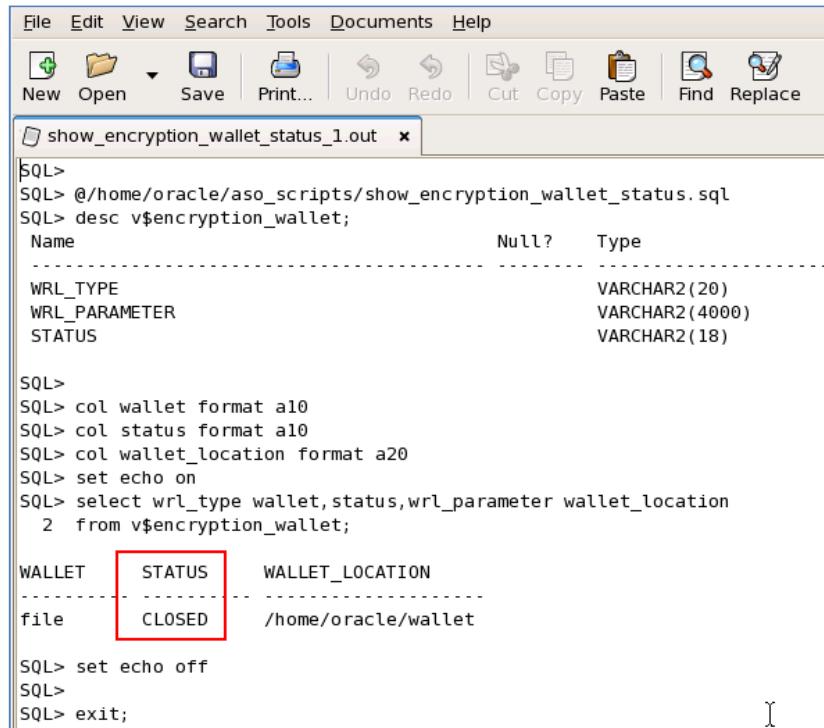
3. Click on the icon **Step 2 – Show Encryption Wallet Status.sh**. In the script, we show the current encryption status and describe the **V\$ENCRYPTION_WALLET** table.



4. Click on the icon, **show_encryption_wallet_status_1.out** to review the output.



5. As you can view from the output, our wallet is **FILE** based (WRL_TYPE), the status is **CLOSED** (STATUS) and the wallet location (WRL_PARAMETER) is located in the **/home/oracle/wallet** directory that we previously configured in the SQLNET.ORA file. Querying the **V\$ENCRYPTION_WALLET** is one way to confirm the proper configuration of the environment.



```

File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
show_encryption_wallet_status_1.out x
SQL>
SQL> @/home/oracle/aso_scripts/show_encryption_wallet_status.sql
SQL> desc v$encryption_wallet;
Name Null? Type
-----
WRL_TYPE          VARCHAR2(20)
WRL_PARAMETER     VARCHAR2(4000)
STATUS            VARCHAR2(18)

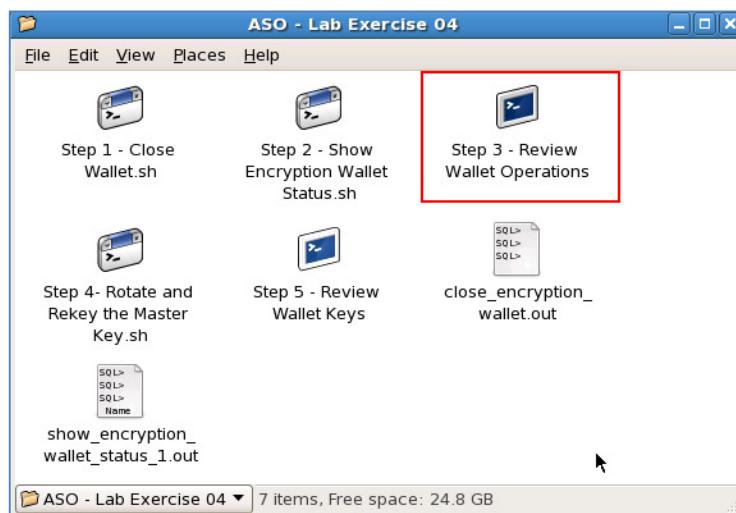
SQL>
SQL> col wallet format a10
SQL> col status format a10
SQL> col wallet_location format a20
SQL> set echo on
SQL> select wrl_type wallet,status,wrl_parameter wallet_location
2  from v$encryption_wallet;

WALLET      STATUS      WALLET_LOCATION
-----      -----
file        CLOSED      /home/oracle/wallet

SQL> set echo off
SQL>
SQL> exit;

```

6. We will now review some wallet management function from the command line. Click on the icon, **Step 3 – Wallet Operations** to open up a terminal window.



7. In the terminal window, set the alias to DB06 and then login to SQLPlus using the user **user_barack/Manager_1**. As user user_barack who has the (GRANT SELECT ANY TABLE TO USER_BARACK) privilege, try and issuing the sql statement, 'SELECT * FROM CUSTOMER'. You will see that the wallet is not open.

```
$ db06
ORACLE_SID=db06
ORACLE_HOSTNAME=cloud.oracle.com
ORACLE_BASE=/u01/oracle
ORACLE_HOME=/u01/oracle/product/11.2.0/dbhome_1
OH=/u01/oracle/product/11.2.0/dbhome_1
oracle@cloud.oracle.com:[/home/oracle/Desktop/Oracle_Open_World_2011/Advanced_Security_Option_-_ASO/ASO - Lab Exercise 04]:DB06
$ sqlplus user_barack/Manager_1

SQL*Plus: Release 11.2.0.2.0 Production on Mon Sep 26 00:02:27 2011

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - Production
With the Partitioning, Oracle Label Security, OLAP, Data Mining,
Oracle Database Vault and Real Application Testing options

SQL> select * from banking.customer;
select * from banking.customer
*
ERROR at line 1:
ORA-28365: wallet is not open
```

Since the wallet is not open, any attempt to access encrypted data will fail. As your current, non-administrative user, attempt to open the wallet with the appropriate statement, 'ALTER SYSTEM SET ENCRYPTION WALLET OPEN IDENTIFIED BY "abcdefg12#". You will notice that the user user_barack does not have the sufficient privileges (the "ALTER SYSTEM" privilege is required) to open the wallet even though he had the wallet password. The wallet password should be treated with the same best practices as any other password, but we wanted to illustrate the point that even if the user user_barack had the compromised password, he still did not have the permission necessary to open up the wallet. A separate individual with the appropriate permissions will need to open up the wallet before normal operations can proceed.

```
SQL> alter system set encryption wallet open identified by
"abcdefg12#";
alter system set encryption wallet open identified by "abcdefg12#"
*
ERROR at line 1:
ORA-01031: insufficient privileges
```

8. We will now login with the user and credentials of **INFOSEC_ISABEL/Manager_1** to open up the wallet.

```
SQL> connect INFOSEC_ISABEL/Manager_1
Connected.
SQL> alter system set encryption wallet open identified by
"abcdefg12#";
System altered.
```

Additional separation of duty can be established in your environment. The DBA might be able to restart the database, but if the wallet is closed and not set to the optional Auto-open mode, the database could require the wallet to be opened by a 'Security DBA', who must know the wallet password for normal operations continue.

9. We will now login back the user and credentials of **user_barack/Manager_1** to query the data. Once the wallet is opened successfully, the query can be completed successfully as well.

```
SQL> conn user_barack/Manager_1
Connected.
SQL> select * from banking.customer;

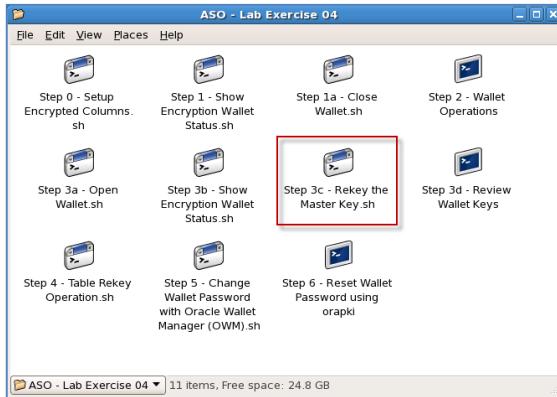
CUSTOMER_ID CUSTOMER_NAME          CUS CUSTOMER_CITY      CU
-----  -----
    101 HERTZ CORPORATION        LO  BERLIN           DE
    102 SUNGARD DATA SYSTEMS     GL  NEW YORK          US
    103 TEMASEK HOLDINGS        GL  SINGAPORE         SG
    104 NORDIC TELEPHONE       GL  STOCKHOLM        SE
    105 ORACLE CORPORATION      GL  REDWOOD SHORES   US
    106 QWEST COMMUNICATIONS   GL  DENVER            US
    107 OLD MUTUAL PRC          GL  LONDON             UK
    108 FRESENIUS MED CARE     GL  LONDON             UK
    109 EMI GERMANY CORPORATION LO  FRANKFURT         DE
    110 DAIMLER                  GL  STUTTGART         DE

10 rows selected.
```

10. As a final step as user **user_barack**, attempt to close the wallet with the appropriate statement, '**ALTER SYSTEM SET ENCRYPTION WALLET CLOSE;**'. As expected, **user_barack** will not have the ability to shut off encryption in the database due to insufficient privileges accidentally or intentionally. Login as **system/oracle1** and close the wallet.

```
SQL> alter system set encryption wallet close identified by
"abcdefg12#";
alter system set encryption wallet close
*
ERROR at line 1:
ORA-01031: insufficient privileges
```

- At this point, the wallet should be open. We will proceed to reset (rekey) the unified master encryption key. Click on the icon **Step 3c – Rekey the Master Key.sh**. Upon reviewing the script, you will notice that we take a backup copy of the wallet file as suggested in the documentation and best practices TDE White Paper.

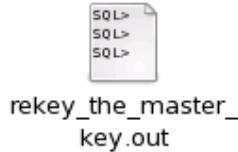


In TDE column encryption, both the master key and table keys can be individually re-keyed, providing for a granular implementation of various security policies. Re-keying of the master key does not impact performance or availability of your application, because it requires only decryption and encryption of the existing associated table keys and not the associated encrypted application data. Re-keying the table keys requires careful planning, since associated application data must first be decrypted and subsequently re-encrypted using the new table encryption key.

A unified master encryption key is used for both Transparent Data Encryption (TDE) Column Encryption and TDE Tablespace Encryption. The unified master encryption key can optionally be stored in a hardware security module. This enables you to use the TDE Tablespace Encryption feature along with hardware security modules. Lastly, as demonstrated in this step, you can reset (rekey) the unified master encryption key. This provides enhanced security and helps meet security and compliance requirements.

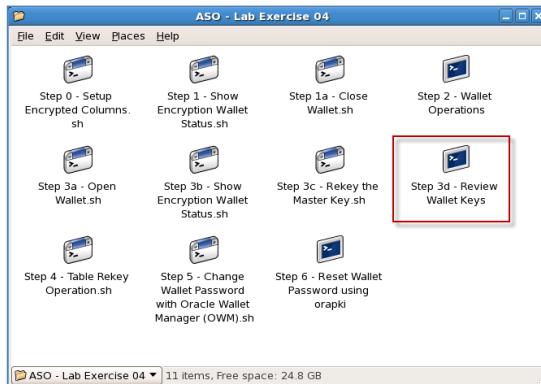
12. Click on output file **rekey_the_master_key.out** file to review the result of the command:

```
alter system set encryption key identified by
"abcdefg12#"
```



Notice that the passphrase for the wallet (abcdefg12#) is unchanged. Oracle Advanced Security will use the same passphrase, but internally has updated the master key in the wallet. This tiered key management allows for maximum flexibility – passphrases, master keys and column/tablespace keys are all managed independently,

13. Click on the icon, **Step 3d – Review Wallet Keys**. We will be showing the utility `mkstore`. The output and the comparison of the output can be used to show that the masterkey has successfully been reset (rekeyed) in addition to the output of the script upon execution.



You have been placed in the location of `/home/oracle/wallet`. In the terminal window, type the commands:

```
[oracle@dbsecurity wallet]$ db06
ORACLE_SID=db06
ORACLE_HOSTNAME=cloud.oracle.com
ORACLE_BASE=/u01/oracle
ORACLE_HOME=/u01/oracle/product/11.2.0/dbhome_1
OH=/u01/oracle/product/11.2.0/dbhome_1
oracle@cloud.oracle.com:[/home/oracle/wallet]:DB06
$ mkstore -wrl . -list
Oracle Secret Store Tool : Version 11.2.0.2.0 - Production
Copyright (c) 2004, 2010, Oracle and/or its affiliates. All rights reserved.

Enter wallet password: abcdefg12#
Oracle Secret Store entries:
```

```
ORACLE SECURITY DB ENCRYPTION AbAicYYyw0+kv+iVDcfAngcAAAAAAAAAAAAAAA  
AAAAAAAAAAAA  
ORACLE SECURITY DB ENCRYPTION AQULACbpOE+Jvyux22eTJwUAAAAAAAAAAAAAAA  
AAAAAAAAAAAA  
ORACLE SECURITY DB ENCRYPTION ARbDNGKasE/Iv1FFKSaoNQAAAAAAAAAAAAAAA  
AAAAAAAAAAA  
ORACLE SECURITY DB ENCRYPTION ASVA6VrUA0/fv9/uZPDssDIAAAAAAAAAAAAAAAA  
AAAAAAAAAAA  
ORACLE SECURITY DB ENCRYPTION ASZCukjal08pv3H4g/17hYcAAAAAAAAAAAAAAA  
AAAAAAAAAAA  
ORACLE SECURITY DB ENCRYPTION ATylG8aVQU+/v0TNbpTsj4AAAAAAAAAAAAAAA  
AAAAAAAAAAA  
ORACLE SECURITY DB ENCRYPTION AV0lHUYy4E8jv+uxCFuSGPwAAAAAAAAAAAAAAA  
AAAAAAAAAAA  
ORACLE SECURITY DB ENCRYPTION MASTERKEY  
ORACLE SECURITY TS ENCRYPTION BUGIRWW7p8BVYAtcz55bqZcCAwAAAAAAAAAAAAAAA  
AAAAAAAAAAA  
oracle@cloud.oracle.com: [/home/oracle/wallet]:DB06  
$
```

As an additional step, return back to **Step 3c – Rekey the Master**

Key.sh. Run that script again and then this current step. You should see an additional entry.

D. Additional Steps

1. Review the additional best practices outlined in the Oracle White Paper – Transparent Data Encryption Best Practices.

E. Summary

You accomplished the following in this lab exercise:

1. Opened wallet and showed the encryption wallet status
2. Performed a rekey of the master key
3. Reviewed the wallet keys

LAB EXERCISE 05 – PROTECTING SENSITIVE DATA AND OPTIMIZING STORAGE WITH ADVANCED COMPRESSION ON DISK & BACKUPS (OPTIONAL)

Identified Challenge –

Data at rest (on disk and backup) is vulnerable to potential breach and potential exposure to global disclosure regulation and must be encrypted.

Optimize the utilization of costly resources including storage and backup infrastructure while not compromising the ability to encrypt data.

Compression and Encryption can be challenging to implement together.

INTRODUCTION

As part of Oracle Advanced Security, we have the ability to protect sensitive data over the network by using Network Encryption and protecting sensitive data at rest (on disk) by using both column-based encryption (10g) and tablespace encryption (11g). One very important consideration that should not be overlooked is ensuring the same levels of protection, to protect sensitive information by means of encryption, goes beyond just network and disk to include backups of portions or the entire contents of the database.

In addition, with the continued storage growth and associated disk costs of database systems, combined with the need to protect sensitive information by using encryption, organizations must also be able to take advantage of compression technologies and use these technologies in combination-- transparently and without additional overhead.

As stated earlier, in Oracle Database 11g, new tablespaces can be defined as encrypted. Defining a tablespace as encrypted means the physical data files created on the operating system will be encrypted. Any tables, indexes and other objects defined in the new tablespace will be encrypted by default with no additional storage space requirements.

Additionally, Oracle Database 11g Advanced Compression Option introduces a comprehensive set of compression capabilities to help maximize resource utilization and reduce costs. It allows IT administrators to significantly reduce their overall database storage footprint by enabling compression for all types of data – be it relational (table), unstructured (file), or backup data.

Both Oracle Database 11g Tablespace Encryption and Advanced Compression can be used in together to achieve two important priorities within IT—protecting sensitive information by using encryption and reducing growing storage costs.

Oracle Data Pump was a new feature introduced in Oracle Database 10g that provides high speed, parallel, bulk data and metadata movement of Oracle database contents. Export (expdp) and Import (impdp) clients use the public interface PL/SQL package, DBMS_DATAPUMP available. In the 11g version of Data Pump, you now have the option of encrypting and compressing data.

A complete high availability and disaster recovery strategy requires dependable data backup, restore, and recovery procedures. Oracle Recovery Manager (RMAN) provides a comprehensive foundation for efficiently backing up and recovering the Oracle database. It is designed to work intimately with the server, providing block-level corruption detection during backup and restore. RMAN optimizes performance and space consumption during backup with file multiplexing and backup set compression, and integrates with Oracle Secure Backup, as well as third party media management products, for tape backup. Both encryption from the Advanced Security Option and compression from the Advanced Compression option can be used in Data Pump and RMAN together.

A. Overview

Organizations are faced with the challenges of encrypting data at rest and trying to optimize the utilization of costly resources including storage infrastructure.

This lab demonstrates the process of both compressing and encrypting data on disk and on backups. You will see the operations to implement Advanced Compression and to implement tablespace encryption are similar in nature and can be used in combination.

This lab provides examples to understand each of the processes and how they can be combined together.

To move data into an encrypted tablespace (thus encrypting the data), the following steps take place.

1. Create encrypted tablespace.
2. Migrate (copy) the tables from the unencrypted tablespace to encrypted tablespace
 - a. Use the ALTER TABLE... MOVE command to move the table into the encrypted tablespace

To implement Advanced Compression, the following steps can be used.

1. Use the ALTER TABLE... COMPRESS command (Multi-state compression affecting data only after the alter command has taken place.)
2. Alternatively, create compressed table in encrypted/unencrypted tablespace as CTAS from un-compressed table by specifying the use of compression during the CREATE TABLE... COMPRESS FOR ALL OPERATIONS

To implement encryption and Advanced Compression, the following steps can be used.

1. Create compressed table in encrypted tablespace as CTAS from un-compressed table by specifying the use of compression during the CREATE TABLE... COMPRESS FOR ALL OPERATIONS
2. Alternatively, create encrypted and compressed tablespace CREATE TABLESPACE... DEFAULT COMPRESS FOR ALL OPERATIONS storage (ENCRYPT)

To implement encrypted and compressed Data Pump exports.

1. Specify the ENCRYPTION and COMPRESSION option in the expdp utility. ENCRYPTION=ALL COMPRESSION=ALL

To implement encrypted and compressed RMAN backups.

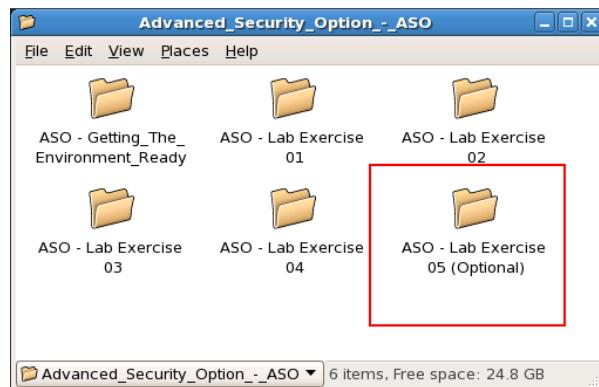
1. Specify the ENCRYPTION and COMPRESSION option in the RMAN script by using `set encryption on;` and `backup as COMPRESSED BACKUPSET`

During this lab you will:

1. Demonstrate the usage and expected characteristics of combining Tablespace Encryption within the Advanced Security Option and Advanced Compression.
2. Encrypt Data Pump archives using encryption from the Advanced Security Option to further protect information in created archives.
3. Encrypt RMAN using encryption from the Advanced Security Option to further protect information in created backups.

B. Setup & Preparation

- All scripts used in this lab exercise can be found in the directory `/home/oracle/ac_scripts`.
 - A directory named `/home/oracle/wallet` has already been created for you. This directory will be used to store the master encryption wallet file. If there is already a `ewallet.p12` file, do not delete this file.
1. You should have already completed **AC Lab Config 00** before using this lab.
 2. Navigate to the folder, **ASO – Lab Exercise 05 (Optional)**.

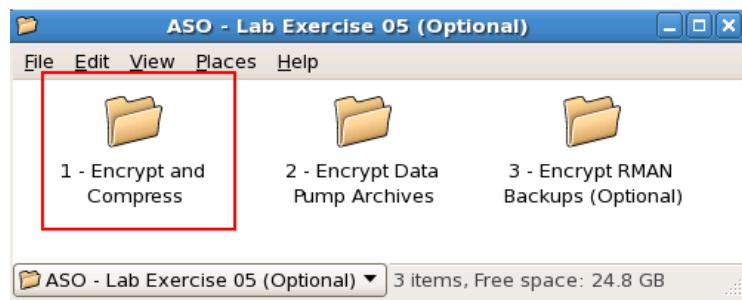


C. Protecting Sensitive Data and Optimizing Storage With Advanced Compression on Disk & Backups

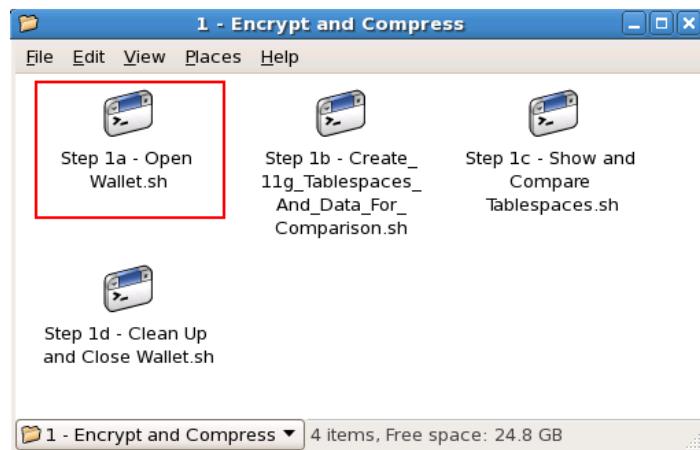
Combining Encryption and Advanced Compression

This portion demonstrates the ability to leverage both capabilities of Advanced Security Option – Tablespace Encryption and Advanced Compression. You will see the operations to implement tablespace encryption and to implement Advanced Compression can be used in combination rather easily and could save downtime by combining the steps into one implementation cycle compared to implementing these independently.

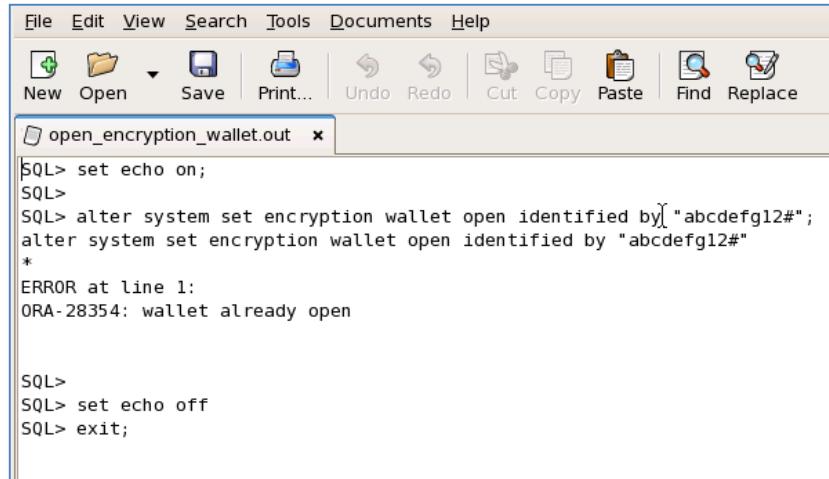
1. Click on the folder **1 – Encrypt and Compress**.



2. Click on the icon, **Step 1a – Open Wallet.sh**.



3. Check the output file **open_encryption_wallet.out**. The wallet may already be open, in which case, you will see the error below.

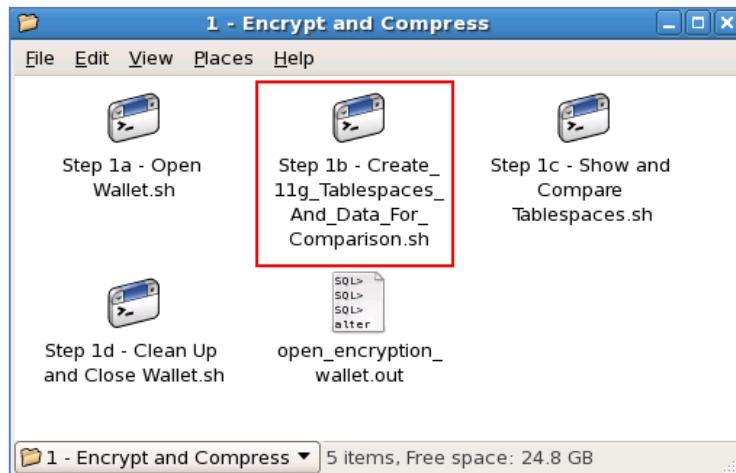


```
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
open_encryption_wallet.out x
SQL> set echo on;
SQL>
SQL> alter system set encryption wallet open identified by "abcdefg12#";
alter system set encryption wallet open identified by "abcdefg12#"
*
ERROR at line 1:
ORA-28354: wallet already open

SQL>
SQL> set echo off
SQL> exit;
```

4. Click on the icon, **Step 1b –**

Create_11g_Tablespace_And_Data_For_Comparison.sh. In this step, we are creating a number of tablespaces, using a combination of no encryption/compression, encryption only, compression only and encryption and compression for comparison purposes.



5. Click on the icon, **Step 1b –**

Create_Tablespace_And_Data_For_Comparisons.out file in the folder to review the output of the previously executed script.



```

SQL> drop tablespace ex_11g_ts including contents and datafiles;
SQL> drop tablespace ex_11g_ts including contents and datafiles;
drop tablespace ex_11g_ts including contents and datafiles
*
ERROR at line 1:
ORA-00959: tablespace 'EX_11G_TS' does not exist

SQL> create tablespace ex_11g_ts
  2  datafile '/u01/oracle/oradata/db06/ex_11g_ts.dbf'
  3  size 100m
  4  /
Tablespace created.

SQL>
SQL> drop tablespace ex_11g_enc_ts including contents and datafiles;
drop tablespace ex_11g_enc_ts including contents and datafiles
*
ERROR at line 1:
ORA-00959: tablespace 'EX_11G_ENC_TS' does not exist

SQL> create tablespace ex_11g_enc_ts
  2  datafile '/u01/oracle/oradata/db06/ex_11g_enc_ts.dbf'
  3  size 100m
  4  encryption using 'AES256'
  5  default storage(encrypt)
  6  /
Tablespace created.

SQL>
SQL> drop tablespace ex_11g_comp_ts including contents and datafiles;
drop tablespace ex_11g_comp_ts including contents and datafiles
*
ERROR at line 1:
ORA-00959: tablespace 'EX_11G_COMP_TS' does not exist

SQL> create tablespace ex_11g_comp_ts
  2  datafile '/u01/oracle/oradata/db06/ex_11g_comp_ts.dbf'
  3  size 100m
  4  default COMPRESS FOR ALL OPERATIONS
  5  /
Tablespace created.

SQL>
SQL> drop tablespace ex_11g_enc_comp_ts including contents and datafiles;
drop tablespace ex_11g_enc_comp_ts including contents and datafiles
*
ERROR at line 1:
ORA-00959: tablespace 'EX_11G_ENC_COMP_TS' does not exist

SQL> create tablespace ex_11g_enc_comp_ts
  2  datafile '/u01/oracle/oradata/db06/ex_11g_enc_comp_ts.dbf'
  3  size 100m
  4  encryption using 'AES256'
  5  default COMPRESS FOR ALL OPERATIONS storage(encrypt)
  6  /
Tablespace created.

SQL>
SQL> conn ACCTS_ADMIN_ACE/Manager_1
Connected.
SQL>
SQL> drop user DBA_sales1 cascade;
drop user DBA_sales1 cascade
*
ERROR at line 1:
ORA-01918: user 'DBA_SALES1' does not exist

```

```

SQL> drop user DBA_sales2 cascade;
drop user DBA_sales2 cascade
*
ERROR at line 1:
ORA-01918: user 'DBA_SALES2' does not exist

SQL> drop user DBA_sales3 cascade;
drop user DBA_sales3 cascade
*
ERROR at line 1:
ORA-01918: user 'DBA_SALES3' does not exist

SQL> drop user DBA_sales4 cascade;
drop user DBA_sales4 cascade
*
ERROR at line 1:
ORA-01918: user 'DBA_SALES4' does not exist

SQL>
SQL> create user DBA_sales1 identified by oracle1 default tablespace ex_11g_ts;
User created.

SQL> create user DBA_sales2 identified by oracle1 default tablespace
ex_11g_enc_ts;
User created.

SQL> create user DBA_sales3 identified by oracle1 default tablespace
ex_11g_comp_ts;
User created.

SQL> create user DBA_sales4 identified by oracle1 default tablespace
ex_11g_enc_comp_ts;
User created.

SQL>
SQL> conn / as sysdba
Connected.
SQL>
SQL> grant dba to DBA_sales1;
Grant succeeded.

SQL> grant dba to DBA_sales2;
Grant succeeded.

SQL> grant dba to DBA_sales3;
Grant succeeded.

SQL> grant dba to DBA_sales4;
Grant succeeded.

SQL>
SQL> conn DBA_sales1/oracle1
Connected.
SQL> create table sales as select * from sh.sales;

Table created.

SQL>
SQL> conn DBA_sales2/oracle1
Connected.
SQL> create table sales as select * from sh.sales;

Table created.

SQL>
SQL> conn DBA_sales3/oracle1

```

```

Connected.
SQL> create table sales as select * from sh.sales;
Table created.

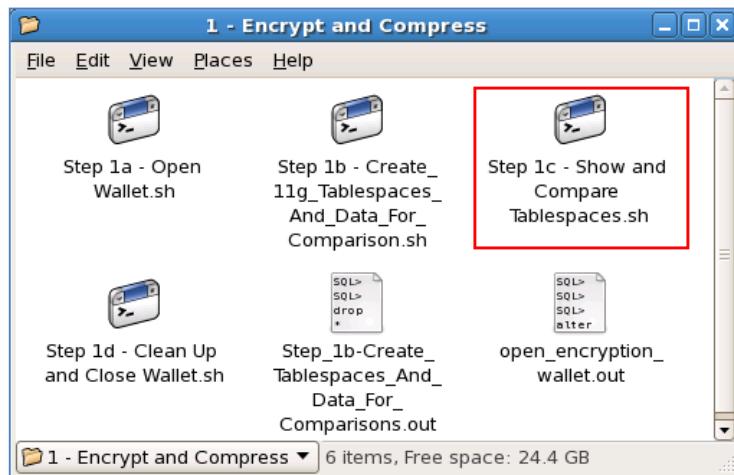
SQL>
SQL> conn DBA_sales4/oracle1
Connected.
SQL> create table sales as select * from sh.sales;

Table created.

SQL>
SQL> set echo off
SQL>
SQL> exit;

```

6. Click on the icon, **Step 1c – Show and Compare Tablespaces.sh**. In this step, we will execute a number of queries to view and compare the different tablespace combinations created in the earlier step.



7. Click on the icon, **Step_1c>Show_and_compare_tablespaces.out** file in the folder to review the output of the previously executed script. There are a number of useful queries to view the configuration and status of these tablespaces.

Step_1c>Show_and_compare_tablespaces.out

```

SQL> @/home/oracle/aso_scripts/show_encrypted_tablespaces.sql
SQL> desc v$encrypted_tablespaces;
Name          Null?    Type
-----
TS#          NUMBER
ENCRYPTIONALG VARCHAR2(7)
ENCRYPTEDTS  VARCHAR2(3)
ENCRYPTEDKEY RAW(32)
MASTERKEYID   RAW(16)
BLOCKS_ENCRYPTED NUMBER
BLOCKS_DECRYPTED NUMBER

```

```

SQL>
SQL> select t.name "TSName", e.encryptioalg "Algorithm", d.file_name
  "File Name"
   2  FROM v$tablespace t, v$encrypted tablespaces e, dba_data_files d
   3 WHERE t.ts# = e.ts# and t.name = d.tablespace_name;

TSName                      Algorit
-----  -----
File Name
-----
-----
EX_11G_ENC_TS          AES256
/u01/oracle/oradata/db06/ex_11g_enc_ts.dbf

EX_11G_ENC_COMP_TS      AES256
/u01/oracle/oradata/db06/ex_11g_enc_comp_ts.dbf

SQL>
SQL> select a.owner "Owner", a.table_name "Table Name", e.encryptioalg
  "Algorithm"
   2  FROM dba_tables a, v$encrypted tablespaces e
   3 WHERE a.tablespace_name in (select t.name from v$tablespace t,
   4 v$encrypted tablespaces e where t.ts# = e.ts#);

Owner                  Table Name          Algorit
-----  -----  -----
DBA_SALES2              SALES            AES256
DBA_SALES2              SALES            AES256
DBA_SALES4              SALES            AES256
DBA_SALES4              SALES            AES256

SQL>
SQL>
SQL>
SQL> connect / as sysdba
Connected.
SQL>
SQL> column TABLE format a20
SQL> column TABLESPACE format a30
SQL> column OWNER format a10
SQL> column ENC format a5
SQL>
SQL> select substr(a.table_name,1,28)
  "TABLE",substr(b.tablespace_name,1,30) "TABLESPACE",
   2  substr(a.owner,1,10) "OWNER",
   3  b.encrypted "ENC?"
   4  from dba_tables a, dba_tablespaces b
   5  where a.tablespace_name=b.tablespace_name
   6  and owner in ('DBA_SALES1','DBA_SALES2','DBA_SALES3','DBA_SALES4')
   7  order by 3,1,2
   8  /

```

TABLE	TABLESPACE	OWNER	ENC
SALES	EX_11G_TS	DBA_SALES1	NO
SALES	EX_11G_ENC_TS	DBA_SALES2	YES
SALES	EX_11G_COMP_TS	DBA_SALES3	NO
SALES	EX_11G_ENC_COMP_TS	DBA_SALES4	YES

```

SQL>
SQL> set echo on
SQL> column compress_for format a30
SQL> column tablespace_name format a30
SQL> select tablespace_name,compress_for from user_tablespaces where
  tablespace_name like '%EX_11G%';

TABLESPACE_NAME          COMPRESS_FOR
-----  -----
EX_11G_TS
EX_11G_ENC_TS
EX_11G_COMP_TS          OLTP
EX_11G_ENC_COMP_TS      OLTP

SQL>
SQL> set linesize 200

```

```

SQL> column segment_name format a30
SQL> select segment_name, owner, blocks, bytes
  2  from dba_segments
  3  where segment_name in ('SALES')
  4  and owner in ('DBA_SALES1','DBA_SALES2','DBA_SALES3','DBA_SALES4')
order by owner;

```

SEGMENT_NAME	OWNER	BLOCKS	BYTES
SALES	DBA_SALES1	4608	37748736
SALES	DBA_SALES2	4608	37748736
SALES	DBA_SALES3	1792	14680064
SALES	DBA_SALES4	1792	14680064

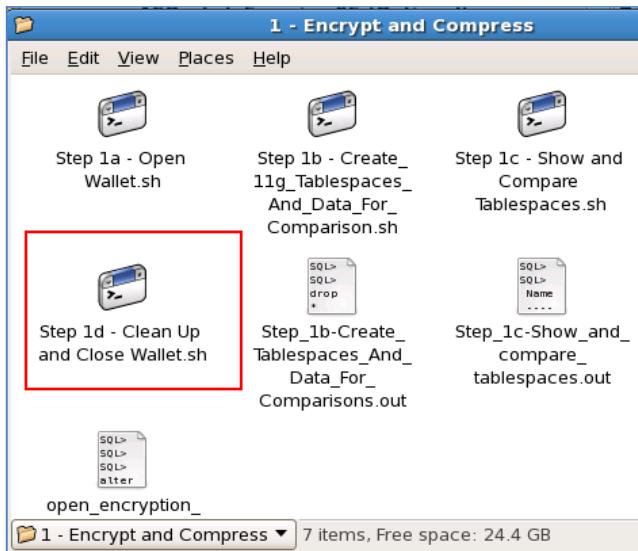
```

SQL> set echo off;
SQL>

```

SQL> exit; One important fact that needs to be pointed out are the results from the last 3 queries. You will see that encrypting data does not increase storage requirements (comparing SALES1 to SALES2), and that the significant efficiencies to be gained by compression (comparing SALES1 to SALES3) are also available with encryption enabled (comparing SALES3 to SALES4).

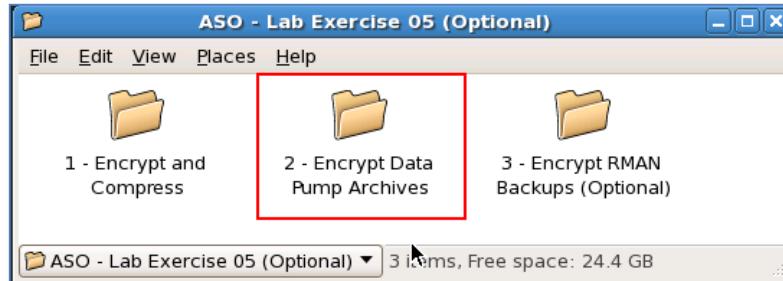
8. Click on the icon, **Step 1d – Clean Up and Close Wallet.sh**. In this step, we are closing the wallet as a housekeeping step for the next exercise.



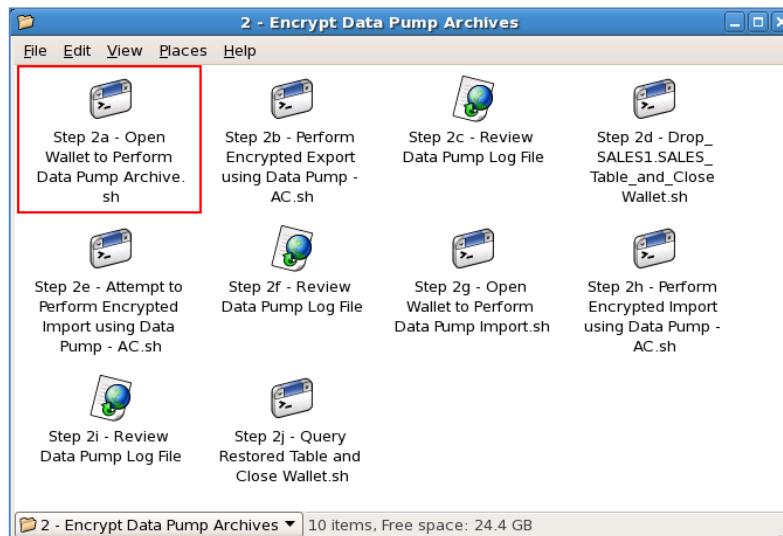
Encrypting Data Pump Archives

In the 11g version of Data Pump, you now have the option of encrypting and compressing data. This section of the lab will demonstrate encryption and compression of Data Pump Archives and how the encryption of these archives will protect sensitive information.

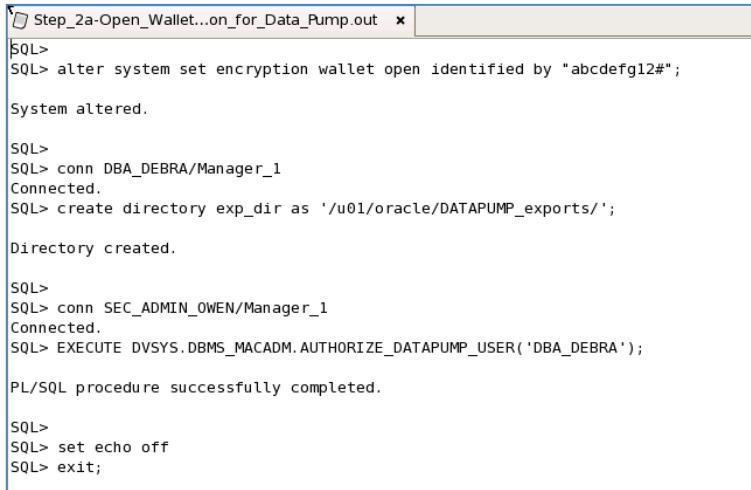
9. Return to the Lab 5 folder and click on **2 – Encrypt Data Pump Archives**.



10. Click on the icon, **Step 2a – Open Wallet to Perform Data Pump Archive.sh**. This step opens the encryption wallet and creates an export directory needed for Data Pump and the expdp utility.



11. Click on the icon, **Step_2a-**
Open_Wallet_in_preparation_for_Data_Pump.out file in the folder
to review the output of the previously executed script, which creates
the datapump directory and grants the appropriate permissions to
perform exports in a Database Vault environment.



```

SQL> alter system set encryption wallet open identified by "abcdefg12#";
System altered.

SQL>
SQL> conn DBA_DEBRA/Manager_1
Connected.
SQL> create directory exp_dir as '/u01/oracle/DATAPUMP_exports/';

Directory created.

SQL>
SQL> conn SEC_ADMIN_OWEN/Manager_1
Connected.
SQL> EXECUTE DV$SYS.DBMS_MACADM.AUTHORIZE_DATAPUMP_USER('DBA_DEBRA');

PL/SQL procedure successfully completed.

SQL>
SQL> set echo off
SQL> exit;

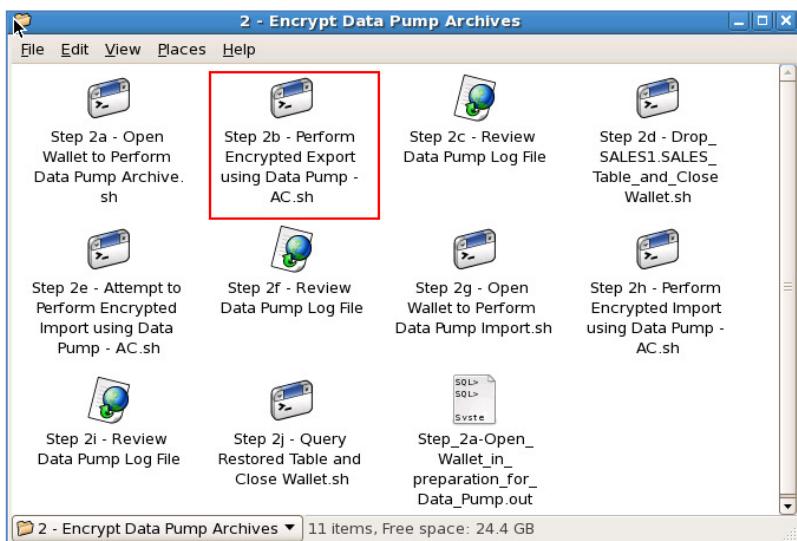
```

12. Click on the icon, **Step 2b – Perform Encrypted Export using Data Pump – AC.sh**. In this step, we will be exporting the SALES1.SALES table and using both compression and encryption.

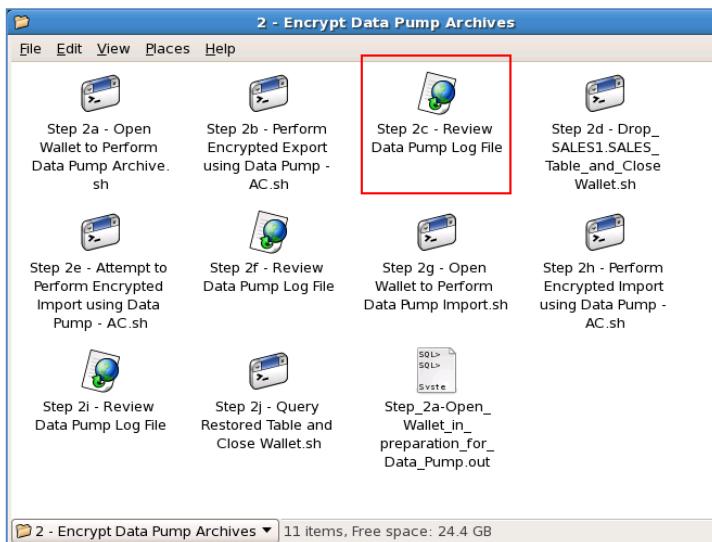
```

expdp system/oracle1 TABLES=SALES1.SALES
DIRECTORY=exp_dir
DUMPFILE=sales_table_export_with_encryption.dmp
LOGFILE=sales_table_export_encryption.log
ENCRYPTION=ALL COMPRESSION=ALL

```

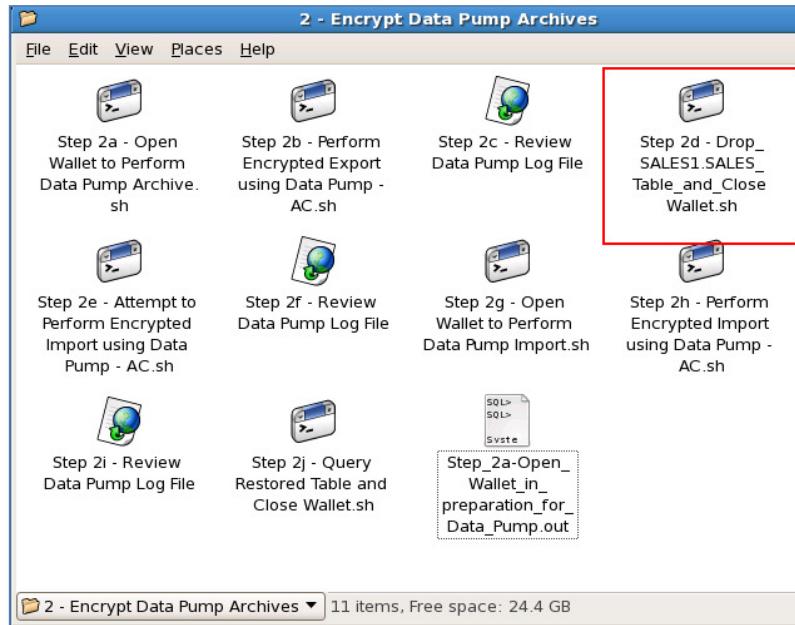


13. Click on the icon, **Step 2c – Review Data Pump Log File** in the folder to review the Data Pump Log File



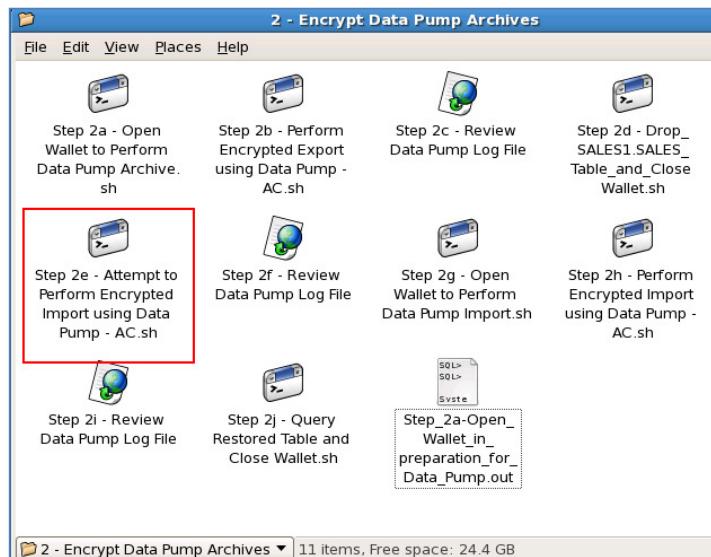
```
sales_table_export_encryption.log
=====
Export: Release 11.2.0.2.0 - Production on Sun Sep 25 22:51:18 2011
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.
=====
Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - Production
With the Partitioning, Oracle Label Security, OLAP, Data Mining,
Data Warehousing and Real Application Testing options
Starting "DBA_DEBRA" "SYS_EXPORT_TABLE_01": DBA_DEBRA/******** TABLES=DEBA_SALES1.SALES DIRECTORY=exp_dir DUMPFILE=sales_table_export_with_encryption.dmp LOGFILE=sales_table_e
Estimate in progress using BLOCKS method...
Processing object type TABLE_EXPORT/TABLE/TABLE_DATA
Total estimation using BLOCKS method: 36 MB
Processing object type TABLE_EXPORT/TABLE/TABLE
Master table "DBA_DEBRA"."SYS_EXPORT_TABLE_01" successfully loaded/unloaded
=====
Dump file set for DBA_DEBRA.SYS_EXPORT_TABLE_01
Job successfully completed at 22:51:34
/u01/oracle/dump/debra_sales_table_export_with_encryption.dmp
Job "DBA_DEBRA"."SYS_EXPORT_TABLE_01" successfully completed at 22:51:34
```

14. Now that the Data Pump export has completed successfully, click on the icon, **Step 2d – Drop_SALES1.SALES_Table_and_Close_Wallet.sh**. In this step, we need to prepare the environment for the import and close the wallet.

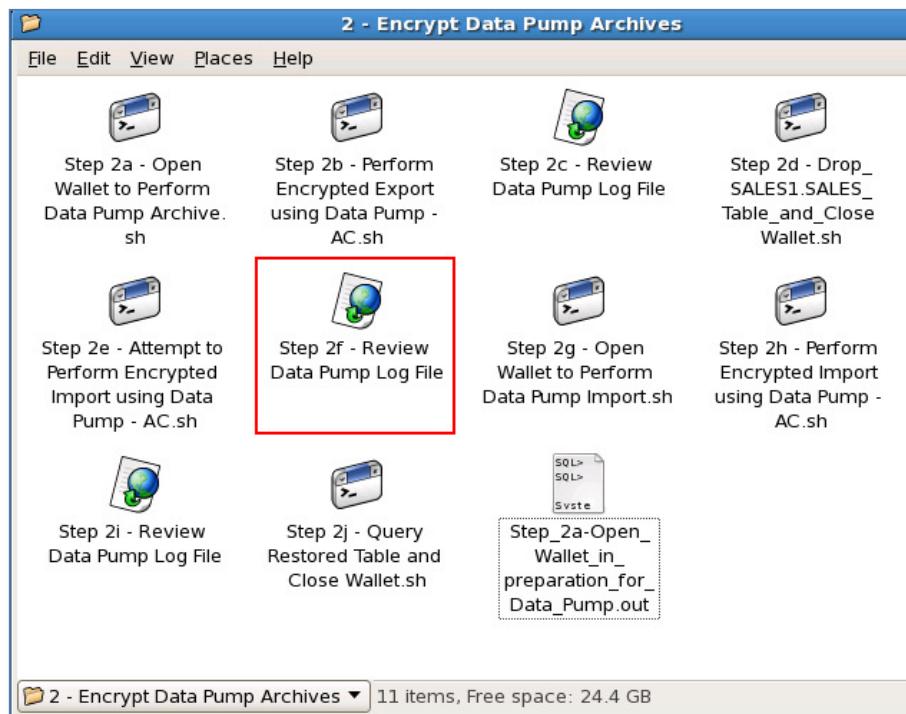


15. Click on the icon, **Step 2e – Attempt to Perform Encrypted Import using Data Pump – AC.sh**. In this step, we are attempting to import the archive created in the previous step.

```
impdp system/oracle1 TABLES=sales1.sales
DIRECTORY=exp_dir
DUMPFILE=sales_table_export_with_encryption.dmp
LOGFILE=sales_table_import_with_encryption.log
```



16. Click on the icon, **Step 2f – Review Data Pump Log File** in the folder to review the Data Pump Log File located in the **/u01/oracle/DATAPUMP_EXPORTS/sales_table_import_with_encryption.log**.



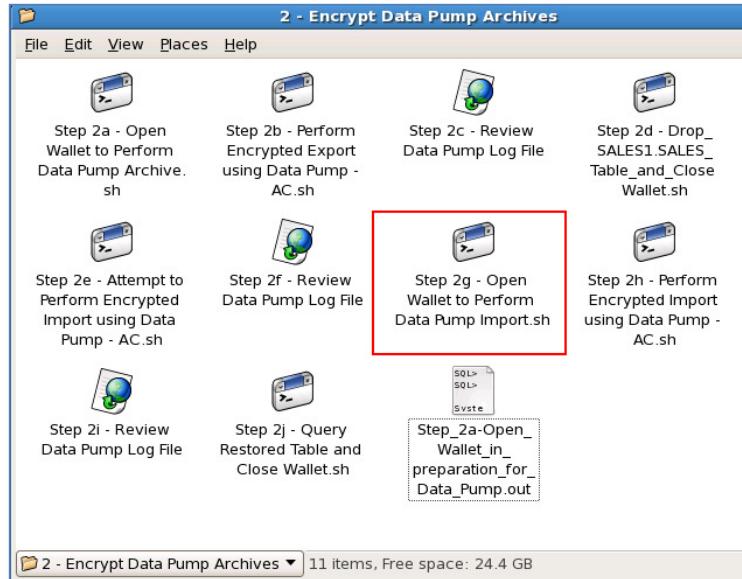
You will notice in the log that while attempting the import without the proper encryption wallet used to encrypt the export and that the matching encryption wallet open, the process will raise an error and fail. We wanted to illustrate that if an encrypted export (.dmp) file found itself in the wrong hands, without the proper encryption wallet, the contents of that .dmp file are rendered useless—protecting the sensitive information inside the .dmp file.

```

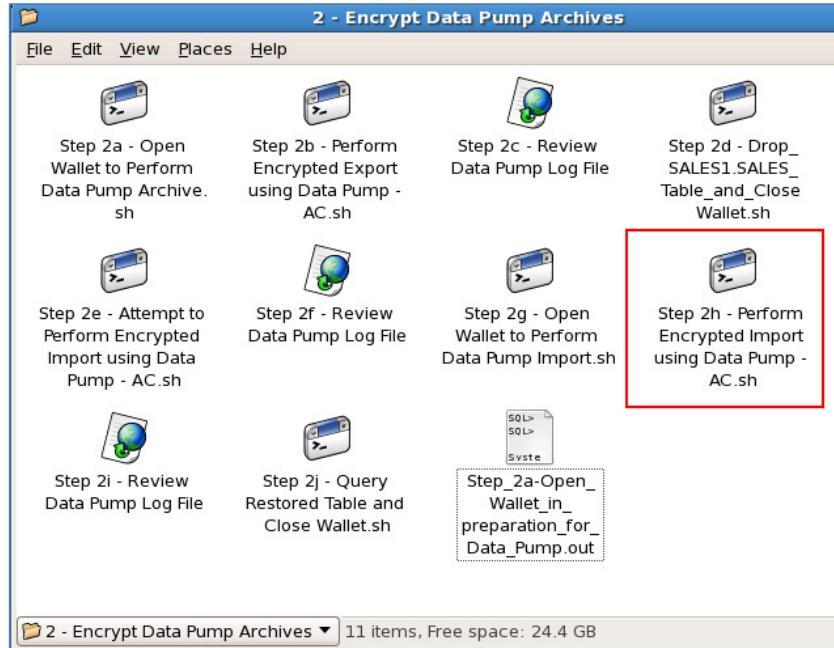
sales_table_import_with_encryption.log * 
;;
Import: Release 11.2.0.2.0 - Production on Sun Sep 25 22:56:48 2011
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.
;;
Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - Production
With the Partitioning, Oracle Label Security, OLAP, Data Mining,
Oracle Database Vault and Real Application Testing options
ORA-39189: unable to decrypt dump file set
ORA-28365: wallet is not open

```

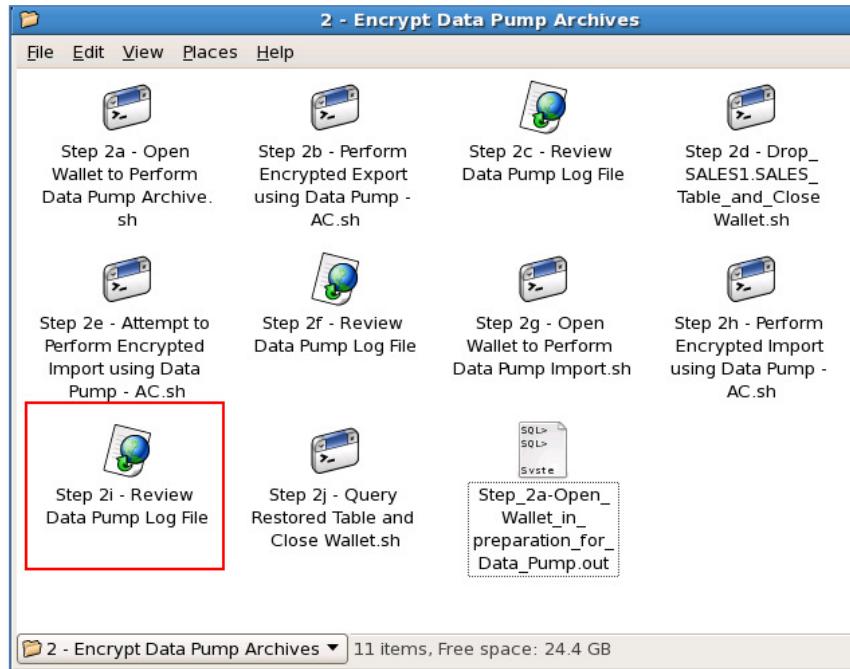
17. Click on the icon, **Step 2g – Open Wallet to Perform Data Pump Import**. In this step, we are opening the proper encryption wallet.



18. Click on the icon, **Step 2h – Perform Encrypted Import using Data Pump – AC.sh**. In this step, we are again executing the import with the important difference that the proper encryption wallet is open.



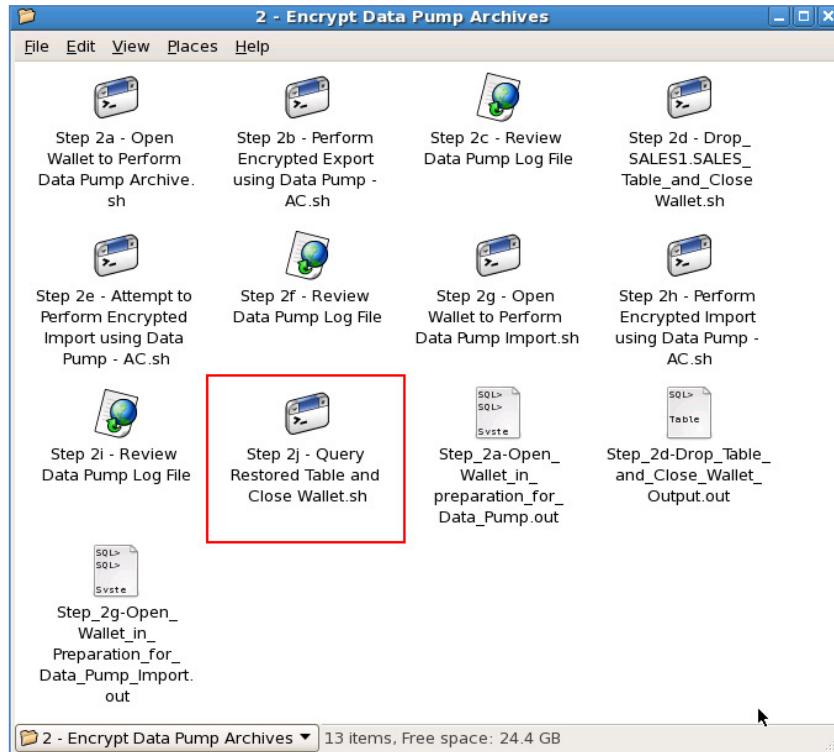
19. Click on the icon, **Step 2i – Review Data Pump Log File** in the folder to review the Data Pump Log File located in the
/u01/oracle/DATAPUMP_EXPORTS/sales_table_import_encryption_after_opening_wallet.log



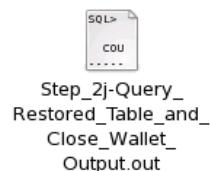
```
sales_table_import...r_opening_wallet.log
=====
;
Import: Release 11.2.0.2.0 - Production on Sun Sep 25 23:02:15 2011
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.
;;
Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - Production
With the Partitioning, Oracle Label Security, OLAP, Data Mining,
Oracle Database Vault and Real Application Testing options
Master table "DBA_DEBRA"."SYS_IMPORT_TABLE_01" successfully loaded/unloaded
Starting "DBA_DEBRA"."SYS_IMPORT_TABLE_01": DBA_DEBRA/********* TABLES=DBA_sales1.sales DIRECTORY=exp_dir DUMPFILE=sales_table_export_with_encryption.dmp LOGFILE=sales_table_im
Processing object type TABLE_EXPORT/TABLE/TABLE
Processing object type TABLE_EXPORT/TABLE/INDEX/INDEX
Imported "DBA_SALES1"."SALES"                                2.586 MB 918843 rows
Job "DBA_DEBRA"."SYS_IMPORT_TABLE_01" successfully completed at 23:02:21
```

Note that with the wallet open, we now can restore the .dmp file.

20. After the successful import of the SALES1.SALES table, Click on the icon, **Step 2j – Query Restored Table and Close Wallet.sh**. In this step, we will query the restored table and finish the steps by closing the wallet.



21. Click on the icon, **Step_2j-Query_Restored_Table_and_Close_Wallet_Output.out** file in the folder to review the output of the previously executed script. Notice that the SALES1.SALES table has been restored properly.



```

Step_2j-Query_Resto...se_Wallet_Output.out x
SQL> select count(*) from DBA_sales1.sales;

COUNT(*)
-----
918843

SQL>
SQL> drop directory exp_dir;

Directory dropped.

SQL>
SQL> conn INFOSEC_ISABEL/Manager_1
Connected.
SQL> alter system set encryption wallet close identified by "abcdefg12#";

System altered.

SQL> exit;

```

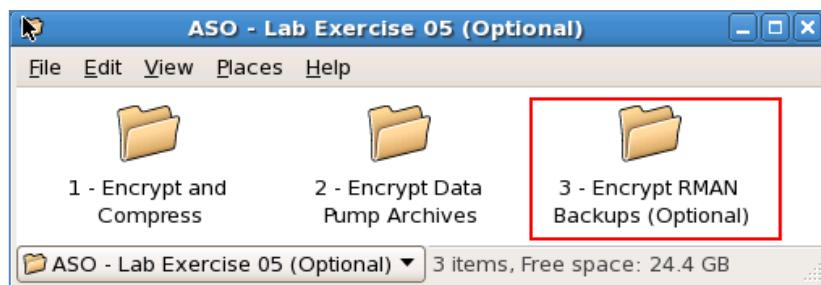
This step demonstrates that we have successfully restored the sales data, then closes the wallet to conclude this set of labs.

Combining Encryption and Advanced Compression

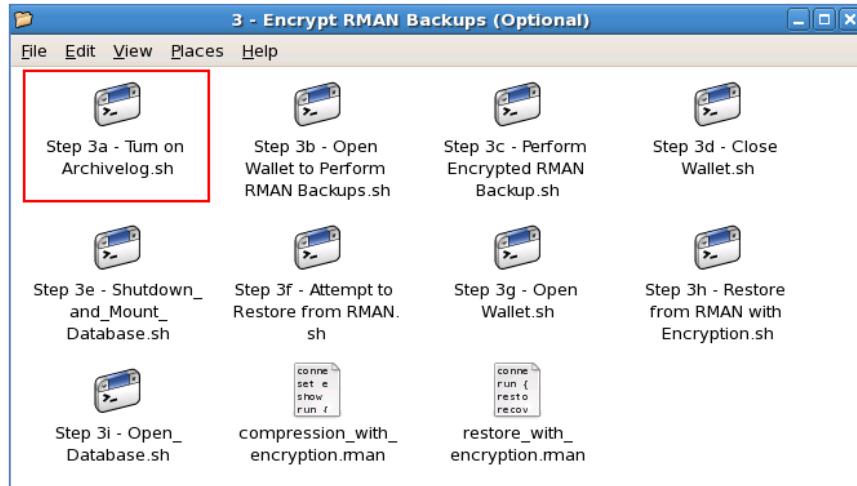
Oracle Recovery Manager (RMAN) provides a comprehensive foundation for efficiently backing up and recovering the Oracle database. It is designed to work intimately with the server, providing block-level corruption detection during backup and restore. Both encryption from the Advanced Security Option and compression from the Advanced Compression option can be used in RMAN together. In this lab section, we demonstrate the use of encryption.

This is an optional lab and does require some additional time (approx. 30 minutes) to complete. This is due to the fact that the lab exercise completes a full database backup.

22. Click on the folder **3 – Encrypt RMAN Backups (Optional)**.



23. Click on the icon, **Step 3a – Turn on Archivelog.sh**. In this step, we are putting the database in ARCHIVELOG mode necessary for RMAN.



24. Click on the icon, **Step_3a-Setup_Database_for_Archivelog.out** file in the folder to review the output of the previously executed script.



```

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount;
ORACLE instance started.

Total System Global Area  418484224 bytes
Fixed Size                  1336932 bytes
Variable Size                343935388 bytes
Database Buffers              67108864 bytes
Redo Buffers                  6103040 bytes
Database mounted.
SQL> alter database archivelog;

Database altered.

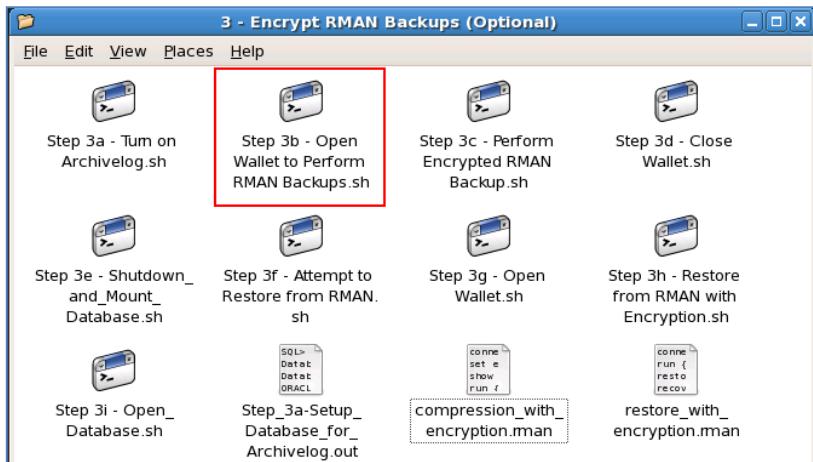
SQL> alter database open;

Database altered.

SQL> exit;

```

25. Click on the icon, **Step 3b – Open Wallet to Perform RMAN Backups.sh**. In this step, we are opening up the encryption wallet in preparation of performing the encrypted RMAN backup.



26. Click on the icon, **Step_3b–Open_Wallet_in_preparation_for_RMAN.out** file in the folder to review the output of the previously executed script.

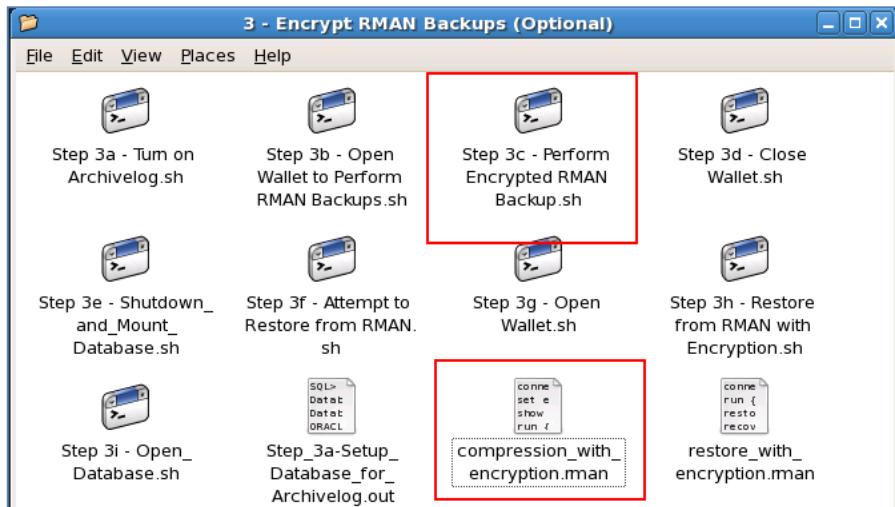


```
SQL> alter system set encryption wallet open identified by "abcdefg12#";
System altered.

SQL> set echo off
SQL> exit;
```

27. Click on the icon, **Step 3c – Perform Encrypted RMAN Backup.sh**.
 This step may take several minutes, and in this step, we are executing an RMAN backup using the option:

```
set encryption on;
```



Click on the icon **compression_with_encryption.rman** to view the full RMAN script being executed.

```
connect target /
set encryption on;
show all;
run {
allocate channel t1 type disk format
'/u01/oracle/RMAN_backups/high_compression_with_encryption_%d_set%s_piece%p_cop
y%c_%t%U';
backup as COMPRESSED BACKUPSET database plus archivelog;
release channel t1;}
```

28. Click on the icon, **Step_3c-Perform_RMAN_with_Encryption.log** file in the folder to review the output of the previously executed script. The output below is abbreviated. View the contents in the file for the complete log output. You will notice the command **CONFIGURE ENCRYPTION FOR DATABASE OFF**; . If set to ON, all RMAN backup sets created by this database will use transparent encryption by default. You will also notice that we are using the default 'AES128' algorithm for encryption. This can be changed by using the command **CONFIGURE ENCRYPTION ALGORITHM TO 'AES256'**;



Recovery Manager: Release 11.2.0.2.0 - Production on Sun Sep 25 23:12:44 2011

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

```
RMAN> connect target *
2> set encryption on;
3> show all;
4> run {
5> allocate channel t1 type disk format
'./u01/oracle/RMAN_backups/high_compression_with_encryption_%d_set%s_piece%p_copy%c_%T%
U';
6> backup as COMPRESSED BACKUPSET database plus archivelog;
7> release channel t1;}
8>
9>
connected to target database: DB06 (DBID=1606160634)

executing command: SET encryption
using target database control file instead of recovery catalog

RMAN configuration parameters for database with db_unique_name DB06 are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP OFF; # default
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; # default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE OFF; # default
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE 'DEFAULT' OPTIMIZE FOR LOAD TRUE
; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO
'/u01/oracle/product/11.2.0/dbhome_1/dbs/snapcf_db06.f'; # default

allocated channel: t1
channel t1: SID=143 device type=DISK

Starting backup at 25-SEP-11
current log archived
channel t1: starting compressed archived log backup set
channel t1: specifying archived log(s) in backup set
input archived log thread=1 sequence=132 RECID=1 STAMP=762822622
input archived log thread=1 sequence=133 RECID=2 STAMP=762822767
channel t1: starting piece 1 at 25-SEP-11
channel t1: finished piece 1 at 25-SEP-11
piece
handle=/u01/oracle/RMAN_backups/high_compression_with_encryption_DB06_set1_piece1_copy
1_2011092501mnfg3g_1_1 tag=TAG20110925T231247 comment=None
channel t1: backup set complete, elapsed time: 00:00:03
Finished backup at 25-SEP-11

Starting backup at 25-SEP-11
channel t1: starting compressed full datafile backup set
channel t1: specifying datafile(s) in backup set
input datafile file number=00001 name=/u01/oracle/oradata/db06/system01.dbf
input datafile file number=00002 name=/u01/oracle/oradata/db06/sysaux01.dbf
input datafile file number=00003 name=/u01/oracle/oradata/db06/undotbs01.dbf
input datafile file number=00004 name=/u01/oracle/oradata/db06/example01.dbf
input datafile file number=00007 name=/u01/oracle/oradata/db06/ex_11g_ts.dbf
input datafile file number=00008 name=/u01/oracle/oradata/db06/ex_11g_enc_ts.dbf
input datafile file number=00009 name=/u01/oracle/oradata/db06/ex_11g_comp_ts.dbf
input datafile file number=00010 name=/u01/oracle/oradata/db06/ex_11g_enc_comp_ts.dbf
input datafile file number=00005 name=/u01/oracle/oradata/db06/users01.dbf
input datafile file number=00006 name=/u01/oracle/oradata/db06/banking01.dbf
channel t1: starting piece 1 at 25-SEP-11
channel t1: finished piece 1 at 25-SEP-11
piece
handle=/u01/oracle/RMAN_backups/high_compression_with_encryption_DB06_set2_piece1_copy
1_2011092502mnfg3j_1_1 tag=TAG20110925T231251 comment=None
channel t1: backup set complete, elapsed time: 00:01:45
channel t1: starting compressed full datafile backup set
channel t1: specifying datafile(s) in backup set
including current control file in backup set
including current SPFILE in backup set
```

```

channel t1: starting piece 1 at 25-SEP-11
channel t1: finished piece 1 at 25-SEP-11
piece
handle=/u01/oracle/RMAN_backups/high_compression_with_encryption_DB06_set3_piece1_copy
1_2011092503mnfg6s_1_1 tag=TAG20110925T231251 comment=NONE
channel t1: backup set complete, elapsed time: 00:00:01
Finished backup at 25-SEP-11

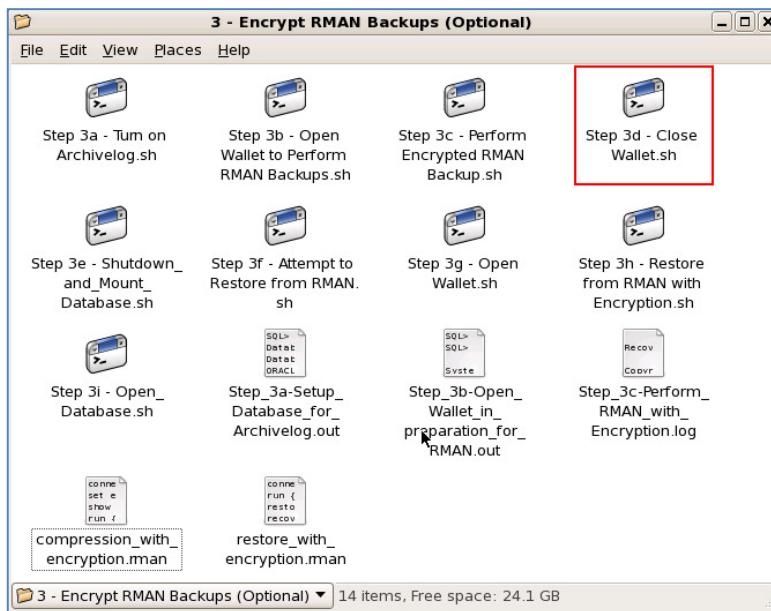
Starting backup at 25-SEP-11
current log archived
channel t1: starting compressed archived log backup set
channel t1: specifying archived log(s) in backup set
input archived log thread=1 sequence=134 RECID=3 STAMP=762822878
channel t1: starting piece 1 at 25-SEP-11
channel t1: finished piece 1 at 25-SEP-11
piece
handle=/u01/oracle/RMAN_backups/high_compression_with_encryption_DB06_set4_piece1_copy
1_2011092504mnfg6v_1_1 tag=TAG20110925T231438 comment=NONE
channel t1: backup set complete, elapsed time: 00:00:01
Finished backup at 25-SEP-11

released channel: t1

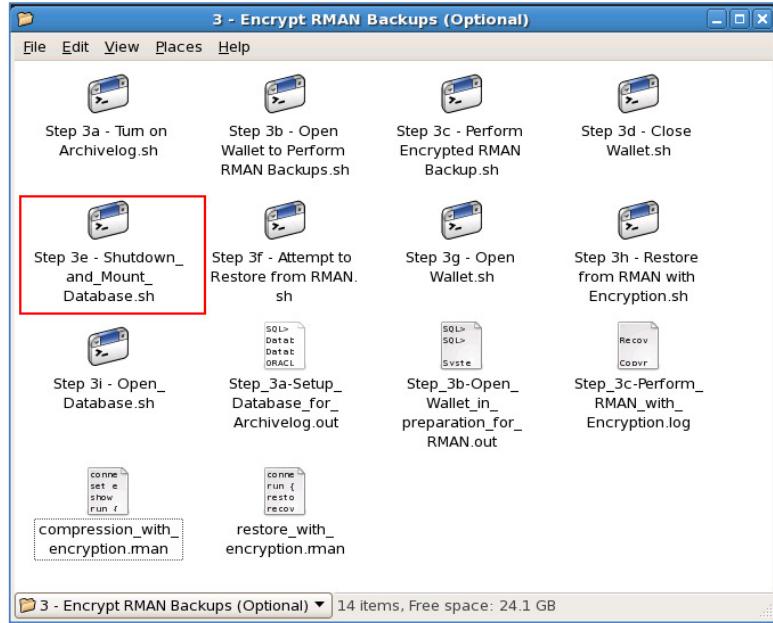
Recovery Manager complete.

```

29. Click on the icon, **Step 3d – Close Wallet.sh**. In this step, we are closing the encryption wallet with the familiar command, alter system set encryption wallet close identified by "abcdefg12#";.



30. Click on the icon, **Step 3e – Shutdown_and_Mount_Database.sh**. In this step, we shut down and mount the database so we can properly attempt to restore it.



31. Click on the icon, **Step_3e-Shutdown_and_Mount_Database.out** file in the folder to review the output of the previously executed script.

Step_3e-Shutdown_and_Mount_Database.out

```

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount;
ORACLE instance started.

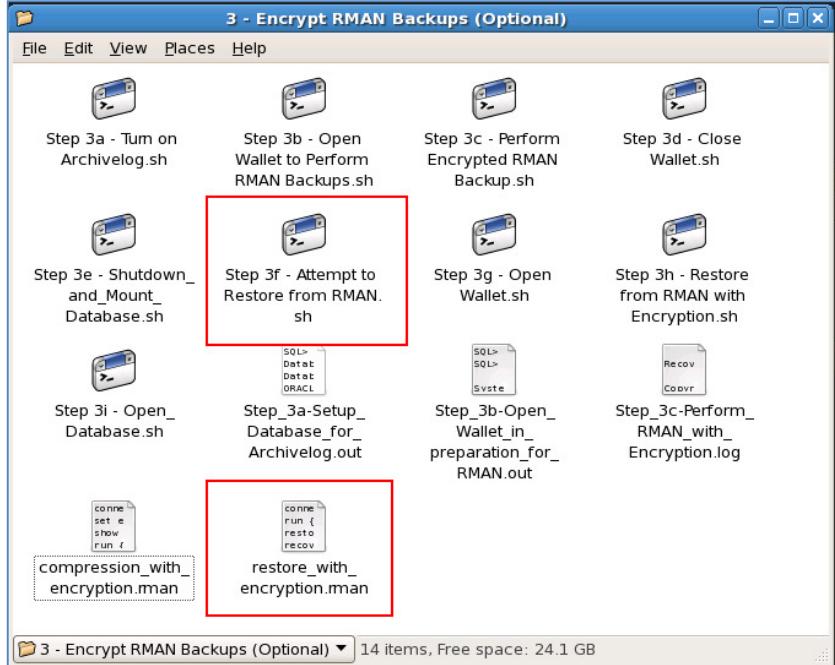
Total System Global Area  418484224 bytes
Fixed Size                  1336932 bytes
Variable Size                352323996 bytes
Database Buffers              58720256 bytes
Redo Buffers                  6103040 bytes
Database mounted.
SQL> alter database noarchivelog;

Database altered.

SQL> exit;

```

32. Click on the icon, **Step 3f – Attempt to Restore from RMAN.sh**. In this step, we are attempting to restore the database using the highlighted script.



Click on the icon **restore_with_encryption.rman** to view the RMAN script being executed.

```
connect target /
run {
restore database;
recover database; }
```

33. Click on the icon, **Step_3f-Attempt_to_Restore_RMAN_with_Encryption.log** file in the folder to review the output of the previously executed script. As expected, you will see the operation failed. Similar to the Data Pump import example, without the proper encryption wallet used to encrypt during the RMAN process and that the matching encryption wallet open, the process will raise an error and fail. Again, we wanted to illustrate that if an encrypted RMAN backup finds itself in the wrong hands or on a lost backup tape, without the proper encryption wallet, the contents of that RMAN backup are rendered useless—protecting the sensitive information inside.

```
Step_3f-Attempt_to_Restore_RMAN_with_Encryption.log
Recovery Manager: Release 11.2.0.2.0 - Production on Sun Sep 25 23:25:37 2011
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.
```

```

RMAN> connect target *
2> run {
3> restore database;
4> recover database;}
5>
6>
connected to target database: DB06 (DBID=1606160634, not open)

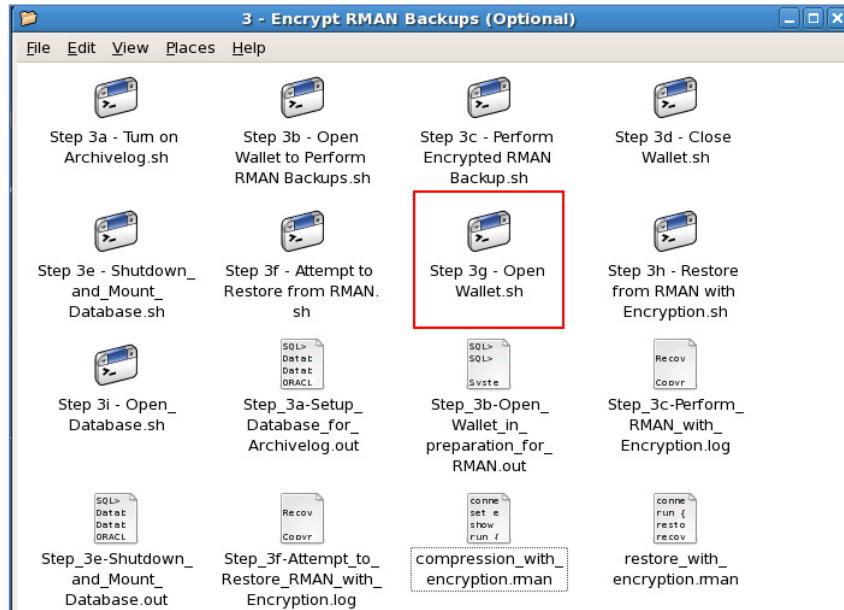
Starting restore at 25-SEP-11
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=133 device type=DISK

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_DISK_1: restoring datafile 00001 to
/u01/oracle/oradata/db06/system01.dbf
channel ORA_DISK_1: restoring datafile 00002 to
/u01/oracle/oradata/db06/sysaux01.dbf
channel ORA_DISK_1: restoring datafile 00003 to
/u01/oracle/oradata/db06/undotbs01.dbf
channel ORA_DISK_1: restoring datafile 00004 to
/u01/oracle/oradata/db06/example01.dbf
channel ORA_DISK_1: restoring datafile 00005 to
/u01/oracle/oradata/db06/users01.dbf
channel ORA_DISK_1: restoring datafile 00006 to
/u01/oracle/oradata/db06/banking01.dbf
channel ORA_DISK_1: restoring datafile 00007 to
/u01/oracle/oradata/db06/ex_11g_ts.dbf
channel ORA_DISK_1: restoring datafile 00008 to
/u01/oracle/oradata/db06/ex_11g_enc_ts.dbf
channel ORA_DISK_1: restoring datafile 00009 to
/u01/oracle/oradata/db06/ex_11g_comp_ts.dbf
channel ORA_DISK_1: restoring datafile 00010 to
/u01/oracle/oradata/db06/ex_11g_enc_comp_ts.dbf
channel ORA_DISK_1: reading from backup piece
/u01/oracle/RMAN_backups/high_compression_with_encryption_DB06_set2_piece1_copy
1_2011092502mnfg3j_1_1
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of restore command at 09/25/2011 23:25:40
ORA-19870: error while restoring backup piece
/u01/oracle/RMAN_backups/high_compression_with_encryption_DB06_set2_piece1_copy
1_2011092502mnfg3j_1_1
ORA-19913: unable to decrypt backup
ORA-28365: wallet is not open

Recovery Manager complete.

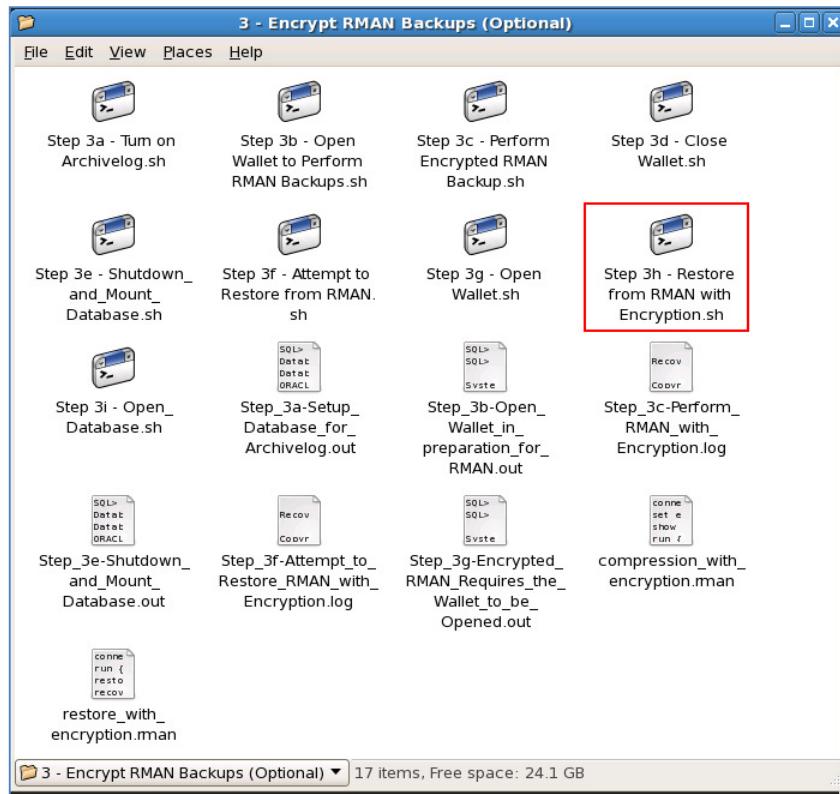
```

34. Click on the icon, **Step 3g – Open Wallet.sh**. In this step, we are opening the encryption wallet with the familiar command, alter system set encryption wallet open identified by "abcdefg12#";.



35. Click on the icon, **Step 3h – Restore from RMAN with Encryption.sh**.

Now that the encryption wallet is open, we can now successfully restore the database.



- 36.

Click on the icon, **Step_3h–Restore_from_RMAN_with_Encryption.log** file in the folder to review the output of the previously executed script. As expected, with the correct encryption wallet file open and open, the RMAN process will successfully complete as expected.



```

Recovery Manager: Release 11.2.0.2.0 - Production on Sun Sep 25 23:35:23 2011
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

RMAN> connect target *
2> run {
3> restore database;
4> recover database;;
5>
6>
connected to target database: DB06 (DBID=1606160634, not open)

Starting restore at 25-SEP-11
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=125 device type=DISK

channel ORA_DISK_1: starting datafile backup set restore

```

```

channel ORA_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_DISK_1: restoring datafile 00001 to
/u01/oracle/oradata/db06/system01.dbf
channel ORA_DISK_1: restoring datafile 00002 to
/u01/oracle/oradata/db06/sysaux01.dbf
channel ORA_DISK_1: restoring datafile 00003 to
/u01/oracle/oradata/db06/undotbs01.dbf
channel ORA_DISK_1: restoring datafile 00004 to
/u01/oracle/oradata/db06/example01.dbf
channel ORA_DISK_1: restoring datafile 00005 to
/u01/oracle/oradata/db06/users01.dbf
channel ORA_DISK_1: restoring datafile 00006 to
/u01/oracle/oradata/db06/banking01.dbf
channel ORA_DISK_1: restoring datafile 00007 to
/u01/oracle/oradata/db06/ex_11g_ts.dbf
channel ORA_DISK_1: restoring datafile 00008 to
/u01/oracle/oradata/db06/ex_11g_enc_ts.dbf
channel ORA_DISK_1: restoring datafile 00009 to
/u01/oracle/RMAN_backups/high_compression_with_encryption_DB06_set2_piece1_copy
1_2011092502mnfg3j_1_1
channel ORA_DISK_1: piece
handle=/u01/oracle/RMAN_backups/high_compression_with_encryption_DB06_set2_piec
e1_copy1_2011092502mnfg3j_1_1 tag=TAG20110925T231251
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:02:25
Finished restore at 25-SEP-11

Starting recover at 25-SEP-11
using channel ORA_DISK_1

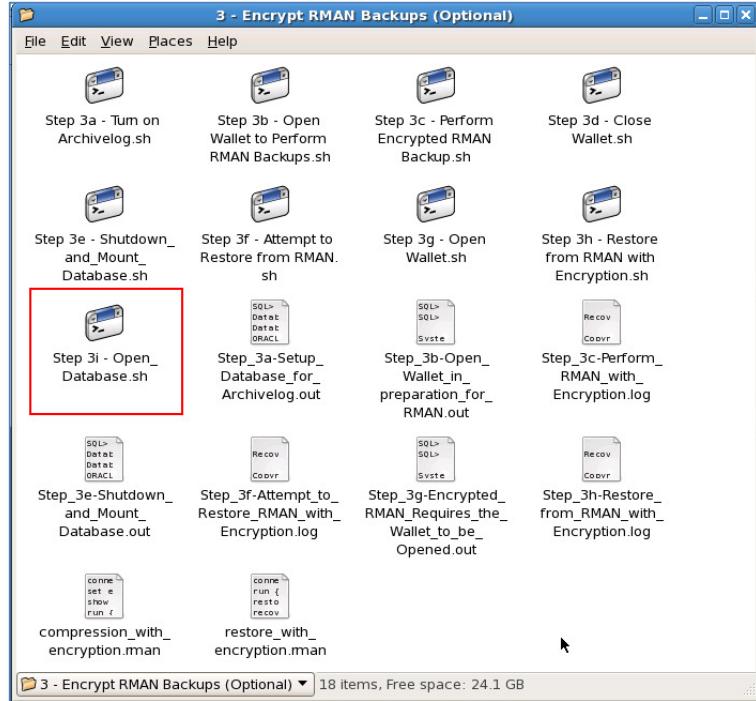
starting media recovery

archived log for thread 1 with sequence 134 is already on disk as file
/u01/oracle/oradata/db06/redo02.log
archived log for thread 1 with sequence 135 is already on disk as file
/u01/oracle/oradata/db06/redo03.log
archived log file name=/u01/oracle/oradata/db06/redo02.log thread=1
sequence=134
archived log file name=/u01/oracle/oradata/db06/redo03.log thread=1
sequence=135
media recovery complete, elapsed time: 00:00:01
Finished recover at 25-SEP-11

Recovery Manager complete.

```

37. Click on the icon, **Step 3i – Open_Databases.sh**. In this step, we are opening the database using the resetlogs option.



38. Click on the icon, **Step_3i-Open_Database.out** file in the folder to review the output of the previously executed script.



```
SQL> alter database open resetlogs;
Database altered.

SQL> exit;
```

D. Summary

In this lab, you completed the following:

1. Demonstrated the usage and expected characteristics of combining Tablespace Encryption within the Advanced Security Option and Advanced Compression.
2. Encrypted Data Pump archives using encryption from the Advanced Security Option to further protect information in created archives.
3. Encrypted RMAN using encryption from the Advanced Security Option to further protect information in created backups.

LAB CONFIGURATION – ORACLE DATABASE VAULT

OVERVIEW

For these lab exercises, we have already started the necessary infrastructure components for you:

- **11gR2 Database: Database DB06**
- All scripts used in this lab exercise can be found in the directory
/home/oracle/dbv_scripts.

Here is a summary of the users and their functions that will be used throughout this lab exercise.

- SYSDBA – Generic Database Administrator Account
- SEC_ADMIN_OWEN – Database Vault Security Administrator
- DBA_DEBRA – Sr. Database Administrator
- DBA_NICOLE – Jr. Database Administrator
- APPS_DBA_HARVEY – Human Resources Applications DBA
- APPS_DBA_SAM – Sales Applications DBA
- APPS_DBA_OLIVER – Order Entry Applications DBA
- SEC_ANALYST_ALLEN – Security Analyst for the Database Environment

Let's get started.

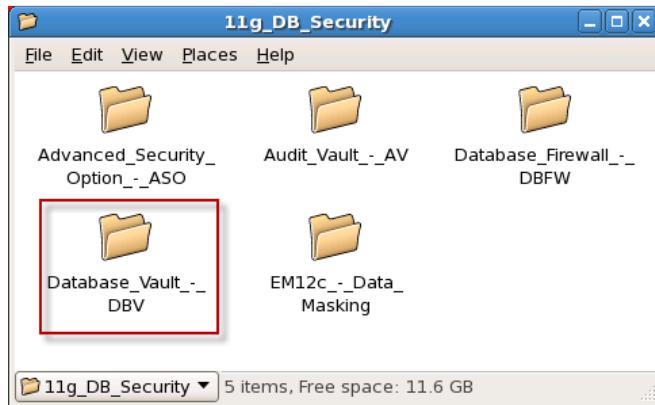
1. On the desktop, navigate to the **Labs** folder, double-click and open the contents.



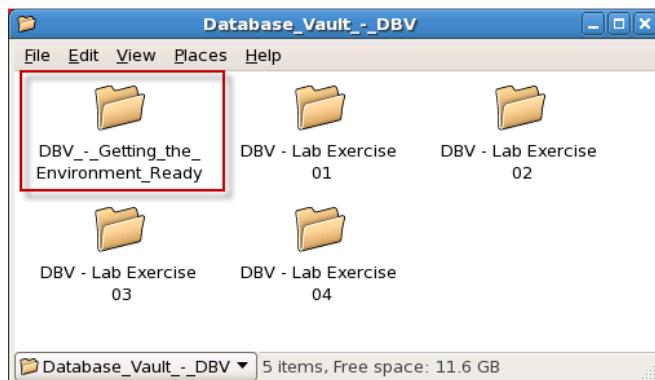
2. Select the folder, **11g_DB_Security**.



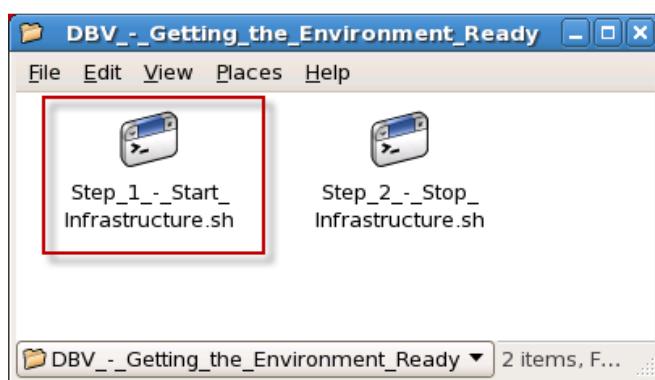
3. Within the **Oracle_Open_World_2011** folder, select the folder **Database_Vault_-_DBV**.



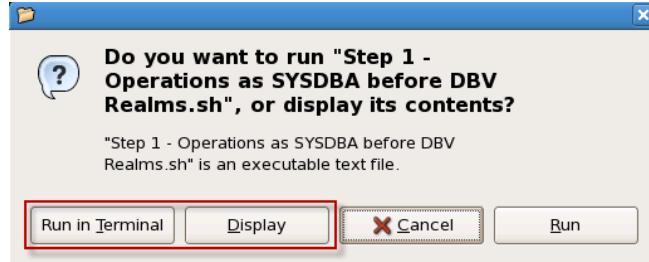
4. Within the **Database Vault – DBV** folder, you can access all of the Lab folders. Select '**DBV – Getting The Environment Ready**'.



5. Select '**Step_1_Start_Infrastructure.sh**'. This script will start the database and initialize the environment used in this lab.



6. In these lab exercises, when you click on the script icons, will have the option to either choose the **Display** button to review the scripts (including SQL) being executed and the **Run in Terminal** to execute them. We wanted to help avoid typing mistakes and provide you the opportunity to better understand what we're looking to accomplish.



You are now ready to begin your Database Vault labs.

LAB EXERCISE 00 – ORACLE DATABASE VAULT OVERVIEW

INTRODUCTION

Oracle Database Vault, part of Oracle's comprehensive portfolio of database security solutions, helps organizations address regulatory mandates and increase the security of existing applications. Regulations such as Sarbanes-Oxley, Payment Card Industry (PCI) Data Security Standard (DSS), Health Insurance Portability and Accountability Act (HIPAA), Gramm-Leach-Bliley Act (GLBA) and similar global directives call for giving access to sensitive data only a need-to-know basis and require separation-of-duties and other preventive controls to ensure data integrity and data privacy. With Oracle Database Vault, organizations can proactively safeguard application data stored in the Oracle database from being accessed by privileged database users. Application data can be further protected using Oracle Database Vault's multi-factor policies that control access based on built-in factors such as time of day, IP address, application name, and authentication method, preventing unauthorized ad-hoc access and application by-pass.

- Pro-actively safeguard application data stored in the Oracle database— Restrict access by unauthorized database users - even privileged users - by using powerful access controls built into the Oracle database.
- Address regulatory requirements—Implement separation-of-duty and other real-time preventive controls.
- Restrict ad-hoc access to application data— Prevent application-bypass with multi-factor policies that are enforced in the database for high security and performance.
- Deploy with confidence—Use certified default policies for Oracle E-Business Suite, Oracle PeopleSoft, and Oracle Siebel CRM applications.

A key challenge for security administrators is protecting enterprise data from insider attacks and external attacks that use compromised privileged database accounts to steal sensitive data. Oracle Database Vault is a database option that can be deployed with Oracle Database Enterprise Edition to help customers build internal controls to help meet regulatory requirements for privacy and compliance.

Oracle Database Vault prevents highly privileged users, including powerful DBAs and other database privileged users, from accessing sensitive applications data in Oracle databases outside their authorized responsibilities. You can use customizable Realms and rules to ensure that users, even administrators, have access only to what they need to do their job.

LAB EXERCISE 01 – PROTECTING SENSITIVE DATA FROM PRIVILEGED USER ACCESS USING ORACLE DATABASE VAULT REALMS

Identified Challenge – Strict Access Controls and Separation of Duties

Database system does not enforce the most restrictive set of rights/privileges or access needed by users (or process acting on behalf of users / business units) for the performance of specified tasks. (e.g. database administrators and other privileged users have access only to the data that they need to do their job and not have access to sensitive data)

Access to sensitive data, Separation of Duties (SOD) and Account Management SOD is uncontrolled and not enforced. Access Control Policy and enforcement needs to be established, documented and reviewed based on business and security requirements for access including separation of access authorization, access administration and audit functions.

INTRODUCTION

Highly Privileged User Controls

Database administrators and other highly privileged users play a critical role in maintaining the database. Backup and recovery, performance tuning, and high availability are all part of the DBA job description. However, the ability to prevent highly privileged users within the database from viewing sensitive application data has become an increasingly important requirement. In addition, application consolidation requires strong boundaries between sensitive business data such as that found in financial and human resource applications.

Oracle Database Vault Realms

Oracle Database Vault Realms prevent DBAs, application owners, and other privileged users from viewing application data using their powerful privileges. Database Vault Realms put in place preventive controls, helping reduce the potential impact when a data breach does occur, enabling the DBA to perform his or her job more effectively. Oracle Database Vault Realms can be used to protect an entire application or a specific set of tables within an application, providing highly flexible and adaptable security enforcement.

Oracle Database Vault Separation of Duty

Oracle Database Vault separation of duty enables a systematic approach to security that strengthens internal controls within the database. Out-of-the-box, Oracle Database Vault creates three distinct responsibilities within the database.

Responsibility	Description
Account Management	A user with the account management responsibility can create, drop, or modify database users. Existing highly privileged users will be prevented from performing account management activities.
Security Administrator	The security administration responsibility is designed to enable a user to become a security administrator (Database Vault Owner) of the database. A security administrator can setup Database Vault Realms, Command Rules, authorize others users to use them, and execute various Database Vault specific security reports. The security administrator is prevented from self-authorizing access to secured business data.
Resource Administration	The resource administration responsibility enables a user with the DBA privileges to continue performing normal management and maintenance associated with the database such backup and recovery, patching, and performance tuning.

Oracle Database Vault extensibility allows separation of duty to be customized to your specific business requirements. For example, you can further subdivide the resource administration responsibility into backup, performance and patching responsibilities. If you have a small company you can consolidate responsibilities, or assign different login accounts for each responsibility, enabling more granular accountability and auditing.

Oracle Database Vault provides numerous out-of-the-box reports that give you the ability to report on such things as attempted data access requests blocked by Realms. For example, if a DBA attempts to access data from an application table protected by a Realm, Database Vault will create an audit record in a specially protected table inside the Database Vault. Oracle Database Vault includes a Realm violation report that makes it easy to view these audit records.

Flexible and Extensible Access Controls

The proliferation of regulations and privacy laws around the globe requires flexible and highly adaptable security policies that can be easily modified to meet existing and newly emerging access control requirements. Further complicating access control requirements are issues such as out-sourcing and hosted or on-demand based applications. Oracle Database Vault introduces powerful capabilities that are uniquely suited to address these and future access control requirements.

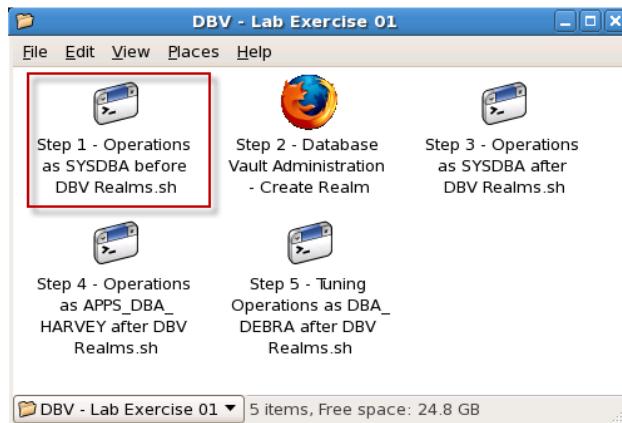
A. Overview

In this lab exercise, you will accomplish the following:

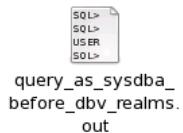
1. *Create a HR Realms to protect all HR objects—taking privilege away from the all powerful DBA.*
2. *Understand some behavior of Realms and Authorized Users.*

B. PROTECTING SENSITIVE DATA FROM PRIVILEGED USER ACCESS USING ORACLE DATABASE VAULT REALMS

1. We want to first show you the access of the SYSDBA user before we implement a Database Vault Realm. Click on the icon **Step 1 – Operations as SYSDBA before DBV Realms.sh** in the **DBV – Lab Exercise 01** Folder.



Click on the **query_as_sysdba_before_dbv_realms.out** to view the output of the scripts.



In this script, the SYSDBA was able to describe and select from the HR.EMPLOYEES table. In addition, the SYSDBA was able to produce an explain plan for analysis, create and drop tables in the HR schema.

```
SQL> show user;
USER is "SYS"
SQL>
SQL> desc hr.employees;
Name                           Null?    Type
-----
EMPLOYEE_ID                  NOT NULL NUMBER(6)
FIRST_NAME                    VARCHAR2(20)
LAST_NAME                     NOT NULL VARCHAR2(25)
EMAIL                          NOT NULL VARCHAR2(25)
PHONE_NUMBER                  VARCHAR2(20)
HIRE_DATE                     NOT NULL DATE
JOB_ID                        NOT NULL VARCHAR2(10)
SALARY                         NUMBER(8, 2)
COMMISSION_PCT                NUMBER(2, 2)
MANAGER_ID                     NUMBER(6)
DEPARTMENT_ID                 NUMBER(4)

SQL>
SQL> select last_name, email, phone_number, salary from hr.employees where
rownum < 5;

LAST_NAME                      EMAIL                            PHONE_NUMBER
-----
```

```

SALARY
-----
King           SKING          515.123.4567
      24000

Kochhar        NKOCHHAR       515.123.4568
      17000

De Haan        LDEHAAN        515.123.4569
      17000

LAST_NAME      EMAIL          PHONE_NUMBER
-----
SALARY
-----
Hunold         AHUNOLD        590.423.4567
      9000

```

```

SQL>
SQL> set autotrace on
SQL> select last_name, email, phone_number, salary from hr.employees where
rownum < 5;

LAST_NAME      EMAIL          PHONE_NUMBER
-----
SALARY
-----
King           SKING          515.123.4567
      24000

Kochhar        NKOCHHAR       515.123.4568
      17000

De Haan        LDEHAAN        515.123.4569
      17000

LAST_NAME      EMAIL          PHONE_NUMBER
-----
SALARY
-----
Hunold         AHUNOLD        590.423.4567
      9000

```

```

Execution Plan
-----
Plan hash value: 1424567464

-----
| Id  | Operation          | Name   | Rows  | Bytes | Cost (%CPU) | Time
|     |                   |        |       |       |             |       |
-----
|   0 | SELECT STATEMENT  |        |       |       |             | 0:00:00:01
| * 1 |  COUNT STOPKEY    |        |       |       |             |
|   2 | TABLE ACCESS FULL | EMPLOYEES |       |       |             | 0:00:00:01
|     |                   |        |       |       |             |
-----
```

```
Predicate Information (identified by operation id):
```

```
1 - filter(ROWNUM<5)
```

Statistics

```
0 recursive calls
0 db block gets
```

```

6  consistent gets
0  physical reads
0  redo size
772 bytes sent via SQL*Net to client
419 bytes received via SQL*Net from client
2  SQL*Net roundtrips to/from client
0  sorts (memory)
0  sorts (disk)
4  rows processed

SQL> set autotrace off
SQL>
SQL> create table hr.junk(id number);

Table created.

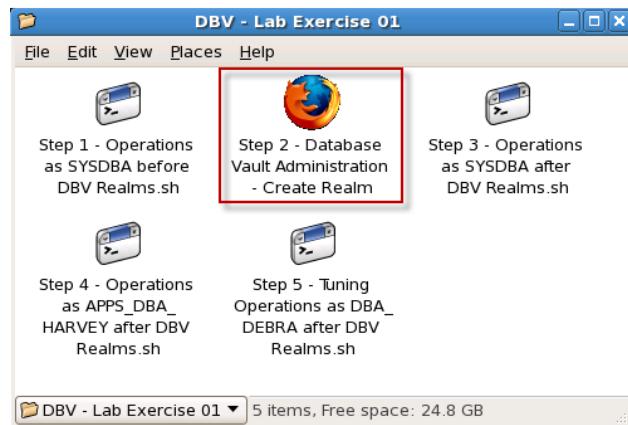
SQL> drop table hr.junk;

Table dropped.

SQL>
SQL> exit;

```

2. We will now step you through the process of creating a Realm to protect the HR schema. Click on the icon **Step 2 – Database Vault Administration – Create Realm.**



3. Provide the following required fields to login to the Database Vault Administration tool and click on the **Login** button to continue.
 - a. User Name: **SEC_ADMIN_OWEN**
 - b. Password: **Manager_1**
 - c. Host: **cloud.oracle.com**
 - d. Port: **1522**
 - e. SID: **db06.oracle.com**

Login

Login to Database:

* User Name

* Password

* Host

* Port

* SID / Service SID
 Service

- We will proceed to the Database Vault Administration to create a realm named **HR Realm**. We will ensure that all objects owned by the user HR are protected by this realm and make **APPS_DBA_HARVEY** the only authorized user. Click on the highlighted **Realms** link.

Administration Database Vault Reports General Security Reports Monitor

The links below allow you to protect applications and data using Oracle Database Vault features that include: Realms, Command Rules, Rule Sets, Factors, and Secure Application Roles.

Database Vault Feature Administration

Realms
[Command Rules](#)
[Factors](#)
[Rule Sets](#)
[Secure Application Roles](#)
[Label Security Integration](#)

Administration Database Vault Reports General Security Reports Monitor

- On the Realms screen, notice the default Oracle-defined Realms that have been installed during the Oracle Database Vault installation. Also notice that each Realm is configured to only “Audit on Failure” and objects and users have been defined for these “active” Realms. Click on the **Create** button to begin creating our HR Realm.

Realms

Database Vault realms provide a capability to classify database schemas and database roles into functional groups in order to provide fine-grained access control of the ability to use system level privileges against these types of database objects.

Select	Name	Audit Options	Oracle Defined Realm?	Objects Protected?	Users Authorized?	Status
<input checked="" type="radio"/>	Database Vault Account Management	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	Oracle Data Dictionary	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	Oracle Database Vault	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	Oracle Enterprise Manager	Audit On Failure	✓	✓	✓	✓

- Enter the **Name** and **Description** as shown for this Realm definition. Notice, but leave all other fields as defaults. Click on the **OK** button.

Create 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 Application

Description: Protect HR Application Data

Status: Enabled Disabled

Audit Options

Audit Disabled
 Audit On Failure
 Audit On Success or Failure

Cancel **OK**

- When you return to the Realm overview page, select the radio button selecting the 'HR Realm' created in the previous step and select the **Edit** button. Notice the red 'X' marks letting you know that neither objects nor users have been specified. We will now add the objects we want to protect and specify the users who are authorized to access this Realm.

Realms

Database Vault realms provide a capability to classify database schemas and database roles into functional groups in order to provide fine-grained access control of the ability to use system level privileges against these types of database objects.

Select Name	Audit Options	Oracle Defined Realm?	Objects Protected?	Users Authorized?	Status
<input type="radio"/> Database Vault Account Management	Audit On Failure	✓	✓	✓	✓
<input checked="" type="radio"/> HR Application	Audit On Failure		X	X	✓
<input type="radio"/> Oracle Data Dictionary	Audit On Failure	✓	✓	✓	✓
<input type="radio"/> Oracle Database Vault	Audit On Failure	✓	✓	✓	✓
<input type="radio"/> Oracle Enterprise Manager	Audit On Failure	✓	✓	✓	✓

Create **Edit** **Remove**

- Scroll down to the section labeled, **Realm Secured Objects** and click on the **Create** button.

Realm Secured Objects

Select Owner: No Items Found

Object Type

Object Name

Create

Realm Authorizations

Select Grantee: No Items Found

Authorization Options

Authorization Rule Set Name

Create

Cancel **OK**

- Select **HR** as the **Object Owner**, “%” as the **Object Type** to select all object types and “%” as the **Object Name** to select all names. Click on the **OK** button to continue.

Create Realm Secured Object

Define a database schema or database role that is protected by the realm.

Object Owner: HR

Object Type: %

Object Name: %

(Cancel) **OK**

As you can see, Database Vault Realms can provide the flexibility to protect both a broad range of Objects or very specific Objects.

- You will be brought back to the Realm definition screen where you can review the previous additions. Scroll down again to the **Realm Authorizations** section. Click on the **Create** button.

Select Owner	Object Type	Object Name
HR	%	%

Realm Authorizations

Select Grantee	Authorization Options	Authorization Rule Set Name
No Items Found		

Create

(Cancel) **OK**

- In the **Create Realm Authorization** screen, select **APPS_DBA_HARVEY [USER]** as the **Grantee**, select the **Participant** option for **Authorization Type** and click on the **OK** button. For this realm, we will not specify an Authorization Rule Set.

Create Realm Authorization

Define a database account or database role as either a realm owner or realm participant. Realm owners and realm participants can use their system privileges against realm secured objects. Only realm owners can grant or revoke realm secured database roles.

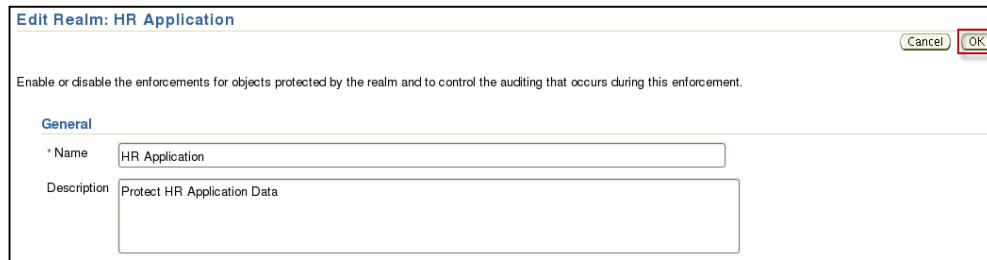
Grantee: APPS_DBA_HARVEY [USER]

Authorization Type: Participant Owner

Authorization Rule Set: <Non Selected>

(Cancel) **OK**

12. When you are brought back to the **Edit Realm: HR Realm** screen, click on the **OK** button to continue after reviewing the **HR Realm** definition.



13. Back in the Realms screen, you will notice the **HR Realm** has been configured with Objects and Users defined with the appropriate green check marks.

Select	Name	Audit Options	Oracle Defined Realm?	Objects Protected?	Users Authorized?	Status
<input type="radio"/>	Database Vault Account Management	Audit On Failure	✓	✓	✓	✓
<input checked="" type="radio"/>	HR Application	Audit On Failure		✓	✓	✓
<input type="radio"/>	Oracle Data Dictionary	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	Oracle Database Vault	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	Oracle Enterprise Manager	Audit On Failure	✓	✓	✓	✓

14. Minimize the browser window and locate the **DBV – Lab Exercise 01** folder again. Click on the icon, **Step 3 – Operations as SYSDBA after DBV Realms.sh** to review and execute the same sql scripts performed earlier to compare the differences.



15. Click on the icon **query_as_sysdba_after_DBV_realms.out** to review the output.

```
SQL>
SQL> show user;
USER is "SYS"

query_as_sysdba_
after_DBV_realms.
out
```

```

SQL>
SQL> desc hr.employees;
Name                           Null?    Type
-----
EMPLOYEE_ID                    NOT NULL NUMBER(6)
FIRST_NAME                      VARCHAR2(20)
LAST_NAME                       NOT NULL VARCHAR2(25)
EMAIL                            NOT NULL VARCHAR2(25)
PHONE_NUMBER                     VARCHAR2(20)
HIRE_DATE                        NOT NULL DATE
JOB_ID                           NOT NULL VARCHAR2(10)
SALARY                            NUMBER(8,2)
COMMISSION_PCT                  NUMBER(2,2)
MANAGER_ID                       NUMBER(6)
DEPARTMENT_ID                   NUMBER(4)

SQL>
SQL> select last_name, email, phone_number, salary from hr.employees where
rownum < 5;
select last_name, email, phone_number, salary from hr.employees where rownum <
5
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL>
SQL> set autotrace on
SQL> select last_name, email, phone_number, salary from hr.employees where
rownum < 5;
select last_name, email, phone_number, salary from hr.employees where rownum <
5
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL> set autotrace off
SQL>
SQL> create table hr.junk(id number);
create table hr.junk(id number)
*
ERROR at line 1:
ORA-47401: Realm violation for CREATE TABLE on HR.JUNK

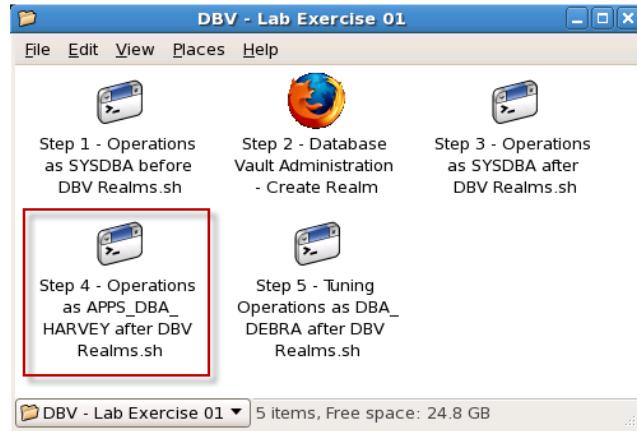
SQL> drop table hr.junk;
drop table hr.junk
*
ERROR at line 1:
ORA-47401: Realm violation for DROP TABLE on HR.JUNK

SQL>
SQL> exit;

```

Notice that Database Vault Realms are now protecting the HR Objects. Other than describing the table, the SYSDBA now does not have access privileges.

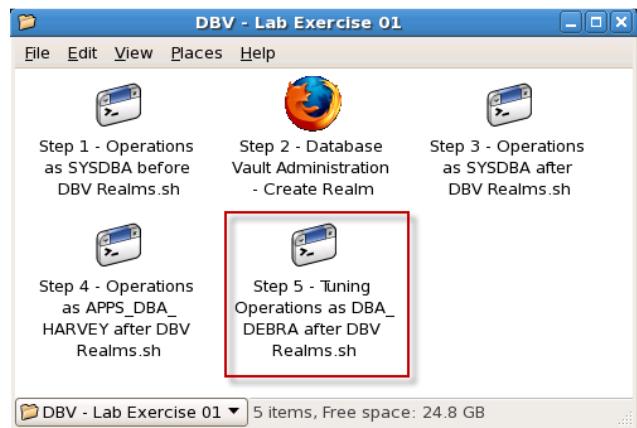
16. Click on the icon, **Step 4 – Operations as SYSDBA after DBV Realms.sh** to review and execute the same scripts. But this time, we are performing this with the user **APPS_DBA_HARVEY**. Remember, **APPS_DBA_HARVEY** was added as an Authorized user of this Realm.



17. Click on the icon **query_as_apps_dba_harvey_after_DBV_realms.out** to review the output. As expected, **APPS_DBA_HARVEY**, who is an Authorized user of the HR Application Realm has the authorization and access to perform the operations in the script.



18. Database Vault Realms are extremely powerful in the effort to enforce strict access controls and separation of duties policies found in numerous regulatory compliance and security best practices. This next example demonstrates how a Database Vault Realm provides protection while still allowing common management tasks to be completed. Click on the icon **Step 5 – Tuning Operations as DBA_DEBRA after DBV Realms.sh**.



19. Click on the icon, **tuning_operations_after_DBV_realms.out**. This Realm limits access to data on a 'Business Need-to-Know' basis-- while still providing the tools necessary to perform performance analysis and tuning. You will notice that the explain plan tables are stored in the DBA_DEBRA user schema and scripts provide the results as expected. Remember, DBA_DEBRA is *not* an authorized user of the HR Application Realm but she can still perform the task necessary without having access to the data.



```
SQL> show user;
USER is "DBA_DEBRA"
SQL>
SQL> drop table PLAN_TABLE;

Table dropped.

SQL> @/u01/oracle/product/11.2.0/dbhome_1/rdbms/admin/utlxplan.sql
SQL> rem
SQL> rem $Header: utlxplan.sql 08-may-2004.12:53:19 bdagevil Exp
$ xplainpl.sql
SQL> rem
SQL> Rem Copyright (c) 1988, 2004, Oracle. All rights reserved.
SQL> Rem NAME
SQL> REM      UTLXPLAN.SQL
SQL> Rem   FUNCTION
SQL> Rem   NOTES
SQL> Rem   MODIFIED
.
.
.
Table created.

.
.

SQL>
SQL> conn DBA_DEBRA/Manager_1
Connected.
SQL>
SQL>
SQL> explain plan set statement_id = 'Employee Count'
2      into plan_table
3      for select last_name, email, phone_number, salary from hr.employees
where rownum < 5;

Explained.

SQL>
SQL> @/u01/oracle/product/11.2.0/dbhome_1/rdbms/admin/utlxpls.sql
SQL> Rem
SQL> Rem $Header: utlxpls.sql 26-feb-2002.19:49:37 bdagevil Exp $
SQL> Rem
SQL> Rem utlxpls.sql
SQL> Rem
SQL> Rem Copyright (c) 1998, 2002, Oracle Corporation. All rights reserved.
SQL> Rem
SQL> Rem   NAME
SQL> Rem      utlxpls.sql - UTILITY eXPPLAIN Serial plans
SQL> Rem
SQL> Rem   DESCRIPTION
SQL> Rem      script utility to display the explain plan of the last
explain plan
SQL> Rem      command. Do not display information related to Parallel Query
SQL> Rem
.
.
```

```

PLAN_TABLE_OUTPUT
-----
-----  

Plan hash value: 1424567464  

-----  

-----  

| Id  | Operation          | Name      | Rows  | Bytes | Cost (%CPU) | Time  

|  

-----  

|   0 | SELECT STATEMENT   |           |       4 |    140 |         2  (0) |  

00:00:01 |  

|* 1 |  COUNT STOPKEY     |           |       |        |  

|  

|   2 | TABLE ACCESS FULL | EMPLOYEES |       4 |    140 |         2  (0) |  

00:00:01 |
-----  

-----  

Predicate Information (identified by operation id):  

PLAN_TABLE_OUTPUT
-----
-----  

-----  

1 - filter(ROWNUM<5)  

14 rows selected.

```

C. Summary

You accomplished the following in this lab exercise:

1. Created a HR Realms to protect all HR objects—taking privilege away from the all powerful DBA.
2. Understood through examples some behavior of Realms and Authorized Users.

LAB EXERCISE 02 – USING ORACLE DATABASE VAULT REALMS TO ENABLE DATABASE CONSOLIDATION

Identified Challenge

Privilege should be allocated only as necessary to carry out their functional role only when needed and additional factors such as IP address, source application, user/role can be enforced.

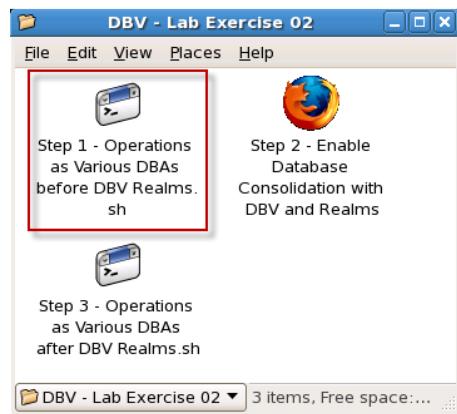
A. Overview

In this lab exercise, you will accomplish the following:

1. *Create Realms to protect the Sales, Order Entry and HR databases*
2. *Demonstrate how Realms limit authorized access to only specified application database administrators and limit their ability to access data beyond ‘Business Need to Know’*

B. USING ORACLE DATABASE VAULT REALMS TO ENABLE DATABASE CONSOLIDATION

1. We want to first show you the access of the SYSDBA user before we implement the Database Vault Realms in the Sales and Order Entry schemas. Click on the icon **Step 1 – Operations as Various DBAs before DBV Realms.sh** in the **DBV – Lab Exercise 02** Folder.



Click on the **operations_as_various_DBAs_before_dbv.out** to view the output of the scripts.



There are a few things we want to point out in the output of the script. First, you will notice that DBA_DEBRA who has DBA privileges attempted unsuccessfully to create the user MALICIOUS_MALFOY.

To meet regulatory, privacy and other compliance requirements, Oracle Database Vault implements the concept of separation of duty. Oracle Database Vault makes clear separation between the account management responsibility, data security responsibility, and database resource management responsibility inside the database. The concept of a privileged user (for example, DBA) is divided among several new database roles to ensure no one user has full control over both the data and configuration of the system.

Oracle Database Vault prevents the SYS user and other accounts with the DBA role and other system privileges from designated protected areas of the database called realms. It also introduces new database roles called the Oracle Database Vault Owner (DV_OWNER) and the Oracle Database Vault Account Manager (DV_ACCTMGR). These new database roles separate the data security and the account management from the traditional DBA role.

```
SQL>
SQL> -----
-----  
SQL> --As the SYSDBA, attempt to create a Malicious User  
SQL> -----
-----  
SQL> create user MALICIOUS_MALFOY identified by Manager_1;  
create user MALICIOUS_MALFOY identified by Manager_1  
          *  
ERROR at line 1:  
ORA-01031: insufficient privileges
```

In this script, you can see where both APPS_DBA_SAM and APPS_DBA OLIVER, who have responsibility for the Sales and the Order Entry schemas respectively, have access to both to each other's data and resources. This is clearly a challenge to address when looking to consolidating databases.

```
SQL>
SQL> -----
-----  
SQL> --As the APPS_DBA_SAM user, demonstrate how this user WITHOUT DBV has  
control to other business data.  
SQL> -----
-----  
SQL> conn APPS_DBA_SAM/Manager_1  
Connected.  
SQL> show user;  
USER is "APPS_DBA_SAM"  
SQL>  
SQL> select cust_id, quantity_sold, amount_sold from sh.sales where rownum < 5;  

      CUST_ID QUANTITY SOLD AMOUNT SOLD  
-----  
      987           1       1232.16  
     1660           1       1232.16  
     1762           1       1232.16  
     1843           1       1232.16
```

```

SQL> select cust_last_name, credit_limit, date_of_birth, income_level from
oe.customers where rownum < 5;

CUST_LAST_NAME      CREDIT_LIMIT DATE_OF_B INCOME_LEVEL
-----
Roberts              600 21-MAR-44 G: 130,000 - 149,999
Steenburghen         600 10-APR-50 F: 110,000 - 129,999
Rampling             600 20-APR-41 D: 70,000 - 89,999
Slater               700 11-MAY-51 D: 70,000 - 89,999

SQL>
SQL> create table oe.junk(id number);

Table created.

SQL> drop table oe.junk;

Table dropped.

SQL>
SQL> -----
SQL> --As the APPS_DBA OLIVER user, demonstrate how this user WITHOUT DBV has
control to other business data.
SQL> -----
SQL> conn APPS_DBA OLIVER/Manager_1
Connected.
SQL> show user;
USER is "APPS_DBA OLIVER"
SQL>
SQL> select cust_id, quantity_sold, amount_sold from sh.sales where rownum < 5;

CUST_ID QUANTITY SOLD AMOUNT SOLD
-----
987          1        1232.16
1660         1        1232.16
1762         1        1232.16
1843         1        1232.16

SQL> select cust_last_name, credit_limit, date_of_birth, income_level from
oe.customers where rownum < 5;

CUST_LAST_NAME      CREDIT_LIMIT DATE_OF_B INCOME_LEVEL
-----
Roberts              600 21-MAR-44 G: 130,000 - 149,999
Steenburghen         600 10-APR-50 F: 110,000 - 129,999
Rampling             600 20-APR-41 D: 70,000 - 89,999
Slater               700 11-MAY-51 D: 70,000 - 89,999

SQL>
SQL> create table oe.junk(id number);

Table created.

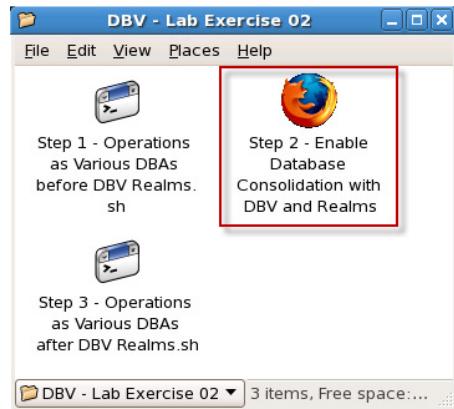
SQL> drop table oe.junk;

Table dropped.

SQL>
SQL> exit;

```

2. We will now use the techniques we used in the previous lab to create two additional Realms. Click on the icon **Step 2 – Enable Database Consolidation with DBV and Realms**.



3. Provide the following required fields to login to the Database Vault Administration tool and click on the **Login** button to continue.
- User Name: **SEC_ADMIN_OWEN**
 - Password: **Manager_1**
 - Host: **cloud.oracle.com**
 - Port: **1522**
 - SID: **db06**

* User Name	<input type="text" value="SEC_ADMIN_OWEN"/>
* Password	<input type="password" value="*****"/>
* Host	<input type="text" value="cloud.oracle.com"/>
* Port	<input type="text" value="1522"/>
* SID / Service	<input type="radio"/> SID <input type="text"/> <input checked="" type="radio"/> Service <input type="text" value="db06.oracle.com"/>
<input type="button" value="Login"/>	

- We will proceed to the Database Vault Administration to create a realm named **Sales Application Realm**. We will ensure that all objects owned by the user **SH** are protected by this realm and make **APPS_DBA_SAM** the only authorized user. Click on the highlighted **Realms** link.

The screenshot shows the Oracle Database Vault Feature Administration interface. At the top, there are tabs: Administration, Database Vault Reports, General Security Reports, and Monitor. Below the tabs, a message states: "The links below allow you to protect applications and data using Oracle Database Vault features that include: Realms, Command Rules, Rule Sets, Factors, and Secure Application Roles." A sidebar on the left lists several options: Realms (which is highlighted with a red box), Command Rules, Factors, Rule Sets, Secure Application Roles, and Label Security Integration. At the bottom, there is another set of tabs: Administration, Database Vault Reports, General Security Reports, and Monitor.

- Click on the **Create** button to begin creating our Sales Application Realm.

The screenshot shows the Realms list page. At the top, it says "Realms" and provides a brief description: "Database Vault realms provide a capability to classify database schemas and database roles into functional groups in order to provide fine-grained access control of the ability to use system level privileges against these types of database objects." Below this is a table with columns: Select, Name, Audit Options, Oracle Defined Realm?, Objects Protected?, Users Authorized?, and Status. The table contains five rows, each with a radio button next to the name. The "Create" button is located at the top right of the table area.

Select	Name	Audit Options	Oracle Defined Realm?	Objects Protected?	Users Authorized?	Status
<input checked="" type="radio"/>	Database Vault Account Management	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	HR Application	Audit On Failure		✓	✓	✓
<input type="radio"/>	Oracle Data Dictionary	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	Oracle Database Vault	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	Oracle Enterprise Manager	Audit On Failure	✓	✓	✓	✓

- Enter the **Name** and **Description** as shown for this Realm definition. Notice, but leave all other fields as defaults. Click on the **OK** button.

The screenshot shows the "Create Realm" dialog box. It has a "General" tab with the following fields: "Name" set to "Sales Application", "Description" set to "Protect Sales (SH) Application Data", and "Status" set to "Enabled". The "OK" button is highlighted with a red box at the top right of the dialog.

- When you return to the Realm overview page, select the radio button selecting the '**Sales Realm**' created in the previous step and select the **Edit** button. Notice the red '**X**' marks letting you know that neither objects nor users have been specified. We will now add the objects we want to protect and specify the users who are authorized to access this Realm.

Select Name	Audit Options	Oracle Defined Realm?	Objects Protected?	Users Authorized?	Status
Database Vault Account Management	Audit On Failure	✓	✓	✓	✓
HR Application	Audit On Failure		✓	✓	✓
Oracle Data Dictionary	Audit On Failure	✓	✓	✓	✓
Oracle Database Vault	Audit On Failure	✓	✓	✓	✓
Oracle Enterprise Manager	Audit On Failure	✓	✓	✓	✓
Sales Application	Audit On Failure		x	x	✓

- Scroll down to the section labeled, **Realm Secured Objects** and click on the **Create** button.

Realm Secured Objects

Select Owner	Object Type	Object Name
No Items Found		

Realm Authorizations

Select Grantee	Authorization Options	Authorization Rule Set Name
No Items Found		

(Cancel) (OK)

- Select **SH** as the **Object Owner**, "%" as the **Object Type** to select all object types and "%" as the **Object Name** to select all names. Click on the **OK** button to continue.

Create Realm Secured Object

Define a database schema or database role that is protected by the realm.

Object Owner	SH
Object Type	%
Object Name	%

(Cancel) (OK)

10. You will be brought back to the Realm definition screen where you can review the previous additions. Scroll down again to the **Realm Authorizations** section. Click on the **Create** button.

Select Owner	Object Type	Object Name
<input checked="" type="radio"/> HR	%	%

Select Grantee	Authorization Options	Authorization Rule Set Name
No Items Found		

Create **Cancel** **OK**

11. In the **Create Realm Authorization** screen, select **APPS_DBA_SAM [USER]** as the **Grantee**, select the **Participant** option for **Authorization Type** and click on the **OK** button. For this realm, we will not specify an Authorization Rule Set.

Create Realm Authorization

Define a database account or database role as either a realm owner or realm participant. Realm owners and realm participants can use their system privileges against realm secured objects. Only realm owners can grant or revoke realm secured database roles.

Grantee
APPS_DBA_SAM [USER]

Authorization Type
 Participant
 Owner

Authorization Rule Set
<Non Selected>

OK **Cancel**

12. When you are brought back to the **Edit Realm: Sales Application Realm** screen, click on the **OK** button to continue after reviewing the **Sales Realm** definition.

Edit Realm: Sales Application

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

General

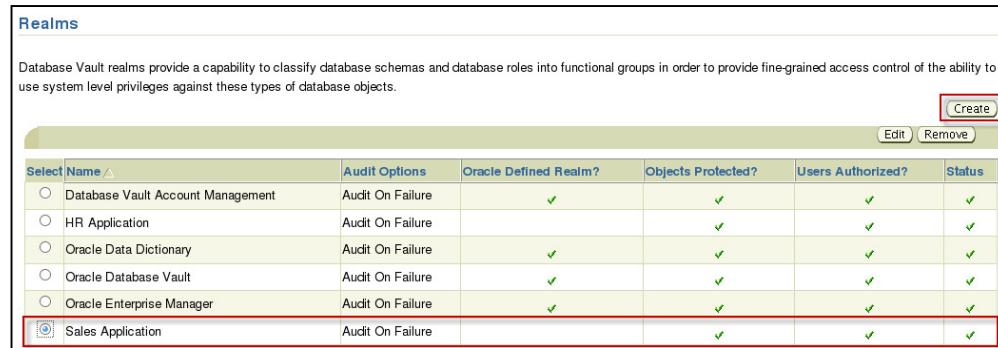
Name: Sales Application

Description: Protect Sales (SH) Application Data

Status: Enabled
 Disabled

OK **Cancel**

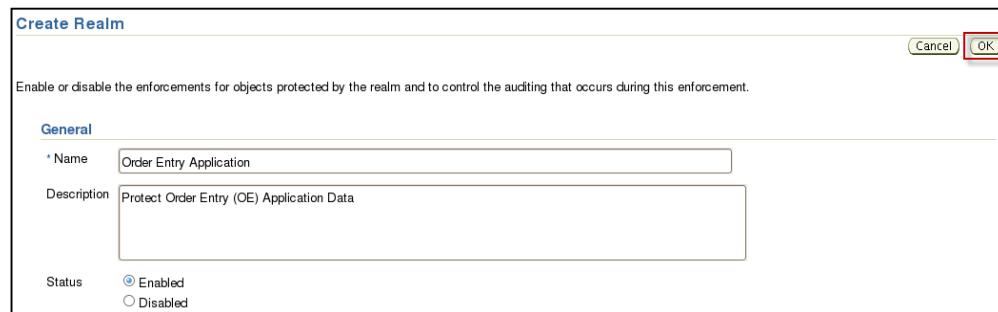
13. Back in the Realms screen, you will notice the **Sales Application Realm** has been configured with Objects and Users defined with the appropriate green check marks. We will repeat these quick steps to protect the Order Entry database.



The screenshot shows the 'Realms' page with a table of realms. The 'Sales Application' row is highlighted with a red border. The table columns are: Select, Name, Audit Options, Oracle Defined Realm?, Objects Protected?, Users Authorized?, and Status. The 'Sales Application' row has 'Audit On Failure' in the Audit Options column and a green checkmark in the Objects Protected? column. The other rows show various Oracle components with green checkmarks in all columns except for the Oracle Data Dictionary which has a red 'X' in the Objects Protected? column.

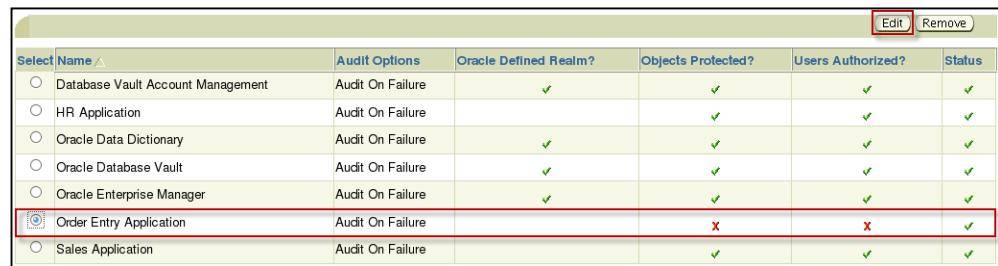
Select	Name	Audit Options	Oracle Defined Realm?	Objects Protected?	Users Authorized?	Status
<input type="radio"/>	Database Vault Account Management	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	HR Application	Audit On Failure		✓	✓	✓
<input type="radio"/>	Oracle Data Dictionary	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	Oracle Database Vault	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	Oracle Enterprise Manager	Audit On Failure	✓	✓	✓	✓
<input checked="" type="radio"/>	Sales Application	Audit On Failure		✓	✓	✓

14. Enter the **Name** and **Description** as shown for this Realm definition. Notice, but leave all other fields as defaults. Click on the **OK** button.



The screenshot shows the 'Create Realm' dialog box. It contains a 'General' section with fields for Name ('Order Entry Application'), Description ('Protect Order Entry (OE) Application Data'), and Status ('Enabled'). The 'OK' button is highlighted with a red border.

15. When you return to the Realm overview page, select the radio button selecting the '**Order Entry Application Realm**' created in the previous step and select the **Edit** button. Notice the red 'X' marks letting you know that neither objects nor users have been specified. We will now add the objects we want to protect and specify the users who are authorized to access this Realm.



The screenshot shows the 'Realms' page with a table of realms. The 'Order Entry Application' row is highlighted with a red border. The table columns are: Select, Name, Audit Options, Oracle Defined Realm?, Objects Protected?, Users Authorized?, and Status. The 'Order Entry Application' row has 'Audit On Failure' in the Audit Options column and red 'X' marks in the Objects Protected? and Users Authorized? columns. The other rows show various Oracle components with green checkmarks in all columns except for the Oracle Data Dictionary which has a red 'X' in the Objects Protected? column.

Select	Name	Audit Options	Oracle Defined Realm?	Objects Protected?	Users Authorized?	Status
<input type="radio"/>	Database Vault Account Management	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	HR Application	Audit On Failure		✓	✓	✓
<input type="radio"/>	Oracle Data Dictionary	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	Oracle Database Vault	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	Oracle Enterprise Manager	Audit On Failure	✓	✓	✓	✓
<input checked="" type="radio"/>	Order Entry Application	Audit On Failure		X	X	✓
<input type="radio"/>	Sales Application	Audit On Failure		✓	✓	✓

16. Scroll down to the section labeled, **Realm Secured Objects** and click on the **Create** button.

The screenshot shows the 'Realm Secured Objects' section of the Oracle Database configuration tool. It consists of two tables. The first table has columns 'Select Owner', 'Object Type', and 'Object Name'. The second table has columns 'Select Grantee', 'Authorization Options', and 'Authorization Rule Set Name'. Both tables show 'No Items Found'. In the top right corner of the first table's header, there is a red-bordered 'Create' button.

17. Select **OE** as the **Object Owner**, “%” as the **Object Type** to select all object types and “%” as the **Object Name** to select all names. Click on the **OK** button to continue.

This is a 'Create Realm Secured Object' dialog box. It contains three input fields: 'Object Owner' (set to 'OE'), 'Object Type' (set to '%'), and 'Object Name' (set to '%'). The 'OK' button in the top right corner is also highlighted with a red box. The text 'Define a database schema or database role that is protected by the realm.' is displayed above the fields.

18. You will be brought back to the Realm definition screen where you can review the previous additions. Scroll down again to the **Realm Authorizations** section. Click on the **Create** button.

This screenshot shows the main realm configuration screen. It includes sections for 'Realm Secured Objects' and 'Realm Authorizations'. In the 'Realm Secured Objects' section, there is a table with columns 'Select Owner', 'Object Type', and 'Object Name'. One row shows 'HR' selected in 'Select Owner', '%' in 'Object Type', and '%' in 'Object Name'. In the 'Realm Authorizations' section, there is another table with columns 'Select Grantee', 'Authorization Options', and 'Authorization Rule Set Name'. The 'Create' button in the top right corner of this table is highlighted with a red box.

19. In the **Create Realm Authorization** screen, select **APPS_DBA OLIVER [USER]** as the **Grantee**, select the **Participant** option for **Authorization Type** and click on the **OK** button. For this realm, we will not specify an Authorization Rule Set.

Create Realm Authorization

Define a database account or database role as either a realm owner or realm participant. Realm owners and realm participants can use their system privileges against realm secured objects. Only realm owners can grant or revoke realm secured database roles.

Grantee
APPS_DBA OLIVER [USER]

Authorization Type
 Participant
 Owner

Authorization Rule Set
<Non Selected>

Cancel **OK**

20. When you are brought back to the **Edit Realm: Order Application Realm** screen, click on the **OK** button to continue after reviewing the **Order Application Realm** definition.

Edit Realm: Order Entry Application

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

General

Name: Order Entry Application

Description: Protect Order Entry (OE) Application Data

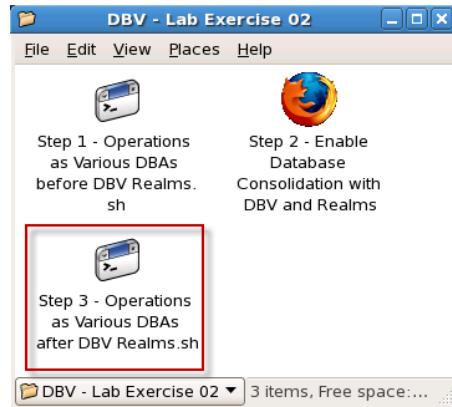
Status: Enabled
 Disabled

Cancel **OK**

21. Back in the Realms screen, you will notice the **Order Entry Realm** has been configured with Objects and Users defined with the appropriate green check marks. Now, all three Realms have been configured to protect the Human Resources (HR), Order Entry (OE) and Sales (SH) applications.

Select	Name	Audit Options	Oracle Defined Realm?	Objects Protected?	Users Authorized?	Status
<input checked="" type="radio"/>	Database Vault Account Management	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	HR Application	Audit On Failure		✓	✓	✓
<input type="radio"/>	Oracle Data Dictionary	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	Oracle Database Vault	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	Oracle Enterprise Manager	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	Order Entry Application	Audit On Failure		✓	✓	✓
<input type="radio"/>	Sales Application	Audit On Failure		✓	✓	✓

22. Minimize the browser window and locate the **DBV – Lab Exercise 01** folder again. Click on the icon, **Step 3 – Operations as Various DBAs after DBV Realms.sh** to review and execute the same sql scripts performed earlier to compare the differences.



23. Click on the icon **query_as_various_dbas_after_DBV.out** to review the output.



You will notice that we have successfully demonstrated how Database Vault Realms are now protecting the Human Resources (HR), Order Entry (OE) and Sales (SH) application databases in a consolidated environment. We accomplished exactly what we set out to do.

```
SQL>
SQL> -----
-----+
SQL> --As the APPS_DBA_SAM user, demonstrate how after DBV, there is control
over data.
SQL> -----
-----+
SQL> conn APPS_DBA_SAM/Manager_1
Connected.
SQL> show user;
USER is "APPS_DBA_SAM"
SQL>
SQL> select cust_id, quantity_sold, amount_sold from sh.sales where rownum < 5;

CUST_ID QUANTITY SOLD AMOUNT SOLD
-----
987          1      1232.16
1660         1      1232.16
1762         1      1232.16
1843         1      1232.16

SQL> select cust_last_name, credit_limit, date_of_birth, income_level from
oe.customers where rownum < 5;
select cust_last_name, credit_limit, date_of_birth, income_level from
oe.customers where rownum < 5
*
ERROR at line 1:
ORA-01031: insufficient privileges
```

```

SQL> select last_name, email, phone_number, salary from hr.employees where
rownum < 5;
select last_name, email, phone_number, salary from hr.employees where rownum <
5
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL>
SQL> create table sh.junk(id number);
create table sh.junk(id number)
*
ERROR at line 1:
ORA-00955: name is already used by an existing object

SQL> drop table oe.junk;
drop table oe.junk
*
ERROR at line 1:
ORA-47401: Realm violation for DROP TABLE on OE.JUNK

SQL>
SQL> create table oe.junk(id number);
create table oe.junk(id number)
*
ERROR at line 1:
ORA-47401: Realm violation for CREATE TABLE on OE.JUNK

SQL> drop table oe.junk;
drop table oe.junk
*
ERROR at line 1:
ORA-47401: Realm violation for DROP TABLE on OE.JUNK

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

SQL> drop table oe.junk;
drop table oe.junk
*
ERROR at line 1:
ORA-47401: Realm violation for DROP TABLE on OE.JUNK

SQL>
SQL> -----
-----  

SQL> --As the APPS_DBA_OLIVER user, demonstrate how after DBV, there is control
over data.  

SQL> -----
-----  

SQL> conn APPS_DBA_OLIVER/Manager_1
Connected.
SQL> show user;
USER is "APPS_DBA_OLIVER"
SQL>
SQL> select cust_last_name, credit_limit, date_of_birth, income_level from
oe.customers where rownum < 5;

CUST_LAST_NAME      CREDIT_LIMIT DATE_OF_B INCOME_LEVEL
-----  -----  -----  -----
Roberts              600 21-MAR-44 G: 130,000 - 149,999
Steenburghen         600 10-APR-50 F: 110,000 - 129,999
Rampling             600 20-APR-41 D: 70,000 - 89,999
Slater               700 11-MAY-51 D: 70,000 - 89,999

```

```

SQL> select last_name, email, phone_number, salary from hr.employees where
rownum < 5;
select last_name, email, phone_number, salary from hr.employees where rownum <
5
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL> select cust_id, quantity_sold, amount_sold from sh.sales where rownum < 5;
select cust_id, quantity_sold, amount_sold from sh.sales where rownum < 5
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL>
SQL> create table oe.junk(id number);

Table created.

SQL> drop table oe.junk;

Table dropped.

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

SQL> drop table oe.junk;
drop table oe.junk
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> create table sh.junk(id number);
create table sh.junk(id number)
*
ERROR at line 1:
ORA-47401: Realm violation for CREATE TABLE on SH.JUNK

SQL> drop table oe.junk;
drop table oe.junk
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> -----
-----  

SQL> --As the APPS_DBA_HARVEY user, demonstrate how after DBV, there is control
over data.
SQL> -----
-----  

SQL> conn APPS_DBA_OLIVER/Manager_1
Connected.
SQL> show user;
USER is "APPS_DBA_OLIVER"
SQL>
SQL> select last_name, email, phone_number, salary from hr.employees where
rownum < 5;
LAST_NAME          EMAIL          PHONE_NUMBER
-----  

-----  

SALARY
-----  

King              SKING          515.123.4567
24000
Kochhar           NKOCHHAR        515.123.4568

```

```

17000

De Haan          LDEHAAN        515.123.4569
17000

LAST_NAME        EMAIL          PHONE_NUMBER
-----
SALARY

Hunold           AHUNOLD       590.423.4567
9000

SQL> select cust_id, quantity_sold, amount_sold from sh.sales where rownum < 5;
select cust_id, quantity_sold, amount_sold from sh.sales where rownum < 5
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL> select cust_last_name, credit_limit, date_of_birth, income_level from
oe.customers where rownum < 5;
select cust_last_name, credit_limit, date_of_birth, income_level from
oe.customers where rownum < 5;

ERROR at line 1:
ORA-01031: insufficient privileges

SQL>
SQL> create table oe.junk(id number);

Table created.

SQL> drop table oe.junk;

Table dropped.

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

SQL> drop table oe.junk;
drop table oe.junk
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> create table sh.junk(id number);
create table sh.junk(id number)
*
ERROR at line 1:
ORA-47401: Realm violation for CREATE TABLE on SH.JUNK

SQL> drop table oe.junk;
drop table oe.junk
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> -----
-----  

SQL> --As the '/' as sysdba' user, demonstrate how after DBV, there is control
over data.
SQL> -----
-----  

SQL> conn / as sysdba
Connected.
SQL> show user;

```

```

USER is "SYS"
SQL>
SQL> select last_name, email, phone_number, salary from hr.employees where
rownum < 5;
select last_name, email, phone_number, salary from hr.employees where rownum <
5
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL> select cust_id, quantity_sold, amount_sold from sh.sales where rownum < 5;
select cust_id, quantity_sold, amount_sold from sh.sales where rownum < 5
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL> select cust_last_name, credit_limit, date_of_birth, income_level from
oe.customers where rownum < 5;
select cust_last_name, credit_limit, date_of_birth, income_level from
oe.customers where rownum < 5
*
ERROR at line 1:
ORA-01031: insufficient privileges

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

SQL> drop table oe.junk;
drop table oe.junk
*
ERROR at line 1:
ORA-47401: Realm violation for DROP TABLE on OE.JUNK

SQL>
SQL> create table sh.junk(id number);
create table sh.junk(id number)
*
ERROR at line 1:
ORA-47401: Realm violation for CREATE TABLE on SH.JUNK

SQL> drop table oe.junk;
drop table oe.junk
*
ERROR at line 1:
ORA-47401: Realm violation for DROP TABLE on OE.JUNK

SQL>
SQL> create table oe.junk(id number);
create table oe.junk(id number)
*
ERROR at line 1:
ORA-47401: Realm violation for CREATE TABLE on OE.JUNK

SQL> drop table oe.junk;
drop table oe.junk
*
ERROR at line 1:
ORA-47401: Realm violation for DROP TABLE on OE.JUNK

SQL>
SQL> exit;

```

C. Summary

You accomplished the following in this lab exercise:

1. *Created Realms to protect the Sales, Order Entry and HR databases.*
2. *Demonstrated how Realms limit authorized access to only specified application database administrators and limit their ability to access data beyond 'Business Need to Know'.*

LAB EXERCISE 03 – ENFORCING OPERATIONAL CONTROLS USING ORACLE DATABASE VAULT MULTI-FACTOR AUTHORIZATION

Identified Challenge

Privilege should be allocated only as necessary to carry out their functional role only when needed and additional factors such as IP address, source application, user/role can be enforced.

INTRODUCTION

Oracle Database Vault Multi-Factor Authorization

Oracle Database Vault Multi-Factor Authorization extends access controls beyond the traditional role based and even more sophisticated label based access control found in the Oracle Database. Using multi-factor authorization, access to databases can be restricted to a specific subnet or application server, creating a virtual trusted path for data access. Limiting data access to approved applications can be achieved using Oracle Database Vault factors in combination with Oracle Database Vault Command Rules. Oracle Database Vault provides a number of built-in Factors, such as IP address, that can be used individually or together in combination with other security rules to significantly raise the level security for an existing application. In addition to the built-in Factors provided by Database Vault, you can add your own custom factors to meet your own business requirements.

Oracle Database Vault Command Rules

Oracle Database Vault Command Rules provide the ability to easily attach security policies to virtually any database operation. Command Rules allow you to strengthen internal controls and enforce industry best practices and secure configuration policies. Command Rules can be used to enforce strong protections on critical business data. For example, a command rule can be used to prevent any user, even the DBA, from dropping application tables in your production environment. Command Rules can be easily managed through the Database Vault GUI or on the command line using the API.

A. Overview

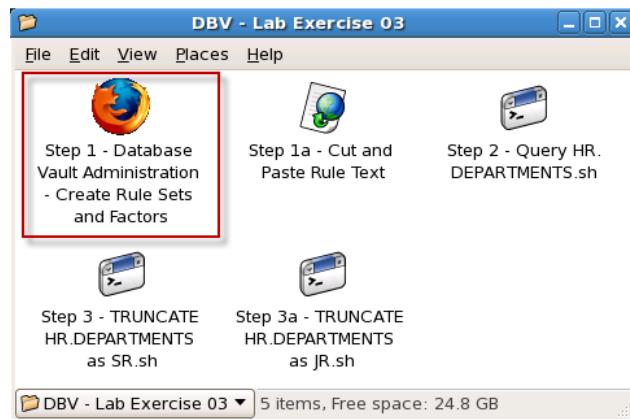
In this lab we will implement the above security requirements using a combination of Database Vault realms, rule sets, and command rules.

In this lab exercise, you will accomplish the following:

1. *Understand some behavior of Factors and Rule Sets and their application to Command Rules.*
2. *Create a Rule set that will restrict the ability of the DBA_NICOLE (Junior Database Administrator) user to TRUNCATE a table only when the following conditions are met:
 - a. The time and date must be within 8-5, Monday through Friday
 - b. Must be logged into the database via localhost and not remotely accessed*

B. ENFORCING OPERATIONAL CONTROLS USING ORACLE DATABASE VAULT MULTI-FACTOR AUTHORIZATION

1. Click on the icon **Step 1 – Database Vault Administration – Create Rule Sets and Factors** in the **DBV – Lab Exercise 03** Folder to log into the Database Vault Web Administration (DVA) Console.



2. Provide the following required fields to login to the Database Vault Administration tool and click on the **Login** button to continue.
 - a. User Name: **SEC_ADMIN_Owen**
 - b. Password: **Manager_1**
 - c. Host: **cloud.oracle.com**
 - d. Port: **1522**
 - e. SID: **db06**

Login

Login to Database:

* User Name: SEC_ADMIN_OWEN

* Password:

* Host: cloud.oracle.com

* Port: 1522

* SID / Service: SID Service db06.oracle.com

Login

- We are going to edit the **HR Application** Realm that we created in the previous exercise. Click on the highlighted **Realms** link.

Administration Database Vault Reports General Security Reports Monitor

The links below allow you to protect applications and data using Oracle Database Vault features that include: Realms, Command Rules, Rule Sets, Factors, and Secure Application Roles.

Database Vault Feature Administration

Realms Command Rules Factors Rule Sets Secure Application Roles Label Security Integration

Administration Database Vault Reports General Security Reports Monitor

- On the **Realms** screen, select the **HR Application** Realm radio button and click on the **Edit** button.

Select	Name	Audit Options	Oracle Defined Realm?	Objects Protected?	Users Authorized?	Status
<input type="radio"/>	Database Vault Account Management	Audit On Failure	✓	✓	✓	✓
<input checked="" type="radio"/>	HR Application	Audit On Failure		✓	✓	✓
<input type="radio"/>	Oracle Data Dictionary	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	Oracle Database Vault	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	Oracle Enterprise Manager	Audit On Failure	✓	✓	✓	✓
<input type="radio"/>	Order Entry Application	Audit On Failure		✓	✓	✓
<input type="radio"/>	Sales Application	Audit On Failure		✓	✓	✓

- Scroll down in the **Edit Realm** screen to the **Realm Authorizations** section. Click on the **Create** button to proceed.

Realm Authorizations

Create

Select	Grantee	Authorization Options	Authorization Rule Set Name
<input checked="" type="radio"/>	APPS_DBA_HARVEY	Participant	

Edit **Remove**

6. We will add a Senior DBA (**DBA_DEBRA [USER]**) and Junior DBA (**DBA_NICOLE [USER]**) to the list of Realm authorizations. Both will be Participants and no Authorization Rule Sets should be applied. After adding each, click on the **OK** button and repeat as necessary.

Create Realm Authorization

Define a database account or database role as either a realm owner or realm participant. Realm owners and realm participants can use their system privileges against realm secured objects. Only realm owners can grant or revoke realm secured database roles.

Grantee
DBA_DEBRA [USER]

Authorization Type
 Participant
 Owner

Authorization Rule Set
<Non Selected>

Cancel **OK**

7. This screen shot is a portion of the **Edit Realm** screen to show the result of the previous step. These are all the users in the **Realm Authorizations** section. You should have added two users **DBA_DEBRA** and **DBA_NICOLE**.

Realm Authorizations

Select Grantee	Authorization Options	Authorization Rule Set Name
<input checked="" type="radio"/> APPS_DBA_HARVEY	Participant	
<input type="radio"/> DBA_DEBRA	Participant	
<input type="radio"/> DBA_NICOLE	Participant	

Create Edit Remove

8. Click the **OK** button on the **Edit Realm** screen to submit the changes and return to the main **Database Vault Administration** screen. Click the highlighted **Rule Sets** link.

Administration Database Vault Reports General Security Reports Monitor

The links below allow you to protect applications and data using Oracle Database Vault features that include: Realms, Command Rules, Rule Sets, Factors, and Secure Application Roles.

Database Vault Feature Administration

- [Realms](#)
- [Command Rules](#)
- [Factors](#)
- Rule Sets**
- [Secure Application Roles](#)
- [Label Security Integration](#)

Administration Database Vault Reports General Security Reports Monitor

9. You will be brought to the **Rule Sets** screen where you can review the existing Rule Sets provided out of the box with Database Vault. We will proceed to create a new rule set by selecting the **Create** button.

Select Name ▾	Evaluation Options	Error Handling	Audit Options	Rules Defined?	Status
<input checked="" type="radio"/> Allow Fine Grained Control of System Parameters	All True	Show Error Message	Audit On Failure	✓	✓
<input type="radio"/> Allow Oracle Data Pump Operation	Any True	Show Error Message	Audit On Failure	✓	✓
<input type="radio"/> Allow Scheduler Job	Any True	Show Error Message	Audit On Failure	✓	✓
<input type="radio"/> Allow Sessions	All True	Show Error Message	Audit On Failure	✗	✓
<input type="radio"/> Allow System Parameters	All True	Show Error Message	Audit On Failure	✓	✓
<input type="radio"/> Can Grant VPD Administration	All True	Show Error Message	Audit On Failure	✓	✓
<input type="radio"/> Can Maintain Accounts/Profiles	Any True	Show Error Message	Audit On Failure	✓	✓
<input type="radio"/> Can Maintain Own Account	Any True	Show Error Message	Audit On Failure	✓	✓
<input type="radio"/> Disabled	All True	Show Error Message	Audit Disabled	✓	✓
<input type="radio"/> Enabled	All True	Show Error Message	Audit Disabled	✓	✓

10. In the **Create Rule Set** screen, provide the **Name (Restrict_HR_DBA_Users)**, a description and select the **Evaluation Options** to be **Any True**. Click on the **OK** button to save and proceed. By selecting the Any True option, if any of the Rules conditions are true (or met), the Rule Set, overall, will evaluate to true.

Create Rule Set

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:

Description: Protects Senior and Junior DBA Activities

Status: Enabled

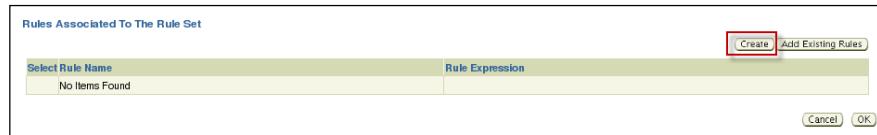
Evaluation Options: All True Any True

Buttons: Cancel, OK

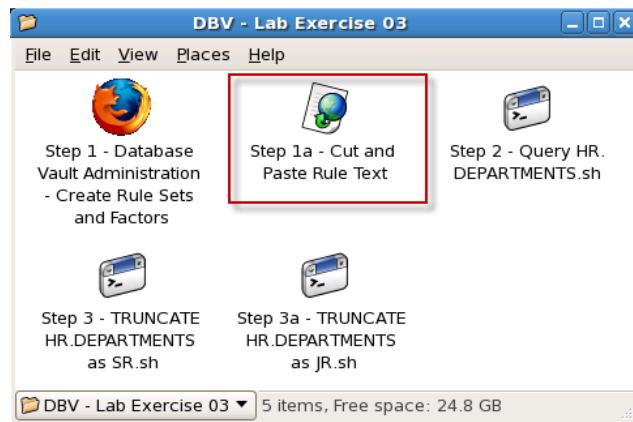
11. When you are brought back to the **Rule Set** screen, select the **Restrict_HR_DBA_Users** Rule Set and click on the **Edit** button to continue.

Select Name ▾	Evaluation Options	Error Handling	Audit Options	Rules Defined?	Status
<input type="radio"/> Allow Fine Grained Control of System Parameters	All True	Show Error Message	Audit On Failure	✓	✓
<input type="radio"/> Allow Oracle Data Pump Operation	Any True	Show Error Message	Audit On Failure	✓	✓
<input type="radio"/> Allow Scheduler Job	Any True	Show Error Message	Audit On Failure	✓	✓
<input type="radio"/> Allow Sessions	All True	Show Error Message	Audit On Failure	✗	✓
<input type="radio"/> Allow System Parameters	All True	Show Error Message	Audit On Failure	✓	✓
<input type="radio"/> Can Grant VPD Administration	All True	Show Error Message	Audit On Failure	✓	✓
<input type="radio"/> Can Maintain Accounts/Profiles	Any True	Show Error Message	Audit On Failure	✓	✓
<input type="radio"/> Can Maintain Own Account	Any True	Show Error Message	Audit On Failure	✓	✓
<input type="radio"/> Disabled	All True	Show Error Message	Audit Disabled	✓	✓
<input type="radio"/> Enabled	All True	Show Error Message	Audit Disabled	✓	✓
<input checked="" type="radio"/> Restrict_HR_DBA_Users	Any True	Show Error Message	Audit On Failure	✗	✓

12. Scroll down to the bottom of the page reviewing the various options and go to the **Rules Associated To The Rule Set** section. We will add two Rules. Click on the **Create** button.



13. Minimize the browser window and locate the **DBV – Lab Exercise 03** folder again. Click on the icon, **Step 1a – Cut and Paste Rule Text**. We have provided the **Rule Expressions** for your convenience.



14. We will be creating (adding) two rules. This particular Rule named **Restrict_HR_DBA_JR_Only_Rule** checks to ensure:
- The time and date must be within 8-5, Monday through Friday.
 - The user must be logged into the database via localhost and not remotely.

- Rule Name: **Restrict_HR_DBA_JR_Only_Rule**
 - o `dvf.f$session_user='DBA_NICOLE' and dvf.f$client_ip is null and to_char(sysdate,'HH24') between '08' and '16' and to_char(sysdate,'d') between '2' and '6'.`

If these conditions are not met, the **DBA_NICOLE** user will not be allowed to perform the operation in which this Rule is associated with. Once you copy the Rule Name and the Rule Expression, click on the **OK** to continue.

Create Rule

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

General

* Name: Restrict_HR_DBA_JR_Only_Rule

* Rule Expression: `dvf.f$session_user='DBA_NICOLE' and dvf.f$client_ip is null and to_char(sysdate,'HH24') between '08' and '16' and to_char(sysdate,'d') between '2' and '6'`

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 DV\$SYS account.

Cancel **OK**

15. Repeat the steps above to add another Rule to the Rule Set and click on the **Create** button.

Rules Associated To The Rule Set

Create **Add Existing Rules**

Select Rule Name ▾ **Rule Expression**

Restrict_HR_DBA_JR_Only_Rule `dvf.f$session_user='DBA_NICOLE' and dvf.f$client_ip is null and to_char(sysdate,'HH24') between '08' and '16' and to_char(sysdate,'d') between '2' and '6'`

Edit **Remove**

16. Add the second Rule. This particular Rule named **Only_HR_DBA_SR_Rule** checks to:

- Ensure that the user trying to perform an operation is the **HR_DBA_SR** user.
- Rule Name: **Only_HR_DBA_SR_Rule**
 - `dvf.f$session_user='DBA_DEBRA'`

If the user is not the **HR_DBA_SR** user, the operation in which this Rule is associated with will not be permitted. Once you copy the Rule Name and the Rule Expression, click on the **OK** to continue.

Create Rule

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

General

* Name: Only_HR_DBA_SR_Rule

* Rule Expression: `dvf.f$session_user='DBA_DEBRA'`

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 DV\$SYS account.

Cancel **OK**

17. After the two Rules have been added and reviewed to the Rule Set, click on the **OK** button to continue.

Rules Associated To The Rule Set

Create **Add Existing Rules**

Select Rule Name ▾ **Rule Expression**

Only_HR_DBA_SR_Rule `dvf.f$session_user='DBA_DEBRA'`

Restrict_HR_DBA_JR_Only_Rule `dvf.f$session_user='DBA_NICOLE' and dvf.f$client_ip is null and to_char(sysdate,'HH24') between '08' and '16' and to_char(sysdate,'d') between '2' and '6'`

Edit **Remove**

18. Returning back to the **Rule Sets** screen, you can now see that Rules are now defined for this Rule Set. The green check mark denotes this.

Select Name	Evaluation Options	Error Handling	Audit Options	Rules Defined?	Status
Allow Fine Grained Control of System Parameters	All True	Show Error Message	Audit On Failure	✓	✓
Allow Oracle Data Pump Operation	Any True	Show Error Message	Audit On Failure	✓	✓
Allow Scheduler Job	Any True	Show Error Message	Audit On Failure	✓	✓
Allow Sessions	All True	Show Error Message	Audit On Failure	✗	✓
Allow System Parameters	All True	Show Error Message	Audit On Failure	✓	✓
Can Grant VPD Administration	All True	Show Error Message	Audit On Failure	✓	✓
Can Maintain Accounts/Profiles	Any True	Show Error Message	Audit On Failure	✓	✓
Can Maintain Own Account	Any True	Show Error Message	Audit On Failure	✓	✓
Disabled	All True	Show Error Message	Audit Disabled	✓	✓
Enabled	All True	Show Error Message	Audit Disabled	✓	✓
Restrict_HR_DBA_Users	Any True	Show Error Message	Audit On Failure	✓	✓

19. We must now associate the created Rule Set with a particular operation. We are going to add a **Command Rule**. Click on the **Command Rule** link highlighted from the main Database Vault Administration screen.

Administration	Database Vault Reports	General Security Reports	Monitor
The links below allow you to protect applications and data using Oracle Database Vault features that include: Realms, Command Rules, Rule Sets, Factors, and Secure Application Roles.			
Database Vault Feature Administration			
Realms			
Command Rules			
Factors			
Rule Sets			
Secure Application Roles			
Label Security Integration			
Administration	Database Vault Reports	General Security Reports	Monitor

20. In the **Command Rules** screen, review the existing Command Rules. Click on the **Create** button to create a new Command Rule.

Select Command	Object Owner	Object Name	Rule Set Name	Status
ALTER PROFILE	%	%	Can Maintain Accounts/Profiles	✓
ALTER SYSTEM	%	%	Allow Fine Grained Control of System Parameters	✓
ALTER USER	%	%	Can Maintain Own Account	✓
CREATE PROFILE	%	%	Can Maintain Accounts/Profiles	✓
CREATE USER	%	%	Can Maintain Accounts/Profiles	✓
DROP PROFILE	%	%	Can Maintain Accounts/Profiles	✓
DROP USER	%	%	Can Maintain Accounts/Profiles	✓

21. We want to limit the **TRUNCATE TABLE** command and associate our newly created Rule Set to this Operation to the HR objects. Select **TRUNCATE TABLE** as the **Command**, **HR** as the **Object Owner** and **Restrict_HR_DBA_Users** as the **Rule Set**. Click on the **OK** button to continue.

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: TRUNCATE TABLE

Status: Enabled

Applicability

Object Owner: HR

Object Name: %

Rule Set

Restrict_HR_DBA_Users

22. You will be brought back to the Command Rules page to review your newly created Command Rule. We will proceed to test the Command Rule that we just created.

Command rules control the ability to process Data Definition Language (DDL) commands and special database operations. Command rules determine whether or not to allow the command to succeed based on the evaluation of a Database Vault rule set.

Select Command	Object Owner	Object Name	Rule Set Name	Status
ALTER PROFILE	%	%	Can Maintain Accounts/Profiles	✓
ALTER SYSTEM	%	%	Allow Fine Grained Control of System Parameters	✓
ALTER USER	%	%	Can Maintain Own Account	✓
CREATE PROFILE	%	%	Can Maintain Accounts/Profiles	✓
CREATE USER	%	%	Can Maintain Accounts/Profiles	✓
DROP PROFILE	%	%	Can Maintain Accounts/Profiles	✓
DROP USER	%	%	Can Maintain Accounts/Profiles	✓
TRUNCATE TABLE	HR	%	Restrict_HR_DBA_Users	✓

23. Minimize the browser window and locate the **DBV – Lab Exercise 03** folder again. Click on the icon, **Step 2 – Query HR.DEPARTMENTS.sh** to review the queries attempted to be executed by both users.

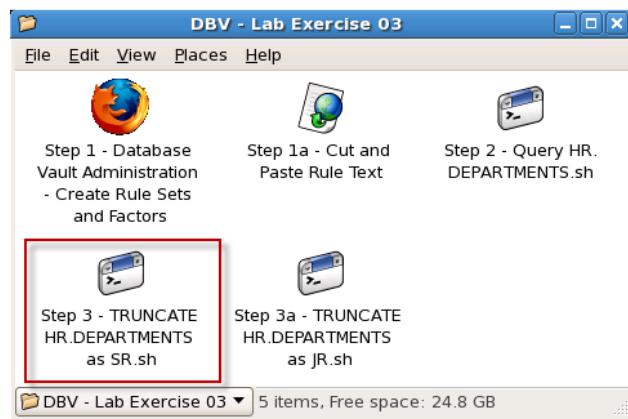


24. Click on the icon, **query_hr_departments_demo.out**. to review the results. You will notice that both users can query the data successfully as intended. These commands have not been restricted.

query_hr_departments_demo.out

```
SQL>
SQL>
Table
```

25. Click on the icon, **Step 3 – TRUNCATE HR.DEPARTMENTS as SR.sh** to review and execute the SQL statements.



26. Click on the icon, **truncate_hr_departments_demo.out**. As you review the output, the user **DBA_DEBRA** can successfully truncate the table because the check to **dvf.f\$session** returns the intended user. For the user **DBA_NICOLE** depending on the time of day (!date), this operation will either fail or succeed.

SQL>
SQL>
SQL> Conne
SQL>

truncate_hr_
departments_demo.
out

SQL>
SQL> @/home/oracle/dbv_scripts/truncate_hr_departments.sql
SQL> -- Operations as USERS: DBA_DEBRA (SR),DBA_NICOLE (JR) ####
SQL> -- DBA_DEBRA can TRUNCATE the hr.departments_demo tables ####
SQL> -- DBA_NICOLE can TRUNCATE the hr.departments_demo tables ONLY ####
SQL> -- when the Rule Set Conditions are met ####
SQL>
SQL>
SQL> connect DBA_DEBRA/Manager_1
connected.
SQL> truncate table hr.departments_demo;

table truncated.

SQL>
SQL> connect DBA_NICOLE/Manager_1
connected.
SQL> select dvf.f\$client_ip from dual;

\$CLIENT_IP

SQL>
SQL> select to_char(sysdate, 'Dy DD-Mon-YYYY HH24:MI:SS') as "Current Time"
 0m dual;

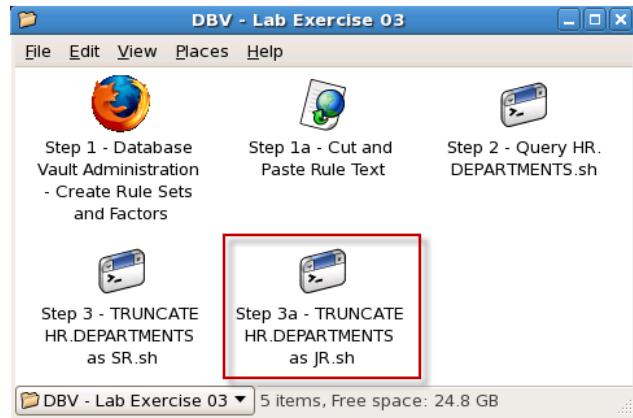
Current Time

Fri 23-Sep-2011 18:21:22

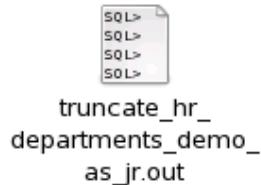
SQL>
SQL> truncate table hr.departments_demo;
truncate table hr.departments_demo
*
RROR at line 1:
RA-47400: Command Rule violation for TRUNCATE TABLE on HR.DEPARTMENTS_DEMO

SQL>
SQL>
SQL>
SQL> exit;

27. Click on the icon, **Step 3a – TRUNCATE HR.DEPARTMENTS as JR.sh** to review and execute the SQL statements.



28. Click on the icon, **truncate_hr_departments_demo.out**. As you review the output, the user **DBA_NICOLE** (JUNIOR DBA) attempts to truncate the table failed due to the result of the **dvf.f\$client_ip**. Since the value was not null (expected in the Rule that we defined), the criteria was not met and the failed as expected.



C. Summary

You accomplished the following in this lab exercise:

1. Understood some behavior of Factors and Rule Sets and their application to Command Rules.
2. Created a Rule set that will restrict the ability of the HR_DBA_JR_DEMO user to TRUNCATE a table only when the following conditions were met:
 - i. The time and date must be within 8-5, Monday through Friday
 - ii. Must be logged into the database via localhost and not remotely accessed

LAB EXERCISE 04 – INCREASE VISIBILITY OF DATABASE ACCESS CONTROLS USING ORACLE DATABASE VAULT MONITORING & REPORTING

INTRODUCTION

One of the biggest side benefits resulting from regulatory compliance has been security awareness. Historically, the focus of the information technology (IT) department has been on high availability and performance. The focus on regulatory compliance has required everyone to take a step back and look at their IT infrastructure, databases, and applications from a security angle. Common questions include:

- Who has access to this information?
- Where is the sensitive information stored?

Regulations such as the Sarbanes-Oxley Act, Health Insurance Portability and Accountability Act (HIPAA), International Convergence of Capital Measurement and Capital Standards: a Revised Framework (Basel II), Japan Privacy Law, Payment Card Industry Data Security Standard (PCI DSS), and the European Union Directive on Privacy and Electronic Communications have common themes that include internal controls, separation of duty, and access control.

Categories of Oracle Database Vault Reports

Oracle Database Vault provides a selection of reports that display security-related information from the database. These reports also show custom Oracle Database Vault audit event information. The reports are in two categories:

- **Database Vault Reports.** These reports allow you to check configuration issues with realms, command rules, factors, factor identities, rule sets, and secure application roles. These reports also reveal realm violations, auditing results, and so on.
- **General Security Reports.** These reports allow you to check the status of object privileges, database account system privileges, sensitive objects, privilege management, powerful database accounts and roles, initialization parameters, profiles, account passwords, security audits, and other security vulnerability reports.

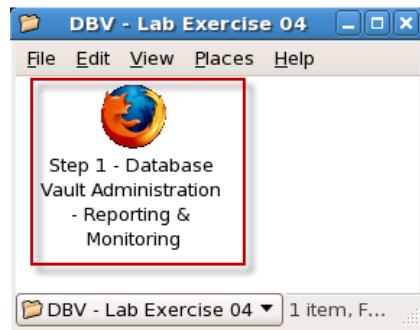
A. Overview

In this lab exercise, you will accomplish the following:

1. *Review available monitoring capabilities and reports that enable security administrators and database administrators to review database access controls.*

B. INCREASE VISIBILITY OF DATABASE ACCESS CONTROLS USING ORACLE DATABASE VAULT MONITORING & REPORTING

1. We will now use the techniques we used in the previous lab to create two additional Realms. Click on the icon **Step 1 – Enable Database Consolidation with DBV and Realms** in the **DBV – Lab Exercise 04** Folder.



2. Provide the following required fields to login to the Database Vault Administration tool and click on the **Login** button to continue.
 - a. User Name: **SEC_ANALYST_ALLEN**
 - b. Password: **Manager_1**
 - c. Host: **cloud.oracle.com**
 - d. Port: **1522**
 - e. SID: **db06**

A screenshot of the Oracle Database Vault login page. The top navigation bar says 'ORACLE Database Vault'. Below it is a 'Login' button. The main area is titled 'Login to Database:' and contains a form with the following fields:

- * User Name: SEC_ANALYST_ALLEN
- * Password: (Redacted)
- * Host: dbsecurity.oracle.com
- * Port: 1521
- * SID / Service:
 - SID: db06
 - Service: (Redacted)

A 'Login' button is located at the bottom right of the form.

It is very important to notice that we are logging in with the user SEC_ANALYST_ALLEN. This user's role is to run Oracle Database Vault reports and monitor Oracle Database Vault. This segregation of duties provides flexibility for reporting and monitoring when needed.

- In the Database Vault Administration screen, you will see three tabs that provide reporting and monitoring. Let's first drill into the **General Security Reports** tab by clicking on the highlighted link. //GEF – Drop one of these screenshots

The screenshot shows the Oracle Database Vault Administration interface. At the top, it says "Database Instance: db06". Below that is a navigation bar with four tabs: "Administration", "Database Vault Reports", "General Security Reports" (which is highlighted with a red border), and "Monitor". A message below the tabs states: "The links below allow you to protect applications and data using Oracle Database Vault features that include: Realms, Command Rules, Rule Sets, and more." Underneath this, there is a section titled "Database Vault Feature Administration". At the bottom of the page, there is a copyright notice: "Copyright (c) 2000, 2010, Oracle. All rights reserved." and links for "About Oracle Database Vault Administrator", "Database", "Help", and "Logout".

- General Security Reports allow you to check the status of object privileges, database account system privileges, sensitive objects, privilege management, powerful database accounts and roles, initialization parameters, profiles, account passwords, security audits, and other security vulnerability reports.

Click on a few reports to explore. Many of these reports are very common to numerous regulatory requirements where this information is required for demonstration during the audit process.

The screenshot shows the "General Security Reports" screen. At the top, there is a navigation bar with tabs: "Administration", "Database Vault Reports", "General Security Reports" (which is highlighted), and "Monitor". Below the tabs, a message reads: "Use this screen to run reports about potential security issues with the existing privilege model, database parameters, application configurations, database accounts, and database roles." There is a "Run Report" button. Below that is a "Expand All | Collapse All" link. A tree view shows the following report categories:

- Reports**
- Object Privilege Reports**
- Database Account System Privileges Reports**
- Sensitive Objects Reports**
- Privilege Management - Summary Reports**
- Powerful Database Accounts and Roles Reports**
- Initialization Parameters and Profiles Reports**
- Database Account Password Reports**
- Security Audit Reports**
- Other Security Vulnerability Reports**

At the bottom, there is another "Run Report" button.

5. Click on tab **Database Vault Reports** to view the specific reports provided. Database Vault Reports allow you to check configuration issues with realms, command rules, factors, factor identities, rule sets, and secure application roles. These reports also reveal realm violations, auditing results, and so on.

The screenshot shows the Oracle Database Vault Reports interface. The top navigation bar includes tabs for Administration, Database Vault Reports (which is highlighted with a red box), General Security Reports, and Monitor. Below the tabs, a message states: "Use this screen to run reports about potential Database Vault configuration issues and Database Vault audit events." A "Run Report" button is present. Under the heading "Select Focus Report Title", there is a tree view with the "Reports" node expanded. Under "Reports", two items are listed: "Database Vault Configuration Issues Reports" and "Database Vault Auditing Reports".

6. Click on the **Monitor** tab. There are three main categories of reports that we will drill into including Security Policy Changes Details, Security Violation Attempts and Database Configuration and Structural Changes. Through these exercises, we have added a number of entries into these categories. As you review each one, think back to the exercises and the activities we performed.

The screenshot shows the Oracle Monitor interface. The top navigation bar includes tabs for Administration, Database Vault Reports, General Security Reports, and Monitor (which is highlighted with a red box). A status message indicates the page was refreshed on Sep 23, 2011 at 7:18:19 PM. A dropdown menu "Show Records For" is set to "Past 24 Hours". Below the tabs, a section titled "Security Policy Changes By Category" shows a chart from Sep 22, 2011 to Sep 23, 2011. The chart tracks the number of changes over time, with bars colored according to the legend. The legend includes: Database Vault Policy (blue), Label Security Policy (light blue), Audit Policy (yellow), Privilege Grants (green), Privilege Revokes (grey), Database Account (orange), and Database Role (red). Below the chart, three links are displayed: "► Security Policy Changes Detail", "► Security Violation Attempts", and "► Database Configuration and Structural Changes".

7. Expand the **Security Policy Changes Details** and review the entries. As this name implies, this report will provide details on the changes to Database Vault policies. This will be useful during any stage of deployment any changes that may compromise intended security policies.

▼ Security Policy Changes Detail						
Sep 22, 2011 7:25:15 PM - Sep 23, 2011 7:25:15 PM						
Timestamp	User Name	User Host	Action Name	Return Code	Object Name	Grantee
Sep 23, 2011 6:59:29 PM	DVACCTMGR	dbsecurity.oracle.com	CREATE USER	0	SEC_ANALYST_ALLEN	
Sep 23, 2011 6:21:07 PM	SEC_ADMIN_OWEN	dbsecurity.oracle.com	INSERT	0 DVSYS	COMMAND_RULE\$	
Sep 23, 2011 6:16:53 PM	DBA_DEBRA	dbsecurity.oracle.com	DELETE	0 LBACSYS	LBAC\$POLT	
Sep 23, 2011 6:16:46 PM	SEC_ADMIN_OWEN	dbsecurity.oracle.com	UPDATE	0 DVSYS	RULE_SET\$	
Sep 23, 2011 6:14:54 PM	SEC_ADMIN_OWEN	dbsecurity.oracle.com	INSERT	0 DVSYS	RULE_SET_RULE\$	
Sep 23, 2011 6:14:54 PM	SEC_ADMIN_OWEN	dbsecurity.oracle.com	INSERT	0 DVSYS	RULE\$	
Sep 23, 2011 6:09:55 PM	SEC_ADMIN_OWEN	dbsecurity.oracle.com	INSERT	0 DVSYS	RULE_SET_RULE\$	
Sep 23, 2011 6:09:55 PM	SEC_ADMIN_OWEN	dbsecurity.oracle.com	INSERT	0 DVSYS	RULE\$	
Sep 23, 2011 6:04:51 PM	SEC_ADMIN_OWEN	dbsecurity.oracle.com	INSERT	0 DVSYS	RULE_SET\$	
Sep 23, 2011 6:04:02 PM	SEC_ADMIN_OWEN	dbsecurity.oracle.com	UPDATE	0 DVSYS	REALM\$	

8. Expand the **Security Violation Attempts** and review the entries. As the name implies, this report provides details on any violation attempt to the Database Vault Policies that are enabled.

▼ Security Violation Attempts						
Sep 22, 2011 7:26:47 PM - Sep 23, 2011 7:26:47 PM						
Timestamp	User Name	User Host	Action Name	Return Action Object	Code Name	Rule Set Name
Sep 23, 2011 7:00:22 PM	DVACCTMGR	dbsecurity.oracle.com	Realm Violation Audit	47410 Oracle Database Vault		GRANT DV_SECANALYST TO SEC_ANALYST_ALLEN
Sep 23, 2011 6:26:32 PM	DBA_NICOLE	dbsecurity.oracle.com	Command Authorization Audit	47400 TRUNCATE TABLE	Restrict_HR_DBA_Users	TRUNCATE TABLE HR.DEPARTMENTS_DEMO
Sep 23, 2011 6:21:22 PM	DBA_NICOLE	dbsecurity.oracle.com	Command Authorization Audit	47400 TRUNCATE TABLE	Restrict_HR_DBA_Users	TRUNCATE TABLE HR.DEPARTMENTS_DEMO
Sep 23, 2011 5:31:53 PM	APPS_DBA_OLIVER	dbsecurity.oracle.com	Realm Violation Audit	1031 Sales Application		SELECT COUNT(*) FROM SH.SALES
Sep 23, 2011 5:31:53 PM	SYS	dbsecurity.oracle.com	Realm Violation Audit	47401 Order Entry Application		DROP TABLE OE.JUNK
Sep 23, 2011 5:31:53 PM	SYS	dbsecurity.oracle.com	Realm Violation Audit	47401 Order Entry Application		CREATE TABLE OE.JUNK(ID NUMBER)
Sep 23, 2011 5:31:53 PM	SYS	dbsecurity.oracle.com	Realm Violation Audit	47401 Order Entry Application		DROP TABLE OE.JUNK
Sep 23, 2011 5:31:53 PM	SYS	dbsecurity.oracle.com	Realm Violation Audit	47401 Sales Application		CREATE TABLE SH.JUNK(ID NUMBER)
Sep 23, 2011 5:31:53 PM	SYS	dbsecurity.oracle.com	Realm Violation Audit	47401 Order Entry Application		DROP TABLE OE.JUNK
Sep 23, 2011 5:31:53 PM	SYS	dbsecurity.oracle.com	Realm Violation Audit	47401 HR Application		CREATE TABLE HR.JUNK(ID NUMBER)

9. Expand the **Database Configuration and Structural Changes** and review the entries. As the name implies, this report provides tracking of these activities with the important details necessary to understand what has occurred.

▼ Database Configuration and Structural Changes

Sep 22, 2011 7:31:19 PM - Sep 23, 2011 7:31:19 PM

Previous 1-10 of 37 Next 10 ▶

Timestamp	User Name	User Host	Action Name	Return Code	Owner	Object Name	Comment Text
Sep 23, 2011 6:16:54 PM	DBA_DEBRA	dbsecurity.oracle.com	CREATE TABLE	0 HR		DEPARTMENTS_DEMO	
Sep 23, 2011 6:16:53 PM	DBA_DEBRA	dbsecurity.oracle.com	DROP TABLE	0 HR		DEPARTMENTS_DEMO	
Sep 23, 2011 5:31:53 PM	APPS_DBA_OLIVER	dbsecurity.oracle.com	DROP TABLE	0 OE		JUNK	
Sep 23, 2011 5:31:53 PM	APPS_DBA_OLIVER	dbsecurity.oracle.com	DROP TABLE	0 OE		JUNK	
Sep 23, 2011 5:31:53 PM	APPS_DBA_OLIVER	dbsecurity.oracle.com	CREATE TABLE	0 OE		JUNK	
Sep 23, 2011 5:31:53 PM	APPS_DBA_SAM	dbsecurity.oracle.com	CREATE TABLE	955 SH		JUNK	
Sep 23, 2011 5:31:53 PM	APPS_DBA_OLIVER	dbsecurity.oracle.com	CREATE TABLE	0 OE		JUNK	
Sep 23, 2011 4:49:54 PM	APPS_DBA_SAM	dbsecurity.oracle.com	CREATE TABLE	0 OE		JUNK	
Sep 23, 2011 4:49:54 PM	APPS_DBA_OLIVER	dbsecurity.oracle.com	CREATE TABLE	0 OE		JUNK	
Sep 23, 2011 4:49:54 PM	APPS_DBA_OLIVER	dbsecurity.oracle.com	DROP TABLE	0 OE		JUNK	

Previous 1-10 of 37 Next 10 ▶

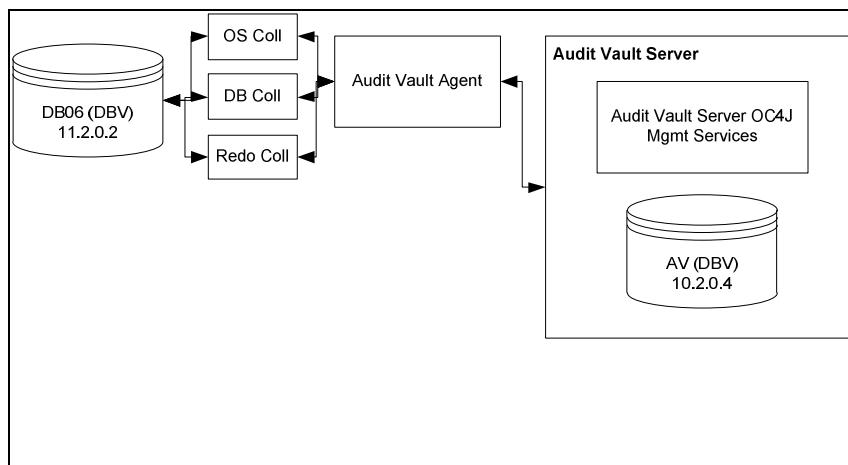
C. Summary

You accomplished the following in this lab exercise:

1. *Reviewed available monitoring capabilities and reports that enable security administrators and database administrators to review database access controls.*

LAB CONFIGURATION – SETUP OF THE AUDIT VAULT ENVIRONMENT

OVERVIEW



The image that you are running has the following installed for Audit Vault:

- Oracle Audit Vault Server 10.3
- Oracle Audit Vault Agent 10.3
- Oracle DB 11.2.0.2 – AV Repository
- Oracle DB 11.2.0.2 – DB06

The image has an Oracle Database (**DB06**) preconfigured to be monitored by Audit Vault, as shown in the diagram above. We are using all three Oracle DB Collectors for DB06: OS Aud Collector, DB Aud Collector and the Redo Collector. The DB06 Database also has native auditing configured to monitor system activity and record user activity using the ‘DB_Extended’ method.

For these lab exercises, the following infrastructure components are running and available.

- **Database to be monitored: Database DB06**
- **Audit Vault Server, Agent and Collectors**
 - Audit Vault DB Listener on Port 1522
 - Audit Vault DB
 - Audit Vault Server Processes – Web Application etc
 - Audit Vault Agent
 - Audit Vault Collectors for DB06 – OS, DB and Redo

Here is a summary of the users and their functions that will be used throughout this lab exercise.

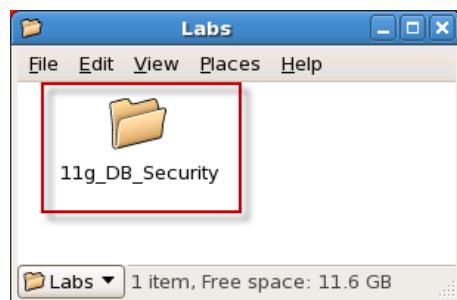
- DBA_DEBRA – Database Administrator Account
- AV_AUD_AUDREY – Audit Vault Administrator

Let's get started.

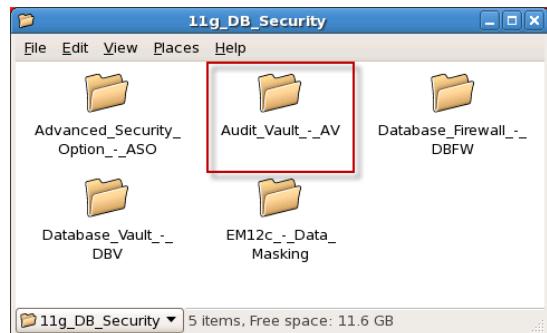
1. On the desktop, navigate to the **Labs** folder, double-click and open the contents.



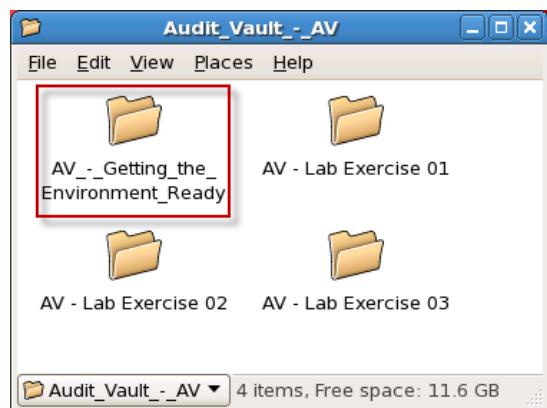
2. Select the folder, **11g_DB_Security**.



3. Select the folder, **Audit_Vault_-_AV**.



4. Within the **Audit_Vault_-_AV**, you can access all of the Lab folders.



Throughout these exercises, when you double click on the script, the OS will prompt you to specify what you want to do with the shell script.



You can select either '**Run in Terminal**' to execute the script, or '**Display**' if you want to see more detail on what is actually being executed. You obviously must run each script to execute the labs, but feel free to display them if you wish to see what is being executed.

You will notice that output files will be saved in the folder after the script executes. You may review this output, as well.

Unless otherwise indicated, the windows will close when the script has completed. Please wait for each script to complete before executing the next script.

You are ready move forward with the Audit Vault labs. Enjoy!!

LAB EXERCISE 00 – AUDIT VAULT OVERVIEW

INTRODUCTION

Oracle Audit Vault, part of Oracle's comprehensive portfolio of database security solutions, reduces the cost and complexity of compliance and the risk of insider threats by automating the collection and consolidation of audit data. It provides a secure and highly scalable audit warehouse, enabling simplified reporting, analysis, and threat detection on audit data. In addition, database audit settings are centrally managed and monitored from within Audit Vault, reducing IT security cost. With Oracle Audit Vault, organizations are in a much better position to enforce privacy policies, guard against insider threats, and address regulatory requirements such as Sarbanes-Oxley and PCI.

- Simplify compliance reporting—Easily analyze audit data and take action in a timely fashion with out-of-the-box reports or custom reporting via the industry's only open warehouse schema for audit information
- Detect threats quickly—Quickly and automatically detect unauthorized activities that violate security and governance policies; thwart perpetrators from covering their tracks
- Lower IT costs with audit policies—Centrally manage audit settings across all databases from a single console
- Transparently collect and consolidate audit data—Collect audit data in a timely fashion across disparate systems
- Provide a secure and scalable repository—Leverage Oracle's industry-leading security and data warehousing technology to provide a secure and scalable audit warehouse

Lab Scenarios and Objectives

For our fictitious company, CashBankTrust is currently evaluating database activity monitoring requirements and challenges within their database environment.

CashBankTrust stores sensitive application data in its Oracle Database environment. It permits administration and development access to a group of privileged users. This access is tiered depending on multiple factors. All access to the data in production systems has to be monitored. CashBankTrust is currently using native database auditing functionality and manually collecting, storing and reporting on this data. Using Oracle Audit Vault they can automate the database activity audit management process. CashBankTrust is replacing a custom legacy system with Oracle Audit Vault.

The Audit Vault labs that you will complete will demonstrate solutions specifically to the identified challenges below.

Product	Identified Challenges
Audit Vault	Monitoring of privileged (sensitive) user activity on the database is insufficient or non-existent - immutability of the data and visibility of the reports is inadequate.
	Monitoring of privileged end-user activity on identified sensitive data being managed in the database is insufficient or non-existent – immutability of the data and visibility of the reports is inadequate.
	Monitoring of privileged user entitlements, privilege grants or revocations on the database is insufficient or non-existent -immutability of the data and visibility of the reports is inadequate
	Inability to track and report on changes to data values in the database for identified sensitive information (e.g. Financial, Salary Data) to ensure confidentiality and integrity.

LAB EXERCISE 01 – EFFECTIVELY MANAGING DATABASE AUDIT POLICY

INTRODUCTION

CashBankTrust is leveraging Oracle Native DB Auditing. They have decided to further investigate what configuration changes can be made to make the audit records more complete. They also want to ensure that the audit configuration is being selective and effective. Ensuring that CashBankTrust is only auditing activity that is considered to be risky is critical to the success of the audit project.

A. Overview

Auditing is the monitoring and recording of selected user database actions. It can be based on individual actions, such as the type of SQL statement run, or on combinations of factors that can include name, application, time, and so on. Security policies can trigger auditing when specified elements in Oracle Database are accessed or altered, including content.

Auditing is generally used to:

- Enable future accountability for current actions taken in a particular schema, table, or row, or affecting specific content
- Investigate suspicious activity. For example, if an unauthorized user is deleting data from tables, then the security administrator could audit all connections to the database and all successful and unsuccessful deletions of rows from all tables in the database.
- Monitor and gather data about specific database activities. For example, the database administrator can gather statistics about which tables are being updated, how many logical I/Os are performed, or how many concurrent users connect at peak times.

You can use Audit Vault to view and configure audit-related initialization parameters and administer audited objects for statement auditing and schema object auditing. For example, Audit Vault shows the properties for current audited statements, privileges, and objects. You can view the properties of each object, and you can search audited objects by their properties. You can also turn on and turn off auditing on objects, statements, and privileges.

Types and Records of Auditing

Oracle Database allows audit options to be focused or broad. You can audit:

- Successful statement executions, unsuccessful statement executions, or both
- Statement executions once in each user session or once every time the statement is run
- Activities of all users or of a specific user

Oracle Database auditing enables the use of several different mechanisms, with the features listed in the table below:

Type of Auditing	Meaning/Description
Statement auditing	Audits SQL statements by type of statement, not by the specific schema objects on which they operate. Typically broad, statement auditing audits the use of several types of related actions for each option. For example, AUDIT TABLE tracks several DDL statements regardless of the table on which they are issued. You can also set statement auditing to audit selected users or every user in the database.
Privilege auditing	Audits the use of powerful system privileges enabling corresponding actions, such as AUDIT CREATE TABLE. Privilege auditing is more focused than statement auditing because it audits only the use of the target privilege. You can set privilege auditing to audit a selected user or every user in the database.
Schema object auditing	Audits specific statements on a particular schema object, such as AUDIT SELECT ON employees. Schema object auditing is very focused, auditing only a specific statement on a specific schema object. Schema object auditing always applies to all users of the database.
Fine-grained auditing	Audits data access and actions based on content. Using DBMS_FGA, the security administrator creates an audit policy on the target table. If any rows returned from a DML statement block match the audit condition, then an audit event entry is inserted into the audit trail.

During this lab you will:

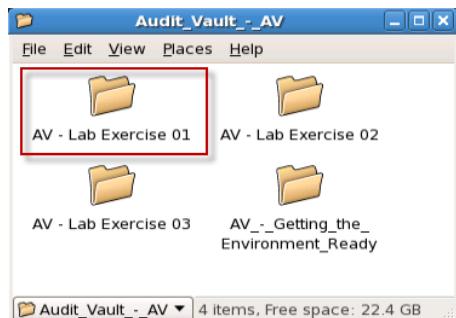
1. Demonstrate native Oracle database auditing and the audit settings configured for this environment.
 - i) Create a table, configure auditing on that table then confirm that the audit records are being generated successfully
2. Review the Oracle Database Secure Configuration 'Best Practice' audit policies
3. Configure the DB06 database with Secure Configuration 'Best Practice' Audit Policy
4. Execute an automated workload generation SQL Script to test the audit policies and confirm that the audit data is being collected by Audit Vault.

B. Setup & Preparation

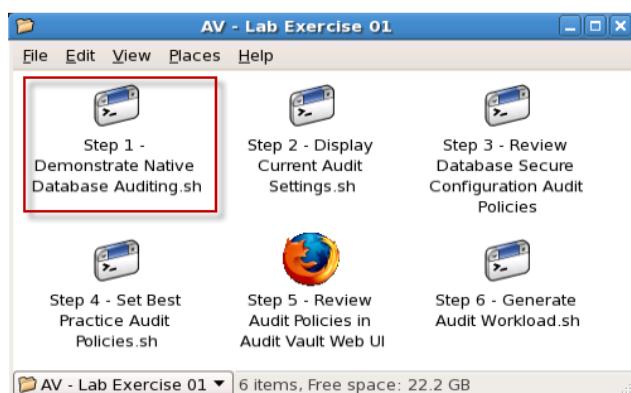
- None.

C. Overview of Native Database Auditing

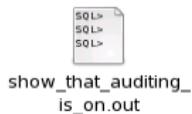
1. Open the 'AV – Lab Exercise 01' Folder



2. We will now step through a simple example of native Oracle database auditing. To start the audit test, click on the icon '**Step 1 – Demonstrate Native Database Auditing**'.



- Click on the icon, **show_that_auditing_is_on.out** and view the output of the script executed.



We first count from **aud\$** table to show the number of audit records in the database audit table before we start our test. Your numbers will be different, but will set a baseline.

```
SQL> select count(*) from aud$;
          COUNT(*)
-----
        27289
SQL> commit;
Commit complete.
```

We will now connect to the database using DBA_DEBRA, CREATE TABLE AS SELECT from the HR.EMPLOYEES table into the HR.EMPLOYEES_TWO table.

```
SQL>
SQL> conn dba_debra/Manager_1 (increment by 2)
Connected.
SQL> create table hr.employees_two as (select * from hr.employees);
(increment by 2)

Table created.

SQL>
SQL> conn system/oracle1 (increment by 2)
Connected.
SQL> select count(*) from aud$;

          COUNT(*)
-----
        27295
```

We check the **aud\$** table again. Notice that the amount of records has increased—incrementing 4 for the connection/session for DBA_DEBRA and SYSTEM users, 1 for the create table and another 1 for the select.

The DB06 has already been configured to monitor the ‘CREATE TABLE’ commands using the ‘AUDIT TABLE’ audit setting/policy.

We will now SELECT the COUNT() on the number of records that are in the newly created HR.EMPLOYEE_TWO table.

```
SQL>
SQL> conn dba_debra/Manager_1 (increment by 2)
Connected.
SQL> select count(*) from hr.employees_two;

          COUNT(*)
```

```

-----
107

SQL>
SQL> conn system/oracle1 (increment by 2)
Connected.
SQL> select count(*) from aud$;

COUNT(*)
-----
27299

```

We check the **aud\$** table again. Notice that the amount of records has only increased by 4—accounting for the connection/session for DBA_DEBRA and SYSTEM users. The SELECT is currently not being monitored.

We will now set the audit policy on the table to monitor SELECTS's.

```

SQL>
SQL> conn dba_debra/Manager_1 (increment by 2)
Connected.
SQL> audit select on hr.employees_two by access; (increment by 1)
Audit succeeded.

SQL>
SQL> conn system/oracle1 (increment by 2)
Connected.
SQL> select count(*) from aud$;

COUNT(*)
-----
27304

```

When we repeat the query to 'SELECT' from the the hr.employees_two table.

```

SQL>
SQL> conn dba_debra/Manager_1 (increment by 2)
Connected.
SQL> select count(*) from hr.employees_two; (increment by 1)

COUNT(*)
-----
107

SQL>
SQL> conn system/oracle1 (increment by 2)
Connected.
SQL> select count(*) from aud$;

COUNT(*)
-----
27309

```

We can now turn off auditing on the HR.EMPLOYEES_TWO table and drop the table.

```

SQL>
SQL> conn dba_debra/Manager_1
Connected.
SQL> noaudit select on hr.employees_two;

Noaudit succeeded.

SQL>
SQL> conn dba_debra/Manager_1
Connected.
SQL> drop table hr.employees_two;

```

```
Table dropped.
```

```
SQL>
SQL> exit;
```

4. Click on the Icon ‘Step 2 – Display Current Audit Settings’. This script will login to DB06 and show audit trail and sys operations parameter, and audit destination parameter.



2. Click on the icon, **display_audit_settings.out**. In the output we can verify that the initialization parameter **AUDIT_TRAIL** is set to the value **DB_EXTENDED** and that **AUDIT_SYS_OPERATIONS** is set to **TRUE**.



You will see that the script checks the parameters for **audit_file_dest**, **audit_sys_operations**, **audit_syslog_level** and **audit_trail**.

```
SQL>
SQL> show parameter audit;

NAME                           TYPE        VALUE
-----                         -----
audit_file_dest                string      /u01/oracle/admin/db06/adump
audit_sys_operations            boolean    TRUE
audit_syslog_level              string
audit_trail                     string      DB_EXTENDED
SQL> exit;
```

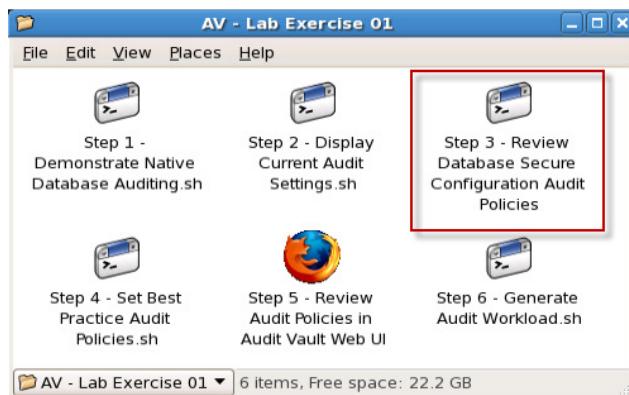
The ‘audit_trail’ parameter can be set to one of several values – this controls where the DB writes audit records. There are three basic options: OS, XML and DB. When ‘audit_trail’ is set to either ‘OS’ or ‘XML’ audit records are written to the location specified in the ‘audit_file_dest’ parameter. This is set to /u01/oracle/admin/db06/adump for our DB06 database.

When the ‘audit_trail’ parameter is set to either ‘DB’ or ‘XML’ you can add the ‘EXTENDED’ parameter to capture SQL Text and SQL Bind variables. This greatly enhances the information that is captured.

When the ‘audit_trail’ parameter is set to ‘DB’ audit records are written to the aud\$ table.

Finally, the parameter ‘audit_sys_operations’ will monitor all activity performed by users logged in as either sysdba or sysoper.

5. Double click on the icon, ‘Step 3 – Review the Oracle DB Audit Best Practice’



You will see the file **\$ORACLE_HOME/demo/seccconf.sql** open in an editor.

This SQL file was derived from the Oracle 11g Database Security Best practice that is implemented by default when you create a new DB with 11g using DBCA. The SQL file contains Oracle Database Audit Policies that are considered to be the best practice for monitoring system activity. For your implementation you can use this Audit Policy as a starting place then tailor the policies to meet your specific business requirements. The policy does not contain any audit policies for monitoring specific objects.

```
Rem
Rem $Header: seccconf.sql 08-mar-2007.10:57:56 vipshah Exp $
Rem
Rem seccconf.sql
Rem
Rem Copyright (c) 2007, Oracle. All rights reserved.
Rem
Rem NAME
Rem   seccconf.sql - SECure CONfiguration script
Rem
Rem DESCRIPTION
Rem   Secure configuration settings for the database
Rem   contains audit settings recommendedas bare minimum
Rem   (enabled, with admin actions audited)
Rem
Rem
Rem NOTES
Rem   Derived from RDBMS 11g Secure Config Settings (seccconf.sql)
Rem
Rem MODIFIED (MM/DD/YY)
Rem   vipshah 03/08/07 - Created
Rem
Audit alter any table by access;
Audit create any table by access;
Audit drop any table by access;
```

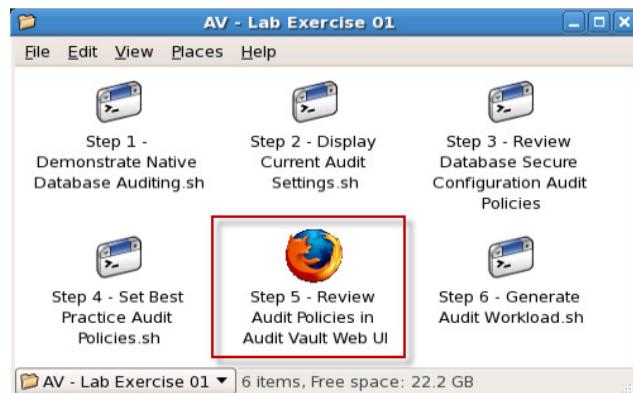
6. Double click on the icon, 'Step 4 – Set Best Practice Audit Policies.sh' icon. As the name implies, we will set these audit policies. We are also setting sample audit policies to be used later in this lab.



7. Click on the icon, **set_audit_policies.out** to review the output.

`SQL>
SQL>
Conne
SQL>`
`set_audit_policies.
out`

8. We will now login to Audit Vault to review the audit policies for DB06. Click on the icon '**Step 5 – Review Audit Policies in Audit Vault Web UI**'.



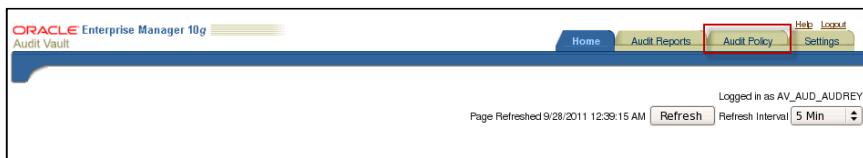
1. Login as the AV Auditor using the username/password of: **AV_AUD_AUDREY/Manager_1** connecting as **AV_AUDITOR**. Click on the **Login** button to continue.

The screenshot shows the 'Login to Audit Vault' page. At the top, it says 'ORACLE Enterprise Manager 10g Audit Vault'. Below that is a 'Login' button. The main area has three input fields: 'User Name' containing 'AV_AUD_AUDREY', 'Password' containing '*****', and 'Connect As' set to 'AV_AUDITOR'. At the bottom right is a yellow 'Login' button.

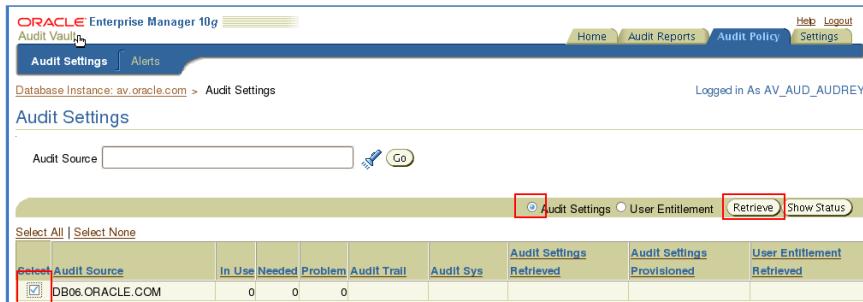
2. Click the '**I agree**' button.



3. Then navigate directly to the **Audit Policy** Tab.



4. Click the checkbox to select the **DB06.ORACLE.COM** Audit Source, ensure the radio button is set to **Audit Settings** and click on the **Retrieve** button. This action will retrieve the Audit Policy from the selected database (DB06).



- You will see a message letting you know that the policy will be retrieved.

We will now also retrieve the **User Entitlement** data from DB06. Change the radio button to **User Entitlement** then click on **Retrieve** button – ensuring that **DB06.ORACLE.COM** is still selected, as shown below.

Select Audit Source	In Use	Needed	Problem	Audit Trail	Audit Sys	Audit Settings Retrieved	Audit Settings Provisioned	User Entitlement Retrieved
<input checked="" type="checkbox"/> DB06.ORACLE.COM	0	0	0					

You will see another message letting you know the user entitlement data is being collected:

- Refresh the page by clicking on the **Audit Settings** tab, after a few moments you will see some audit settings appear in the summary.

You will notice that the database already has a defined policy, which was defined for the DB06 database during an earlier lab.

Select Audit Source	In Use	Needed	Problem	Audit Trail	Audit Sys	Audit Settings Retrieved	Audit Settings Provisioned	User Entitlement Retrieved
<input checked="" type="checkbox"/> DB06.ORACLE.COM	2180	0	2180	DB_EXTENDED TRUE		Sep 28, 2011 11:17:05 PM GMT+00:00		Sep 28, 2011 11:20:17 PM GMT+00:00

7. Click on the **DB06.ORACLE.COM** hyperlink

The screenshot shows the Oracle Enterprise Manager 10g Audit Vault interface. The title bar reads "ORACLE Enterprise Manager 10g Audit Vault". The top navigation bar includes "Home", "Audit Reports", "Audit Policy", "Help", and "Logout". The user is logged in as "AV_AUD_AUDREY". The main content area is titled "Audit Settings" and shows a table of audit sources. One row for "DB06.ORACLE.COM" is selected, indicated by a red box around the checkbox in the "Audit Source" column. The table columns include "Audit Source", "In Use", "Needed", "Problem", "Audit Trail", "Audit Sys", "Audit Settings Retrieved", "Audit Settings Provisioned", and "User Entitlement Retrieved". The "Audit Source" row has values: 2180, 0, 2180, DB_EXTENDED, TRUE, Sep 28, 2011 11:17:05 PM GMT+00:00, and Sep 28, 2011 11:20:17 PM GMT+00:00.

8. Review the detail for the audit policy.

The screenshot shows the "Save Audit Settings" section of the Audit Settings page. It includes a note about saving work and a "Save All Audit Settings" button. Below is the "Apply Audit Settings" section, which contains a table of audit settings types and their counts. The table has columns: "Audit Settings Type", "In Use", "Needed", and "Problem". The rows show: Statement (77, 0, 77), Object (1816, 0, 1816), Privilege (286, 0, 286), FGA (1, 0, 1), and Capture Rule (0, 0, 0).

Audit Settings Type	In Use	Needed	Problem
Statement	77	0	77
Object	1816	0	1816
Privilege	286	0	286
FGA	1	0	1
Capture Rule	0	0	0

9. Scroll down to the lower half of the summary screen.

The screenshot shows the "Copy Audit Settings from Another Source" section. It includes fields for "Verify", "Export as SQL", "Audit Source User Name", "Provision", "Audit Source Password", and "From". Below is the "Overview" tab, which is currently selected.

You will notice that this screen has several sub tabs to help you see the audit policies in place on DB06. *If you do not see any audit policies in any of the tabs, please repeat the previous step to retrieve DB06's information.*

If you want a description of each category of audit policy, please continue to **Section D – Additional Information** in this guide.

17. As the last step of this lab, return back to the **AV – Lab Exercise 01** folder and double click on the icon, '**Step 6 – Generate Audit Workload**' icon. The script generates a workload that triggers audit records to be generated. You may see some errors when this script is run, however these can be ignored. The errors in the script are used to demonstrate how Oracle DB auditing can capture both successful and unsuccessful transactions. In the next lab, you will be reviewing the audit records this script generates in the Audit Vault Reporting console.



18. Click on the icon, **generate_audit_workload_and_inject_to_av.out** to review the output of this script.



D. Additional Steps

- Review the following information to understand the different categories of audit policy.

Understanding Statement Auditing

The screenshot shows the Oracle Enterprise Manager 10g Audit Vault interface. The URL is [av.oracle.com](#). The page title is "Audit Settings". The navigation bar includes Home, Audit Reports, Audit Policy, and Settings. The user is logged in as AV_AUD_AUDREY. The main content area is titled "Statement" and shows a table of audit configurations:

Statement	User	Proxy User	Execution Condition	Audit Granularity	In Use	Needed
ALTER SEQUENCE			Both	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ALTER TABLE			Both	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
COMMENT TABLE			Both	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
DEBUG PROCEDURE			Both	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
DELETE TABLE			WHENEVER NOT SUCCESSFUL	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
EXECUTE PROCEDURE			WHENEVER NOT SUCCESSFUL	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SYSTEM AUDIT			Both	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ROLE			Both	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Statement auditing audits SQL statements by type of statement, not by the specific schema objects on which the statement operates. Statement auditing can be broad or focused, for example, by auditing the activities of all database users or of only a select list of users. Typically broad, statement auditing, audits the use of several types of related actions for each option. These statements are in the following categories:

- (a) Data definition statements (DDL). For example, AUDIT TABLE audits all CREATE TABLE and DROP TABLE statements. AUDIT TABLE tracks several DDL statements regardless of the table on which they are issued. You can also set statement auditing to audit selected users or every user in the database.
- (b) Data manipulation statements (DML). For example, AUDIT SELECT TABLE audits all SELECT ... FROM TABLE or SELECT ... FROM VIEW statements, regardless of the table or view.

Understanding Object Auditing

The screenshot shows the Oracle Enterprise Manager 10g Audit Vault interface. The top navigation bar includes links for Home, Audit Reports, Audit Policy, and Settings. The main content area is titled 'DB06.ORACLE.COM' and shows the 'Object' tab selected under 'Audit Settings'. A table lists various database statements and their audit configurations. The columns include Statement, Schema, Object, Execution Condition, Audit Granularity, and In Use Needed. Most statements are audited by access ('BY ACCESS') and are marked as needed ('In Use' checked). The statements listed are INSERT, RENAME, DELETE, AUDIT, UPDATE, and SELECT.

	Statement	Schema	Object	Execution Condition	Audit Granularity	In Use	Needed
1	INSERT	DVSYS	DV\$RULE_SET_RULE	Both	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	RENAME	DVSYS	DV\$RULE_SET_RULE	Both	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	DELETE	DVSYS	DV\$RULE_SET_RULE	Both	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	AUDIT	DVSYS	DV\$SYS_GRANTEE	Both	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	UPDATE	DVSYS	DV\$SYS_GRANTEE	Both	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	SELECT	DVSYS	DV\$SYS_GRANTEE	WHENEVER	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Schema **object** auditing is the auditing of specific statements on a particular schema object, such as AUDIT SELECT ON HR.EMPLOYEES. Schema object auditing is very focused, auditing only a specific statement on a specific schema object for all users of the database.

- (a) For example, object auditing can audit all SELECT and DML statements permitted by object privileges, such as SELECT or DELETE statements on a given table. The GRANT and REVOKE statements that control those privileges are also audited.
- (b) Object auditing lets you audit the use of powerful database commands that enable users to view or delete very sensitive and private data. You can audit statements that reference tables, views, sequences, standalone stored procedures or functions, and packages.
- (c) Oracle Database and Oracle Audit Vault always set schema object audit options for all users of the database. You cannot set these options for a specific list of users.

Understanding Privilege Auditing

The screenshot shows the Oracle Enterprise Manager 10g Audit Vault interface, similar to the previous one but for 'Privilege' auditing. The main content area is titled 'DB06.ORACLE.COM' and shows the 'Privilege' tab selected under 'Audit Settings'. A table lists various database privileges and their audit configurations. The columns include Privilege, User, Proxy User, Execution Condition, Audit Granularity, and In Use Needed. Most privileges are audited by access ('BY ACCESS') and are marked as needed ('In Use' checked). The privileges listed are CREATE PUBLIC SYNONYM, CREATE ROLE, CREATE ROLLBACK SEGMENT, CREATE SEQUENCE, CREATE ANY SEQUENCE, CREATE MATERIALIZED VIEW, and CREATE SYNONYM.

	Privilege	User	Proxy User	Execution Condition	Audit Granularity	In Use	Needed
1	CREATE PUBLIC SYNONYM			Both	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	CREATE ROLE			Both	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	CREATE ROLLBACK SEGMENT			Both	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	CREATE SEQUENCE			Both	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	CREATE ANY SEQUENCE			Both	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	CREATE MATERIALIZED VIEW			Both	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	CREATE SYNONYM			Both	BY ACCESS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Privilege auditing is the auditing of SQL statements that use a system privilege. You can audit the use of any system privilege. Like statement auditing, privilege auditing can audit the activities of all database users or of only a specified list of users.

- (a) For example, if you enable AUDIT SELECT ANY TABLE, Oracle Database audits all SELECT tablename statements issued by users who have the SELECT ANY TABLE privilege. This type of auditing is very important for the Sarbanes-Oxley (SOX) Act compliance requirements. Sarbanes-Oxley and other compliance regulations require the privileged user be audited for inappropriate data changes or fraudulent changes to records.
- (b) Privilege auditing audits the use of powerful system privileges enabling corresponding actions, such as AUDIT CREATE TABLE. If you set both similar statement and privilege audit options, then only a single audit record is generated. For example, if the statement clause TABLE and the system privilege CREATE TABLE are both audited, then only a single audit record is generated each time a table is created. The statement auditing clause, TABLE, audits CREATE TABLE, ALTER TABLE, and DROP TABLE statements. However, the privilege auditing option, CREATE TABLE, audits only CREATE TABLE statements, because only the CREATE TABLE statement requires the CREATE TABLE privilege.
- (c) Privilege auditing does not occur if the action is already permitted by the existing owner and schema object privileges. Privilege auditing is triggered only if these privileges are insufficient, that is, only if what makes the action possible is a system privilege.
- (d) Privilege auditing is more focused than statement auditing for the following reasons:
 - a. It audits only a specific type of SQL statement, not a related list of statements.
 - b. It audits only the use of the target privilege.

Understanding Fine-Grained Auditing

The screenshot shows the Oracle Enterprise Manager 10g Audit Vault interface. The top navigation bar includes links for Home, Audit Reports, Audit Policy, Help, and Logout. The main title is "Audit Vault" and the sub-section is "Audit Settings". The URL is "Database Instance: av.oracle.com > Audit Settings > DB06.ORACLE.COM". The user is logged in as "AV_AUD_AUDREY". Below this, there are tabs for Overview, Statement, Object, Privilege, FGA (which is selected), and Capture Rule. A toolbar at the top right has buttons for "Mark All as Needed", "Mark All as Not Needed", and "Create". A table below lists audit policies. One row is highlighted with a red border: "Policy Name: CHK_HR_EMP", "Schema: HR", "Object: EMPLOYEES", "Statement: S", "Columns: SALARY", and "In Use: Needed". There are also up and down arrow buttons for sorting.

Fine-grained auditing (FGA) enables you to create a policy that defines specific conditions that must take place for the audit to occur. For example, fine-grained auditing lets you audit the following types of activities:

- An IP address from outside the corporate network being used
- A table being accessed between 9 p.m. and 6 a.m. or on Saturday and Sunday.
- A table column being selected or updated
- A value in a table column being modified

A fine-grained audit policy provides granular auditing of select, insert, update, and delete operations. Furthermore, because you are auditing only very specific conditions, you reduce the amount of audit information generated and can restrict auditing to only the conditions that you want to audit. This creates a more meaningful audit trail that supports compliance requirements. For example, a central tax authority can use fine-grained auditing to track access to tax returns to guard against employee snooping, with enough detail to determine what data was accessed. It is not enough to know that a specific user used the SELECT privilege on a particular table. Fine-grained auditing provides a deeper audit, such as when the user queried the table or the computer IP address user who performed the action.

If you drill into our FGA rule, you will see that we've added auditing for any select statements issued against the HR.EMPLOYEES table, when the SALARY column is greater than 10000.

The screenshot shows the Oracle Enterprise Manager 10g Audit Vault interface, specifically the "Fine Grained Audit Detail" section. The top navigation bar and user information are identical to the previous screenshot. The main title is "Fine Grained Audit Detail". The URL is "Database Instance: av.oracle.com > Audit Settings > DB06.ORACLE.COM". The user is logged in as "AV_AUD_AUDREY". The form displays the configuration for the FGA rule: "FGA Policy Name: CHK_HR_EMP", "Audit Trail: Database with SQL Text", "Schema: HR", "Object: EMPLOYEES", "Statements: SELECT", "Columns: SALARY", and "Condition: salary>10000". There are radio buttons for "All" and "Any" conditions, with "All" selected. At the bottom right of the form is a "Cancel" button.

Understanding Capture Rules

You can create a **capture rule** to track changes in the database redo log files. The capture rule specifies DML and DDL changes that should be checked when Oracle Database scans the database redo log. You can apply the capture rule to an individual table, a schema, or globally to the entire database. Unlike statement, object, privilege, and fine-grained audit policies, you do not retrieve and activitate capture rule settings from a source database, because you cannot create them there. You only can create the capture rule in the Audit Vault Console.

We have not defined any capture rules as part of these labs.

E. Summary

In this lab, you completed the following:

1. *Demonstrated native Oracle database auditing and the audit settings configured for this environment.*
 - i) *Created a table, configure auditing on that table then confirm that the audit records are being generated successfully*
2. *Reviewed the Oracle Database Secure Configuration ‘Best Practice’ audit policies*
3. *Configured the DB06 database with Secure Configuration ‘Best Practice’ Audit Policy*
4. *Executed an automated workload generation SQL Script to test the audit policies and confirm that the audit data is being collected by Audit Vault.*

LAB EXERCISE 02 – REDUCE TIME TO COMPLIANCE USING ORACLE AUDIT VAULT REPORTING

INTRODUCTION

The custom solution CashBankTrust previously used for monitoring database activity did not provide any reporting interface. Each time the CashBankTrust internal IT audit team requested report data the Oracle DBA's had to write SQL and generate a new report. This obstacle meant that CashBankTrust had limited visibility into the database activity, it also created a dependency between the consumer of the data and the provider – the Oracle DBA's and the IT Audit Team. Using Oracle Audit Vault the IT Audit team can have real time access to the audit data and not have to use the DBA's time.

A. Overview

Oracle Audit Vault provides powerful built-in reports to monitor a wide range of activity including privileged user activity and changes to database structures. The reports provide visibility into activities and provide detailed information the who, what, when and where of database access. The latest release of Oracle Audit Vault provides an exciting new reports interface built on the widely popular Oracle Application Express technology. The new reports provide an easy-to-use interface with the ability to create colourful charts and graphs as well as the ability to customize the report format. Report columns can be re-ordered as well as removed. Rules can be put in place to automatically highlight specific rows so that report users can quickly spot suspicious or unauthorized activity. Reports will include audit information from Oracle, Sybase, IBM DB2 and Microsoft SQL Server databases, providing a holistic picture of activity across the enterprise. Oracle Audit Vault provides numerous standard audit assessment reports categorized into areas such as compliance and alerts. Compliance reports themselves are further categorized into those relating to cardholder data, financial data and healthcare data. Out-of-the-box reports include information on database account management, roles and privileges, object management, and login failures. Oracle Business Intelligence, Oracle BI Publisher and other 3rd party reporting tools can be used to build additional reports to meet specific compliance and security requirements. Detailed information on the repository tables can be found in the Oracle Audit Vault administrator's guide.

In this lab exercise, you will:

1. Generate a variety of reports available in Audit Vault including:
 - i. Data Access Report
 - ii. Database Vault Audit Report
 - iii. System Management
 - iv. Entitlement Snapshot
 - v. Data Access Report with Redo (Before and After Values)
 - vi. Compliance Reports

B. Setup & Preparation

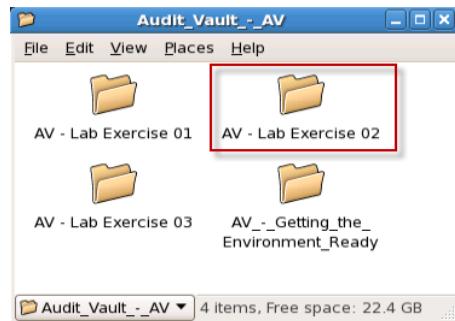
- None

C. REDUCE TIME TO COMPLIANCE USING ORACLE AUDIT VAULT REPORTING

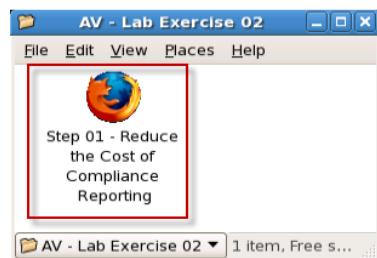
Once you determine what you need to audit, you will have a better understanding of the requirements to which reports are needed to demonstrate compliance to the various requirements. In this exercise, we will demonstrate how to generate useful reports to these Database Audit Requirements.

What Do You Need To Audit?						
Database Audit Requirements	SOX	PCI DSS	HIPAA	Basel II	FISMA	GLBA
Accounts, Roles & Permissions Do you have visibility of GRANT and REVOKE activities?	●	●	●	●	●	●
Failed Logins Do you have visibility of failed logins and other exception activities?	●	●	●	●	●	●
Privileged User Activity Do you have visibility of users activities?	●	●	●	●	●	●
Access to Sensitive Data Can you have visibility into what information is being queried (SELECTs)?		●	●	●	●	●
Schema Changes Are you aware of CREATE, DROP and ALTER Commands that are occurring on Identified Tables / Columns?	●	●	●	●	●	●
Data Changes Do you have visibility into Insert, Update, Merge, Delete commands?	●			●		

1. Open the '**AV – Lab Exercise 02**' Folder



2. Double-click on the '**Step 1 – Reduce the Cost of Compliance Reporting**' icon. This will open up the browser.



3. Login to the Audit Vault console as user **av_aud_audrey/Manager_1**. Be sure to select the value **AV_AUDITOR** in the **Connect As** menu option. Click on the **Login** button to continue.

ORACLE Enterprise Manager 10g
Audit Vault

Login

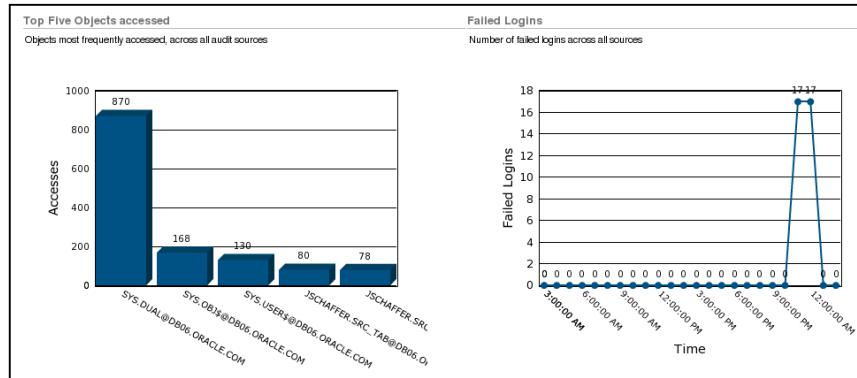
Login to Audit Vault

* User Name

* Password

Connect As

4. Scroll down the Audit Vault Home page and verify that in the lower portion of the **Overview** screen there are some entries for frequently accessed objects. Also, you will likely see an indication of failed logins at the far right of the **Failed Logins** graph. Sample output is shown below.



DATA ACCESS REPORTS

Database Audit Requirements	SOX	PCI DSS	HIPAA	Basel II	FISMA	GLBA
Access to Sensitive Data Can you have visibility into what information is being queried (SELECTs)?		●	●	●	●	●

5. Click on the **Audit Reports** tab and select the **Data Access** report within the **Access Reports** section. View the report by clicking on the **Data Access** link and verify that you see audit records that were collected on today's date.



The screenshot shows the 'Access Reports' section of the Oracle Enterprise Manager 10g Audit Vault. It includes tabs for Default Reports, Compliance Reports, Custom Reports, Generated Reports, Report Schedules, and Entitlement Snapshots. Under the 'Access Reports' tab, there are three main sections: 'Activity Overview' (with a computer monitor icon), 'Data Access' (which is highlighted with a red box), 'Database Vault', 'Distributed Database', 'Procedure Executions', and 'User Sessions'. To the right, there are sections for 'Entitlement Reports' (User Accounts, User Accounts by Source, User Privileges, User Privileges by Source) and 'Alert Reports' (All Alerts, Critical Alerts, Warning Alerts).

6. Review the **Data Access Report**. The Data Access Report displays audited data manipulation language (DML) activities (i.e. SELECT, INSERT, UPDATE or DROP SQL) statements.

Source	Target	Event	Event Status	User	Host	Event Time
DB06.ORACLE.COM	EMPLOYEES	UPDATE	SUCCESS	DBA_DEBRA	dsecurity.oracle.com	9/28/2011 12:59:46 AM
DB06.ORACLE.COM	EMPLOYEES	UPDATE	SUCCESS	DBA_DEBRA	dsecurity.oracle.com	9/28/2011 12:59:46 AM
DB06.ORACLE.COM	EMPLOYEES	INSERT	SUCCESS	DBA_DEBRA	dsecurity.oracle.com	9/28/2011 12:59:46 AM
DB06.ORACLE.COM	EMPLOYEES	DELETE	SUCCESS	DBA_DEBRA	dsecurity.oracle.com	9/28/2011 12:59:46 AM
DB06.ORACLE.COM	EMPLOYEES	DELETE	SUCCESS	DBA_DEBRA	dsecurity.oracle.com	9/28/2011 12:59:46 AM
DB06.ORACLE.COM	EMPLOYEES	INSERT	SUCCESS	DBA_DEBRA	dsecurity.oracle.com	9/28/2011 12:59:46 AM
DB06.ORACLE.COM	EMPLOYEES	UPDATE	SUCCESS	DBA_DEBRA	dsecurity.oracle.com	9/28/2011 12:59:46 AM
DB06.ORACLE.COM	EMPLOYEES	DELETE	SUCCESS	DBA_DEBRA	dsecurity.oracle.com	9/28/2011 12:59:46 AM
DB06.ORACLE.COM	EMPLOYEES	SELECT	UNKNOWN:FGA	DBA_DEBRA	dsecurity.oracle.com	9/28/2011 12:59:45 AM
DB06.ORACLE.COM	MY_EMP	SELECT	SUCCESS	JSCHAFFER	dsecurity.oracle.com	9/28/2011 12:59:45 AM
DB06.ORACLE.COM	EMPLOYEES	SELECT	SUCCESS	JSCHAFFER	dsecurity.oracle.com	9/28/2011 12:59:45 AM
DB06.ORACLE.COM	LBAC\$POLS	SELECT	SUCCESS	JSCHAFFER	dsecurity.oracle.com	9/28/2011 12:59:45 AM
DB06.ORACLE.COM	DBA_TAB_COLUMNS	SELECT	SUCCESS	JSCHAFFER	dsecurity.oracle.com	9/28/2011 12:59:45 AM
DB06.ORACLE.COM	DBA_TAB_COLS	SELECT	SUCCESS	JSCHAFFER	dsecurity.oracle.com	9/28/2011 12:59:45 AM
DB06.ORACLE.COM	_BASE_USER	SELECT	SUCCESS	JSCHAFFER	dsecurity.oracle.com	9/28/2011 12:59:45 AM

7. Return to the **Default Reports** page, click on the **Data Access** category. All DML activity is captured here. Here are some sample audit policies that will generate records here.

- AUDIT DELETE;
- AUDIT INSERT;
- AUDIT SELECT;
- AUDIT TRUNCATE TABLE;
- AUDIT UPDATE;

DATABASE VAULT REPORTS

Database Audit Requirements	SOX	PCI DSS	HIPAA	Basel II	FISMA	GLBA
Access to Sensitive Data Can you have visibility into what information is being queried (SELECTs)?	●	●	●	●	●	●

8. Select the **Database Vault** report within the **Access Reports** section. View the report by clicking on the **Database Vault** link and verify that you see audit records that were collected on today's date.

The screenshot shows the Oracle Enterprise Manager 10g Audit Vault interface. The top navigation bar includes links for Home, Audit Reports, Audit Policy, Help, Logout, and Settings. Below the navigation bar, there are three main sections: Access Reports, Entitlement Reports, and Alert Reports. The Access Reports section is active, displaying icons for Activity Overview, Data Access, Database Vault (which is highlighted with a red box), Distributed Database, Procedure Executions, and User Sessions. The Database Vault link under Access Reports is also highlighted with a red box.

9. This **Database Vault Report** displays audited Oracle Database Vault Activity. These audit records are collected from the Oracle Database Vault audit trail.

The screenshot shows the Oracle Enterprise Manager 10g Audit Vault interface with the Database Vault report selected. The top navigation bar includes links for Home, Audit Reports, Audit Policy, Help, Logout, and Settings. The Default Reports tab is selected. The Database Vault report page displays a search bar, a row limit of 15, a Go button, and a Create PDF button. A filter option "Event Time is in the last 24 hours" is checked. Below the search bar is a table listing audit events. The table columns are: Source, Event, Event Status, User, Associated Target, Owner, Target, and Event Time. The table contains 15 rows of audit log entries, all of which occurred on 9/28/2011 between 12:59:36 AM and 11:15:17 PM. The last row of the table has a page number indicator "1 - 15" with a right arrow.

Source	Event	Event Status	User	Associated Target	Owner	Target	Event Time
DB06.ORACLE.COM	REALM VIOLATION	47401	DBA_NICOLE	DVSYS	REALM\$		9/28/2011 12:59:36 AM
DB06.ORACLE.COM	COMMAND AUTHORIZATION	47400	SYS	DVSYS	CODE\$		9/28/2011 12:59:36 AM
DB06.ORACLE.COM	COMMAND AUTHORIZATION	47400	SYS	DVSYS	CODE\$		9/28/2011 12:59:36 AM
DB06.ORACLE.COM	COMMAND AUTHORIZATION	47400	SYS	DVSYS	CODE\$		9/28/2011 12:59:36 AM
DB06.ORACLE.COM	COMMAND AUTHORIZATION	47400	APPS_DBA_HARVEY	DVSYS	CODE\$		9/28/2011 12:59:35 AM
DB06.ORACLE.COM	COMMAND AUTHORIZATION	47400	APPS_DBA_HARVEY	DVSYS	CODE\$		9/28/2011 12:59:35 AM
DB06.ORACLE.COM	COMMAND AUTHORIZATION	47400	APPS_DBA_HARVEY	DVSYS	CODE\$		9/28/2011 12:59:35 AM
DB06.ORACLE.COM	COMMAND AUTHORIZATION	1031	APPS_DBA_HARVEY	DVSYS	CODE\$		9/28/2011 12:59:35 AM
DB06.ORACLE.COM	REALM VIOLATION	47401	DBA_NICOLE	DVSYS	REALM\$		9/27/2011 11:15:18 PM
DB06.ORACLE.COM	COMMAND AUTHORIZATION	47400	SYS	DVSYS	CODE\$		9/27/2011 11:15:17 PM
DB06.ORACLE.COM	COMMAND AUTHORIZATION	47400	SYS	DVSYS	CODE\$		9/27/2011 11:15:17 PM
DB06.ORACLE.COM	COMMAND AUTHORIZATION	47400	APPS_DBA_HARVEY	DVSYS	CODE\$		9/27/2011 11:15:17 PM
DB06.ORACLE.COM	COMMAND AUTHORIZATION	47400	APPS_DBA_HARVEY	DVSYS	CODE\$		9/27/2011 11:15:17 PM
DB06.ORACLE.COM	COMMAND AUTHORIZATION	47400	APPS_DBA_HARVEY	DVSYS	CODE\$		9/27/2011 11:15:17 PM

SYSTEM MANAGEMENT REPORTS

Database Audit Requirements	SOX	PCI DSS	HIPAA	Basel II	FISMA	GLBA
Privileged User Activity Do you have visibility of users activities?	●	●	●	●	●	●

10. Return to the **Default Reports** page.



The **System Management Report** displays audited system management activity. For example, it lists activities such as STARTUP and SHUTDOWN operations on a database, ROLLBACK operations, ENABLE and DISABLE operations on all triggers, and so forth. It also lists user-related operations, such as unlocking a user account.

Source	Event	User	OS User	Event Status	Event Time
DB06.ORACLE.COM	SUPER USER DDL	/	oracle	SUCCESS	9/28/2011 12:59:47 AM
DB06.ORACLE.COM	SUPER USER TRANSACTION CONTROL	/	oracle	SUCCESS	9/28/2011 12:59:46 AM
DB06.ORACLE.COM	SUPER USER DDL	/	oracle	SUCCESS	9/28/2011 12:59:46 AM
DB06.ORACLE.COM	SUPER USER DDL	/	oracle	SUCCESS	9/28/2011 12:59:46 AM
DB06.ORACLE.COM	SUPER USER TRANSACTION CONTROL	/	oracle	SUCCESS	9/28/2011 12:59:46 AM
DB06.ORACLE.COM	SUPER USER TRANSACTION CONTROL	/	oracle	SUCCESS	9/28/2011 12:59:45 AM
DB06.ORACLE.COM	SUPER USER TRANSACTION CONTROL	/	oracle	SUCCESS	9/28/2011 12:59:45 AM
DB06.ORACLE.COM	SUPER USER TRANSACTION CONTROL	/	oracle	SUCCESS	9/28/2011 12:59:44 AM
DB06.ORACLE.COM	SUPER USER TRANSACTION CONTROL	/	oracle	SUCCESS	9/28/2011 12:59:44 AM
DB06.ORACLE.COM	SUPER USER DDL	/	oracle	SUCCESS	9/28/2011 12:59:44 AM
DB06.ORACLE.COM	SUPER USER DDL	/	oracle	SUCCESS	9/28/2011 12:59:44 AM
DB06.ORACLE.COM	SUPER USER DDL	/	oracle	SUCCESS	9/28/2011 12:59:44 AM
DB06.ORACLE.COM	SUPER USER DDL	/	oracle	SUCCESS	9/28/2011 12:59:44 AM

ENTITLEMENT SNAPSHOTS & REPORTS

The Entitlement Snapshots page displays information about the entitlement snapshots that are taken each time user entitlement settings are retrieved.

11. Finally we will review the **User Entitlement** reports. Navigate to the **Audit Reports** tab. Once there click on the **User Accounts** report.

The **User Accounts Report** displays the latest snapshot of Database users with their account profile and values for sources registered with Oracle Audit Vault.

This screenshot shows the Oracle Enterprise Manager 10g Audit Vault interface. The top navigation bar includes links for Home, Audit Reports, Audit Policy, Help, Logout, and Settings. Below the navigation bar, the Audit Reports tab is selected. The main content area is divided into three sections: Access Reports, Entitlement Reports, and Alert Reports. Under Access Reports, there are links for Activity Overview, Data Access, Database Vault, Distributed Database, Procedure Executions, and User Sessions. Under Management Activity Reports, there are links for Account Management, Audit Commands, Object Management, Procedure Management, Role and Privilege Management, and System Management. Under Entitlement Reports, the 'User Accounts' link is highlighted with a red box. Other links in this section include User Privileges, User Profiles, Database Roles, and System Privileges. Under Alert Reports, there are links for All Alerts, Critical Alerts, and Warning Alerts.

12. Initially you will not see any reporting data. Click on '**Go**' (next to the '**Label** drop down). This will bring back the latest Entitlement snapshot that you just retrieved from the source database. You can take multiple snapshots of user entitlement data, this report lets you look at the data for any of these snapshots. You can also compare different snapshots to see how the data has changed over time.

This screenshot shows the 'User Accounts' report page. The top navigation bar is identical to the previous screenshot. The main content area is titled 'User Accounts'. It features a 'Label' dropdown menu set to 'LATEST', a 'compare' checkbox, and a 'Go' button, all of which are highlighted with a red box. Below these controls is a search bar with a magnifying glass icon, a 'Rows' dropdown set to 15, a 'Go' button, and a 'Create PDF' button. A message at the bottom states 'No User Accounts found.'

COMPLIANCE REPORTS

13. Click on the **Compliance Reports** tab. The reports shown here are intended to help you meet your compliance reporting requirements as quickly as possible, across PCI, Sarbanes-Oxley and HIPAA (healthcare-related) areas.

The screenshot shows the Oracle Enterprise Manager 10g Audit Vault interface. At the top, there is a navigation bar with tabs: Default Reports, Compliance Reports (which is highlighted with a red box), Custom Reports, Generated Reports, Report Schedules, and Entitlement Snapshots. Below the navigation bar, there are three main sections: Credit Card, Financial, and Health Care. Each section contains several audit-related links. In the Credit Card section, the 'Database Failed Logins' link is highlighted with a red box.

Credit Card	Financial	Health Care
Credit Card Related Data Access Audit Setting Changes Before/After Values Database Failed Logins Database Login/Logout Database Logoff Database Logon Database Startup/Shutdown Deleted Objects Program Changes Schema Changes System Events User Privilege Change Activity	Financial Related Data Access Financial Related Data Modifications Audit Setting Changes Before/After Values Database Failed Logins Database Login/Logout Database Logoff Database Logon Database Startup/Shutdown Deleted Objects Program Changes Schema Changes System Events User Privilege Change Activity	EPHI Related Data Access Audit Setting Changes Before/After Values Database Failed Logins Database Login/Logout Database Logoff Database Logon Database Startup/Shutdown Deleted Objects Schema Changes System Events User Privilege Change Activity

14. Click on the '**Database Failed Logins**' link under the **Credit Card** section. This report lists all of the DB login failures across the audited sources. You are able to alter the scope of the report to a specific set of objects for the databases or objects they are covered by your regulatory requirement. To do this click on the **Change Definition** button.

The screenshot shows the 'Database Failed Logins' report. At the top, there is a search bar, a 'Rows' dropdown set to 15, a 'Go' button, a 'Create PDF' button, and a 'Change Definition' button (which is highlighted with a red box). Below the search bar, there are filter checkboxes for 'Event Time is in the last 24 hours' and 'X'. A table follows, displaying a list of failed logins with columns: Source, Event, User, Authentication Method, OS User, Event Status, and Event Time. All entries show a source of 'DB06.ORACLE.COM', an event of 'LOGON', a user of 'SCOTT' or 'PJONES' or 'APPB', an authentication method of 'DATABASE', an OS user of 'oracle', an event status of '1017', and an event time of '9/28/2011 12:59:45 AM'.

Source	Event	User	Authentication Method	OS User	Event Status	Event Time
DB06.ORACLE.COM	LOGON	APPB	DATABASE	oracle	1017	9/28/2011 12:59:45 AM
DB06.ORACLE.COM	LOGON	SCOTT	DATABASE	oracle	1017	9/28/2011 12:59:45 AM
DB06.ORACLE.COM	LOGON	PJONES	DATABASE	oracle	1017	9/28/2011 12:59:45 AM
DB06.ORACLE.COM	LOGON	APPB	DATABASE	oracle	1017	9/28/2011 12:59:45 AM
DB06.ORACLE.COM	LOGON	APPB	DATABASE	oracle	1017	9/28/2011 12:59:45 AM
DB06.ORACLE.COM	LOGON	APPB	DATABASE	oracle	1017	9/28/2011 12:59:45 AM

You will see that you are able to change the definition of the report.

The screenshot shows the 'Database Failed Logins' report. At the top, there is a search bar and several buttons: 'Create PDF', 'Save Definition', and 'Cancel'. Below the search bar, there are two filter options: 'Event Time is in the last 24 hours' (checked) and 'User contains 'ListOfUsers''. The second filter is highlighted with a red box. The main table has columns: Source, Event, User, Authentication Method, OS User, Event Status, and Event Time. The data in the table is as follows:

Source	Event	User	Authentication Method	OS User	Event Status	Event Time
DB06.ORACLE.COM	LOGON	APPS	DATABASE	oracle	1017	9/28/2011 12:59:45 AM
DB06.ORACLE.COM	LOGON	SCOTT	DATABASE	oracle	1017	9/28/2011 12:59:45 AM
DB06.ORACLE.COM	LOGON	PJONES	DATABASE	oracle	1017	9/28/2011 12:59:45 AM
DB06.ORACLE.COM	LOGON	APPS	DATABASE	oracle	1017	9/28/2011 12:59:45 AM

15. You will then see that there is a filter called “**User Contains ListOfUsers**”. You will notice that the filter is not applied. Before applying a filter, replace the “**ListOfUsers**” entry with some specific objects. Click on the **User contains ‘ListOfUsers’** link, you will see the following.

The screenshot shows the 'Database Failed Logins' report with a 'Filter' dialog box open. The 'Filter' dialog has three fields: 'Column' (set to 'User'), 'Operator' (set to 'contains'), and 'Expression' (set to 'ListOfUsers'). This entire dialog is highlighted with a red box. Below the dialog, the report interface shows the same filters ('Event Time is in the last 24 hours' checked, 'User contains 'ListOfUsers'' unchecked) and the same table data as the previous screenshot.

16. Change the filter to show only activity that occurred on the DB06 source. Change the filter to have a ‘**Column**’ of ‘**Source**’ and the ‘**Expression**’ to ‘**DB06.ORACLE.COM**’, as shown below. Click on the **Apply** button.

Using this scoping functionality you will be able to tailor the Compliance Reports to meet your businesses requirements.

The screenshot shows the 'Database Failed Logins' report with a 'Filter' dialog box open. The 'Filter' dialog has three fields: 'Column' (set to 'Source'), 'Operator' (set to 'contains'), and 'Expression' (set to 'DB06.ORACLE.COM'). This entire dialog is highlighted with a red box. Below the dialog, the report interface shows the same filters ('Event Time is in the last 24 hours' checked, 'User contains 'ListOfUsers'' unchecked) and the same table data as the previous screenshots.

17. In addition, by clicking on the **Actions Menu** (Gear icon), you can further customize your reports.

You will be able to:

- Select more columns for the report
- Filter the report
- Sort rows
- Highlight rows
- Generate a chart
- Save the report for future use
- Download the report to CSV

The screenshot shows the 'Database Failed Logins' report interface. At the top right, there is a gear icon representing the 'Actions Menu'. A context menu is open over this icon, with a red border highlighting the 'Select Columns' option. The main report area displays a table of failed logins with columns: Source, Event, User, Authentication Method, OS User, Event Status, and Event Time. The table contains eight rows of data. Above the table, there are two filter options: 'Event Time is in the last 24 hours' and 'Source contains 'DB06.ORACLE.COM''. The 'Save Definition' button is also visible at the top right of the report area.

18. When you have finished changing the definition, click the **Save Definition** button.

This screenshot shows the same 'Database Failed Logins' report interface as the previous one, but with the 'Save Definition' button highlighted by a red box. The rest of the interface, including the report table and filters, appears identical to the previous screenshot.

19. Once you have completed this step, navigate back to the **Compliance Reports** tab.

The screenshot shows the Oracle Enterprise Manager 10g interface. The top navigation bar has tabs for Home, Audit Reports, Default Reports, and Compliance Reports. The Compliance Reports tab is highlighted with a red box. Below the tabs is a search bar and a row of buttons: Create PDF, Save Definition, and Cancel. Underneath is a table with columns: Source, Event, User, Authentication Method, OS User, Event Status, and Event Time. A single row is visible: DB06.ORACLE.COM, LOGON, APPS, DATABASE, oracle, 1017, 9/28/2011 12:59:45 AM.

20. Select the report **Schema Changes Report** in the Credit Cards section.

Database Audit Requirements	SOX	PCI DSS	HIPAA	Basel II	FISMA	GLBA
Schema Changes Are you aware of CREATE, DROP and ALTER Commands that are occurring on identified Tables / Columns?	●	●	●	●	●	●

The screenshot shows the Credit Card section of the Oracle Enterprise Manager interface. It includes three tabs: Credit Card, Financial, and Health Care. Under the Credit Card tab, there is a list of audit events: Credit Card Related Data Access, Audit Setting Changes, Before/After Values, Database Failed Logins, Database Login/Logoff, Database Logoff, Database Logon, Database Startup/Shutdown, Deleted Objects, Program Changes, Schema Changes (which is highlighted with a red box), System Events, and User Privilege Change Activity. The Financial and Health Care tabs show similar lists of audit events.

21. We will create a PDF version of a report, which can then be sent to people who require it. After clicking the **Create PDF** button you are taken to the **Create or Schedule PDF Report** screen.

The screenshot shows the Schema Changes report interface. It has a search bar and a row of buttons: Create PDF (highlighted with a red box) and Change Definition. Below is a table with columns: Source, Event, Owner, Target, User, OS User, Event Status, and Host. Three rows are listed: DB06.ORACLE.COM, CREATE TABLE, JSCHAFFER, MY_EMP, JSCHAFFER, oracle, SUCCESS, dbsecurity.oracle.co; DB06.ORACLE.COM, ALTER TABLE, SCOTT, EMP1, JTAYLOR, oracle, 942, dbsecurity.oracle.co; DB06.ORACLE.COM, DROP MATERIALIZED VIEW, JSCHAFFER, MVW1, JSCHAFFER, oracle, SUCCESS, dbsecurity.oracle.co.

22. From here you can schedule the report to be run immediately or on a schedule. We will schedule to run this report on a weekly basis.

In the **Schedule** section, select the radio button **Specify Schedule** option. Then in the **Repeat** section select **Weekly** and leave the remaining fields default, as shown below.

23. In the '**Attestation**' section select the '**KZENG', '**MWALDRON', and '**TBEDNAR**' users and move the user to the right hand pane using the '**>**' button.****

24. We will leave the Report Formatting section all default.

25. Finally, click on the **Schedule** button at the top right hand side of the page.

26. You will be re-directed to the **Report Schedules** where you will see that your report is now scheduled to run every week.

The screenshot shows a 'Report Schedules' interface. At the top, there's a search bar, a row counter set to 15, and a 'Create' button. Below is a table with columns: Category Name, Report Name, Schedule, and Last Run. One entry is visible: 'Credit Card' under 'Category Name', 'Schema Changes' under 'Report Name', 'Weekly (1) Wed 12:00 AM -12:00 Start 28-09-2011' under 'Schedule', and a trash can icon under 'Last Run'. The bottom of the table shows a page number '1 - 1'.

27. We will now create a PDF report immediately and demonstrate the attestation process. Select the report **User Privilege Change Activity** in the Financial section.

Database Audit Requirements	SOX	PCI DSS	HIPAA	Basel II	FISMA	GLBA
Schema Changes Are you aware of CREATE, DROP and ALTER Commands that are occurring on identified Tables / Columns?	●	●	●	●	●	●

The screenshot shows the 'Financial' section of the application. It lists several audit categories under three main headings: Credit Card, Financial, and Health Care. Under 'Financial', the 'User Privilege Change Activity' report is highlighted with a red box. Other reports listed include 'Audit Setting Changes', 'Before/After Values', 'Database Failed Logins', 'Database Login/Logoff', 'Database Logoff', 'Database Logon', 'Database Startup/Shutdown', 'Deleted Objects', 'Program Changes', 'Schema Changes', 'System Events', and 'User Privilege Change Activity'.

28. You will see the following **User Privilege Change Activity** report, click on the **Create PDF** button.

The screenshot shows the 'User Privilege Change Activity' report table. The table has columns: Source, Event, Target, Admin_Option, Grantee_User, Event_Status, User, OS_User, Event_Time, Host, and SQL_Text. A 'Create PDF' button is located at the top right of the table area. The table data includes various privilege changes like grants, drops, and alterations across different Oracle databases.

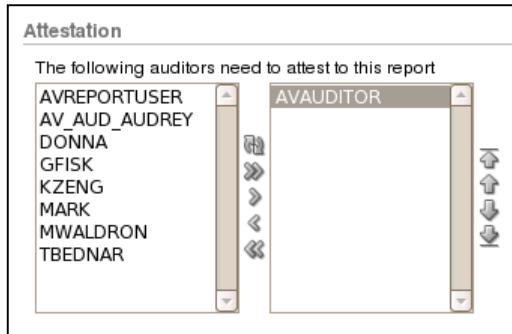
Source	Event	Target	Admin_Option	Grantee_User	Event_Status	User	OS_User	Event_Time	Host	SQL_Text
DB06.ORACLE.COM	SET ROLE	ALL			SUCCESS	JSCANNER	oracle	9/28/2011 12:59:42 AM	dbsecurity.oracle.com	set role all
DB06.ORACLE.COM	GRANT ROLE	DBA		JSMITH	47410	DBA_NICOLE	oracle	9/28/2011 12:59:36 AM	dbsecurity.oracle.com	grant dba to jsmith
DB06.ORACLE.COM	SYSTEM GRANT					DBA_NICOLE	oracle	9/28/2011 12:59:36 AM	dbsecurity.oracle.com	grant dba to jsmith
DB06.ORACLE.COM	DROP PROFILE	NPPFILE				APPS_DB_A_HARVEY	oracle	9/28/2011 12:59:35 AM	dbsecurity.oracle.com	DROP PROFILE NPPFILE
DB06.ORACLE.COM	ALTER PROFILE	NPPFILE				APPS_DB_A_HARVEY	oracle	9/28/2011 12:59:35 AM	dbsecurity.oracle.com	ALTER PROFILE NPPFILE LIMIT FAILED_LOGIN_ATTEMPTS 5
DB06.ORACLE.COM	CREATE PROFILE	NPPFILE				APPS_DB_A_HARVEY	oracle	9/28/2011 12:59:35 AM	dbsecurity.oracle.com	CREATE PROFILE NPPFILE LIMIT PASSWORD_REUSE_MAX 10 PASSWORD_REUSE_TIME 50
DB06.ORACLE.COM	GRANT ROLE	NEW_ROLE	RMTUSR	1081		APPS_DB_A_HARVEY	oracle	9/28/2011 12:59:35 AM	dbsecurity.oracle.com	GRANT NEW_ROLE TO rmtusr IDENTIFIED BY *
DB06.ORACLE.COM	ALTER ROLE	NEW_ROLE			SUCCESS	APPS_DB_A_HARVEY	oracle	9/28/2011 12:59:35 AM	dbsecurity.oracle.com	ALTER ROLE NEW_ROLE IDENTIFIED BY *
DB06.ORACLE.COM	CREATE ROLE	NEW_ROLE			SUCCESS	DBA_NICOLE	oracle	9/28/2011 12:59:35 AM	dbsecurity.oracle.com	CREATE ROLE NEW_ROLE IDENTIFIED BY *
DB06.ORACLE.COM	DROP ROLE	NEW_ROLE			SUCCESS	DBA_NICOLE	oracle	9/28/2011 12:59:35 AM	dbsecurity.oracle.com	DROP ROLE NEW_ROLE

29. You will be brought to the Create or Schedule PDF Report screen. Hopefully, this screen will be familiar from the steps we conducted earlier.

Create or Schedule PDF Report

Category Name: **Financial** Report Name: **User Privilege Change Activity**

30. In the **Attestation** section, move the AVAUDITOR user to the right hand pane.



31. Leave all other fields as default and click the **Create PDF** button.

Create or Schedule PDF Report

Category Name: **Financial** Report Name: **User Privilege Change Activity**

32. You will be taken to the **Generated Reports** sub tab. You may see that the report status is either 'BEING GENERATED' or 'SUCCESS'. To refresh the page, click either the 'Go' button, or click on the **Generated Reports** tab.

Generated Reports

Notify	Details	Category	Report	Generated	Last Attested	Last Notified	Last Notification Sent	Status
<input type="checkbox"/>		Financial	User Privilege Change Activity	9/28/2011 07:41:52 AM	9/28/2011 07:41:52 AM			SUCCESS
<input type="checkbox"/>		Access Reports	Data Access	9/27/2011 11:32:20 PM	9/27/2011 11:39:25 PM	9/27/2011 11:39:25 PM		SUCCESS
<input type="checkbox"/>		Access Reports	Data Access	9/27/2011 11:32:1 PM		9/27/2011 11:32:1 PM		SUCCESS

1 - 3

33. Click on the '**Details**' button associated with the report.

Notify	Details	Category	Report	Generated	Last Attested	Last Notified	Last Notification Sent	Status
<input type="checkbox"/>		Financial	User Privilege Change Activity	9/28/2011 07:41:52 AM	9/28/2011 07:41:52 AM			SUCCESS

34. Click on the ‘View Report’ button to review the PDF. If this shows you a link to the PDF, just click on the link.

Details for Generated Report
Report: User Privilege Change Activity Generated 9/28/2011 07:41:52 AM Retained till 3/28/2012
Save Save & Attest Done View Report
New Note
Previous Notes
*** AV_AUD_AUDREY *** 9/28/2011 07:41:52 AM *** Report Generated ***
Attestations
Attested by On
AUDITOR

35. Click on the Details button.

Report View - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://dbsecurity.oracle.com:5707/av/console/f?p=7700:4:255571279368394::NO::P4_REPORT_ID:
Most Visited Audit Vault Grid Control DB01 EM DB01 DBV DB02 EM DB03 EM DB04 EM DB06 EM DB06 DBV
ORACLE Enterprise Manager 10g Audit Vault Home Audit Reports Help Logout Settings
Detail Reports Compliance Reports Custom Reports Generated Reports Report Schedules Entitlement Snapshots
Downloads Details
Oracle Audit Vault
Source Event Target Admin Option Grantee User Event Status User OS User Event Time Host SQL Text
DB06.OR SET ROLE ALL SUCCEJS JSCHAFFER oracle 9/28/2011 12:59:42 AM dbsecurity set role all
DB06.OR GRANT ROLE DBA JSMITH 47410 DBA_NICOLE oracle 9/28/2011 12:59:36 AM dbsecurity grant dba to jsmith
DB06.OR SYSTEM ROLE 47410 DBA_NICOLE oracle 9/28/2011 12:59:36 AM dbsecurity grant dba to jsmith
DB06.OR DROP PROFILE NPFILE 47400 APPS_DB_A_HARVEY oracle 9/28/2011 12:59:35 AM dbsecurity DROP PROFILE NPFILE
DB06.OR ALTER PROFILE NPFILE 47400 APPS_DB_A_HARVEY oracle 9/28/2011 12:59:35 AM dbsecurity ALTER PROFILE

36. Once back at the Details for Generated Report add a new attestation note such as, ‘I have reviewed and attest to the data in this Report’ and click on the ‘Save & Attest’ button

Details for Generated Report
Report: User Privilege Change Activity Generated 9/28/2011 07:41:52 AM Retained till 3/28/2012
Save Save & Attest Done View Report
New Note
I have reviewed and attest to the data in this report.

After clicking on the ‘Save & Attest’ button you will see that the attestation note is stored with the report, as shown below.

The screenshot shows a section titled 'Attestations' within a larger interface. It includes a table with two rows:

Attested by	On
AV_AUD_AUDREY	9/28/2011 07:49:49 AM
AVAUDITOR	

Click on **Done** once you have finished.

D. Additional Steps

1. Quickly review other reports and the audit data they collect.

Account Management Activity

In this section you will find activity related to the following audit policies:

- AUDIT ALTER PROFILE;
- AUDIT ALTER USER;
- AUDIT CREATE PROFILE;
- AUDIT CREATE USER;
- AUDIT DROP PROFILE;
- AUDIT DROP USER;

Command Activity

In this category you will see activity related to the issuing of audit commands themselves.

Object Management Activity

This will have all DDL activity. Here are some sample audit policy statements that will generate activity in this category.

- AUDIT CREATE DIMENSION;
- AUDIT CREATE DIRECTORY;
- AUDIT CREATE INDEX;
- AUDIT CREATE MATERIALIZED VIEW;
- AUDIT CREATE MATERIALIZED VIEW LOG;
- AUDIT CREATE OUTLINE;
- AUDIT CREATE PUBLIC DATABASE LINK;
- AUDIT CREATE PUBLIC SYNONYM;
- AUDIT CREATE SCHEMA;
- AUDIT CREATE SEQUENCE;

- AUDIT CREATE SYNONYM;
- AUDIT CREATE TABLE;
- AUDIT CREATE VIEW;

Peer Association (DB Link) Activity

This category contains all activity related to Database Links:

- AUDIT CREATE DATABASE LINK;
- DROP DATABASE LINK;

Role and Privilege Category Activity

Role and Privilege Management captures:

- AUDIT ALTER ROLE;
- AUDIT CREATE ROLE;
- AUDIT DROP ROLE;
- AUDIT GRANT OBJECT;
- AUDIT GRANT ROLE;
- AUDIT REVOKE OBJECT;
- AUDIT REVOKE ROLE;

System Management Activity

Here are some examples of the audit policy that would trigger activity in this category.

- AUDIT ALTER CLUSTER;
- AUDIT ALTER DATABASE;
- AUDIT ALTER ROLLBACK SEG;
- AUDIT ALTER SYSTEM;
- AUDIT ALTER TABLESPACE;
- AUDIT ANALYZE CLUSTER;
- AUDIT CREATE CLUSTER;
- AUDIT CREATE CONTROL FILE;
- AUDIT CREATE DATABASE;
- AUDIT CREATE ROLLBACK SEG;
- AUDIT CREATE TABLESPACE;
- AUDIT DISABLE ALL TRIGGERS;
- AUDIT DROP CLUSTER;
- AUDIT DROP ROLLBACK SEG;
- AUDIT DROP TABLESPACE;
- AUDIT ENABLE ALL TRIGGERS;
- AUDIT FLASHBACK;
- AUDIT FLASHBACK DATABASE;
- AUDIT PURGE DBA_RECYLEBIN;
- AUDIT PURGE TABLESPACE;
- AUDIT SHUTDOWN;

- AUDIT STARTUP;
- AUDIT SUPER USER DDL;
- AUDIT SUPER USER DML;
- AUDIT SYSTEM GRANT;
- AUDIT SYSTEM REVOKE;
- AUDIT TRUNCATE CLUSTER;

User Session Activity

You will see all the user login/logoff/session information in this category.

BEFORE / AFTER VALUES (REDO) REPORTS

Database Audit Requirements	SOX	PCI DSS	HIPAA	Basel II	FISMA	GLBA
Data Changes Do you have visibility into Insert, Update, Merge, Delete commands?	●			●		

1. Click on the **Compliance Reports** tab.



2. Click on the **Before/After Values** link under the **Financial** section. The Before / After Values Reports displays changes to row data when an INSERT, UPDATE, or DELETE operation occurs on the Oracle Database. This report is especially useful if you are utilizing the REDO collector to extract the before or after values of data updates.

Credit Card	Financial	Health Care
 Credit Card Related Data Access Audit Setting Changes Before/After Values Database Failed Logins Database Login/Logout Database Logoff Database Logon Database Startup/Shutdown Deleted Objects Program Changes Schema Changes System Events User Privilege Change Activity	 Financial Related Data Access Financial Related Data Modifications Audit Setting Changes Before/After Values Database Failed Logins Database Login/Logout Database Logoff Database Logon Database Startup/Shutdown Program Changes Schema Changes System Events User Privilege Change Activity	 EPHI Related Data Access Audit Setting Changes Before/After Values Database Failed Logins Database Login/Logout Database Logoff Database Logon Database Startup/Shutdown Deleted Objects Schema Changes System Events User Privilege Change Activity

Before/After Values									
<input type="text"/> Rows 15 <input type="button" value="Go"/> <input type="button" value="Create PDF"/> <input type="button" value="Change Definition"/>									
<input checked="" type="checkbox"/> Event Time is in the last 24 hours <input type="checkbox"/> <input type="checkbox"/>									
Source	Target	Event	Event Status	User	Host	Event Time	Data Trace Values		
Column	Old Value	New Value							
DB06.ORACLE.COM	EMPLOYEES	UPDATE	SUCCESS	DBA_DEBRA	dbsecurity.oracle.com	9/30/2011 03:50:9 PM	EMPLOYEE_ID	1000	1000
							EMAIL	tammy.bednar@oracle.com	tbednar@oracle.com
DB06.ORACLE.COM	EMPLOYEES	INSERT	SUCCESS	DBA_DEBRA	dbsecurity.oracle.com	9/30/2011 03:50:9 PM	EMPLOYEE_ID	1000	1000
							FIRST_NAME	Tammy	
							LAST_NAME	Bednar	
							EMAIL	tammy.bednar@oracle.com	
							PHONE_NUMBER	000-000-0000	
							HIRE_DATE	07/09/2003 00:00:00	
							JOB_ID	SALESMAN	
							SALARY	2800	
							COMMISSION_PCT	0	
							MANAGER_ID	124	
							DEPARTMENT_ID	50	

E. Summary

In this lab, you:

1. *Generated a variety of reports available in Audit Vault including:*
 - a. *Data Access Report*
 - b. *Database Vault Audit Report*
 - c. *System Management*
 - d. *Entitlement Snapshot*
 - e. *Data Access Report with Redo (Before and After Values)*
 - f. *Compliance Reports*

LAB EXERCISE 03 – GAIN REAL-TIME DATABASE ACTIVITY MONITORING USING AUDIT VAULT ALERTING

INTRODUCTION

After monitoring and collecting database activity CashBankTrust wants to be alerted when certain activities occur. They have identified several high-risk activities that the security team needs to be made aware of, these include account management and database structural changes. The alerts will let the security team know when there has been activity thereby triggering auditing.

A. Overview

Oracle Audit Vault provides security personnel with the ability to detect and alert on activities that may indicate attempts to gain unauthorized access and/or abuse system privileges. Oracle Audit Vault can generate alerts for system defined and user defined audit events. Oracle Audit Vault continuously monitors the audit data collected, evaluating the activities against defined alert conditions. Alerts can be associated with any auditable database event including system events such as changes to application tables and creating privileged users. For instance, an alert could be generated when someone attempts to access sensitive business information. The Oracle Audit Vault interface provides graphical summaries of activities causing alerts. These include a summary of alert activity and top sources by number of alerts. Oracle Audit Vault users can click on the summary graphs and drill down to a more detailed report. Alerts for the purpose of reporting are grouped by the sources with which they are associated. Alerts can also be grouped by the event category to which the event belongs, and by the severity level of the alert (warning or critical).

During this lab you will:

1. *Configure Audit Vault to send emails on the cloud*
2. *Create a Distribution List for Audit Vault Alerts*
3. *Modify the email template for Audit Vault Alerts*
4. *Add a new Audit Vault Alert Status*
5. *Create an Audit Vault Alert with the Web Interface*
6. *Test that the alert is functioning*
7. *View the near real-time nature of alert functionality*

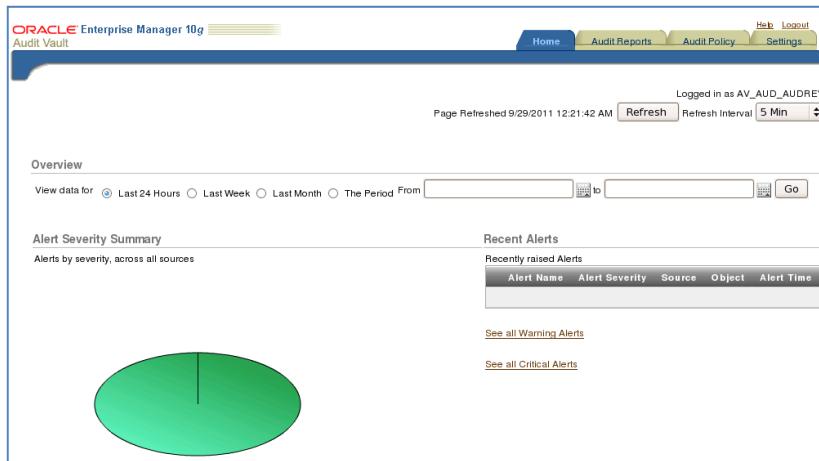
B. Setup & Preparation

- You should have already completed **Lab Configuration 02 – REDUCE TIME TO COMPLIANCE USING ORACLE AUDIT VAULT REPORTING** before using this lab.

C. GAIN REAL-TIME DATABASE ACTIVITY MONITORING USING AUDIT VAULT ALERTING

Viewing Activity and Alerts

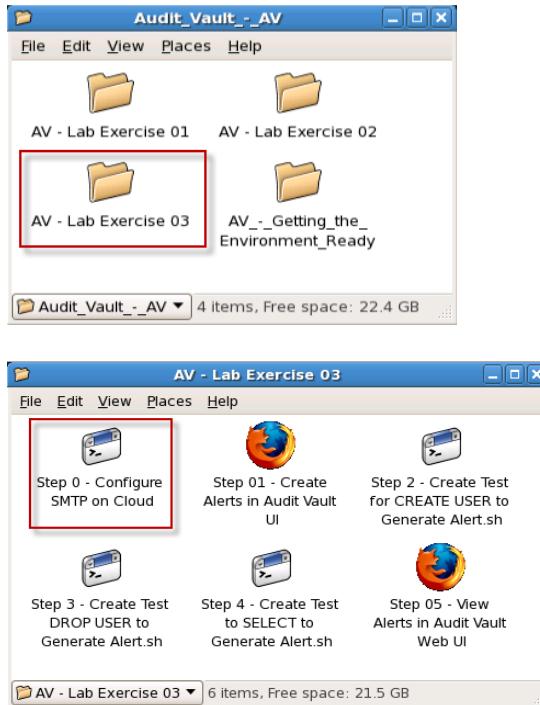
1. The “Home” screen shows an overview of activity across all sources in our environment. You may need to click the **Refresh** button to show the details of activity that you generated in an earlier lab. You should see a screen that looks similar to the following:



The Dashboard is organized into four horizontal sections. The first includes a pie chart displaying all alerts across all sources (we have just one source – db06 – in our environment) as well as a tabular display of recent alerts. The second section shows a bar chart of audit sources with the highest number of alerts, and another bar chart showing the number of alerts according to audit category. The third section contains a chart of database objects that are being most frequently accessed, and another chart displaying failed logins across all sources. The fourth and last section shows a list of all audit reports requiring the current user’s attestation. There will be no alerts or reports shown here at this stage.

All audit activity that is collected from your sources, and is defined as an alert, will be categorized then made available in this chart. There are 14 categories. During this lab we will review each of the categories and the activity that is captured in them.

- Open the **AV – Lab Exercise 03** Folder and double-click the icon for **Step 0 – Configure SMTP on Cloud** to set the SMTP Email server address in Audit Vault. This address will be used to send email alerts later in the lab.



After clicking you the icon you will see that the message ‘SMTP Server registered successfully’ appears in the shell window. This means that Audit Vault can now send emails when an alert is configured to do so.

The command that was sent to configure the SMTP server on your machine is:

```
oracle@cloud.oracle.com: [/home/oracle]:BASE
$ av
ORACLE_SID=av
ORACLE_HOSTNAME=cloud.oracle.com
ORACLE_BASE=/u01/oracle
ORACLE_HOME=/u01/oracle/product/10.2/av
OH=/u01/oracle/product/10.2/av
oracle@cloud.oracle.com: [/home/oracle]:AV
$ avca register_smtp -server localhost:25 -sender_id avadmin -
sender_email avadmin@localhost.com -noauth
```

The output of this script/command is as follows:



```
Terminal
File Edit View Terminal Tabs Help
ORACLE_SID=av
ORACLE_BASE=/u01/app
ORACLE_HOSTNAME=dbsecurity.oracle.com
ORACLE_HOME=/u01/oracle/product/10.2/av
OH=/u01/oracle/product/10.2/av
SMTP server registered successfully.
Please Press the Enter Key to Continue
```

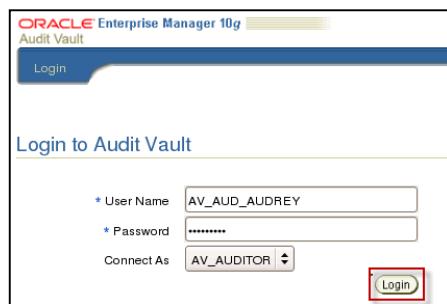
Documentation on configuring the SMTP Server can be found at:

http://download.oracle.com/docs/cd/E14472_01/doc.102/e14459/av_adm_mng_manage.htm#CACJFCGI

3. Click the icon, **Step 01 – Create Alerts in Audit Vault UI** to open the browser.



4. Log into the Audit Vault console as **AV_AUD_AUDREY/Manager_1**.



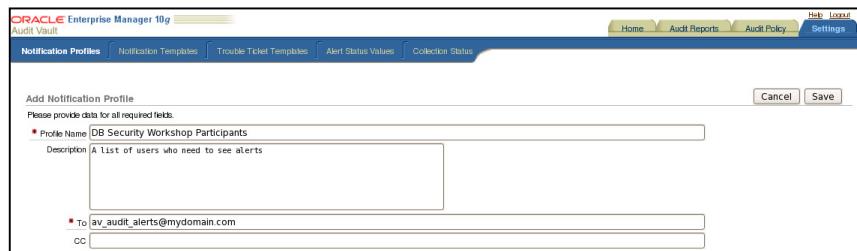
5. We will be adding a new Alert Email Notification Profile. This Notification Profile will serve as a Distribution List for the Alert we will create later in the lab. Once logged in navigate to the **Settings** Tab, then the **Notification Profile** sub tab.



Click on the **Create** button. You will see the **Add Notification Profile** Screen as shown below. Add the following pieces of information:

Profile Name: DB Security Workshop Participants
Description: A List of people who need to see alerts
To: av_audit_alerts@mydomain.com

Please replace the 'av_audit_alerts@mydomain.com' with your email address.



Once the data is entered, click the **'Save'** button. You will be taken back to the **'Notification Profile'** Tab, where you will see the new entry.

Notification Profiles			
Profile Name	Description	To	CC
DB Security Workshop Participants	A list of people who need to see alerts	mark.d.waldron@oracle.com	
1 - 1			

6. Navigate to the **Settings** tab (which you should already be on), then to the **Notification Templates** sub tab. From here will be able to manage the existing template definitions and create new ones. Once at this page you will see the following:

Name	Description	Subject	Format	Type
Alert Notification Template	Default Alert Notification Template	Audit Vault Alert: #AlertName#, #AlertTime#	Plain Text	Alert
Report Notification Template	Default Report Notification Template	Audit Vault report ready for your review: #ReportName#, #DateCreated#	Plain Text	Report Notification
Report Attached Template	Default Report Attachment Template	Audit Vault report: #ReportName#, #DateCreated#	Plain Text	Report Attachment

We will edit the **Alert Notification Template**, which is the default template used for sending emails. You could create a new template, but for the purpose of this lab we will just edit the existing one. Click on the **Alert Notification Template** link on the left hand side of the page.

Edit Notification Template

Please provide data for all required fields.

Type: Alert Report Attachment Report Notification

Name: Alert Notification Template

Description: Default Alert Notification Template

Subject: Audit Vault Alert: #AlertName#, #AlertTime#

Format: Plain Text HTML

Body: #AlertBody#

Please do not reply to this email. This is an automated message.

Available Tags

- #AlertName#
- #AlertTime#
- #AlertStatus#
- #Object#
- #AlertSeverity#
- #ClientHost#
- #ClientHostIP#
- #Event#
- #OSUserName#
- #UserName#
- #SourceName#
- #Description#
- #TroubleTicketID#
- #TroubleTicketTime#
- #URL#
- #AlertBody#

Add in the **#AlertStatus#** and **#SourceName#** fields into the email subject, as shown in the screen below. Click the **Save** button once completed. We will test the new template later in the lab.

The screenshot shows the 'Edit Notification Template' dialog. The 'Subject' field contains the placeholder '#AuditVault Alert: #AlertName#, #AlertTime#, #AlertStatus#, #SourceName#'. The 'Save' button is highlighted with a red box.

7. Navigate to the **Settings** tab (which you should already be on), then to the **Alert Status Values** sub tab. You will see that there are two default Alert status values. These values are used to maintain a status for each alert that is created in Audit Vault. You can then manage alerts according to your business requirements. We will be adding a new status to record that we are reviewing a given alert. Click on the '**Create**' button.

The screenshot shows the 'Alert Status Values' list. The 'Create' button is highlighted with a red box.

You will see the following:

The screenshot shows the 'Add Alert Status Value' dialog. The 'Status Value' field is empty and highlighted with a red box.

Now add a new Alert Status, enter the following information

Status Value:	REVIEWING
Description:	Alert being reviewed

Once completed, click the **Save** button.

The screenshot shows the 'Add Alert Status Value' dialog in Oracle Enterprise Manager 10g. The 'Status Value' field is set to 'REVIEWING' and the 'Description' field contains 'Alert being reviewed'. There are 'Cancel' and 'Save' buttons at the top right.

8. We will now create a new alert in Audit Vault. This alert will let us know when a new Oracle DB User account has been created. Click on the **Audit Policy > Alerts** sub tab. Click on the **Create** button.

The screenshot shows the 'Alerts' list page in Oracle Enterprise Manager 10g. A red box highlights the 'Create' button. The page also includes a search bar, a 'Rows' dropdown set to 15, and a 'Go' button.

9. Enter the following information for the Alert. You will have to scroll down to complete this form.

- Alert: **CREATE_USER**
- Alert Severity: **Critical**
- Audit Source Type: **ORCLDB**
- Audit Source: **DB06.ORACLE.COM**
- Audit Event Category: **ACCOUNT MANAGEMENT**
- Audit Event: **CREATE USER**
- Audit Event Status: **Both**
- Notification Action: **<Select the 'Alert Notification Template'>**
- Profile: **< Select the 'DB Security Workshop Participants'>**

The screenshot shows the 'Create Alert Rule' dialog in Oracle Enterprise Manager 10g. The 'Alert' field is set to 'CREATE_USER'. The 'Description' field is empty. Under 'Alert Severity', 'Audit Source Type', 'Audit Source', and 'Audit Event Category' are all set to their respective values from the previous list. The 'Basic' radio button is selected under 'Specify additional alert conditions in'. The 'Basic Alert Condition' section includes fields for 'User' and 'Table', both set to empty. The 'Audit Event' dropdown is set to 'CREATE USER'. The 'Audit Event Status' radio buttons show 'Success' and 'Failure' as options, with 'Both' selected.

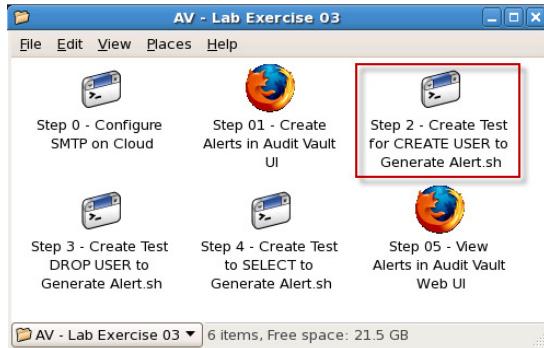
10. After selecting the Notification Action, click the **Add To List** button. Note that we will not add any details in the **Alert Action** area at the bottom of this screen but note that this is where, given the occurrence of an alert, we could email individuals and/or create a trouble ticket in our Help Desk system if we so wished.

11. Click OK to create the alert.

12. Verify that you should see the newly created alert in the summary screen, as shown below.

Alert Name	Description	Audit Source	Audit Source Type	Audit Event Category	Delete
CREATE_USER		DB06.ORACLE.COM	ORCLDB	ACCOUNT MANAGEMENT	

13. Go back to your open folder for **AV – Lab Exercise 03** and execute the script to create some new users in the db06 database by clicking on the **Step 2 – Create Test for CREATE USER to Generate Alert.sh** icon as shown below.



14. Click on the icon, **create_user_for_alert.out** to view the results of the executed scripts.



15. In the Audit Vault console, select the **Audit Reports** tab then select the **All Alerts** report, as shown below.



16. Your alert report will look similar to that shown below. Notice that the report is being filtered to show only alerts in the past 24 hours and alerts that have a status of anything other than **CLOSED**. Click on the entry for **TNUGENT** in the list of alerts.

All Alerts										
<input type="button" value="Set Status to"/> <input type="button" value="CLOSED"/> <input type="button" value="Apply"/> <input type="button" value="Notify"/> <input type="button" value="Log Trouble Tickets"/>										
<input type="button" value="Create PDF"/>										
<input type="checkbox"/> <input checked="" type="checkbox"/> Audit Vault Alert Time is in the last 24 hours <input type="checkbox"/> <input checked="" type="checkbox"/>										
Select	Details	Alert Name	Object	Event	Event Category	User	Source	Alert Severity	Event Time	Alert Status
<input type="checkbox"/>	<input checked="" type="checkbox"/>	CREATE_USER	TNUGENT	CREATE USER	ACCOUNT MANAGEMENT	DVACCTMGR	DB06.ORACLE.COM	Critical	9/28/2011 08:43:15 AM	NEW
<input type="checkbox"/>	<input type="checkbox"/>	CREATE_USER	PRAYER	CREATE USER	ACCOUNT MANAGEMENT	DVACCTMGR	DB06.ORACLE.COM	Critical	9/28/2011 08:43:15 AM	NEW
<input type="checkbox"/>	<input type="checkbox"/>	CREATE_USER	HBRADBURY	CREATE USER	ACCOUNT MANAGEMENT	DVACCTMGR	DB06.ORACLE.COM	Critical	9/28/2011 08:43:15 AM	NEW
<input type="checkbox"/>	<input type="checkbox"/>	CREATE_USER	LIMH	CREATE USER	ACCOUNT MANAGEMENT	DVACCTMGR	DB06.ORACLE.COM	Critical	9/28/2011 08:43:15 AM	NEW

17. Examine the detail alert record and all of the sections including the Notes, Notification and Trouble Ticket Information.

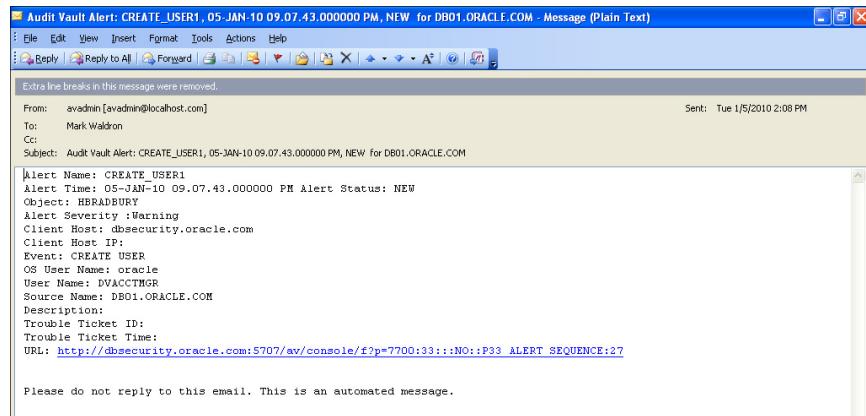
Alert Details	
Event Time 9/28/2011 08:43:15 AM Audit Vault Alert Time 9/28/2011 08:43:40 AM Name CREATE_USER Status NEW <input type="button" value="CLOSED"/> <input type="button" value="Update"/> Severity Critical	
Audit Record Details	
SCN 5654061 Source Type ORCLDB Source DB06.ORACLE.COM Host dbsecurity.oracle.com Version 11.2.0.2.0 IP Address 10.245.110.79 Audit Vault Time 9/28/2011 08:43:39 AM Event Time 9/28/2011 08:43:15 AM Event Status SUCCESS Event Name CREATE_USER Category ACCOUNT MANAGEMENT Owner Package Information User DVACCTMGR Enduser Session Login Name SQL Text create user tnugent identified by * Statement ID 64 Privilege Name 20 Context 570207 Sub Context 01000700B8060000	

You will notice that you can set the status of this alert to the **REVIEWING** Status we had previously created. After you have selected the new status, click on the **Update** button. You can then filter your alert report using this new status.

<input style="border: none; background-color: inherit; color: inherit; font-size: inherit; width: 100%; height: 100%;" type="button" value="< Report View"/>	
Alert Details	
Event Time 9/28/2011 08:43:15 AM Audit Vault Alert Time 9/28/2011 08:43:40 AM	
Name CREATE_USER Status REVIEWING <input style="border: 1px solid #ccc; padding: 2px 10px; margin-right: 10px;" type="button" value="REVIEWING"/> <input style="border: 1px solid #ccc; padding: 2px 10px;" type="button" value="Update"/>	
Severity Critical	

18. If you configured the Cloud SMTP Server to deliver emails, check the email account you specified in the alert to see whether you have any new alert emails. You should see an email delivered by Audit Vault and initiated by this alert.

Date: Today				
	avadmin	Audit Vault Alert: CREATE_USER, 28-SEP-11 08.43.15.130023 AM - N...	Wed 9/28...	2 KB
	avadmin	Audit Vault Alert: CREATE_USER, 28-SEP-11 08.43.15.307603 AM - N...	Wed 9/28...	2 KB
	avadmin	Audit Vault Alert: CREATE_USER, 28-SEP-11 08.43.15.178066 AM - N...	Wed 9/28...	2 KB
	avadmin	Audit Vault Alert: CREATE_USER, 28-SEP-11 08.43.15.046582 AM - N...	Wed 9/28...	2 KB



19. We will proceed by creating a two more alerts. Repeat the steps above to create another alert for the DROP USER command and another alert that will provide notification when a DBA user account attempts to directly access data outside of the intended application user account.

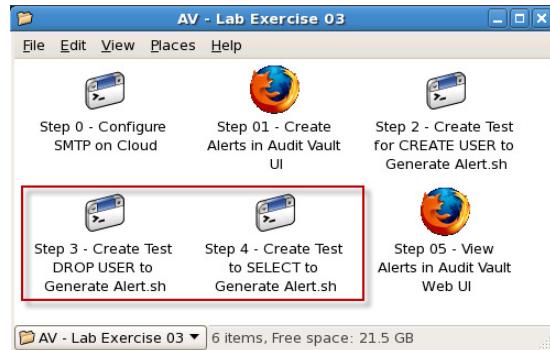
Enter the following information for the **DROP_USER** Alert.

- Alert: **DROP_USER**
- Alert Severity: **Critical**
- Audit Source Type: **ORCLDB**
- Audit Source: **DB06.ORACLE.COM**
- Audit Event Category: **ACCOUNT MANAGEMENT**
- Audit Event: **DROP USER**
- Audit Event Status: **Both**
- Notification Action: <Select the 'Alert Notification Template'>
- Profile: < Select the 'DB Security Workshop Participants'>

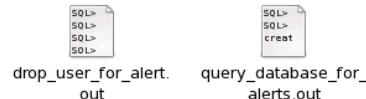
Enter the following information for the **EMPLOYEE_SELECT_NON_APP** Alert.

- Alert: **EMPLOYEE_SELECT_NON_APP**
- Alert Severity: **Warning**
- Audit Source Type: **ORCLDB**
- Audit Source: **DB06.ORACLE.COM**
- Audit Event Category: **DATA ACCESS**
- Specify additional alert conditions in **Advanced**
- Alert Condition:
**(#SOURCE_EVENTID#=’3’ or
#SOURCE_EVENTID#=’6’) and
Upper(#USERNAME#) like ‘DBA_%’**
- Audit Event Status: **Both**
- Notification Action: <Select the 'Alert Notification Template'>
- Profile: < Select the 'DB Security Workshop Participants'>

20. Click 'Add to List' to add the notification action and profile, then select Return back to folder for AV – Lab Exercise 03. Click on the icons, **Step 3 – Create Test DROP USER to Generate Alert.sh** and **Step 4 – Create Test to SELECT to Generate Alert.sh** to generate database activity on DB06 to test both alert conditions.



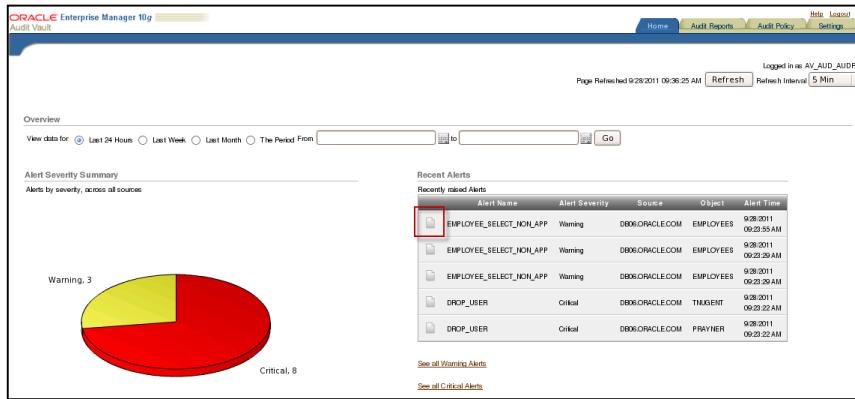
21. Click on the icons, **drop_user_for_alert.out** and **query_database_for_alert.out** to view the results of the executed scripts.



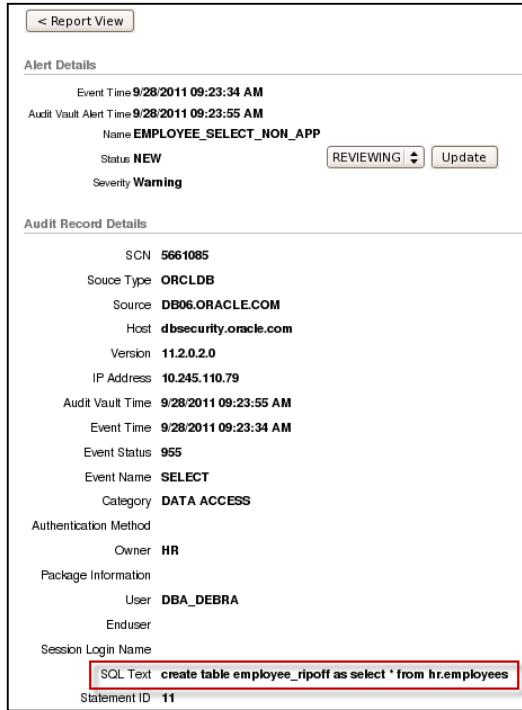
23. Return to your browser window or go to the folder AV – Lab Exercise 03 and click on the icon **Step 05 – View Alerts in Audit Vault**. We will review the alerts generated.



24. On the Home tab, review the **Alert Severity Summary** and **Recent Alerts**.



25. On the screen above, click on details icon for the Alert Name **EMPLOYEE_SELECT_NON_APP**. We will review the details of this alert. Notice that the alert helped us identify that an unauthorized activity took place by a user other than the expected application user.



D. Summary

In this lab, you:

1. Configured Audit Vault Alerting
2. Created an Audit Vault Alert.
3. Triggered the alert via several SQL actions.
4. Triggered the alert via a single SQL action.

LAB CONFIGURATION – ORACLE DATABASE FIREWALL

OVERVIEW

For these lab exercises, you will need to have the following infrastructure components running and available. We have already started the necessary infrastructure for you. You will be connecting to a Windows Desktop client.

- **Database: 11gR2 (DB06)**
- **Oracle Database Firewall**
- **Windows Client**

Your instructor will be providing you with information about the environment, including all hostnames, IP address and username/passwords. Use this table to write down the IP address information if necessary.

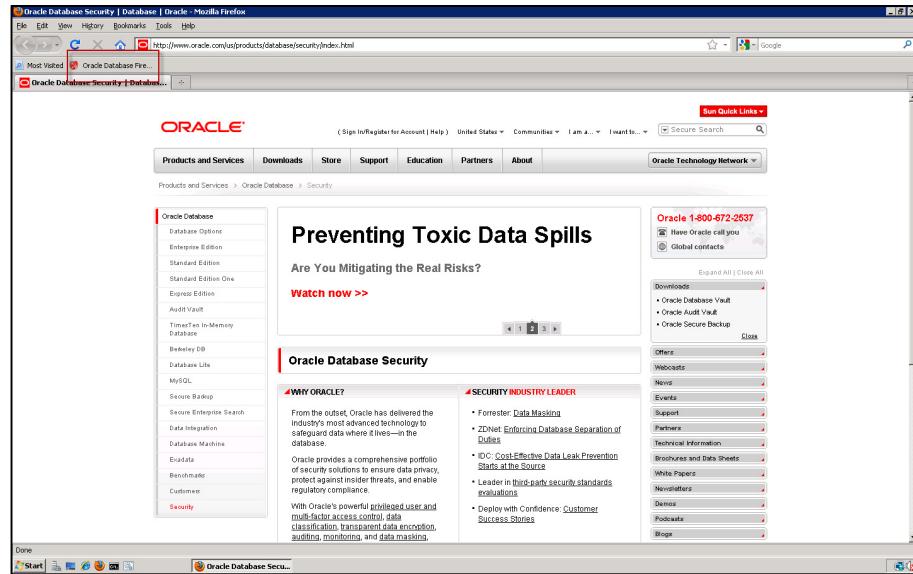
IP Addresses to be used in these Lab Exercises
Database IP:
Database Firewall IP:
Windows Client IP:

We will step through the simplified version connecting to the three environments you will be using for this lab.

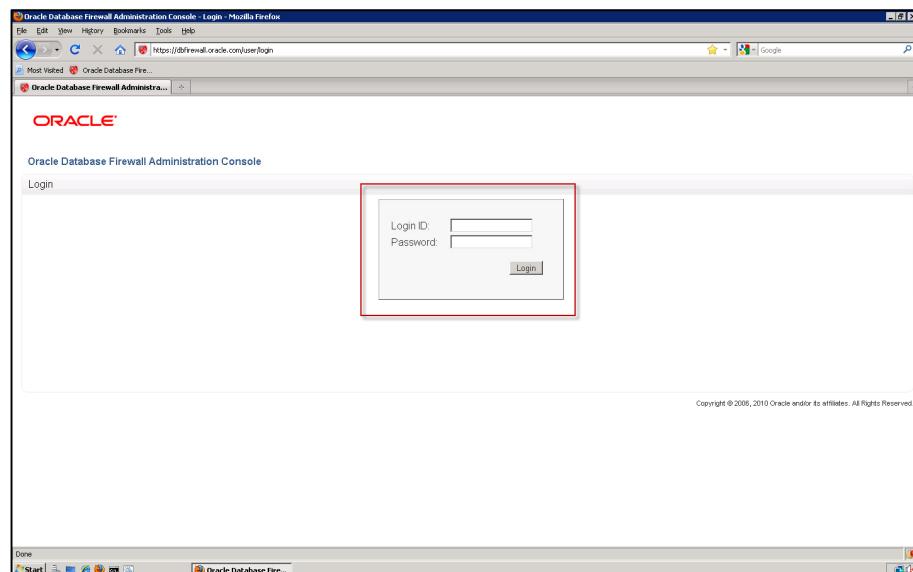
1. Connect to the Windows Environment using the information provided by the instructor. On the desktop, open the Firefox browser.



2. In the Browser, Click on the ‘Oracle Database Firewall’ Link.



You should see a login page for the Oracle Database Firewall Administration Console.



You are now ready to start the hands-on labs.

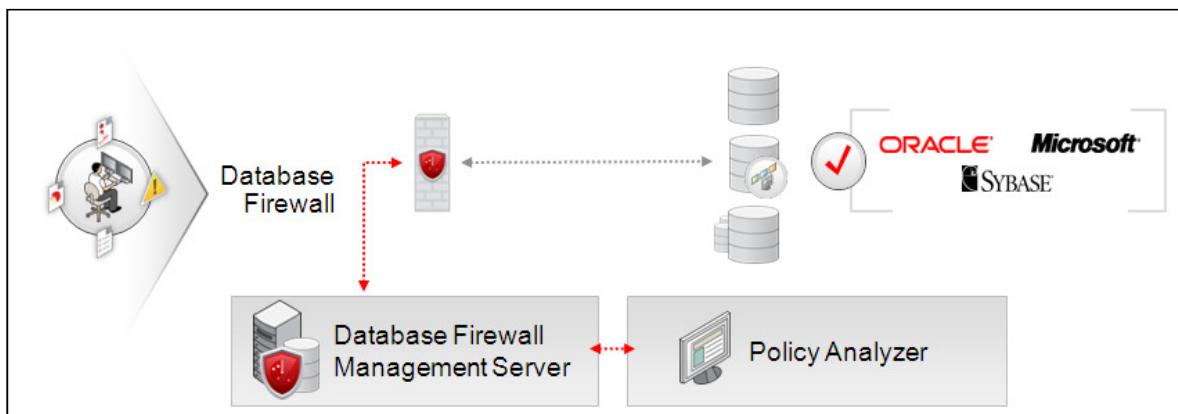
LAB EXERCISE 00 – ORACLE DATABASE FIREWALL OVERVIEW

INTRODUCTION

The Oracle Database Firewall system secures and protects data in SQL databases. It blocks attempted attacks, logs activity, and produces related warnings and provides tools to assess vulnerabilities. The Database Firewall system enhances existing database security features, such as encryption and user authentication, and brings significant advantages over traditional database firewall systems. Traditional systems usually test the syntax of statements passed to the database, recognizing predefined expressions. Creating a set of rules using this technique requires a hand-crafted approach and can be very time-consuming and complex, even for someone very knowledgeable about the database. Even if significant resources create satisfactory protection for known threats, little protection may be offered for unknown threats. The Database Firewall addresses these challenges.

The Oracle Database Firewall system works by analyzing the meaning of the SQL statements that database clients send to the database. This provides a much higher degree of protection than traditional database firewalls, because it does not depend on the source of an attack or on recognizing the syntax of known security threats. The database firewall can block previously unseen attacks (known as "zero-day" attacks), including those targeted individually against your organization. Zero-day attacks are becoming more widespread, and there is great need to protect databases against such attacks.

The Oracle Database Firewall protects without affecting the performance of the database server or its client applications. The system protects from attacks originating from inside firewalls, as well as from external sources. The central feature of an Oracle Database Firewall system is the ability to scan and log SQL traffic to the monitored databases. The Database Firewall system scans all SQL statements passed to the databases in real time. You can configure enforcement points to monitor traffic, generate warnings of potential attacks, and block harmful statements.



The 'Database Firewall' is comprised of the following components:

- Oracle Database Firewall
- Oracle Database Firewall Management Server
- Oracle Database Firewall Analyzer

Oracle Database Firewall

The system employs at least one Database Firewall for up to 20 Protected Databases which can optionally report to the Management Server. The Database Firewall handles real-time recording and analysis of SQL transaction requests and responses from a protected database, which may be an Oracle, Microsoft SQL Server, Sybase Adaptive Server Enterprise (ASE), or Sybase SQL Anywhere database. The Database Firewall enforces data policies, known as baseline policies, that the Oracle Database Firewall Analyzer creates. You can use Oracle Database Firewall to configure pairs of network ports. Each pair of ports can connect to a separate database network or to a different point in the same network. The default number of enforcement points is 20, but can vary depending on the platform specification. See the Oracle Database Firewall Installation Guide for a list of supported platforms.

Oracle Database Firewall Management Server

The Database Firewall Management Server remotely manages all Database Firewalls that are connected to it. It accumulates SQL from these firewalls, stores and manages About the Oracle Database Firewall System Introducing the Oracle Database Firewall System 1-3 log files, provides business reports, and integrates with third-party applications as needed.

Oracle Database Firewall Analyzer

You use the Oracle Database Firewall Analyzer to create the baseline, which is the policy that the Database Firewalls use to block, alert, log or permit SQL statements for the database. The analyzer does this by reading logs that the Database Firewalls create. Oracle Database Firewall Analyzer is installed on a Microsoft Windows client computer.

A. Lab Scenarios and Objectives

In our lab scenario we have a fictional company called ACME Corp which is evaluating its security posture in preparation for a consolidation program. ACME Corp currently supports multiple applications serving multiple external customers. Due to current security policy each new application has to be hosted on a new server. This has caused the IT infrastructure to expand rapidly and incur significant administration overhead. The consolidation program will migrate multiple applications to a single platform. However, after reviewing the new architecture ACME's security team has raised a number of concerns. One of the chief concerns is about SQL Injection being able to effect databases managed on a single platform. ACME has decided to implement Oracle Database Firewall (DBFW) to address these security concerns. By implementing DBFW to block unauthorized SQL traffic, ACME can continue to progress the consolidation project.

LAB EXERCISE 01 – ORACLE DATABASE FIREWALL ENFORCEMENT POINTS TO MONITOR AND PROTECT DATABASES

INTRODUCTION

Oracle Database Firewall, part of Oracle's comprehensive portfolio of database security solutions, is the first line of defence for both Oracle and non-Oracle databases. It monitors database activity on the network to help prevent unauthorized access, SQL injections, privilege or role escalation, and other external and internal attacks - all in real time. Based on innovative SQL grammar technology that can reduce millions of SQL statement into a small number of SQL characteristics, Oracle Database Firewall offers unmatched accuracy, scalability, and performance. Enforcement of positive (white lists) and negative (black lists) security models provides protection from threats without time consuming and costly false positives. Oracle Database Firewall also enables organizations to address SOX, PCI, HIPAA/HITECH, and other regulatory requirements without changes to existing applications or databases, and demonstrate compliance with over a hundred built-in customizable reports.

A. Lab Scenarios and Objectives

In this lab exercise, you will accomplish the following:

1. *Create and setup an enforcement point in Oracle Database Firewall*
2. *Generate simulated database activity on expected authorized traffic*
3. *Monitor Oracle Database traffic in the Database Firewall Management console*
4. *Prepare to train the Database Firewall with acceptable and expected behavior*

B. Setup and Preparation

- You should have completed LAB EXERCISE 00 – ORACLE DATABASE FIREWALL OVERVIEW
- Connected to the Windows Environment.
- Connected to the Oracle Database Firewall Web Administration Console.

C. CONFIGURE DBFW ENFORCEMENT POINTS TO MONITOR AND PROTECT DATABASES

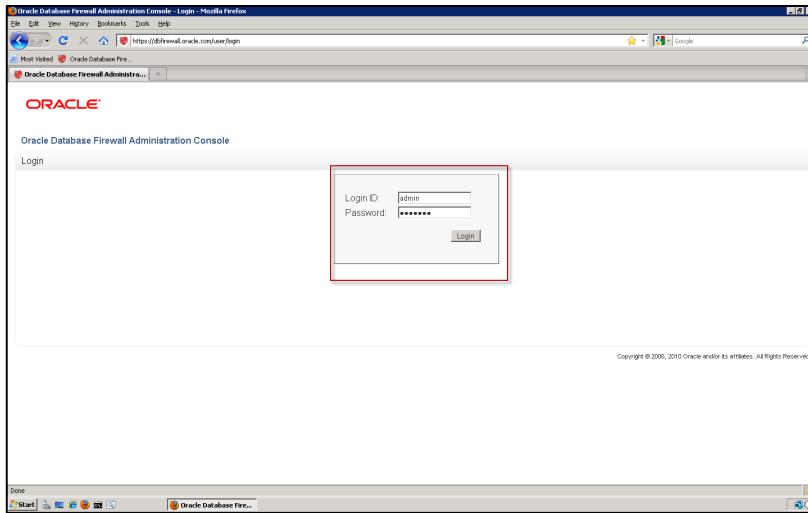
1. If you have not already connected to the Windows Desktop and started the Firefox browser, do so now.



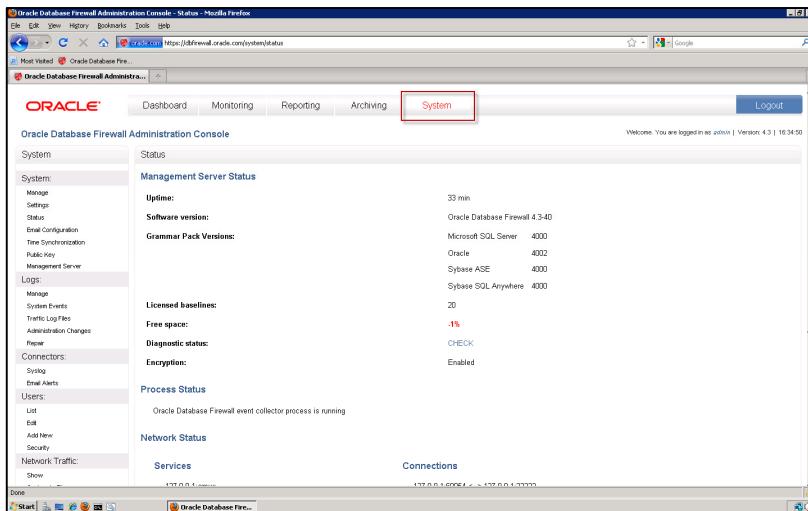
2. Once Firefox has started, click on the 'Oracle Database Firewall' link in the Favorites toolbar.

A screenshot of a Firefox browser window displaying the Oracle Database Security page. The URL in the address bar is http://www.oracle.com/us/products/database/security/index.html. The page content includes a sidebar with links to various Oracle products like Database, MySQL, and Oracle Database Firewall. The main content area features a section titled "Preventing Toxic Data Spills" with a sub-section "Oracle Database Security". It highlights Oracle's role as a "SECURITY INDUSTRY LEADER" and lists several security features: Forester, Data Masking, ZFS, Enforcing Database Separation of Duties, IDC, Cost-Effective Data Leak Prevention, Starts at the Source, Leader in third-party security standards publications, and Deploy with Confidence. A sidebar on the right contains links to Oracle's Technology Network, such as Oracle Database Vault, Audit Vault, and Secure Backup.

3. You will be taken to the Oracle Database Firewall (DBFW) Administration Console login page. Login with the default administration user. The username is '**admin**', and the password is '**tdsdbfw01**'. Click the **login** button after entering the credentials.



4. We will start by creating a new user to administer the Oracle Database Firewall (DBFW). If you are not already on the '**System**' tab, navigate to it.



- Click on the ‘List’ link in the ‘Users’ section. This will list all of the users allowed to access DBFW. You will see the ‘admin’ user that we have just authenticated in with.

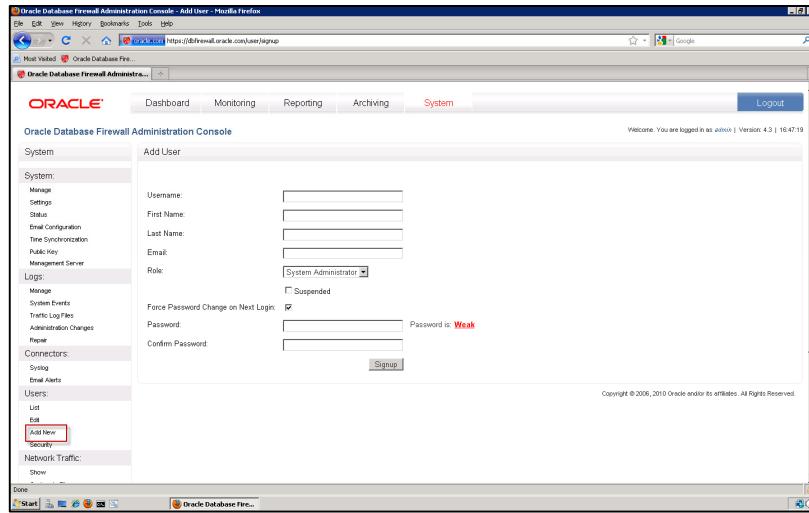
You will notice there are three different user roles. They are:

- View-only User:** Enables the user to view log data, change his/her password, and connect from the Analyzer.
- Log Administrator:** Enables a user of the Administration Console to view log data, change his/her password, configure logging, run archive or restore jobs, and connect from the Analyzer.
- System Administrator:** Gives the user full access to all options in the Administration Console, and to connect from the Analyzer.

Login	First name	Last name	Role	Created	Suspended
admin	Admin	Account	System Administrator	2010-09-09 13:07:08	no

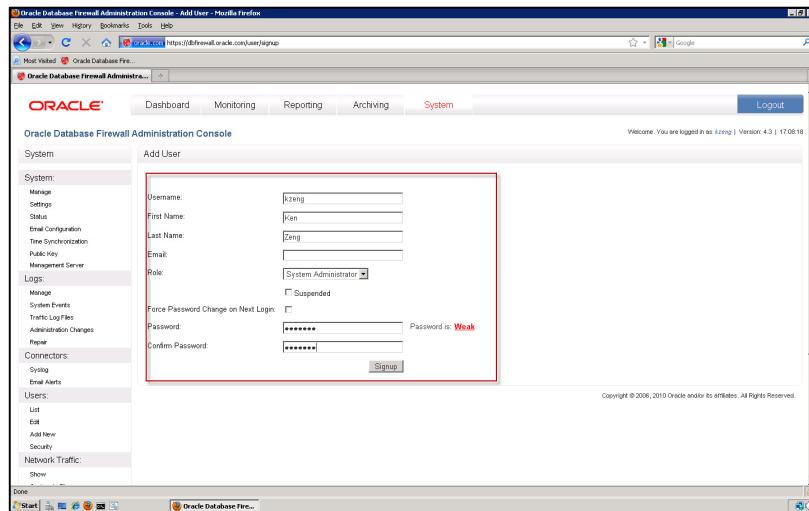
The screenshot shows the Oracle Database Firewall Administration Console interface. The main menu includes Dashboard, Monitoring, Reporting, Archiving, and System. Under the System category, the 'Logs' section is selected. On the left, a sidebar lists various management tasks: System (Manage, Settings, Status, Email Configuration, Time Synchronization, Public Key, Management Server), Logs (Manage, System Events, Traffic Log Files, Administration Changes, Repair), Connectors (Syslog, Email Alerts), and Users (List, Add New, Security, Network Traffic, Show). The 'List' link under 'Users' is highlighted with a red box. The top status bar indicates the user is logged in as 'admin' and the version is 4.3 | 16:47:08. The bottom status bar shows the operating system as 'Windows'.

Click on the 'Add New' link in the 'Users' section to create a new user.



Enter the following information:

Username:	kzeng
First Name:	Ken
Last Name:	Zeng
Email:	<<LEAVE BLANK>>
Role:	System Administrator
Force Password Change on Next Login:	Uncheck
Password:	oracle1



Once the data has been entered, click 'Signup' to create the user.

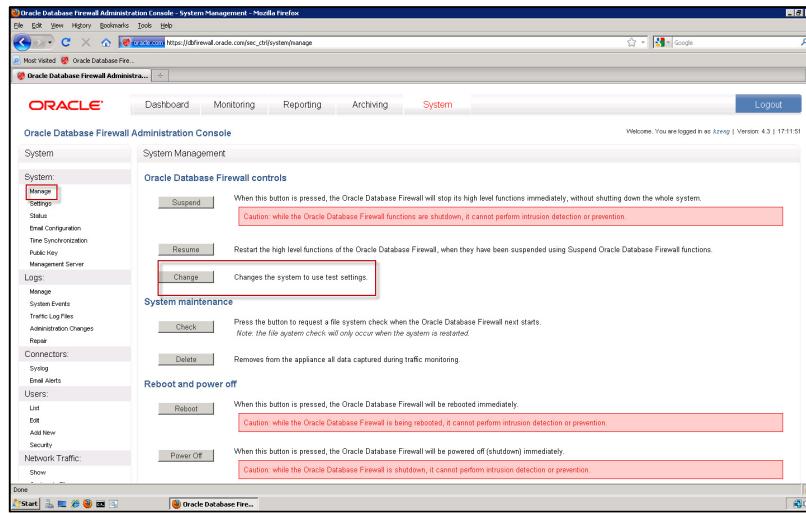
You will be taken back to the screen showing a list of all users in DBFW. Notice that there is now an entry for the 'kzeng' user we have created.

The screenshot shows the Oracle Database Firewall Administration Console interface. The main menu on the left includes System, Logs, Connectors, and Users. Under Users, options like List, Edit, Add New, and Security are available. The central area displays a table of users. A message box at the top right states 'User 'kzeng' successfully created'. The user 'kzeng' is listed in the table with a red border around it. The 'Logout' button in the top right corner is also highlighted with a red box.

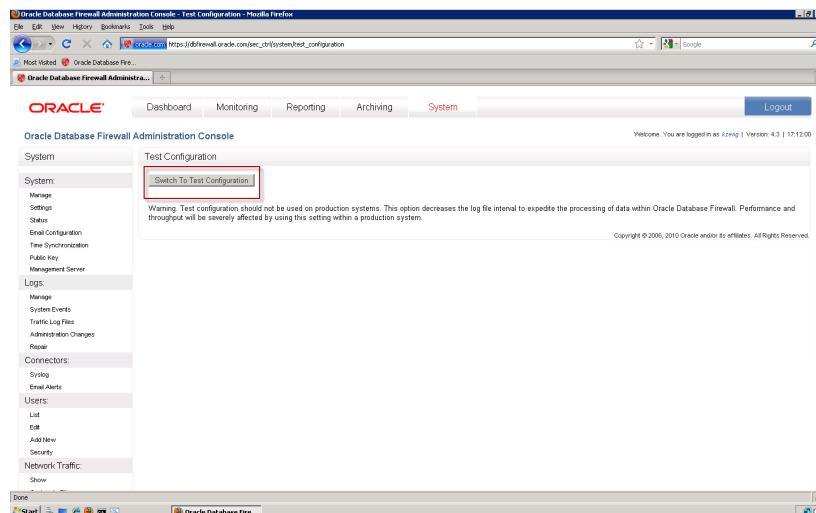
Click the 'Logout' button. We will now login with the 'kzeng' user. Enter 'kzeng' for the Login ID and 'oracle1' in the password field.

The screenshot shows the Oracle Database Firewall Administration Console login page. It features a simple form with 'Login ID:' and 'Password:' fields, each containing 'kzeng' and 'oracle1' respectively. A large red box surrounds the entire login form. Below the form, a note says 'Copyright © 2006, 2010 Oracle and/or its affiliates. All Rights Reserved.'

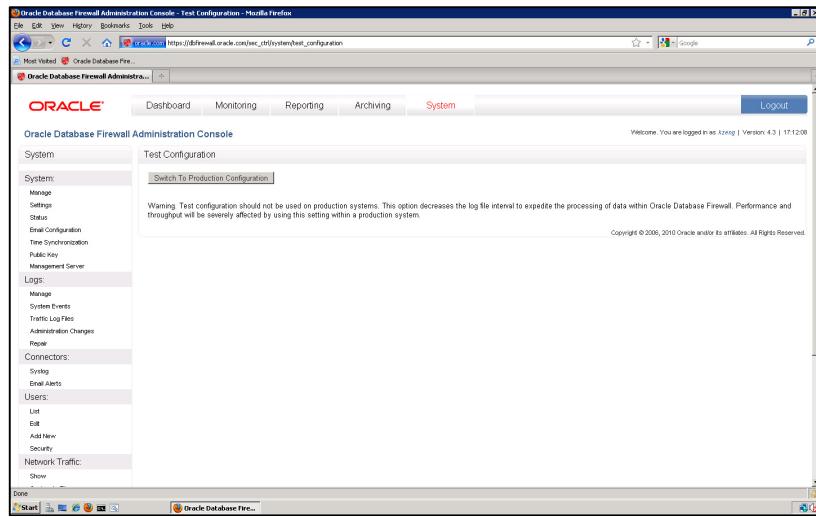
6. We will now change the DBFW system setting to '**test mode**'. This mode writes out log files and makes reports available at a quicker frequency—which is convenient for testing and our hands-on-labs, but not ideal or necessary in a production environment. Click on the '**System**' tab, then on the '**Manage**' link in the '**System**' section, as shown below. Once there, click the '**Change**' button in the '**Oracle Database Firewall controls**' section.



Click on the '**Switch the Test Configuration**' button to toggle the setting.



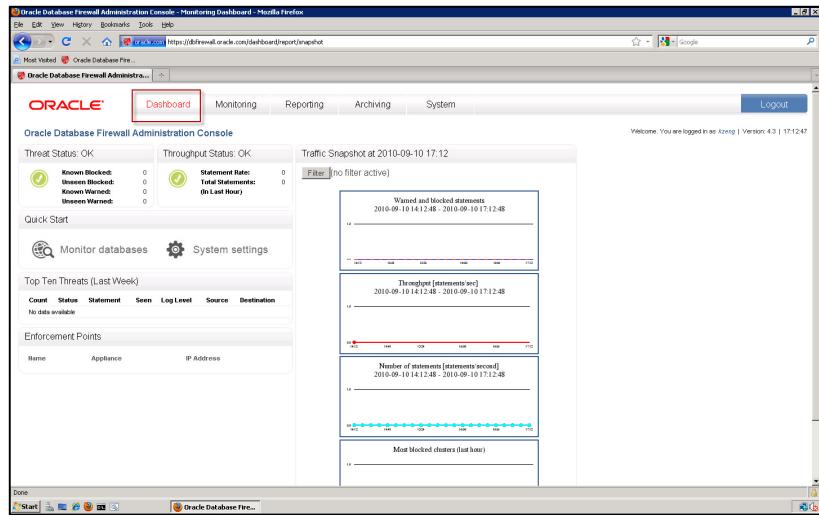
Once completed, you will see the screen changes to show you the '**Switch to Production Configuration**' button. You are now in the correct Test Configuration mode.



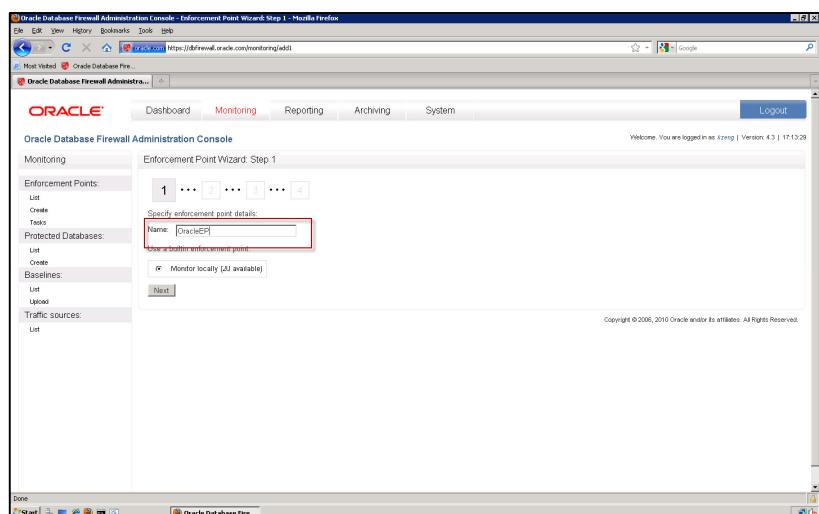
7. We will now create an Enforcement Point and a Protected Database. An Enforcement Point is the Oracle Database Firewall object that is responsible for monitoring and logging SQL statements passed to the database. Multiple enforcement points can be used to monitor traffic to different databases or at different locations in the network.

We will be monitoring an Oracle 11g set of databases hosted on a remote server. The DBFW server will monitor and block activity to the databases based on policy that we will create in the next lab. Again, you will need the Database IP Address provided by the instructor to create the Enforcement Point and Protected Database.

Click on the ‘Dashboard’ tab at the top of the page, once there, click on the ‘Monitor databases’ link in the middle of the page.

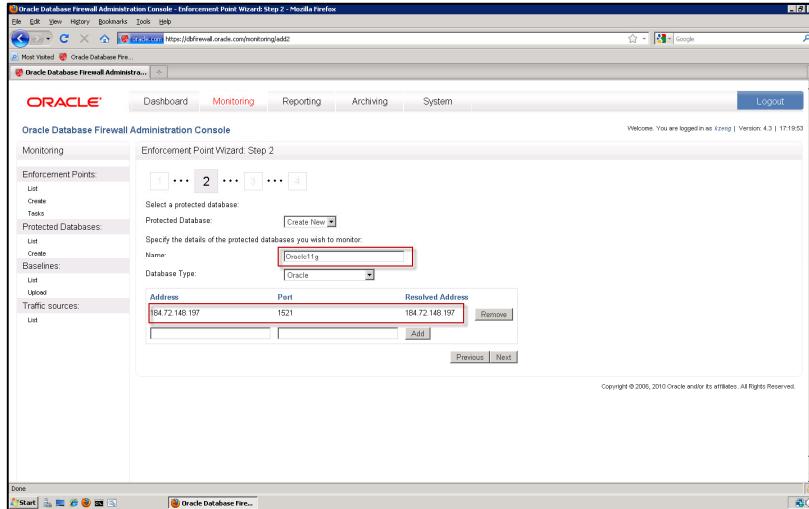


You will be taken into the Enforcement Point Setup Wizard. Type in ‘OracleEP’ for the name of the Enforcement Point.

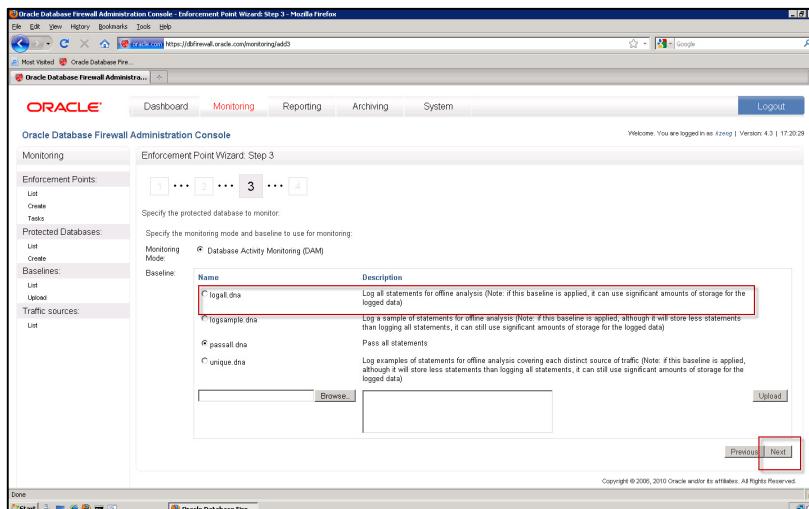


Enter the following information, then click ‘Next’:

Name:	Oracle11g
Database:	Oracle
Address:	<<IP of the DB Server>>
Port:	1522

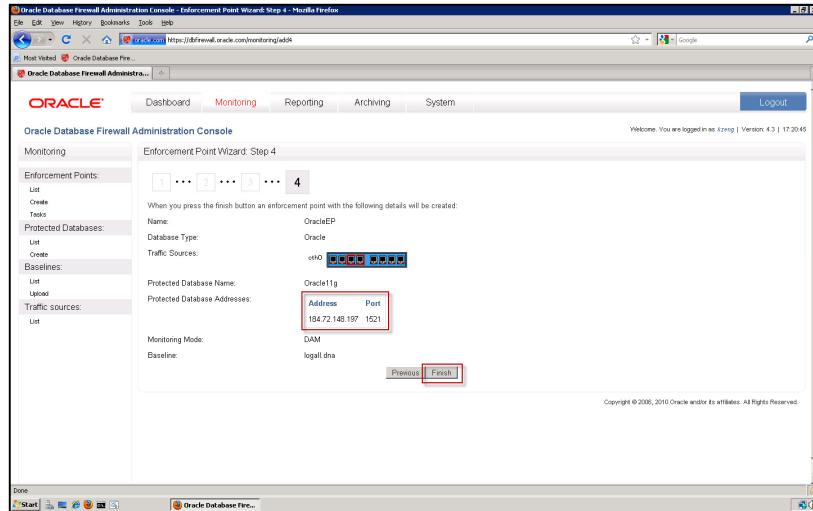


On the third step of the wizard select the ‘**logall.dna**’ Baseline radio button. This will configure the Enforcement Point to monitor everything. The default for the monitoring mode is set to ‘**Database Activity Monitoring (DAM)**’. This current/default mode will not allow us to block any SQL traffic. We will configure the Enforcement Point to block in a later step.

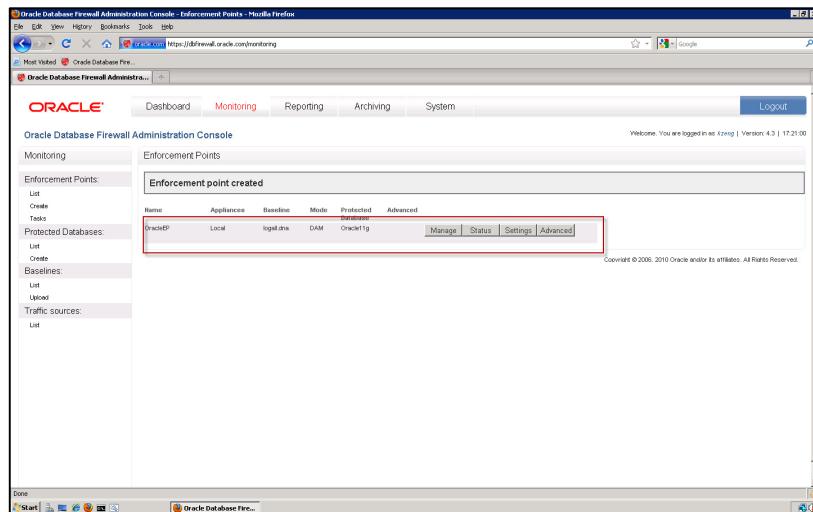


You will see a confirmation page that shows the configuration. Check that the Protected Database Address has the correct IP and port (as

specified by your instructor). Click ‘Finish’ to create the Enforcement Point.



We will configuration the Enforcement Point to block SQL traffic. In the Enforcement Point Summary page you will see the ‘OracleEP’ entry we just created. Click on the ‘Settings’ button for this Enforcement Point.



Check each of the ‘Activate Database Response Monitoring’ and the ‘Full error message annotation...’ check-boxes. This will enable DBFW to monitor the response to a given piece of SQL and monitor error messages. Also switch the radio button to the ‘Database Policy Enforcement (DPE)’. This will allow us to block SQL traffic. Click ‘Save’ once complete.

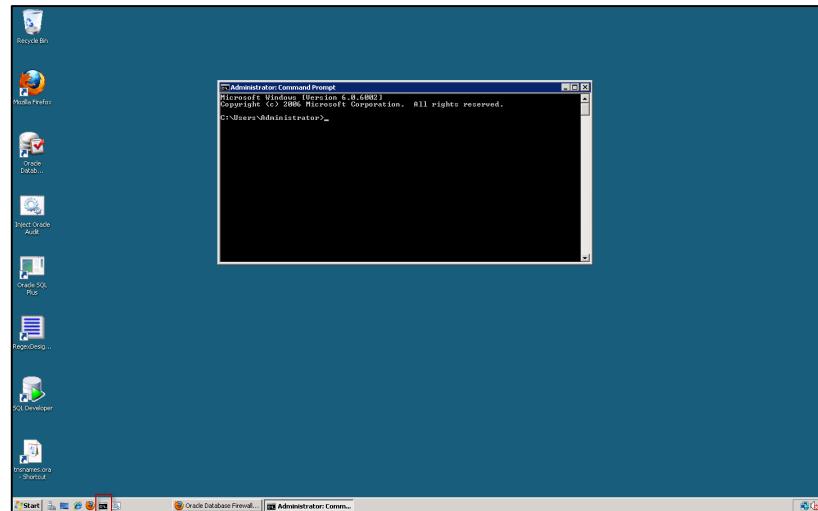
You will notice that the ‘OracleEP’ Enforcement Point is now configured to ‘DPE’ mode.

Name	Appliances	Baseline	Mode	Protected Database	Advanced
OracleEP	Local	logall.dba	DPE	Oracle11g	Manage Status Settings Advanced

Click on the '**Dashboard**' tab at the top of the screen. Notice that in the Enforcement Point summary section you will see the '**OracleEP**' entry.

The screenshot shows the Oracle Database Firewall Administration Console interface. At the top, there are tabs for Dashboard, Monitoring, Reporting, Archiving, and System. The Dashboard tab is selected. On the left, there are sections for Threat Status (OK), Throughput Status (OK), and Enforcement Points (listing OracleEP). The main area displays four traffic snapshots: 'Blocked and Blocked statements' (2010-09-10 14:12:48 - 2010-09-10 17:12:48), 'Throughput [statements/sec]' (2010-09-10 14:12:48 - 2010-09-10 17:12:48), 'Number of statements [statements/second]' (2010-09-10 14:12:48 - 2010-09-10 17:12:48), and 'Most blocked queries (last hour)'. The bottom of the screen shows the Windows taskbar with the Oracle Database Firewall icon.

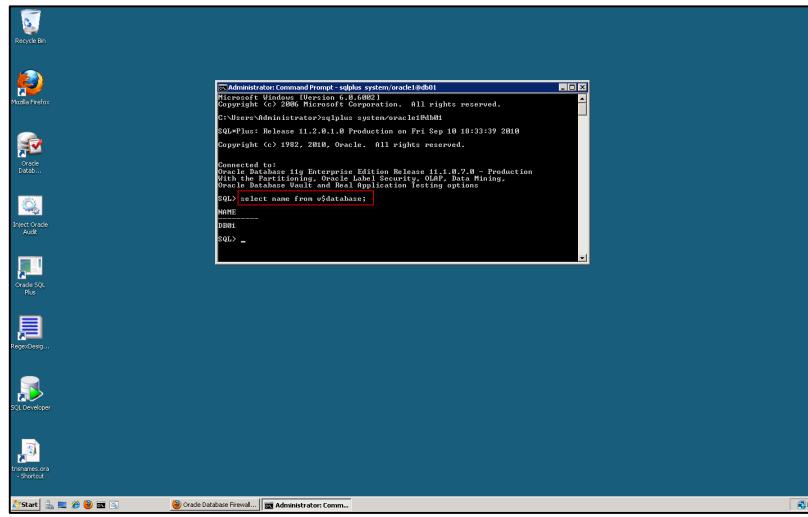
8. We will now test that DBFW is monitoring traffic to the Oracle 11g Database. Open a command prompt window by clicking on the icon at the bottom of the screen, as shown below. You can also go to the Start Button → Run... → and type in 'cmd'.



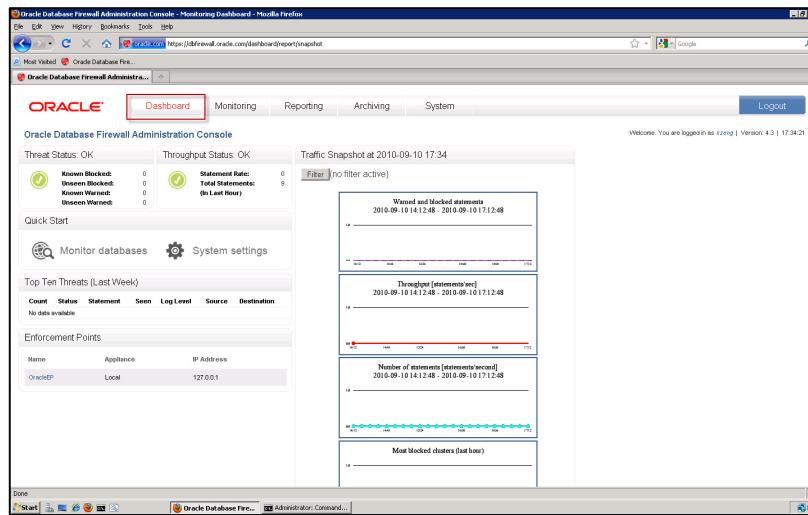
We will connect to the Oracle 11g database called **DB06**. Enter the following:

```
> sqlplus system/oracle1@db06
> select name from v$database;
```

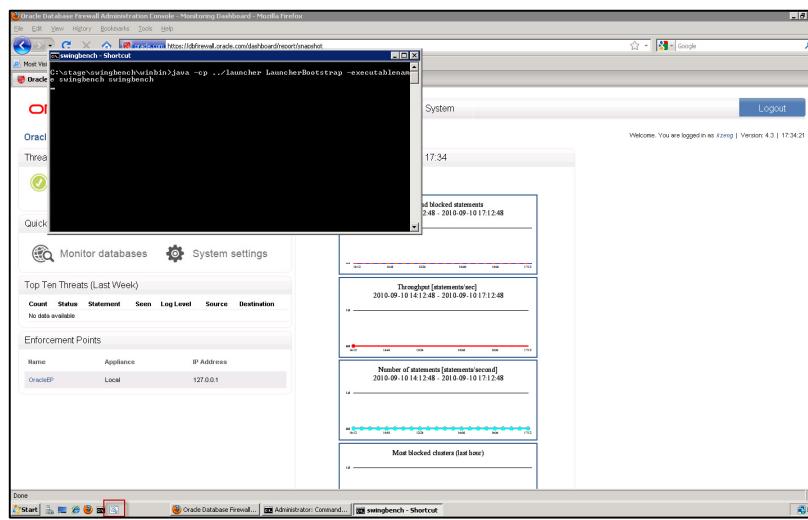
This will connect you to the database and confirm that we are able to complete SQL statements. You will see that the SQL returns 'DB06'.



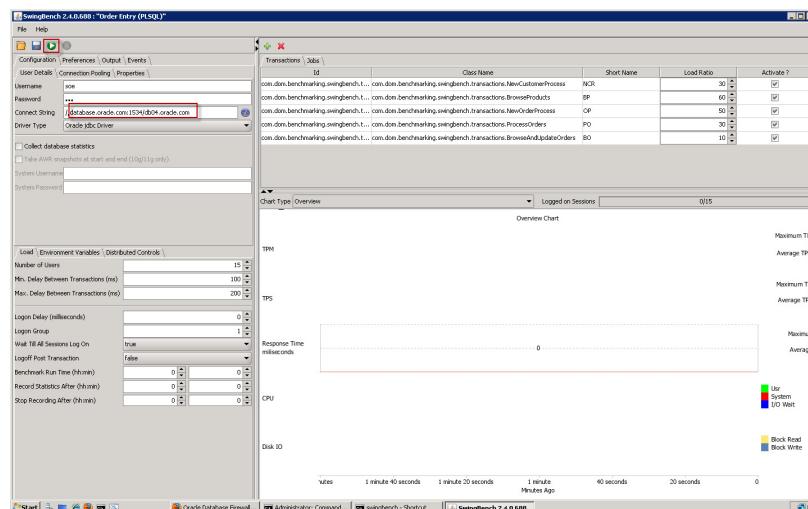
9. Return to the Database Firewall Administration Console. Re-open Firefox if necessary and navigate to the '**Oracle Database Firewall**' and login with 'kzeng'. You will notice that there are now some '**Total Statements**' in the '**Throughput Status: OK**' section.



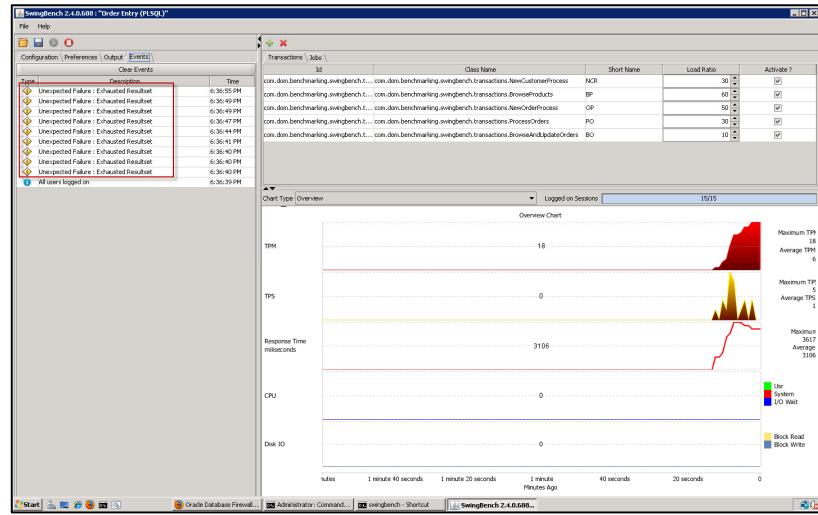
10. We will now simulate some application traffic for the DBFW to train what is considered to be normal and authorized traffic for our White List policy. Again, White List policies are simply the set of approved SQL commands that the firewall expects to see. For this simulation we will be using a tool called Swingbench. It runs SQL statements that simulate an order entry application. It performs different DML statement using SQL bind variables to query and change customer and order information. Click on the '**Swingbench**' icon at the bottom of the screen to start the tool, as shown below.



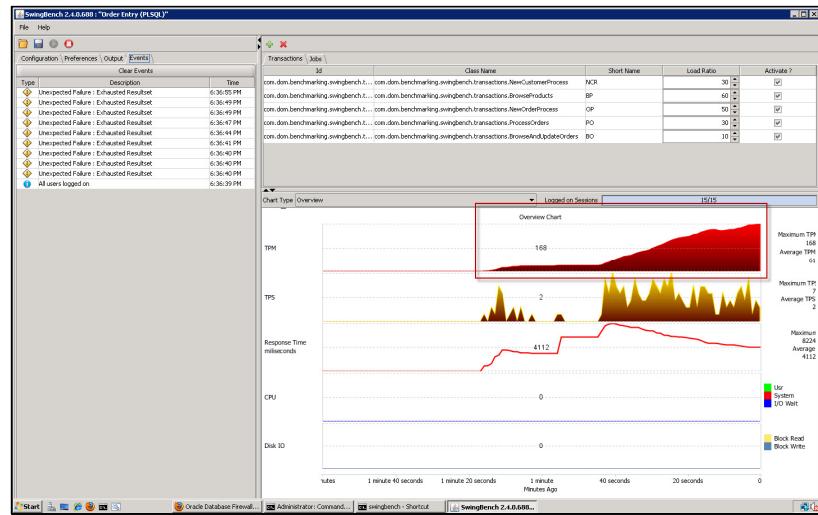
Swingbench is java application that will open after a few moments. Notice that the '**Connection String**' is pre-configured to connect to our Oracle 11g Database DB06. Click on the '**Start**' green arrow icon, as shown below. This will start the simulated application load. We will run roughly 10 minutes of application traffic.



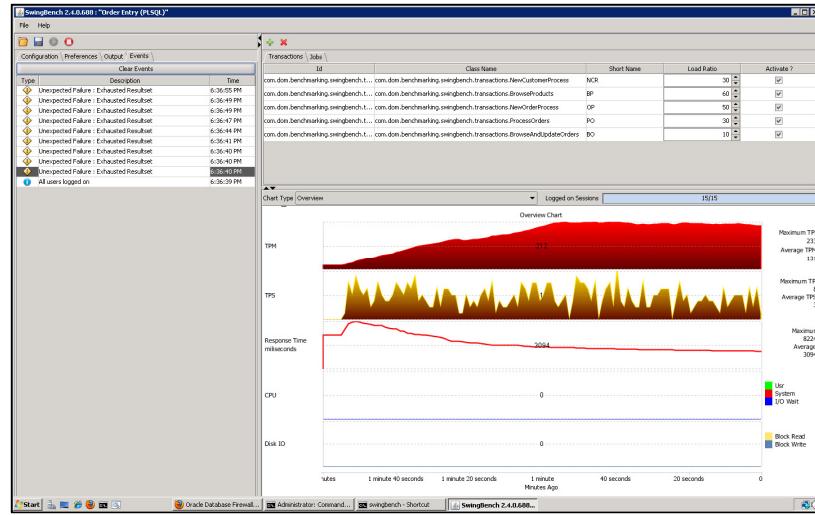
Navigate to the ‘Events’ tab in Swingbench. Ignore any ‘Exhausted Resultset’ messages as shown below. This has no affect on what we are doing.



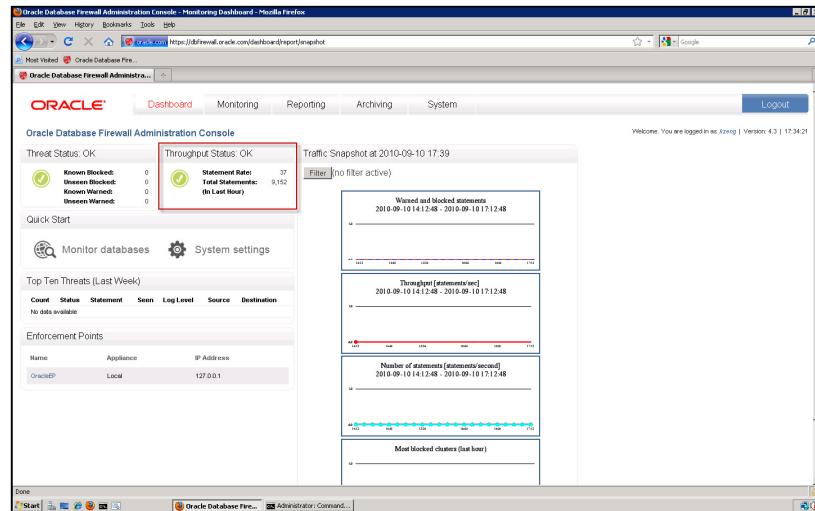
After a few moments you will see the transactions per minute (TPM) graph show an increase in transactions.



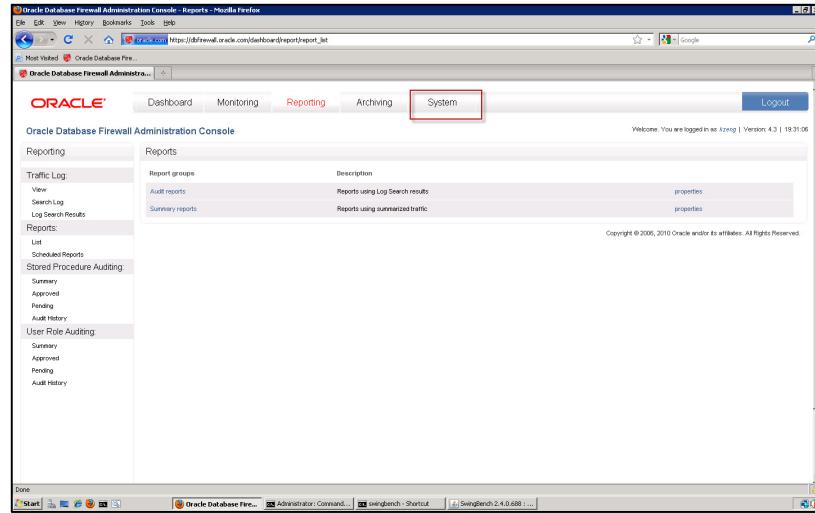
Leave the Swingbench application running. We will switch to the DBFW and monitor the traffic generated by the application.



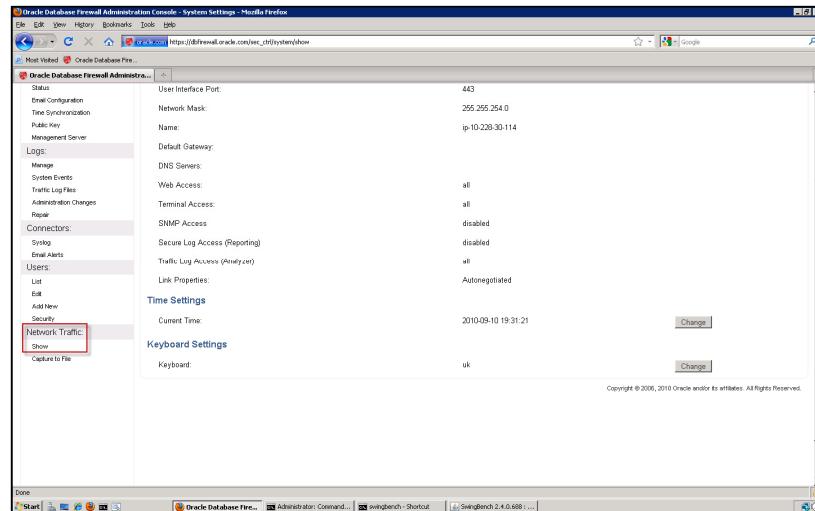
11. Return to the DBFW Administration Console. If you are not already on the 'Dashboard' tab, click on it. Notice that there are a significantly increased 'Total Statements' being monitored.



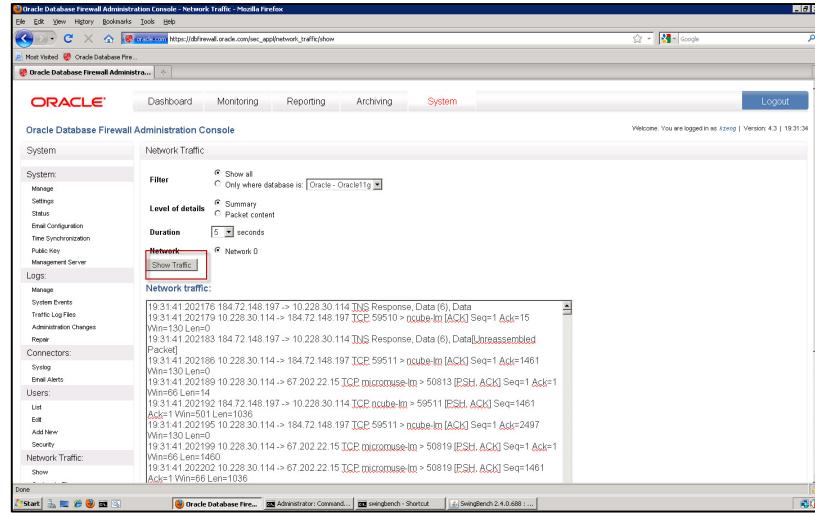
We will confirm that DBFW is monitoring traffic using a few different techniques. Click on the 'System' tab highlighted.



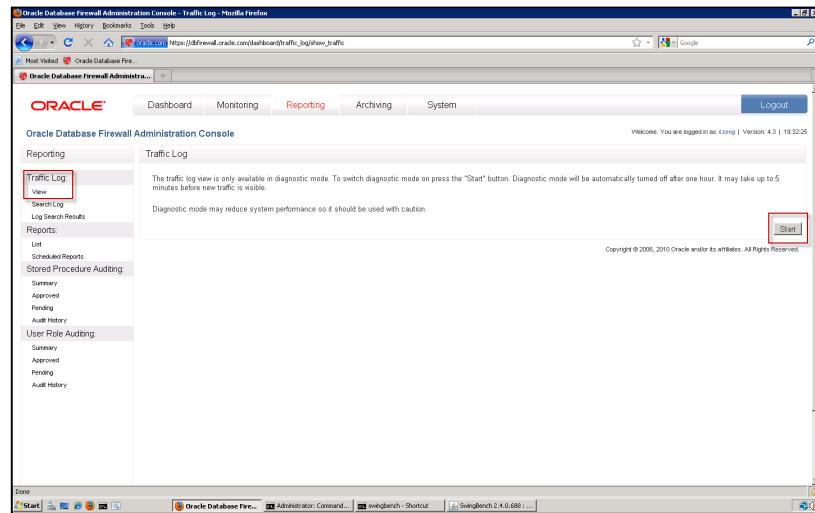
Click on the 'Show' link under the 'Network Traffic' section. This will allow us to view a snapshot of the raw network traffic that is being processed by DBFW.



Click on the '**Show Traffic**' button, ensuring that the '**Filter**' is set to '**Show All**'. You should see lots of information in the network traffic window. You can use this screen to confirm that DBFW is configured properly to collect raw network traffic to needed to process.



12. You can also view the DBFW logs to confirm that SQL traffic is being monitored. From time to time, you may want to use the Traffic Log menu in the Reporting page to recover data from the traffic log for auditing purposes, forensic analysis, or to investigate possible attempted attacks. The traffic log stores details of all logged SQL statements. Navigate to the '**Reporting**' tab. Once there, click on the '**View**' link under the '**Traffic Log**' section on the left hand navigation menu. Click on the '**Start**' button to view the logs that DBFW is creating.



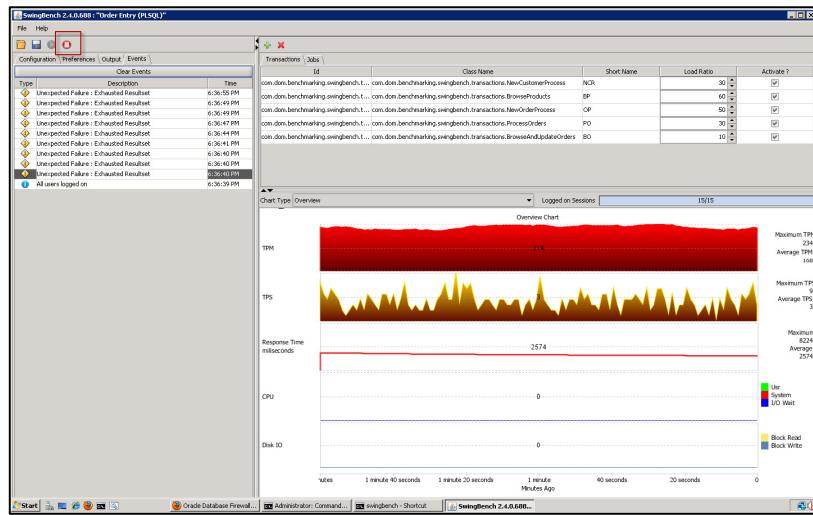
After a few moments you should see numerous statements that have been captured. If you do not see this information immediately, please be patient and allow a few minutes for DBFW to process the logs.

Time	Type	Statement	Logging Code	User	Action Code	Threat Severity
2010-09-10 19:32:49	data manipulation read only	select p.product_id,product_name,pr...	always	SOE	pass	unassigned
2010-09-10 19:32:49	data manipulation read only	select c.customer_id,cust_fn,ln...	always	SOE	pass	unassigned
2010-09-10 19:32:49	data manipulation read only	SELECT /*+ use_nl */o.order_id,...	always	SOE	pass	unassigned
2010-09-10 19:32:49	data manipulation read only	select orders_total,nettotal from dual	always	SOE	pass	unassigned
2010-09-10 19:32:49	data manipulation	insert into logon (value,0,0)	always	SOE	pass	unassigned
2010-09-10 19:32:49	data manipulation	update inventories set quantity_on_han...	always	SOE	pass	unassigned
2010-09-10 19:32:49	data manipulation read only	select customer_id,cust_fn,ln...	always	SOE	pass	unassigned
2010-09-10 19:32:49	data manipulation	insert into order_(ORDER_ID, ORDER_DAT...	always	SOE	pass	unassigned
2010-09-10 19:32:49	data manipulation read only	select customer_id,cust_fn,ln...	always	SOE	pass	unassigned
2010-09-10 19:32:49	data manipulation	update orders set order_mode = 0, or...	always	SOE	pass	unassigned
2010-09-10 19:32:49	data manipulation	update inventories set quantity_on_han...	always	SOE	pass	unassigned
2010-09-10 19:32:49	data manipulation read only	select product_id,product_name,pr...	always	SOE	pass	unassigned
2010-09-10 19:32:49	data manipulation read only	select product_id,product_name,pr...	always	SOE	pass	unassigned
2010-09-10 19:32:49	data manipulation read only	select product_id,product_name,pr...	always	SOE	pass	unassigned
2010-09-10 19:32:49	data manipulation read only	select product_id,product_name,pr...	always	SOE	pass	unassigned

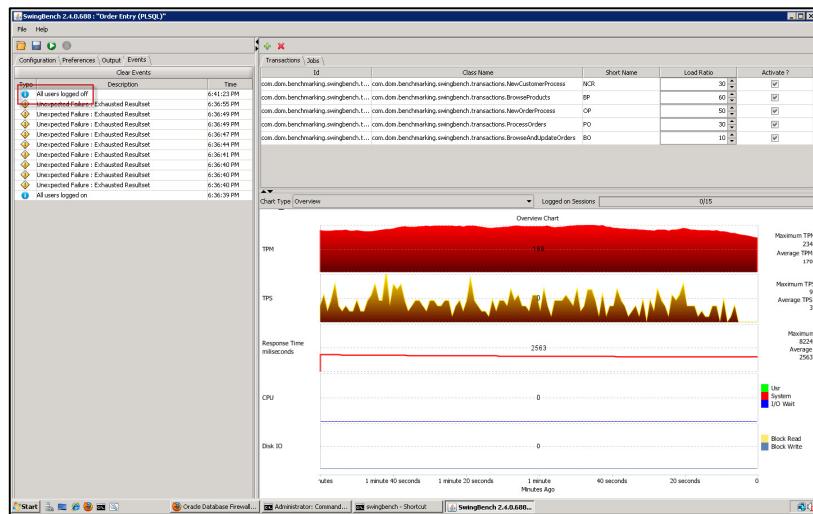
Please feel free to navigate through the logs to review their contents. Click on one of the available report page links (e.g. page 6 shown below) to see more data.

Time	Type	Statement	Logging Code	User	Action Code	Threat Severity
2010-09-10 19:32:46	data manipulation	insert into order_(ORDER_ID, ORDER_DAT...	always	SOE	pass	unassigned
2010-09-10 19:32:46	data manipulation read only	select customer_id,cust_fn,ln...	always	SOE	pass	unassigned
2010-09-10 19:32:46	data manipulation read only	select customer_id,cust_fn,ln...	always	SOE	pass	unassigned
2010-09-10 19:32:46	data manipulation	insert into logon (value,0,0)	always	SOE	pass	unassigned
2010-09-10 19:32:46	data manipulation read only	select p.product_id,product_name,pr...	always	SOE	pass	unassigned
2010-09-10 19:32:46	data manipulation	update inventories set quantity_on_han...	always	SOE	pass	unassigned
2010-09-10 19:32:46	data manipulation	update inventories set quantity_on_han...	always	SOE	pass	unassigned
2010-09-10 19:32:46	data manipulation read only	select product_id,product_name,pr...	always	SOE	pass	unassigned
2010-09-10 19:32:46	data manipulation	select /*+ first_nest */p.product_id...	always	SOE	pass	unassigned
2010-09-10 19:32:46	data manipulation	insert into order_(ORDER_ID, ORDER_DAT...	always	SOE	pass	unassigned
2010-09-10 19:32:46	data manipulation read only	select product_id,product_name,pr...	always	SOE	pass	unassigned
2010-09-10 19:32:46	data manipulation	insert into logon (value,0,0)	always	SOE	pass	unassigned
2010-09-10 19:32:46	data manipulation	insert into order_(ORDER_ID, ORDER_DAT...	always	SOE	pass	unassigned
2010-09-10 19:32:46	data manipulation read only	select p.product_id,product_name,pr...	always	SOE	pass	unassigned

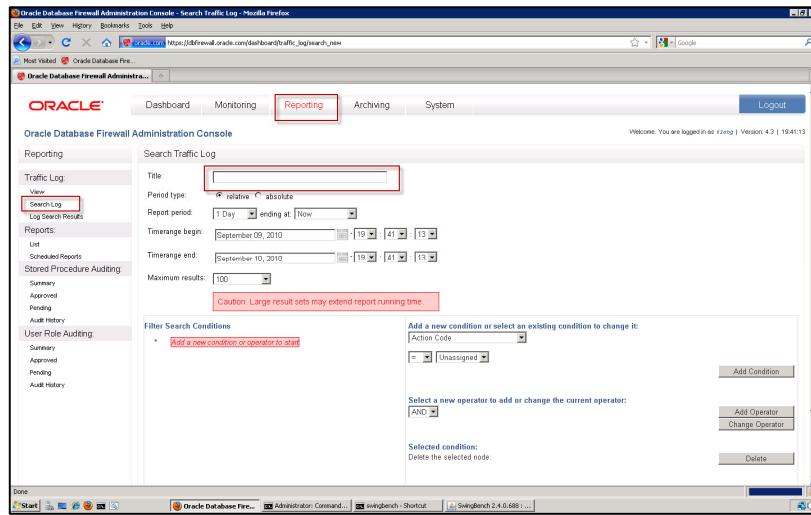
13. After you have seen network traffic and some log data (and roughly 10 minutes has passed) please stop the Swingbench application. Click on the red stop icon at the top of the application, as shown below



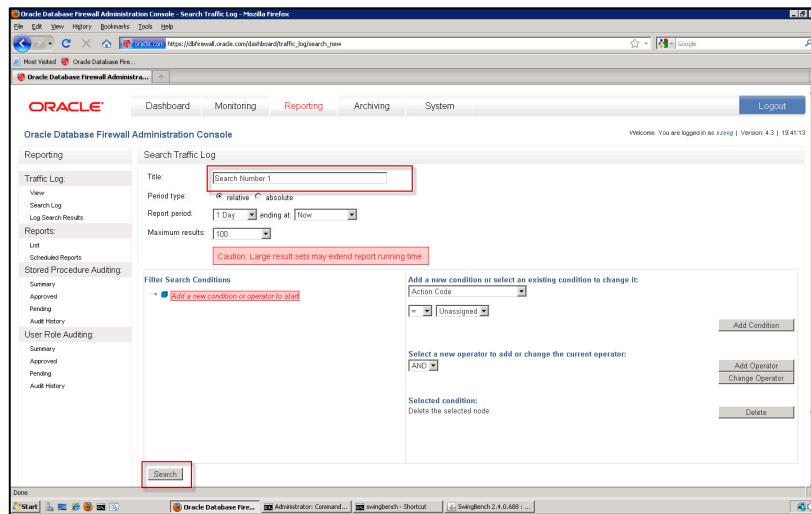
After a few moments you will see that all users will be logged off, check the ‘Events’ tab to confirm that there is no more traffic. Then close Swingbench.



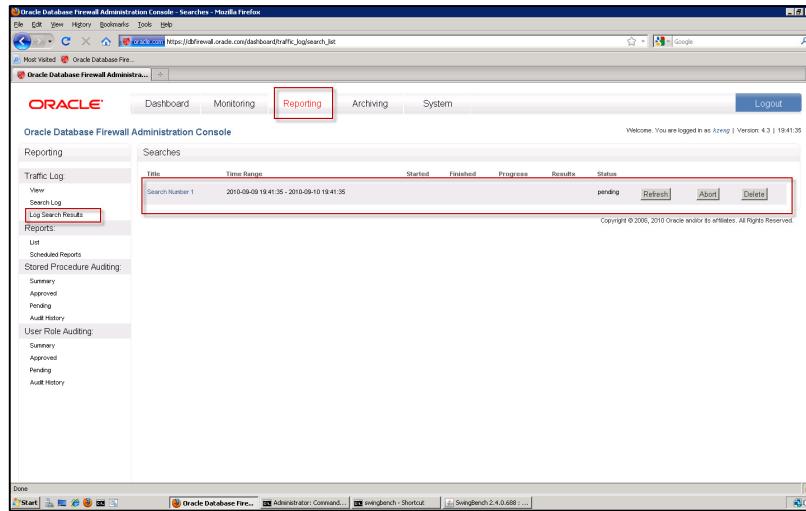
14. Finally, we will create a new log search to review all of the traffic that has been captured. Navigate to the ‘**Reporting**’ tab then click on ‘**Search Log**’ in the ‘**Traffic Log**’ section.



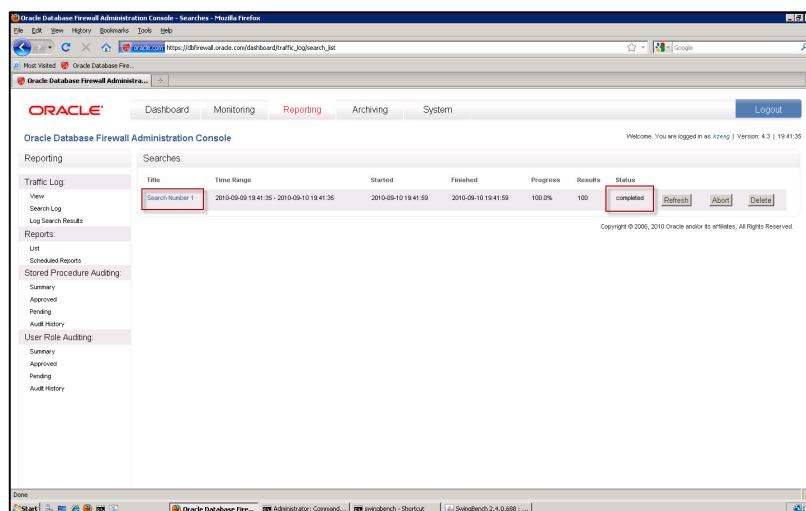
Enter ‘**Search Number 1**’ in the ‘**Title**’ field. Then click on the ‘**Search**’ button. This will search everything for the past one day, showing us 100 results.



You will be taken to the ‘**Log Search Results**’ page and show you the report being generated. It will show a status of ‘**pending**’ until the report is completed and ready for you to access. The page will automatically refresh, you do not need to refresh the web page.



Once the ‘**Status**’ is ‘**completed**’, click on the name link, ‘**Search Number 1**’ under report ‘**Title**’ section.



You will see all of the SQL statements that have been monitored by DBFW. These are the normal statements that we expect from our application. Notice that, by default, we are not blocking anything.

The screenshot shows the Oracle Database Firewall Administration Console interface. The main area displays a table of 'Search Results' under the 'Reporting' tab. The table has columns: Time, Type, Origin, Description, DB Client, DB User, Action Code, and Threat Severity. The 'Description' column contains various SQL statements, such as 'CONNECTED,LOGIN', 'SELECT USER FROM DUAL', and 'BEGIN DEMS_OUTPUT_DISABLE; END;'. Most entries have a green checkmark in the 'Action Code' column and are labeled 'no alert' in the 'Threat Severity' column. One entry at the bottom, 'CONNECTED,LOGOUT', has a red exclamation mark in the 'Action Code' column and is labeled 'no alert' in the 'Threat Severity' column.

Click on any one of the '**Descriptions**' to review more details about any of the specific statements.

This screenshot shows the same Oracle Database Firewall Administration Console interface, but it has zoomed into a specific row in the 'Search Results' table. The row corresponds to the 'CONNECTED,LOGOUT' statement from the previous screenshot. The expanded view includes additional details: 'Name' (product.product_id), 'Origin' (product), 'Value' (product.product_name), 'DB Client' (SOE), 'DB User' (pass), 'Action Code' (no alert), and 'Threat Severity' (unsigned). Below this, the 'Transaction Status' section shows the SQL request: 'select product_id,product_name,product_description,category_id,weight_class,supplier_id,product_status list_price,min_price,catalog_url from product_information where category_id = 0'. It also shows 'Response Status' (statement.success), 'Response Code' (0), and 'Record Type' (statement). The 'Performance' section provides execution times: 'Request Time' (2010-09-10 17:36:42.700), 'Response Time' (2010-09-10 17:36:43.523), and 'Transaction Time' (0.730). The 'Context' section lists traffic sources and client programs: 'Traffic Source' (network), 'DB User Name' (SOE), 'DB User Name Origin' (network), 'DB User Name (new)' (SOE), 'DB Client Program Name' (JDBC Thin Client), and 'DB Client Program Name Origin' (network).

You will see that we are capturing a lot of information about the SQL statements.

The screenshot shows the Oracle Database Firewall Administration Console interface. The main area displays a table of network and database connection details. Below this, there are sections for 'Database Firewall Action' and 'Database Firewall Analysis'. At the bottom, two specific SQL requests are listed with their execution details.

Action Code	pass
Threat Severity	unassigned
Log Cause	novelty
Logging Level	always

Cluster Type	data manipulation read only
Cluster ID	387422950
Protected Database	Oracle11g
Source Name	OracleREP
Baseline	oracle-logdata
Grammar Pack Version	4002
Failure Count	0
SQL Request ID	4C3A6CAAD2B00002

Statement	select product_id,product_name,prod...	07.202.22.15:49870	SOE	pass	unassigned
Statement	SELECT * FROM order_id_line	07.202.22.15:49873	SOE	pass	unassigned

D. Summary

You accomplished the following in this lab exercise:

1. Created and setup an enforcement point in Oracle Database Firewall
2. Generated simulated database activity on expected authorized traffic
3. Monitored Oracle Database traffic in the Database Firewall Management console
4. Prepared to train the Database Firewall with acceptable and expected behavior

LAB EXERCISE 02 – ORACLE DATABASE FIREWALL – USE THE TRAFFIC ANALYZER TO CONFIGURE POLICIES AND BLOCK UNAUTHORIZED TRAFFIC

INTRODUCTION

White list, Black list, Exception list policies

Oracle Database Firewall examines the grammar of the SQL statements being sent to the database, analyzes their meaning, and determines the appropriate security policy to apply. This highly accurate approach provides a significantly higher degree of protection than first-generation database monitoring technologies that relied on recognizing the "signature" of known security threats. By enforcing normal application behaviour, Oracle Database Firewall helps organizations avoid the costly and disruptive false positives and false negatives common with other approaches. Oracle Database Firewall recognizes SQL injection attacks on compromised applications and blocks them before they reach the database.

Iterative Development Cycle of the Baseline

Oracle Database Firewall Analyzer enables you to design baselines efficiently in minimum time. Successful deployment of a Database Firewall system depends on an effective baseline. The process of developing a baseline involves an iterative process that keeps refining and improving the baseline.

The cycle is as follows:

1. Oracle Database Firewall collects and logs SQL statements from clients in Training Mode.
2. The Database Firewall then analyzes and tests the logged SQL statements against the statements used to build the current baseline. This creates a better understanding of how the database is used and an awareness of areas where the baseline needs to be improved or application programming changed.
3. The Analyzer allocates the new statements to the appropriate clusters, and when necessary, creates new clusters. If new clusters have been created, action and logging levels for these should be assigned, either automatically or manually.
4. Once you have made modifications, you can deploy the baseline in the typical way.

A. Lab Scenarios and Objectives

In this lab exercise, you will accomplish the following:

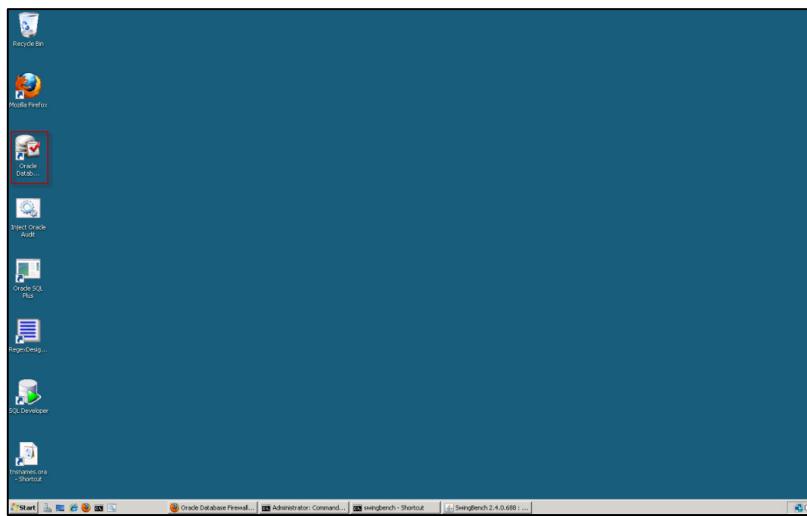
1. *Complete an iterative development cycle of the baseline*
2. *Use the Oracle Database Firewall (DBFW) Analyzer to analyze and train on traffic logs.*
3. *Develop and deploy a baseline policy*
4. *Modify and re-deploy the baseline policy*
5. *Verify that policy is enforced and ensure that unseen traffic is blocked*

B. Setup and Preparation

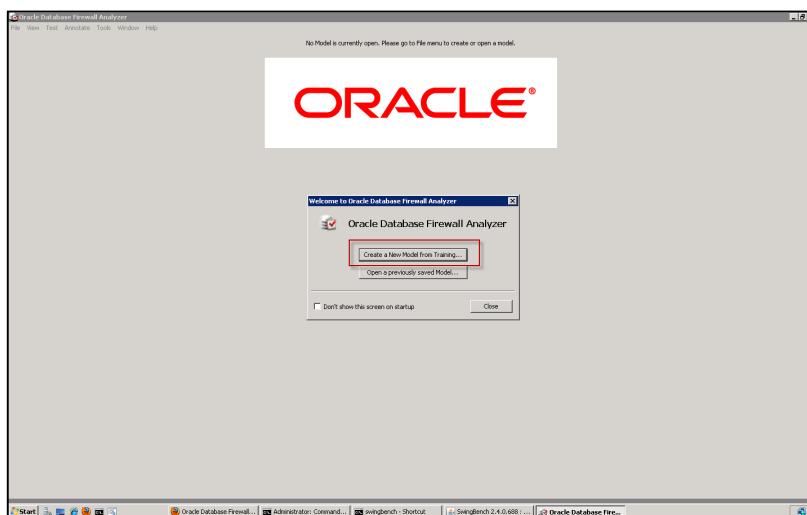
- Completion of LAB EXERCISE 01 – ORACLE DATABASE FIREWALL ENFORCEMENT POINTS TO MONITOR AND PROTECT DATABASES

C. USE THE TRAFFIC ANALYZER TO CONFIGURE POLICIES AND BLOCK UNAUTHORIZED TRAFFIC

1. In order to create a baseline policy that DBFW uses to block SQL statements will use the Oracle DBFW Analyzer product. Start the DBFW Analyzer by double-clicking it's icon on the desktop, as shown below:

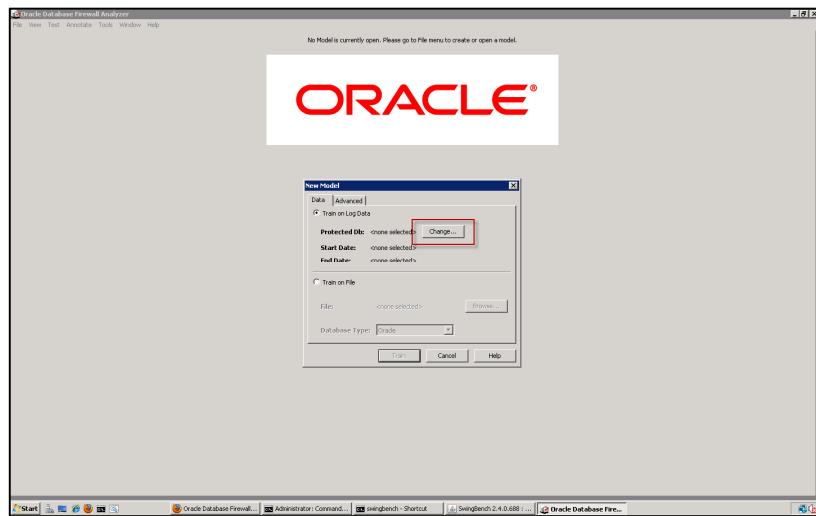


The Oracle Database Firewall system must understand the normal way that client applications use the database. This is accomplished by supplying logged data to the Analyzer before you start developing a new baseline. We will create a new model based on the simulated application traffic we just monitored. Click on the '**Create a New Model from Training**' button.

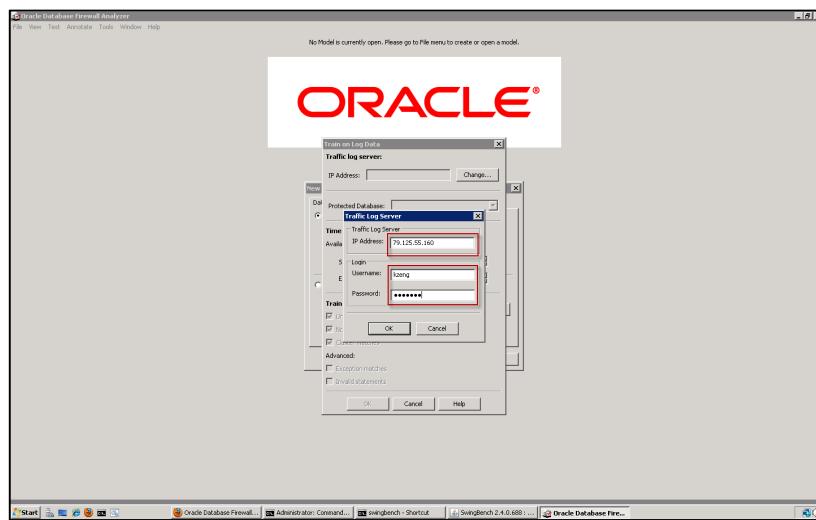


Click on the '**Change**' button in the '**Train on Log Data**' section. This will allow us to login to DBFW.

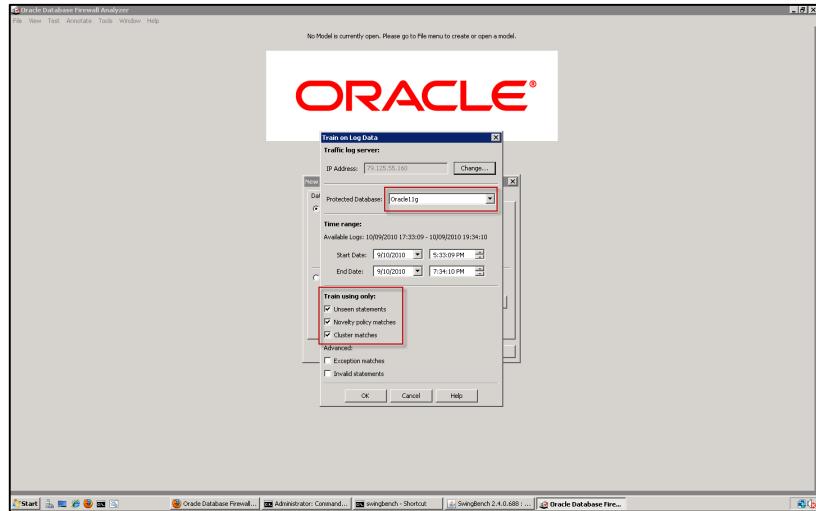
Training Mode logs SQL traffic specifically for developing a new baseline. The logged data enables the Analyzer to understand how client applications use the database and enables rapid development of a baseline that reflects actual use of the database and its client applications.



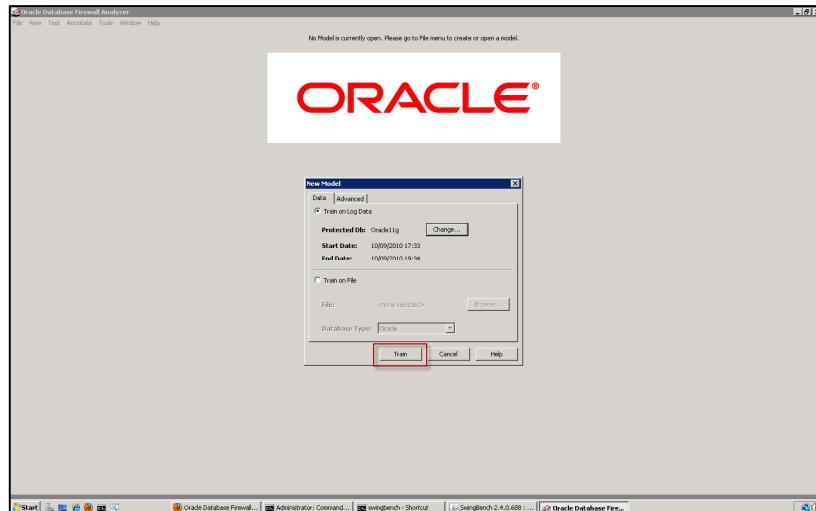
Enter the IP address of the DBFW server as provided by your instructor. Enter the username of 'kzeng' and password is 'oracle1'. Click 'OK'.



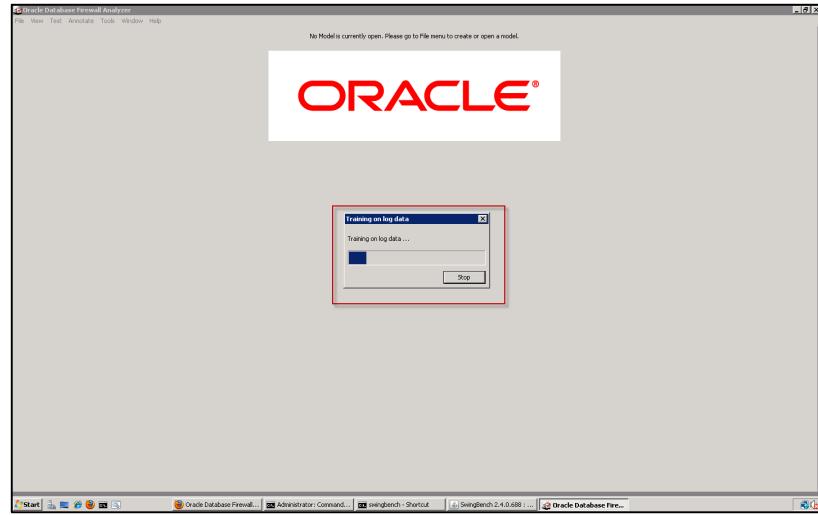
You will then see the Protected Databases listed and the timeframe with which you wish to create your new model. We only have one protected database configuration, '**Oracle11g**', that you created earlier. Ensure that all of the Training Options are set, as shown below. Click '**OK**' to start creating the model.



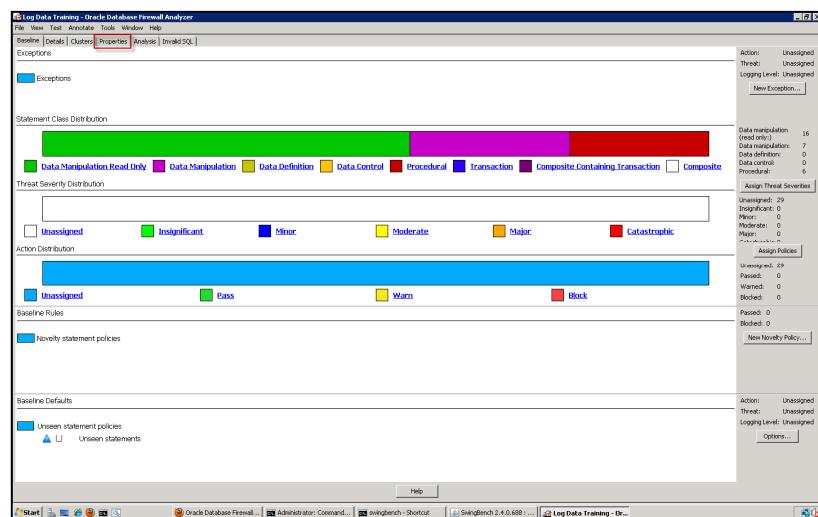
Click on '**Train**'. This will start the analysis of the monitored traffic captured by DBFW.



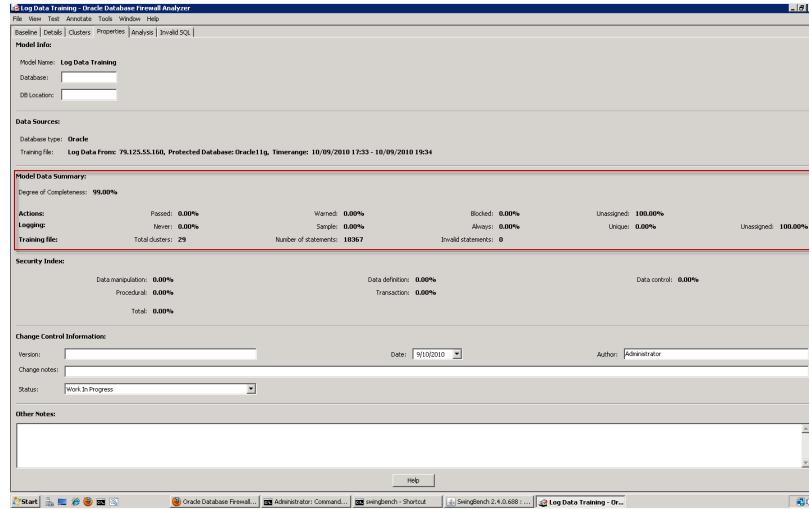
The time taken to train depends on the size of the captured monitored data and the resources available to the DBFW Analyzer.



2. The baseline tab will be shown initially. This shows you a summary of the model that we are creating. Click on the 'Properties' tab.



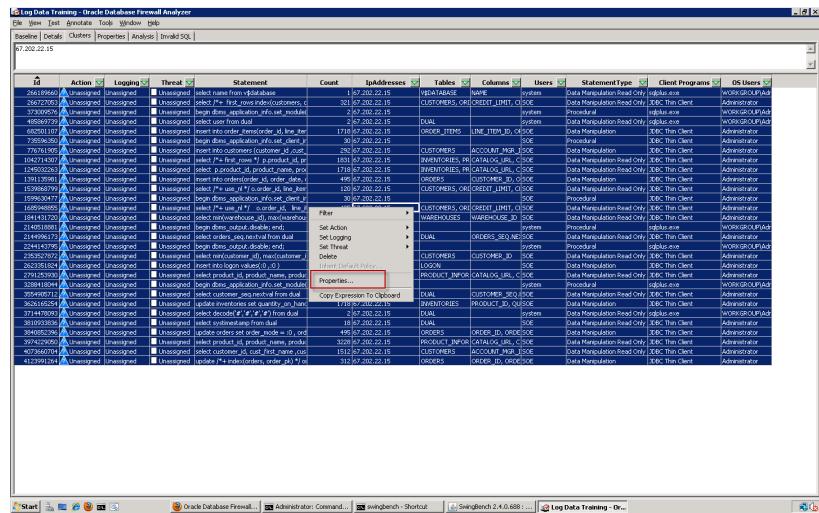
This will show a summary of the information we're using to create a model.



Click on the '**Clusters**' tab. Using its built-in knowledge of the SQL language, the Analyzer automatically groups the SQL statements into sets of semantically similar statements known as clusters. Clustering simplifies baseline construction and helps the operator analyze the data to understand how client applications use the database. Clusters allow administrators to create policies with a high degree of confidence that they have control of the environment.

#	Action	Logon	Thread	SQL	Statement	Count	IP Addresses	Tables	Columns	Users	Statement Type	SQL Client Programs	OS Users
264199440	Unassigned	Unlogged		Unassigned	select name from v\$database	1	67.202.22.15	VB\$DATABASE	NAME	system	Data Manipulation Read Only	sqlplus.exe	WORKGROUP\AD
266727205	Unassigned	Unlogged		Unassigned	select /*+ first_rows index(customers, c */	321	67.202.22.15	CUSTOMERS	ORACLE_CREDIT_LIMIT	0.50E	Data Manipulation Read Only	JDBC-The Client	Administrator
373905954	Unassigned	Unlogged		Unassigned	begin dbms_application_info.set_module	2	67.202.22.15	system			Procedural	sqlplus.exe	WORKGROUP\AD
425997000	Unassigned	Unlogged		Unassigned	begin dbms_application_info.set_module	1	67.202.22.15	DBMS_APPLICATION_INFO	MODULE_NAME	0.50E	Data Manipulation Read Only	sqlplus.exe	WORKGROUP\AD
658201101	Unassigned	Unlogged		Unassigned	insert into order_header(order_id, line_item	1710	67.202.22.15	ORDER_ITEMS	LINE_ITM_ID	0.50E	Data Manipulation	JDBC-The Client	Administrator
739596395	Unassigned	Unlogged		Unassigned	begin dbms_application_info.set_client_id	251	67.202.22.15	DBMS_APPLICATION_INFO	CLIENT_ID	0.50E	Procedural	sqlplus.exe	Administrator
104274150	Unassigned	Unlogged		Unassigned	select /*+ first_rows */ p.product_id, p.pr	1831	67.202.22.15	INVENTORIES	ACCOUNT_PGR	0.50E	Data Manipulation	JDBC-The Client	Administrator
149532264	Unassigned	Unlogged		Unassigned	select p.product_id, product_name, pro	1718	67.202.22.15	INVENTORIES	PR_CATALOG_URL	0.50E	Data Manipulation Read Only	JDBC-The Client	Administrator
151919920	Unassigned	Unlogged		Unassigned	begin dbms_output.enable(10000)	449	67.202.22.15	DBMS_OUTPUT	ENABLED	0.50E	Data Manipulation	sqlplus.exe	Administrator
153966775	Unassigned	Unlogged		Unassigned	select /*+ use_nl */ o.order_id, l.line_item	120	67.202.22.15	CUSTOMERS	ORACLE_CREDIT_LIMIT	0.50E	Data Manipulation Read Only	JDBC-The Client	Administrator
199430471	Unassigned	Unlogged		Unassigned	begin dbms_application_info.set_module	30	67.202.22.15	DBMS_APPLICATION_INFO	MODULE_NAME	0.50E	Procedural	sqlplus.exe	Administrator
200869440	Unassigned	Unlogged		Unassigned	select /*+ use_nl */ o.order_id, l.line_item	495	67.202.22.15	CUSTOMERS	ORACLE_CREDIT_LIMIT	0.50E	Data Manipulation Read Only	JDBC-The Client	Administrator
184131720	Unassigned	Unlogged		Unassigned	begin dbms_output.enable(10000)	2	67.202.22.15	DBMS_OUTPUT	ENABLED	0.50E	Data Manipulation	sqlplus.exe	Administrator
214953880	Unassigned	Unlogged		Unassigned	begin dbms_output.disable end	2	67.202.22.15	DBMS_OUTPUT	ENABLED	0.50E	Procedural	sqlplus.exe	WORKGROUP\AD
214969373	Unassigned	Unlogged		Unassigned	select order_id, seq_recnum from dual	495	67.202.22.15	ORDER_SEQ_NBR	SEQ_NBR	0.50E	Data Manipulation Read Only	JDBC-The Client	Administrator
224742950	Unassigned	Unlogged		Unassigned	begin dbms_output.enable(10000)	1	67.202.22.15	DBMS_OUTPUT	ENABLED	0.50E	Data Manipulation	sqlplus.exe	WORKGROUP\AD
238272675	Unassigned	Unlogged		Unassigned	select min(customer_id), max(customer_id)	2	67.202.22.15	CUSTOMERS	CUSTOMER_ID	0.50E	Data Manipulation Read Only	JDBC-The Client	Administrator
362391530	Unassigned	Unlogged		Unassigned	insert into logon values(0, 0)	151	67.202.22.15	LOGON	LOGON	0.50E	Data Manipulation	JDBC-The Client	Administrator
372759200	Unassigned	Unlogged		Unassigned	begin dbms_output.enable(10000)	1718	67.202.22.15	DBMS_OUTPUT	ENABLED	0.50E	Data Manipulation	sqlplus.exe	Administrator
388818904	Unassigned	Unlogged		Unassigned	begin dbms_application_info.set_module	2	67.202.22.15	DBMS_APPLICATION_INFO	MODULE_NAME	0.50E	Procedural	sqlplus.exe	WORKGROUP\AD
395495715	Unassigned	Unlogged		Unassigned	select customer_seq.nextval from dual	292	67.202.22.15	CUSTOMERS	CUSTOMER_ID	0.50E	Data Manipulation Read Only	JDBC-The Client	Administrator
396296000	Unassigned	Unlogged		Unassigned	begin dbms_output.enable(10000)	1718	67.202.22.15	DBMS_OUTPUT	ENABLED	0.50E	Data Manipulation	sqlplus.exe	Administrator
371447895	Unassigned	Unlogged		Unassigned	select decode('X','Y') from dual	2	67.202.22.15	INVENTORIES	PRODUCT_ID	0.50E	Data Manipulation	sqlplus.exe	WORKGROUP\AD
381093385	Unassigned	Unlogged		Unassigned	select systimestamp from dual	18	67.202.22.15	DUAL	SYSTIMESTAMP	0.50E	Data Manipulation	sqlplus.exe	Administrator
399295000	Unassigned	Unlogged		Unassigned	select product_id, product_name, produ	495	67.202.22.15	PRODUCTS	ORDER_ID	0.50E	Data Manipulation Read Only	JDBC-The Client	Administrator
397429500	Unassigned	Unlogged		Unassigned	select product_id, product_name, produ	3228	67.202.22.15	PRODUCT_INFO_CATALOG_URL	C_SOLE	0.50E	Data Manipulation Read Only	JDBC-The Client	Administrator
407366074	Unassigned	Unlogged		Unassigned	select customer_id, cust_first_name, cus	1512	67.202.22.15	CUSTOMERS	ACCOUNT_PGR	0.50E	Data Manipulation Read Only	JDBC-The Client	Administrator
412399124	Unassigned	Unlogged		Unassigned	update order_header set order_id = o	312	67.202.22.15	ORDERS	ORDER_ID	0.50E	Data Manipulation	JDBC-The Client	Administrator

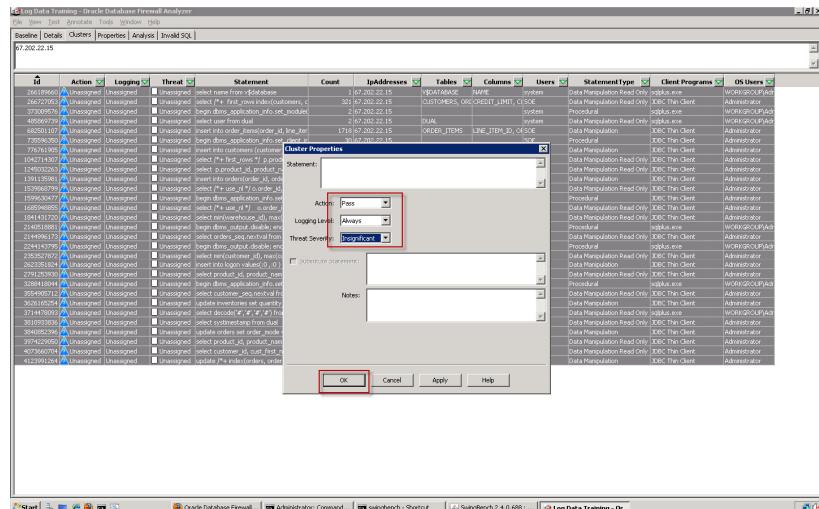
Select all of the clusters on the page, use control-A to select everything. Then, right click on the page and select 'Properties'.



We will set all of this simulated application traffic to be allowed to access the protected database. Select the following:

Actions:	Pass
Logging Level:	Always
Threat Severity:	Insignificant

Click 'OK' to set these properties for all the SQL statement clusters.



Notice that the action, logging and threat columns have been set for each SQL statement clusters.

Action	Logging	Threat	Statement	Count	IpAddress	Tables	Columns	Users	StatementType	Client Programs	OS Users
26619949 ✓ Pass	Always	Diagnostic	select name from v\$database	1	67.202.22.15	SYSTEM			Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
26677209 ✓ Pass	Always	Diagnostic	select /*+ first */ row_id, customer_id	321	67.202.22.15	CUSTOMERS	NAME, CREDIT_LIMIT, DISC		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
37309576 ✓ Pass	Always	Diagnostic	select /*+ first */ row_id, user_name, user_id	2	67.202.22.15	DBA			Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
4058891 ✓ Pass	Always	Diagnostic	select into order_header_id, line_id	1710	67.202.22.15	ORDER_ITEMS	LINE_ITEM_ID, DISC		Data Manipulation	sqlplus.exe	WORKGROUP\PA
6827001 ✓ Pass	Always	Diagnostic	select into order_header_id, line_id	1710	67.202.22.15	ORDER_ITEMS	LINE_ITEM_ID, DISC		Data Manipulation	sqlplus.exe	WORKGROUP\PA
7293636 ✓ Pass	Always	Diagnostic	select into order_header_id, line_id	1710	67.202.22.15	ORDER_ITEMS	LINE_ITEM_ID, DISC		Data Manipulation	sqlplus.exe	WORKGROUP\PA
7767616 ✓ Pass	Always	Diagnostic	select into customers(customer_id, cost)	296	67.202.22.15	CUSTOMERS	ACCOUNT_MGR_ID, DISC		Data Manipulation	sqlplus.exe	WORKGROUP\PA
1042714 ✓ Pass	Always	Diagnostic	select /*+ first */ * from product_id, pr	1831	67.202.22.15	INVENTORIES	PR_CATALOG_URL, DISC		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
1042715 ✓ Pass	Always	Diagnostic	select /*+ first */ * from product_id, pr	1831	67.202.22.15	INVENTORIES	PR_CATALOG_URL, DISC		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
1291129 ✓ Pass	Always	Diagnostic	select /*+ first */ * from customer_id, cs, voter, date	496	67.202.22.15	OPTIONS	CUSTOMER_ID, DISC		Data Manipulation	sqlplus.exe	WORKGROUP\PA
15996889 ✓ Pass	Always	Diagnostic	select /*+ first */ use /*?*/ order_id, line_id	120	67.202.22.15	CUSTOMERS	CREDIT_LIMIT, DISC		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
15996890 ✓ Pass	Always	Diagnostic	select /*+ first */ use /*?*/ order_id, line_id	120	67.202.22.15	CUSTOMERS	CREDIT_LIMIT, DISC		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
1894431 ✓ Pass	Always	Diagnostic	select into rawwarehouse_id, maxwarehouse	2	67.202.22.15	WAREHOUSES	WAREHOUSE_ID, SOCIE		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
21409338 ✓ Pass	Always	Diagnostic	select /*+ first */ * from dual	496	67.202.22.15	DUAL			Procedure	sqlplus.exe	WORKGROUP\PA
21409339 ✓ Pass	Always	Diagnostic	select /*+ first */ * from dual	496	67.202.22.15	DUAL			Procedure	sqlplus.exe	WORKGROUP\PA
2244143 ✓ Pass	Always	Diagnostic	begin /*+ output disable; end;	2	67.202.22.15	CUSTOMERS	CUSTOMER_ID, SOCIE		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
2385271 ✓ Pass	Always	Diagnostic	begin /*+ output disable; end;	2	67.202.22.15	CUSTOMERS	CUSTOMER_ID, SOCIE		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
2385272 ✓ Pass	Always	Diagnostic	begin /*+ output disable; end;	2	67.202.22.15	CUSTOMERS	CUSTOMER_ID, SOCIE		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
27915579 ✓ Pass	Always	Diagnostic	select product_id, product_name, produc	151	67.202.22.15	USERS			Data Manipulation	sqlplus.exe	WORKGROUP\PA
2888189 ✓ Pass	Always	Diagnostic	select /*+ first */ * from product_id, pr	1719	67.202.22.15	PRODUCT_INFOR	PR_CATALOG_URL, DISC		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
2888190 ✓ Pass	Always	Diagnostic	select /*+ first */ * from product_id, pr	1719	67.202.22.15	PRODUCT_INFOR	PR_CATALOG_URL, DISC		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
3625165 ✓ Pass	Always	Diagnostic	select /*+ first */ * from dual, user_id	1719	67.202.22.15	INVENTORIES	PR_CATALOG_URL, DISC		Data Manipulation	sqlplus.exe	WORKGROUP\PA
3714789 ✓ Pass	Always	Diagnostic	select /*+ first */ * from dual	2	67.202.22.15	DUAL			Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
3714790 ✓ Pass	Always	Diagnostic	select /*+ first */ * from dual	2	67.202.22.15	DUAL			Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
3840502 ✓ Pass	Always	Diagnostic	select /*+ first */ * from dual, user_id	496	67.202.22.15	OPTIONS	OPTION_ID, DISC		Data Manipulation	sqlplus.exe	WORKGROUP\PA
3974296 ✓ Pass	Always	Diagnostic	select product_id, product_name, produc	3229	67.202.22.15	PRODUCT_INFOR	PR_CATALOG_URL, DISC		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
40736670 ✓ Pass	Always	Diagnostic	select /*+ first */ * from product_id, pr	1511	67.202.22.15	CUSTOMERS	ACCOUNT_MGR_ID, DISC		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
4212991 ✓ Pass	Always	Diagnostic	select /*+ first */ * from dual, user_id	316	67.202.22.15	OPTIONS	OPTION_ID, DISC		Data Manipulation	sqlplus.exe	WORKGROUP\PA

- We will also create a Login/Logout policy to allow users to authenticate to the protected Oracle Database. Click on the ‘Tools’ menu, then on the ‘Login/Logout Policy’.

Action	Logging	Threat	Statement	Count	IpAddress	Tables	Columns	Users	StatementType	Client Programs	OS Users
26619949 ✓ Pass	Always	Diagnostic	name from v\$database	1	67.202.22.15	SYSTEM			Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
26677209 ✓ Pass	Always	Diagnostic	select /*+ first */ row_id, customer_id	321	67.202.22.15	CUSTOMERS	CREDIT_LIMIT, DISC		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
37309576 ✓ Pass	Always	Diagnostic	select /*+ first */ row_id, user_name, user_id	2	67.202.22.15	DBA			Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
4058891 ✓ Pass	Always	Diagnostic	select /*+ first */ use /*?*/ from dual	1710	67.202.22.15	CUSTOMERS	CREDIT_LIMIT, DISC		Data Manipulation	sqlplus.exe	WORKGROUP\PA
6827001 ✓ Pass	Always	Diagnostic	select /*+ first */ use /*?*/ from dual	1710	67.202.22.15	ORDER_ITEMS	LINE_ITEM_ID, DISC		Data Manipulation	sqlplus.exe	WORKGROUP\PA
7293636 ✓ Pass	Always	Diagnostic	select /*+ first */ use /*?*/ from dual	1710	67.202.22.15	ORDER_ITEMS	LINE_ITEM_ID, DISC		Data Manipulation	sqlplus.exe	WORKGROUP\PA
7767616 ✓ Pass	Always	Diagnostic	select /*+ first */ use /*?*/ from customers(customer_id, cost)	296	67.202.22.15	CUSTOMERS	ACCOUNT_MGR_ID, DISC		Data Manipulation	sqlplus.exe	WORKGROUP\PA
1042714 ✓ Pass	Always	Diagnostic	select /*+ first */ * from product_id, pr	1831	67.202.22.15	INVENTORIES	PR_CATALOG_URL, DISC		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
1042715 ✓ Pass	Always	Diagnostic	select /*+ first */ * from product_id, pr	1831	67.202.22.15	INVENTORIES	PR_CATALOG_URL, DISC		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
1291129 ✓ Pass	Always	Diagnostic	select /*+ first */ * from customer_id, cs, voter, date	496	67.202.22.15	OPTIONS	CUSTOMER_ID, DISC		Data Manipulation	sqlplus.exe	WORKGROUP\PA
15996889 ✓ Pass	Always	Diagnostic	select /*+ first */ use /*?*/ order_id, line_id	120	67.202.22.15	CUSTOMERS	CREDIT_LIMIT, DISC		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
15996890 ✓ Pass	Always	Diagnostic	select /*+ first */ use /*?*/ order_id, line_id	120	67.202.22.15	CUSTOMERS	CREDIT_LIMIT, DISC		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
1894431 ✓ Pass	Always	Diagnostic	select /*+ first */ rawwarehouse_id, maxwarehouse	2	67.202.22.15	WAREHOUSES	WAREHOUSE_ID, SOCIE		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
21409338 ✓ Pass	Always	Diagnostic	select /*+ first */ * from dual	496	67.202.22.15	DUAL			Procedure	sqlplus.exe	WORKGROUP\PA
21409339 ✓ Pass	Always	Diagnostic	select /*+ first */ * from dual	496	67.202.22.15	DUAL			Procedure	sqlplus.exe	WORKGROUP\PA
2244143 ✓ Pass	Always	Diagnostic	begin /*+ output disable; end;	2	67.202.22.15	CUSTOMERS	CUSTOMER_ID, SOCIE		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
2385271 ✓ Pass	Always	Diagnostic	begin /*+ output disable; end;	2	67.202.22.15	CUSTOMERS	CUSTOMER_ID, SOCIE		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
2385272 ✓ Pass	Always	Diagnostic	begin /*+ output disable; end;	2	67.202.22.15	CUSTOMERS	CUSTOMER_ID, SOCIE		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
27915579 ✓ Pass	Always	Diagnostic	select product_id, product_name, produc	151	67.202.22.15	USERS			Data Manipulation	sqlplus.exe	WORKGROUP\PA
2888189 ✓ Pass	Always	Diagnostic	select /*+ first */ * from product_id, pr	1719	67.202.22.15	PRODUCT_INFOR	PR_CATALOG_URL, DISC		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
2888190 ✓ Pass	Always	Diagnostic	select /*+ first */ * from product_id, pr	1719	67.202.22.15	PRODUCT_INFOR	PR_CATALOG_URL, DISC		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
3625165 ✓ Pass	Always	Diagnostic	select /*+ first */ * from dual, user_id	1719	67.202.22.15	INVENTORIES	PR_CATALOG_URL, DISC		Data Manipulation	sqlplus.exe	WORKGROUP\PA
3714789 ✓ Pass	Always	Diagnostic	select /*+ first */ * from dual	2	67.202.22.15	DUAL			Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
3714790 ✓ Pass	Always	Diagnostic	select /*+ first */ * from dual	2	67.202.22.15	DUAL			Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
3840502 ✓ Pass	Always	Diagnostic	select /*+ first */ * from dual, user_id	496	67.202.22.15	OPTIONS	OPTION_ID, DISC		Data Manipulation	sqlplus.exe	WORKGROUP\PA
3974296 ✓ Pass	Always	Diagnostic	select product_id, product_name, produc	3229	67.202.22.15	PRODUCT_INFOR	PR_CATALOG_URL, DISC		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
40736670 ✓ Pass	Always	Diagnostic	select /*+ first */ * from product_id, pr	1511	67.202.22.15	CUSTOMERS	ACCOUNT_MGR_ID, DISC		Data Manipulation Read Only	sqlplus.exe	WORKGROUP\PA
4212991 ✓ Pass	Always	Diagnostic	select /*+ first */ * from dual, user_id	316	67.202.22.15	OPTIONS	OPTION_ID, DISC		Data Manipulation	sqlplus.exe	WORKGROUP\PA

Enter the following into the Login/Logout Policy Dialog:

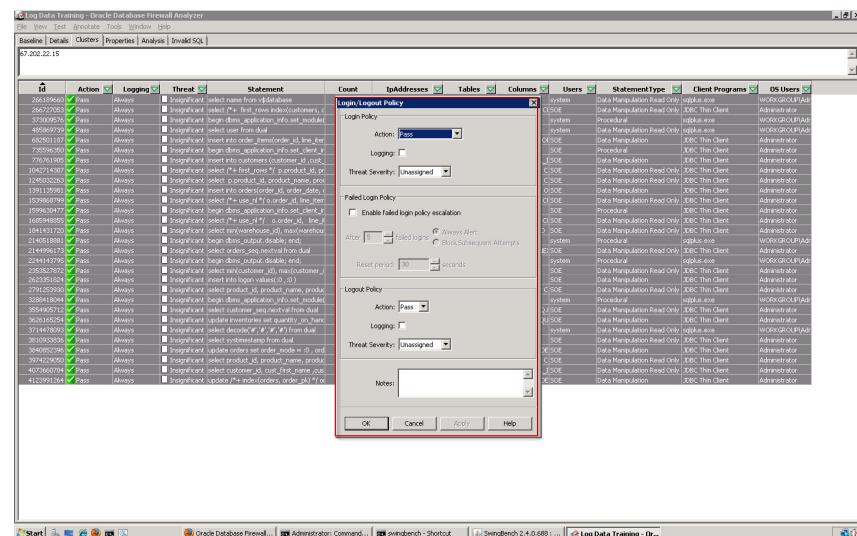
Login Policy

Action:	Pass
Threat Severity:	Unassigned

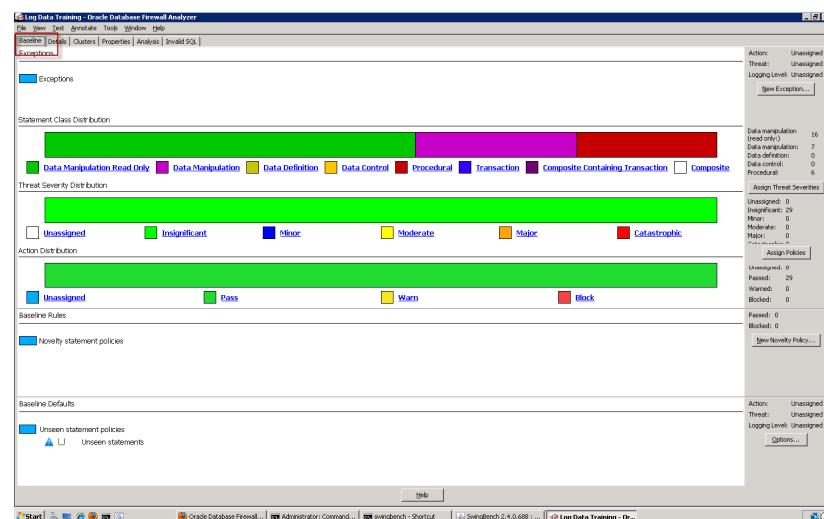
Logout Policy

Action:	Pass
Threat Severity:	Unassigned

Click 'OK' when completed.

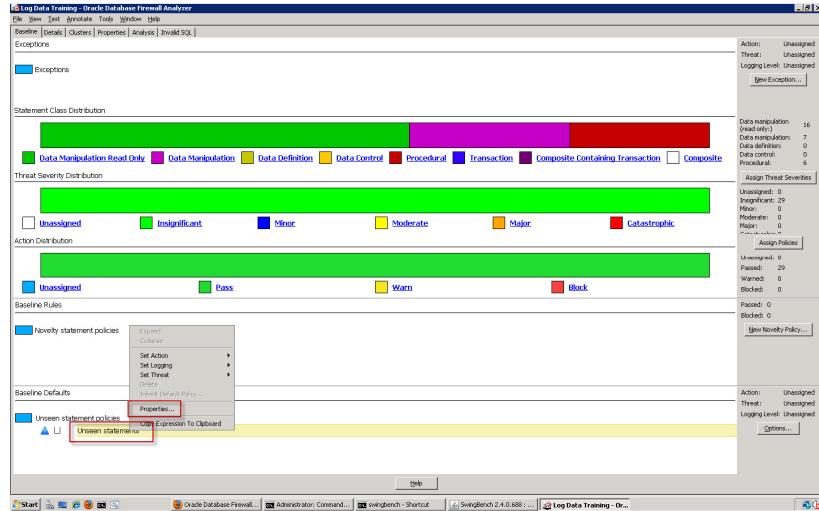


Return to the 'Baseline' tab.



- Now that we have classified all of our *authorized* SQL traffic we will let DBFW what to do in the event of *unauthorized* SQL traffic. DBFW classifies unauthorized as being *unseen*, in that, as far as DBFW is concerned the SQL traffic has not been added to the White List and is

therefore not allowed. Right click on the ‘**Unseen statements**’ section at the bottom of the screen. Then, select the ‘**Properties**’ section.



Enter the following in the **Unseen Statement Dialog**:

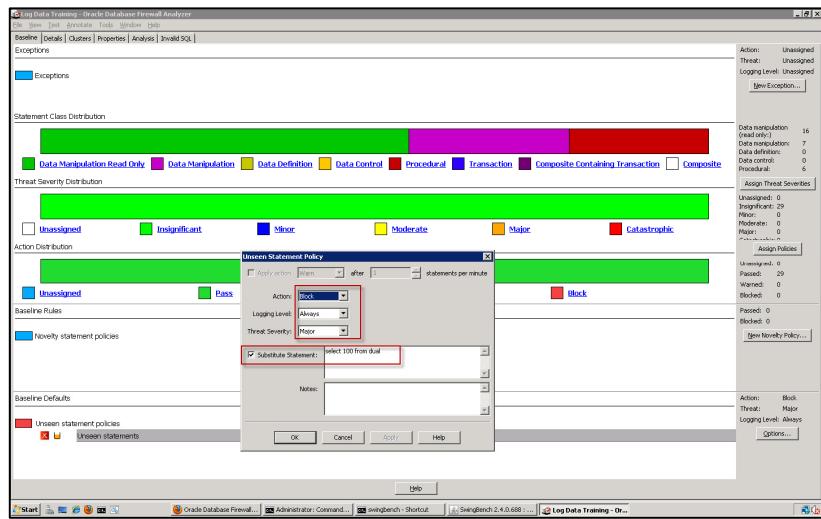
Action:	Block
Logging Level:	Always
Threat Severity:	Major

Then, select the ‘**Substitute Statement**’ checkbox and enter the following SQL:

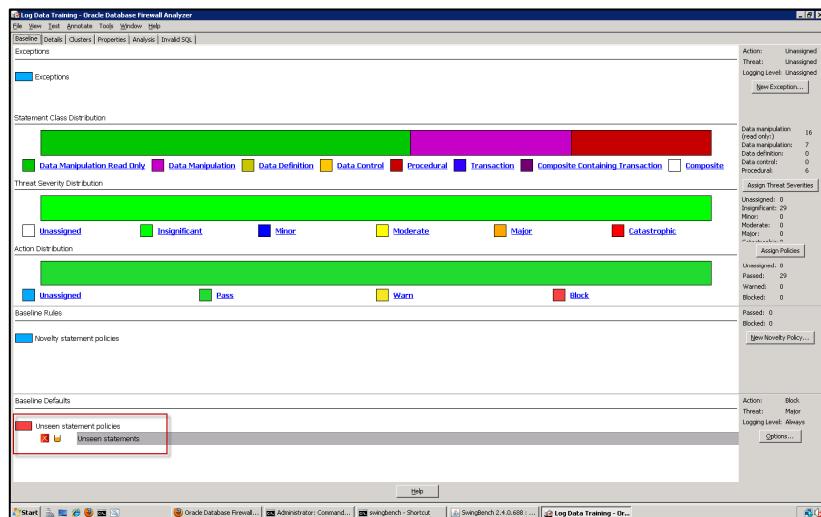
```
select 100 from dual
```

NOTE: Do not enter a semi-colon.

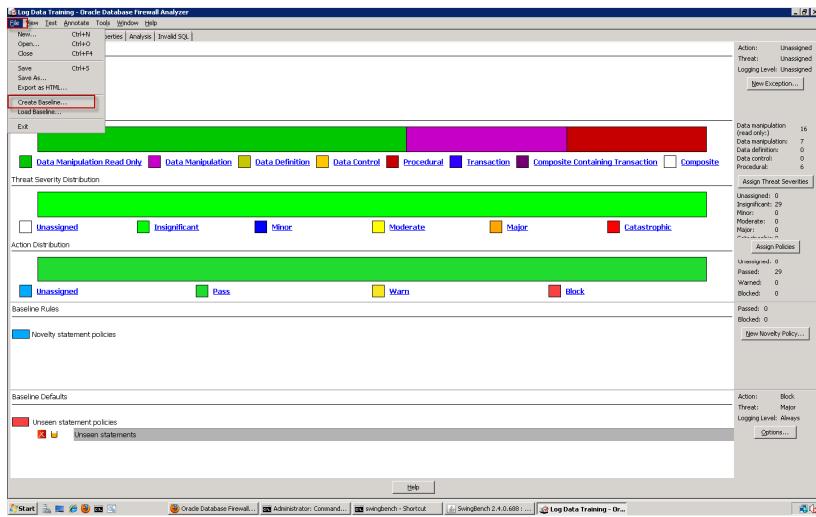
This substitution capability is a key differentiator for Oracle Database Firewall. In Database Policy Enforcement (DPE) mode only, the Analyzer enables you to define a substitute statement for each blocked cluster. When a SQL statement that matches the cluster is blocked, the substitute statement is used instead. This can prevent undesirable effects (e.g. avoiding a communication loop) and may display a suitable error to the database user who originated the statement.



After entering this policy you will see it represented at the bottom of the page.

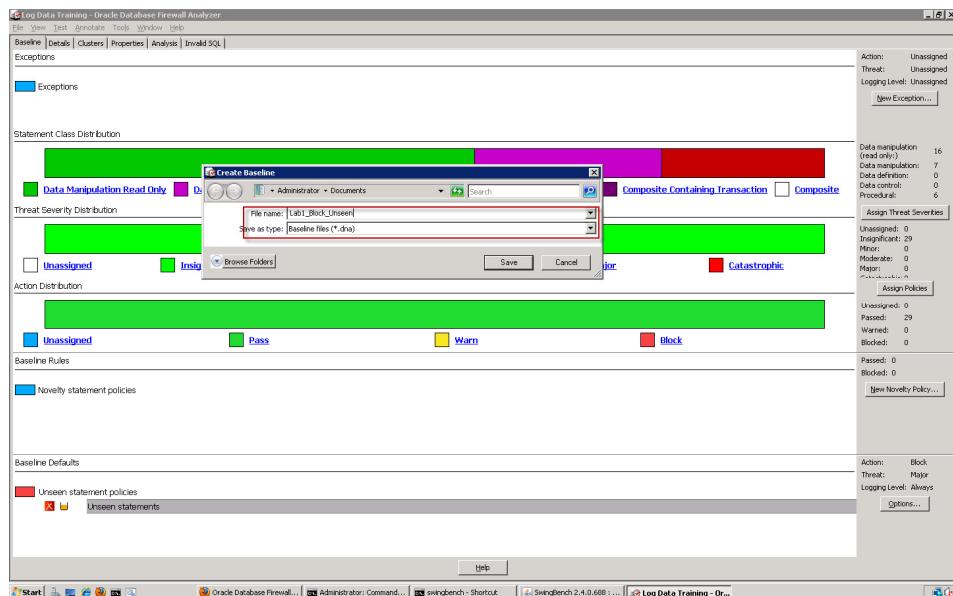


5. This policy is now ready to be uploaded to the DBFW server and enforced. Click on the ‘File’ menu and then select ‘Create Baseline’.



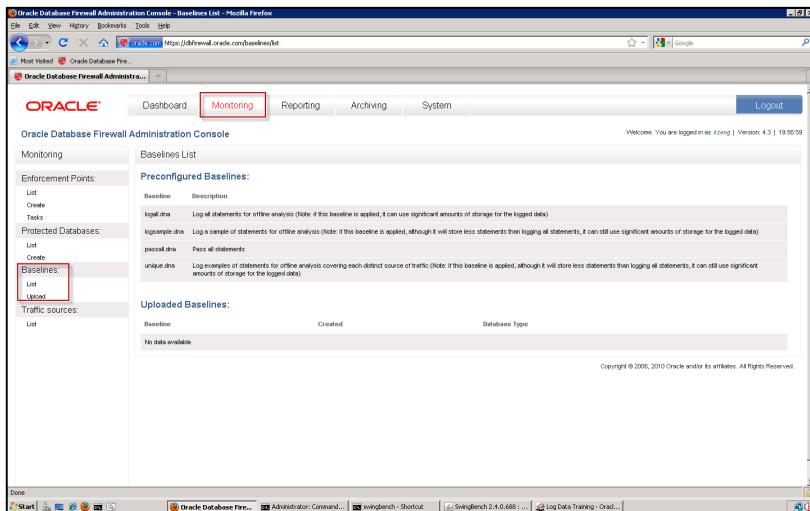
Enter the following file name: **Lab1_Block_Unseen**

This will be saved in the Administrator’s Document folder.

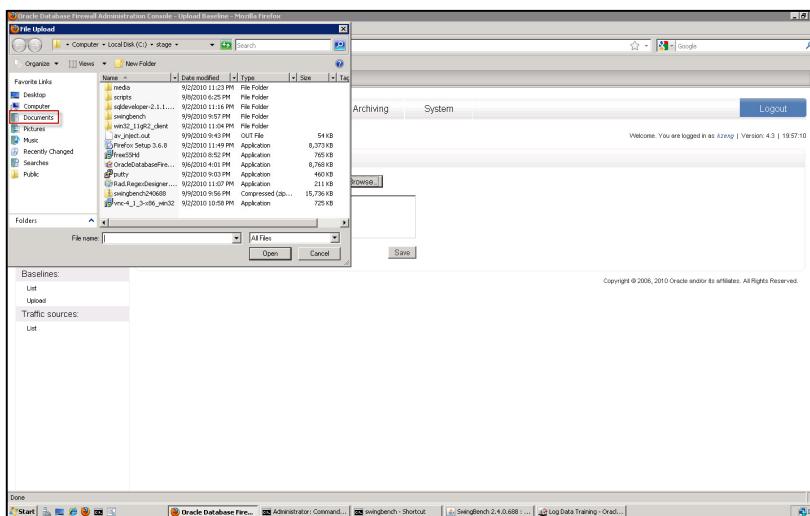


Leave the Oracle DBFW Analyzer open.

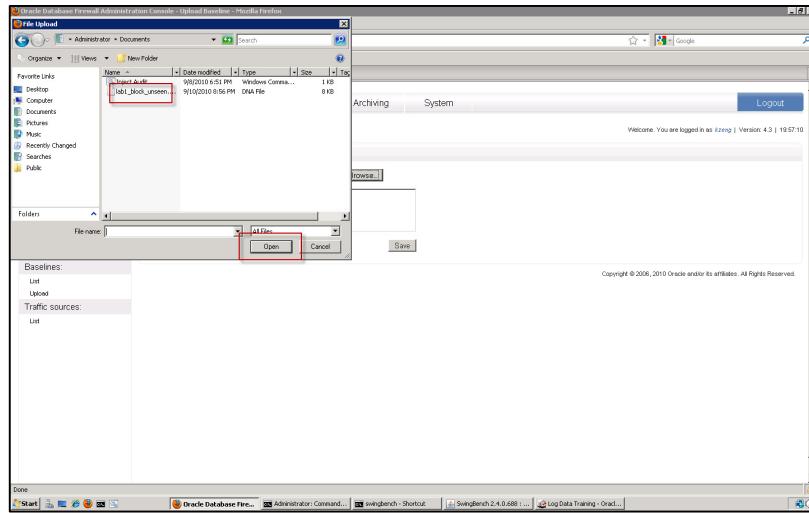
6. Return to the DBFW Web Administration Console. Login with the kzeng user if necessary. Click on the ‘Monitoring’ tab, then on the ‘List’ link in the ‘Baselines’ section. You will see all of the default baseline policies and a section with uploaded policies. Click on the ‘Upload’ link in the ‘Baselines’ section to upload our new baseline.



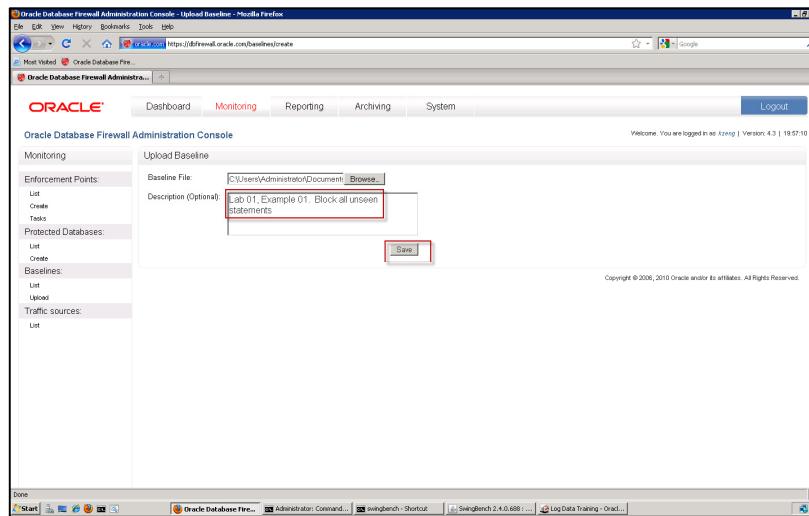
Click on the ‘browse’ button, then select the ‘Documents’ folder on the left hand navigation pane. You will then be able to select the Lab1_Block_Unseen.dna file we just created.



Click 'Open' once you have selected the file.



Enter a meaningful description. We entered – **'Lab 01. Example 01. Block All Unseen Statements'**. Informing us that DBFW will block everything that has not already been authorized. In our case this is only the simulated application traffic generated from Swingbench. Click on 'Save'.



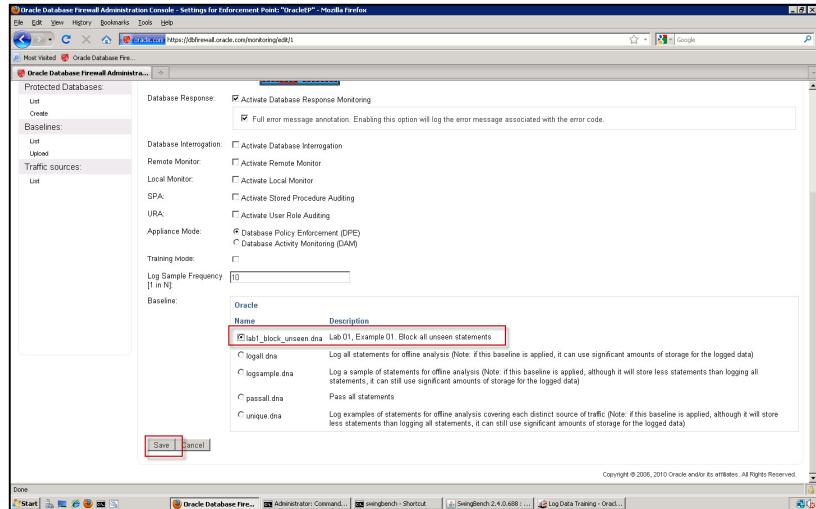
You will see that the policy has been uploaded successfully and is visible in the ‘Uploaded Baselines’ section.

Baseline	Created	Database Type	Description
Lab_01_Example_01_Block_all_unseen_statements	2010-09-10	Oracle	Lab_01_Example_01_Block all unseen statements

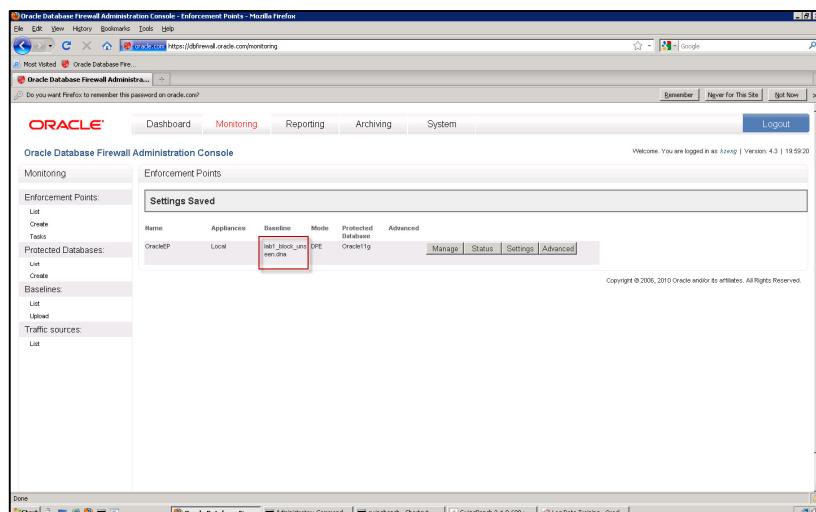
- To implement the baseline policy we will have to modify the Enforcement Point ‘OracleEP’ that we created earlier. Click on the ‘Monitoring’ tab, followed by the ‘List’ link in the ‘Enforcement Points’ section on the left hand side. Then click on the ‘Settings’ button for the ‘OracleEP’ Enforcement Point.

Name	Appliances	Baseline	Mode	Protected Database	Advanced
OracleEP	Local	legal.dra	DPE	Oracle11g	Manage Status Settings Advanced

Select the new baseline policy '**Lab1_block_unseen.dna**' from the radio button list. Then click 'Save'.



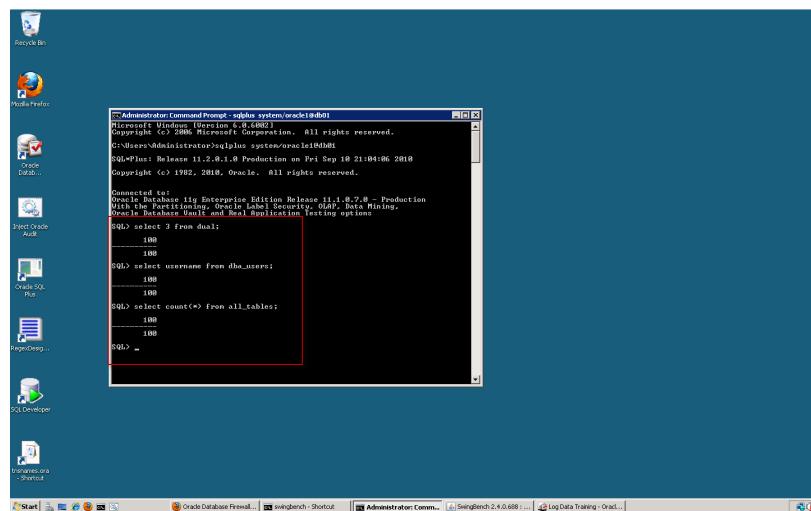
In the summary screen you will see that the baseline is now set to our new policy.



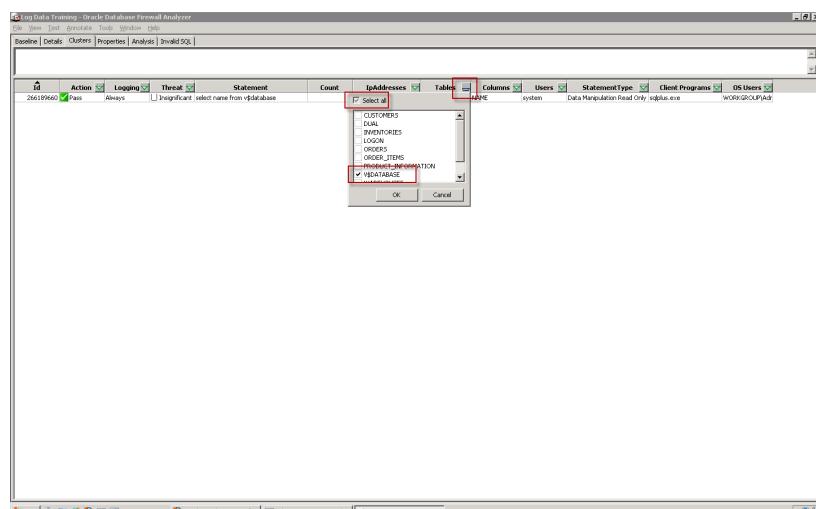
8. To test the policy is working we will open a command prompt screen and use SQL*Plus to login to the protected Oracle database. If you have not already got a command prompt open click on the icon at the bottom of the screen. Then enter the following:

```
sqlplus system/oracle1@db06
select 3 from dual;
select username from dba_users;
select count(*) from all_users;
```

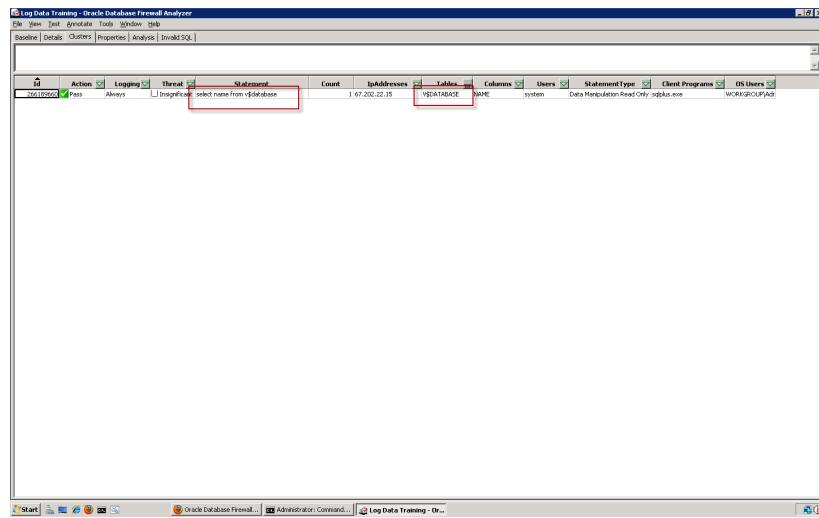
Notice that each SQL command returns '100'. This is because each SQL Statement is considered unseen and will trigger our SQL substitution.



Return to the DBFW Analyzer. Click on the '**Clusters**' tab. Click on the button to the right of the '**Tables**' column heading. We will be checking that there is an authorized (seen) SQL statement to test. Select the '**V\$DATABASE**' table.



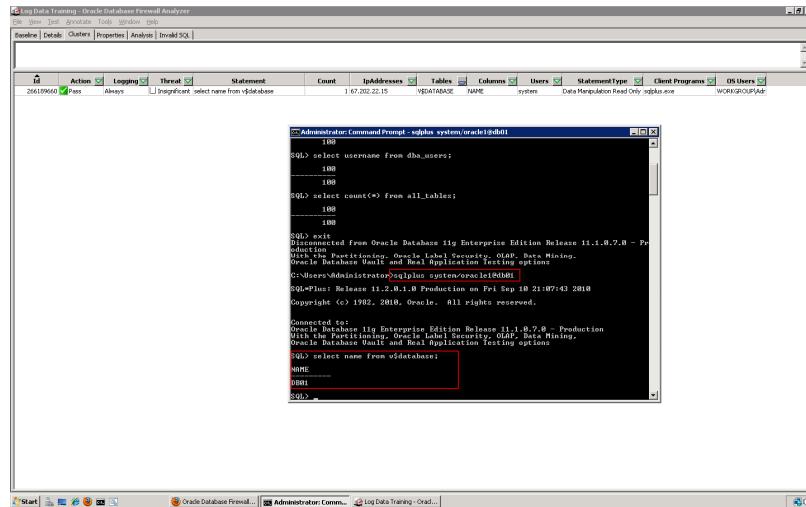
You should see a single entry showing the 'select name from v\$database' statement that we made initially to test the DBFW configuration.



Return to the SQL Plus session. If you have closed it, please re-open the Windows Command Prompt and login to DB06 as system/oracle1, as follows:

```
sqlplus system/oracle1@db06
select name from v$database;
```

You will see that the name 'DB06' is returned. This is because this SQL Statement is being allowed since it was in our baseline.



Now enter the following SQL:

```
select * from v$database;
```

Notice that this is blocked and a substitute statement (returning 100) has been used. This is because DBFW understands that more columns are being potentially returned and the SQL is grammatically different.

The screenshot shows the Oracle Database Firewall Log Data Training interface. A table titled 'Statement' lists a single row: '26619960 Pass Always 1 Insignificant select name from v\$database'. Below this is a command prompt window showing the following session:

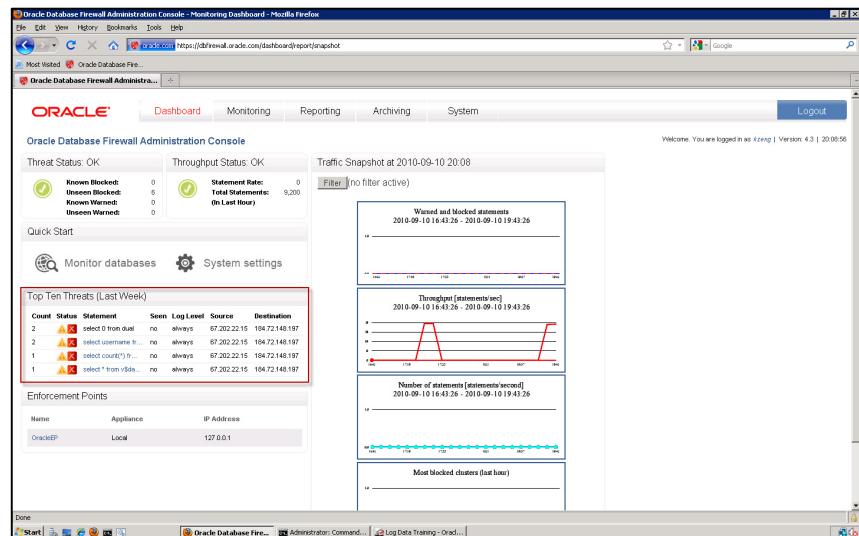
```

Administrator: Command Prompt - sqlplus system/oracle@db01
SQL> select count(*) from all_tables;
          100
          100
          100
SQL> exit
Disconnected From Oracle Database 11g Enterprise Edition Release 11.1.0.7.8 - Production
With Advanced Multitenant Features, Oracle Label Security, OLAP, Data Mining,
Oracle Database Vault and Real Application Testing options
C:\Users\Administrator>sqlplus system/oracle@db01
SQL*Plus: Release 11.2.0.1.0 Production on Fri Sep 10 21:07:43 2010
Copyright (c) 1982, 2010, Oracle. All rights reserved.

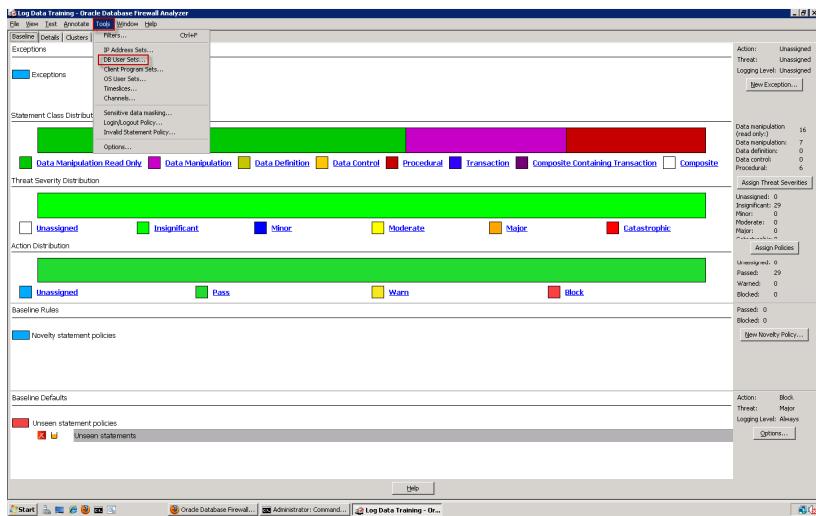
Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.7.8 - Production
With Advanced Multitenant Features, Oracle Label Security, OLAP, Data Mining,
Oracle Database Vault and Real Application Testing options
SQL> select name from v$database;
NAME
DB01
SQL> select * from v$database;
          100
          100
SQL> 

```

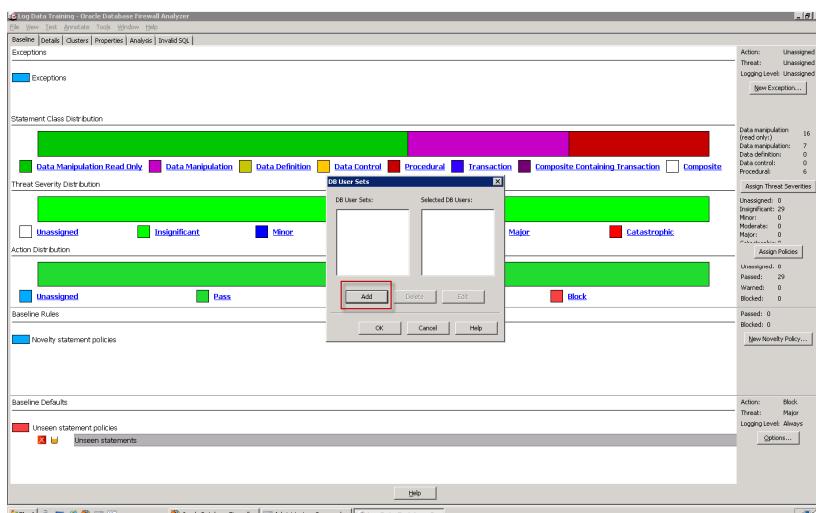
Return to the DBFW Administration Console. Notice that in the '**Dashboard**' tab that there is a list of the Top Ten Threats – containing SQL Statements that you have just seen as being blocked.



9. We will now refine the baseline policy to make an exception allowing SQL traffic from the some administrator users. Return to the DBFW Analyzer. Once there click on the ‘Tools’ menu. Then click on the ‘DB User Sets’ link. This will allow us to define what users we are going to have as our administrators.



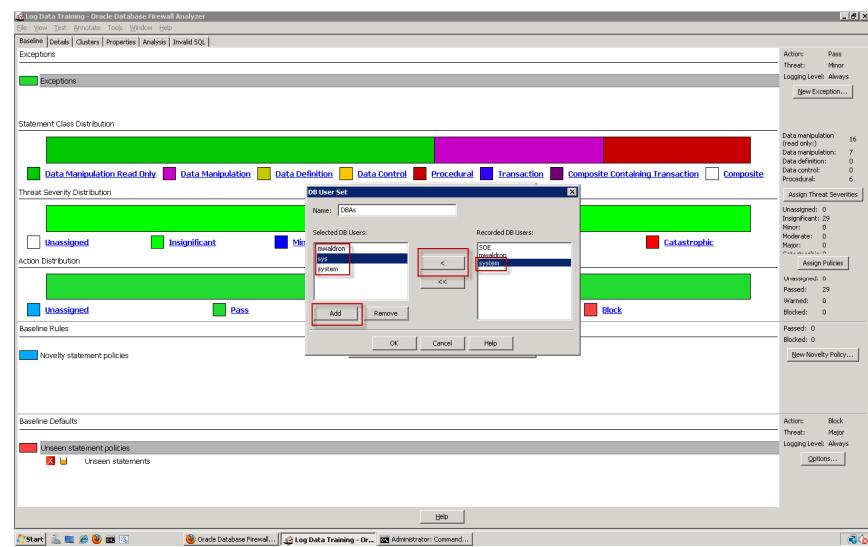
Click on the ‘Add’ button.



Add the following three users in the DB User Set dialog:

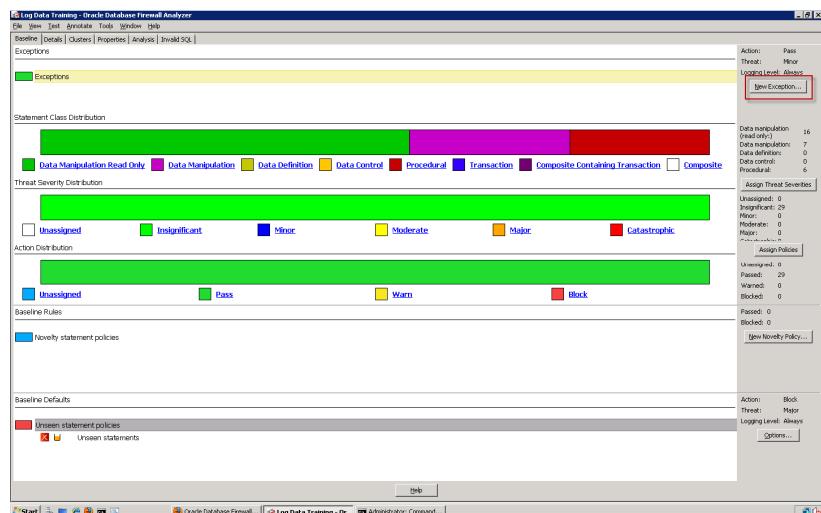
- sys
- system
- mwaldron

If the users are present in the right-hand panel select the '<' button to copy them over, alternately, click on the 'Add' button then enter the username. Once all the three users are added, click 'OK'. Then, click 'OK' again in the 'DB User Set' summary pane.

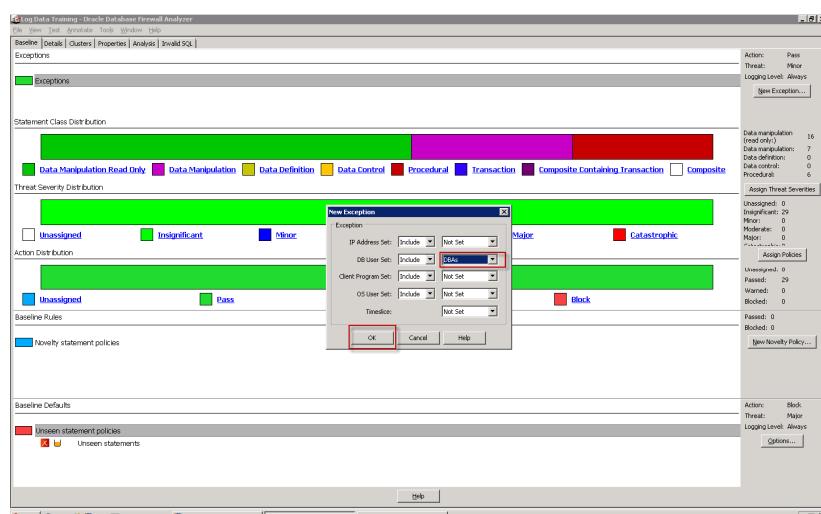


10. Navigate back to the ‘Baseline’ tab. Click on the ‘New Exception’ button. This will add a new exception – we will add the DB User set to the exception, meaning that these users in that group will be able to access anything.

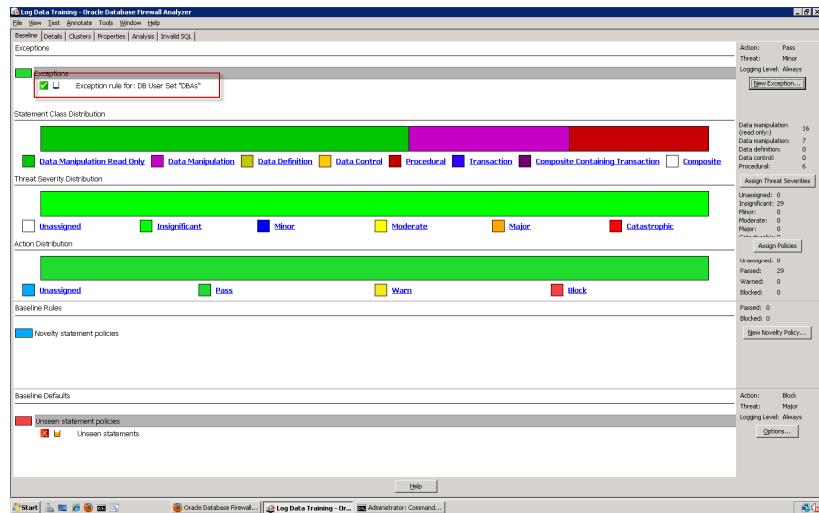
An exception determines the action level, logging level, and threat severity to use for statements that occur during specific times of the day, or originate (or do not originate) from selected client IP addresses or user names. Exceptions override all other baseline rules. You may, for example, want to set up an exception that overrides standard baseline rules for SQL statements originating from an administrator or for statements originating from anyone other than the administrator.



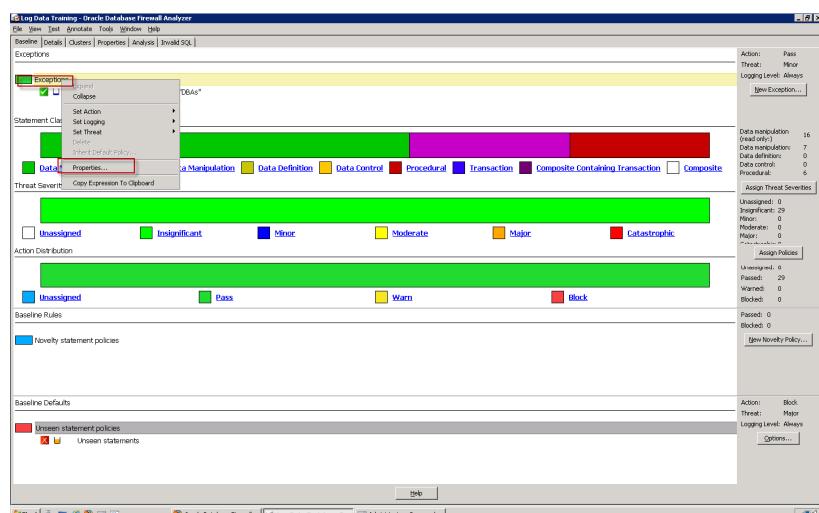
In the ‘DB User Set’ configuration, change the group to DBA’s.



Observe that the exception has now been created.



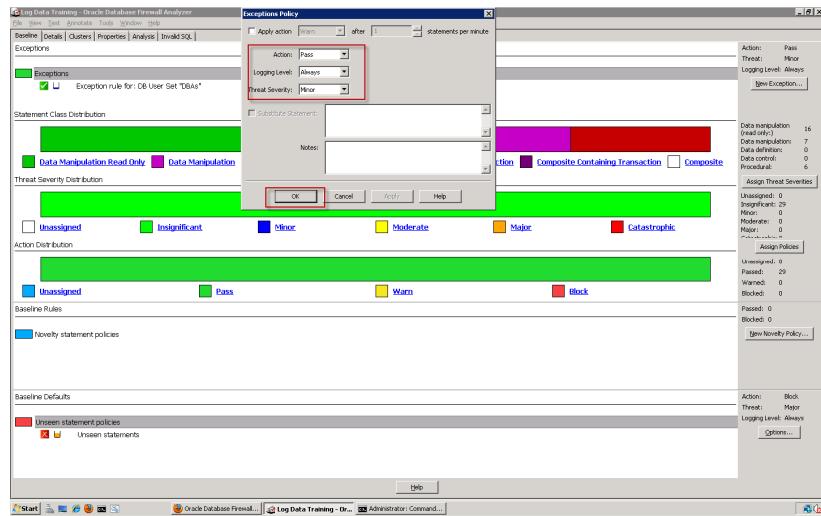
Right click on the 'Exceptions' word, and then click on the 'Properties' link. We will configure the exception policy for our admin users.



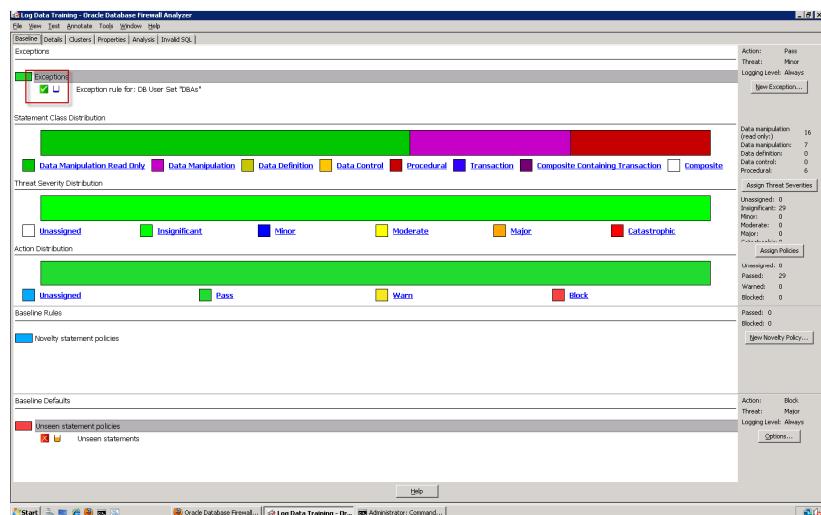
Enter the following in the **Exception Policy** dialog:

Action:	Pass
Logging Level:	Always
Threat Severity:	Minor

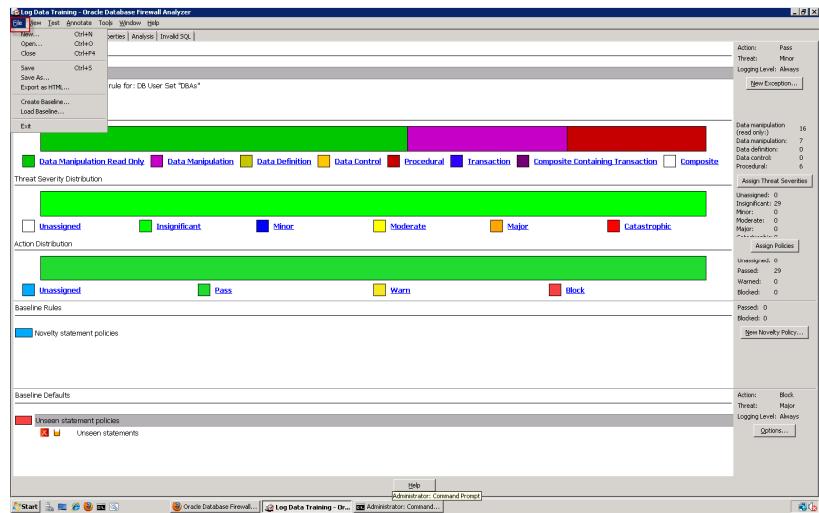
This will allow all traffic from our admin users (sys, system and mwaldron) to pass to the protected database.



Notice that the exception is now set in the baseline screen.

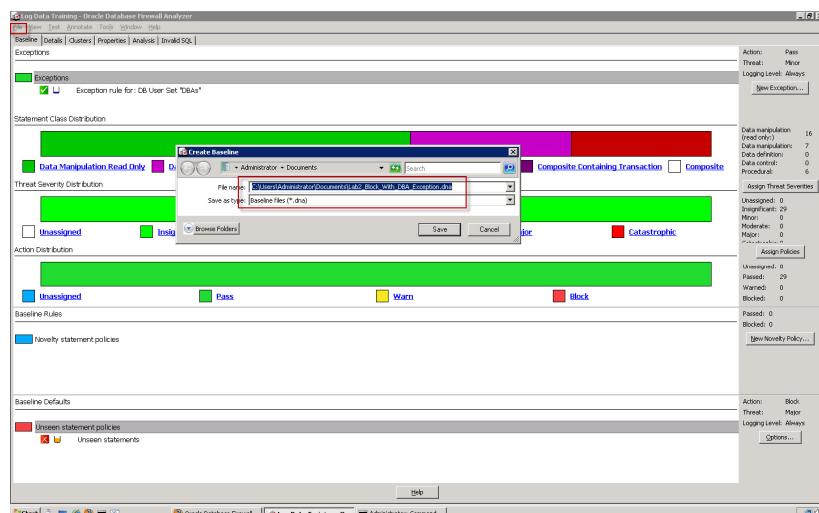


We can now save a new baseline and implement it in DBFW. Click on the ‘File’ menu, followed by ‘Create Baseline’.



Enter the following file name: **Lab2_Block_With_DBA_Exception**

Click ‘Save’.



11. Open the DBFW Administration Console. Ensure that you are still logged in as 'kzeng'. Click on the '**Monitoring**' tab. Then click on the '**List**' link in the '**Baselines**' section. You will see that the previous uploaded baseline policy is in place. Click on the '**Upload**' link in the '**Baselines**' section.

Baseline	Created	Database Type	Description
Lab_01_Block_unseen.dna	2010-09-10	Oracle	Lab 01, Example 01: Block all unseen statements

Select the **Lab2_Block_With_DBA_Exception.dna** file from the Administrator Documents folder. Then enter a meaningful description, we've entered '**Lab 01. Example 02. Block all except admin**'. Click '**Save**'.

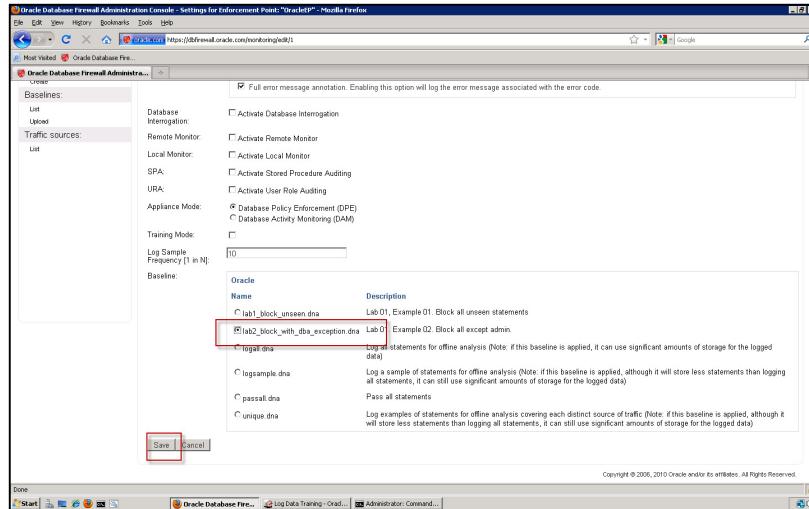
You will see that the new baseline policy is uploaded and available.

The screenshot shows the Oracle Database Firewall Administration Console. The title bar reads "Oracle Database Firewall Administration Console - Baselines List - Mozilla Firefox". The main menu at the top includes "File", "Edit", "View", "History", "Bookmarks", "Tools", "Help", "Logout", and "Welcome: You are logged in as zhang | Version: 4.3 | 10:28:14". The left sidebar has sections for "Monitoring", "Enforcement Points", "Protected Databases", "Baselines", and "Traffic sources". Under "Monitoring", the "List" link is selected. The main content area is titled "Preconfigured Baselines" and "Uploaded Baselines". The "Preconfigured Baselines" section lists four baselines: "legal.dba", "logsample.dba", "passall.dba", and "unique.dba". The "Uploaded Baselines" section lists two baselines: "m01_block_all_unseen.dba" and "m02_block_all_except_watermark.dba". Both rows have "Edit" and "Delete" buttons. At the bottom right of the page is the copyright notice "Copyright © 2006, 2010 Oracle and/or its affiliates. All Rights Reserved."

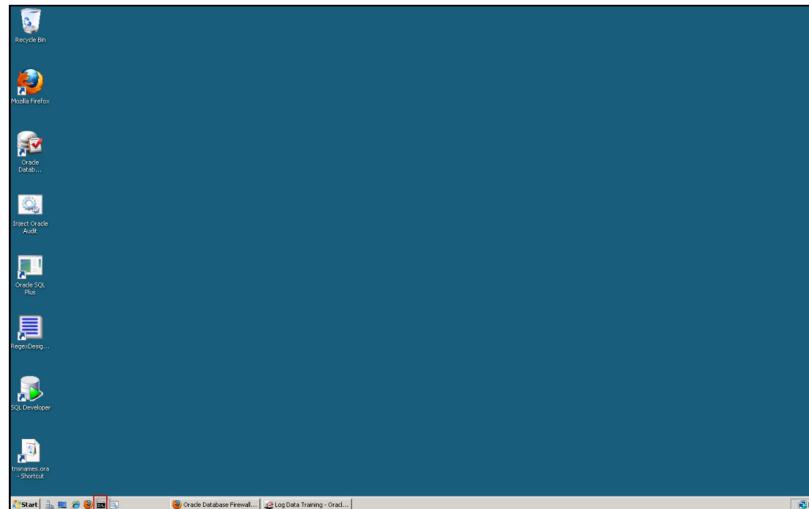
Navigate to the '**Monitoring**' tab, then to the '**List**' link in the '**Enforcement Points**' section. Click on the '**Settings**' button. We will be enforcing our new baseline policy by configuring our '**OracleEP**' to use it.

The screenshot shows the Oracle Database Firewall Administration Console. The title bar reads "Oracle Database Firewall Administration Console - Enforcement Points - Mozilla Firefox". The main menu at the top includes "File", "Edit", "View", "History", "Bookmarks", "Tools", "Help", "Logout", and "Welcome: You are logged in as zhang | Version: 4.3 | 10:28:14". The left sidebar has sections for "Monitoring", "Enforcement Points", "Protected Databases", "Baselines", and "Traffic sources". Under "Monitoring", the "List" link is selected. The main content area is titled "Enforcement Points". It shows a table with one row for "OracleEP". The columns are Name, Appliances, Baseline, Mode, Protected Database, and Advanced. The "Baseline" column for OracleEP is set to "legal.dba". Below the table are "Manage", "Status", "Settings", and "Advanced" buttons. The "Settings" button is highlighted with a red box. At the bottom right of the page is the copyright notice "Copyright © 2006, 2010 Oracle and/or its affiliates. All Rights Reserved."

Select the new '**Lab2_block_with_dba_exception.dna**' baseline policy radio button from the list. Then, click '**Save**'.



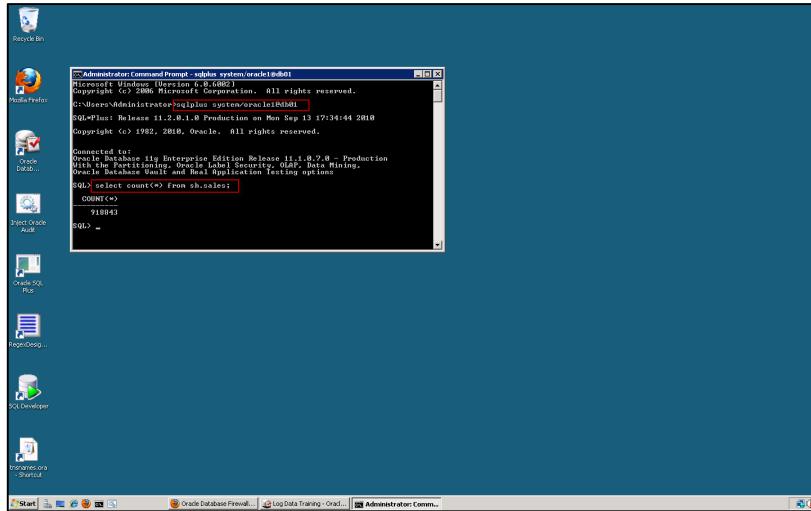
12. Finally, we can test that the new exception is taking effect. On the Desktop, open a Windows Command Prompt. Then login to SQL*Plus at system/oracle1@db06.



Enter the following:

```
sqlplus system/oracle1@db06
select count(*) from sh.sales;
```

The last command is NOT in our whitelist, as it is not in the simulated application load generated by Swingbench. The exception policy is working as our system user is now able to execute SQL statements that are not within our whitelist policy.



D. Summary

You accomplished the following in this lab exercise:

1. Completed an iterative development cycle of the baseline
2. Used the Oracle Database Firewall (DBFW) Analyzer to analyze and train on traffic logs.
3. Developed and deploy a baseline policy
4. Modified and re-deploy the baseline policy
5. Verified that policy is enforced and ensure that unseen traffic is blocked

LAB EXERCISE 03 – ORACLE DATABASE FIREWALL – GAIN VISIBILITY AND SATISFY REQUIREMENTS THROUGH REPORTING

INTRODUCTION

Flexible reporting and alerting

Oracle Database Firewall includes over 125 prebuilt reports that can be easily customized for regulations such as PCI, HIPAA and SOX. Real-time alerts can also be setup for fast response to any policy exception. For privacy and compliance requirements, personally identifiable information contained in logged SQL can be masked.

A. Lab Scenarios and Objectives

In this lab exercise you will accomplish the following:

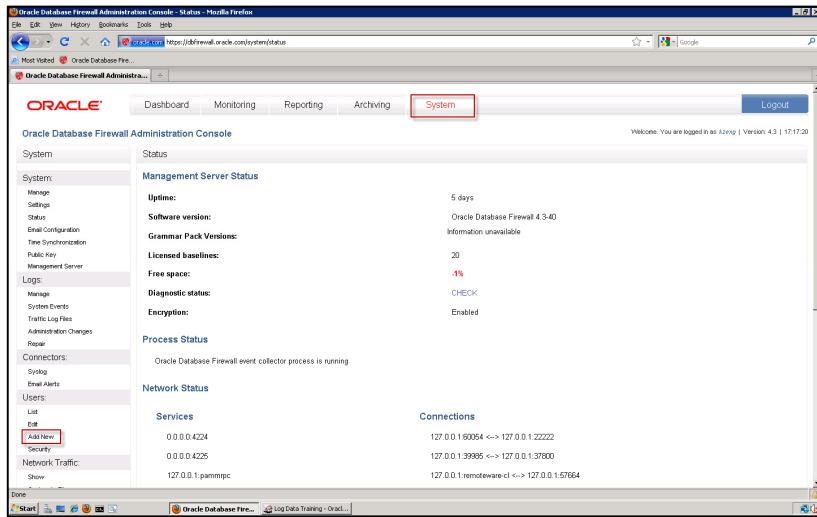
1. *Create a Read-only Report User*
2. *Generate and review available reports*
3. *Gain experience with report groups and hierarchies*

B. Setup and Preparation

- Completion of LAB EXERCISE 01 – ORACLE DATABASE FIREWALL ENFORCEMENT POINTS TO MONITOR AND PROTECT DATABASES
- Completion LAB EXERCISE 02 – ORACLE DATABASE FIREWALL – USE THE TRAFFIC ANALYZER TO CONFIGURE POLICIES AND BLOCK UNAUTHORIZED TRAFFIC

C. GAIN VISIBILITY AND SATISFY REQUIREMENTS THROUGH REPORTING

1. Return to the DBFW Administration Console. If you are not already logged in, please login as 'kzeng' using the password 'oracle1'. We will create a new user that is solely responsible for managing DBFW reports. Click on the '**Add New**' link in the 'Users' section.



2. We will create a user called 'mwaldron' with a password of 'oracle1'. This user will have the '**View-Only User**' Role.

Enter the following information in the **Add User** screen:

Username:	mwaldron
First Name:	Mark
Last Name:	Waldron
Role:	View-Only User
Force Password Change:	<<Uncheck>>
Password:	oracle1

Oracle Database Firewall Administration Console - Add User - Mozilla Firefox

ORACLE Dashboard Monitoring Reporting Archiving System

Welcome: You are logged in as Kong | Version: 4.3 | 17:27:40

Oracle Database Firewall Administration Console

System Add User

Manage Username: mwaldron

Status First Name: Mark

Email Configuration Last Name: Waldron

Time Synchronization Email:

Public Key Role: View-only User

Management Server Logs Force Password Change on Next Login:

System Events Suspended:

Traffic Log Files Password: *****

Administration Changes Confirm Password: *****

Regular Signup

Copyright © 2009, 2010 Oracle and/or its affiliates. All Rights Reserved.

3. You will be returned to the user list screen, showing that the new user has been created successfully. Click on the 'Logout' button. We will login as the 'mwaldron' user to access DBFW reports.

Oracle Database Firewall Administration Console - Users - Mozilla Firefox

ORACLE Dashboard Monitoring Reporting Archiving System

Welcome: You are logged in as Kong | Version: 4.3 | 17:28:09

Oracle Database Firewall Administration Console

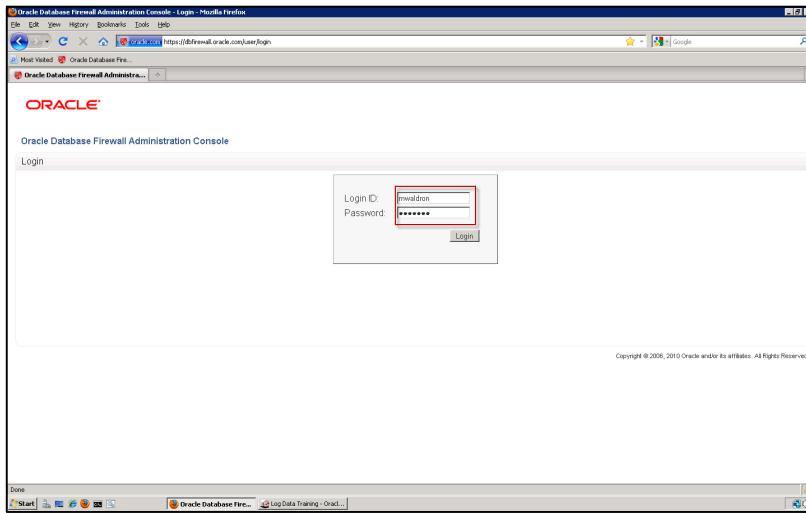
System Users

User 'mwaldron' successfully created

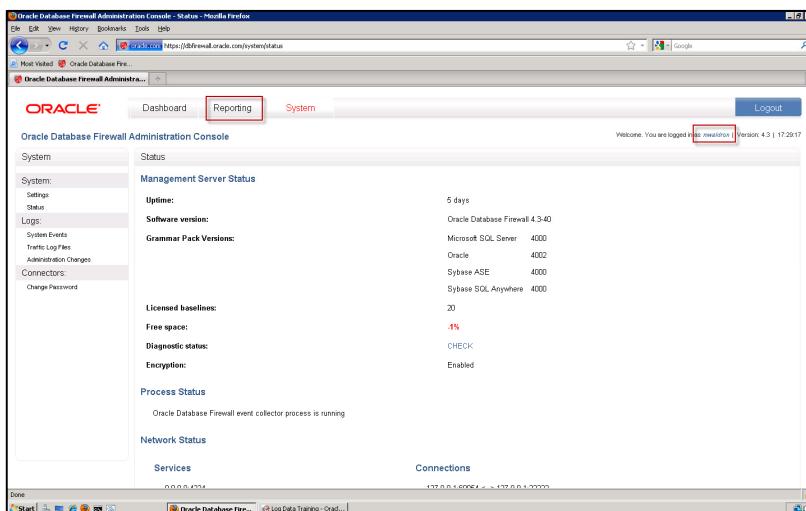
Login	First name	Last name	Role	Created	Suspended
admin	Admin	Account	System Administrator	2010-09-09 13:07:08	no
kzeng	Ken	Zeng	System Administrator	2010-09-10 17:06:37	no
mwaldron	Mark	Waldron	View-only User	2010-09-15 17:28:09	no

Copyright © 2009, 2010 Oracle and/or its affiliates. All Rights Reserved.

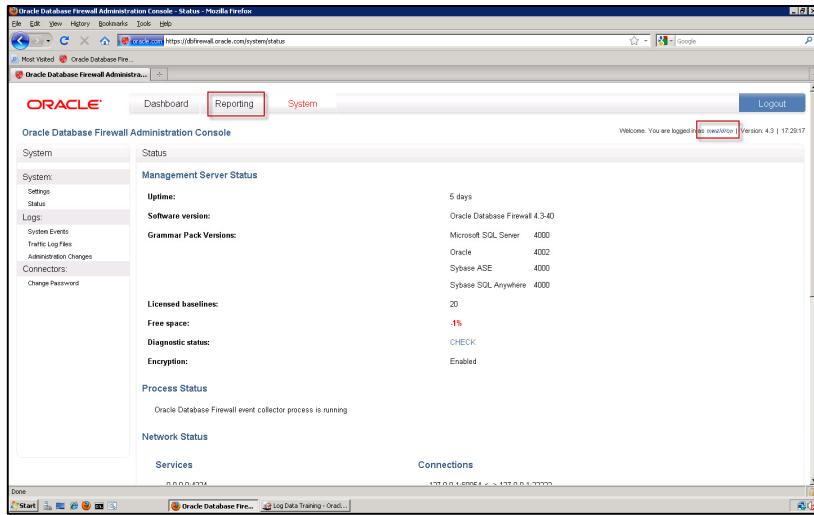
4. Enter the login credentials you just created – username ‘mwaldron’ and password ‘oracle1’, then click ‘Login’.



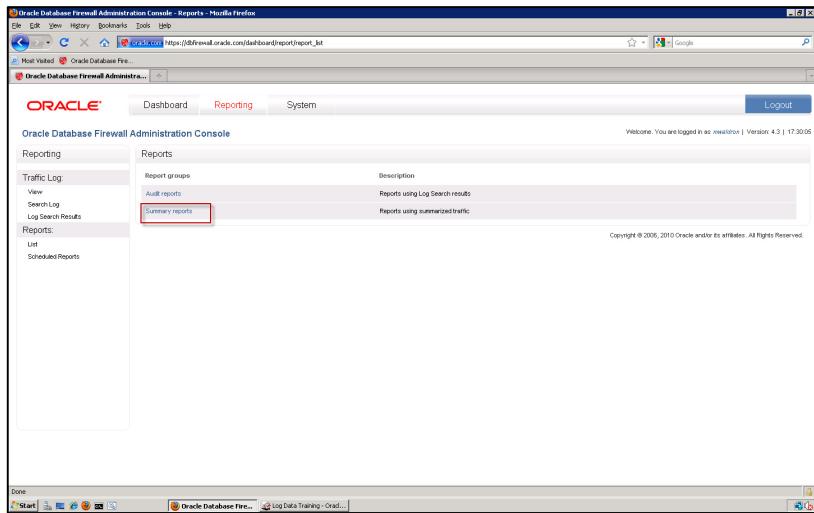
5. Notice that your welcome message now shows the use as ‘mwaldron’.



6. Navigate to the reports, by clicking on the ‘Reporting’ tab.



7. We will start by viewing a report showing all DB access attempts grouped by database user. There are dozens of reports provided by default in DBFW. The reports can be customized to meet your specific requirements. We will review a few reports to help you build comfort with the selection. Click on the ‘Summary Reports’ link.



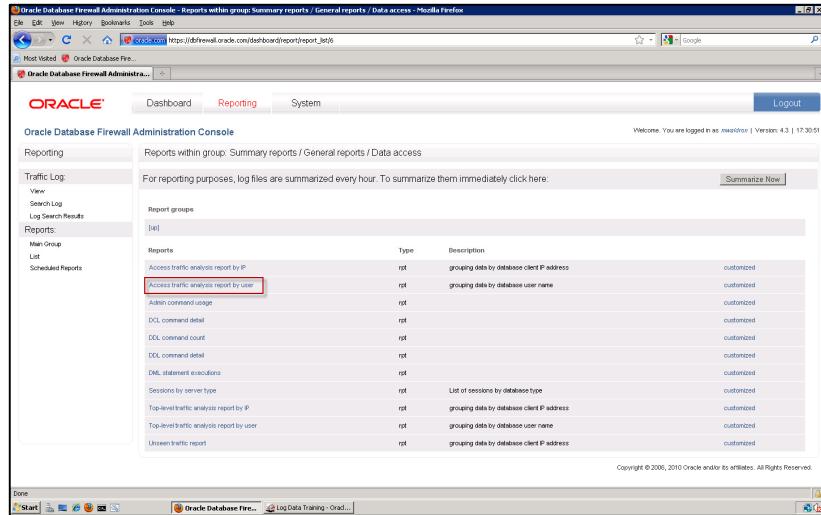
8. Notice that you can also see report groups for some common regulations. DBFW includes reports supporting HIPPA, GLBA, SOX and other regulations. Click on the '**General Reports**' link.

The screenshot shows the Oracle Database Firewall Administration Console interface. The main menu bar includes File, Edit, View, History, Bookmarks, Task, Help, and a browser address bar showing https://dbfrewall.oracle.com/dashboard/report/report_id/4. The top navigation bar has tabs for Oracle, Dashboard, Reporting, and System, with a Logout button. The Reporting section is active, displaying 'Reports within group: Summary reports'. Under 'Report groups', there is a list of categories: General reports (highlighted with a red box), User's own reports, UPA reports, GLBA reports, HIPAA reports, PCI reports, Performance reports, SOX reports, and Summary reports. A 'Summarize Now' button is located at the top right of this list. Below the report groups, there is a 'Reports' section with a 'Type' column. At the bottom of the page, a note says 'No reports available in this group'. The footer includes a copyright notice: 'Copyright © 2006, 2010 Oracle and/or its affiliates. All Rights Reserved.'

9. Click on '**Data Access**'. This report group will show information for protected databases from an access perspective.

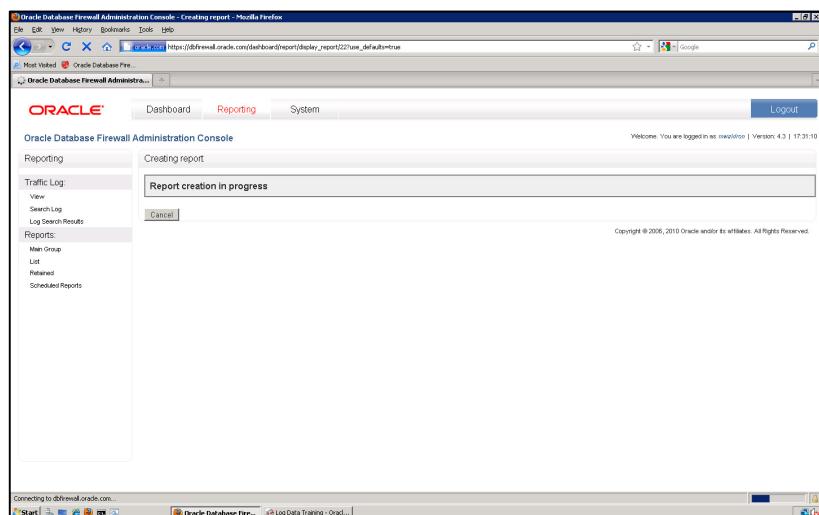
This screenshot is similar to the previous one but shows a different report group. The 'Data access' link under 'Report groups' is highlighted with a red box. The rest of the interface is identical to the previous screenshot, including the top navigation bar, the 'Reports within group: General reports' section, and the 'Reports' section below it. The footer also remains the same.

10. Click on the 'Access traffic analysis report by user'. This will generate a PDF report showing our access information.



The screenshot shows the Oracle Database Firewall Administration Console interface. The main menu on the left includes 'Reporting', 'Traffic Log', and 'Reports'. Under 'Reports', there are several options like 'Main Group', 'List', 'Retained', and 'Scheduled Reports'. A list of reports is displayed with columns for 'Reports', 'Type', and 'Description'. One report, 'Access traffic analysis report by user', is highlighted with a red box. The 'Description' column for this report states 'grouping data by database client IP address' and 'customized'. Other reports listed include 'Admin command usage', 'DDL command detail', 'DCL command count', 'DDX command detail', 'DML statement executions', 'Sessions by server type', 'Top-level traffic analysis report by IP', 'Top-level traffic analysis report by user', and 'Unknown traffic report'. The status bar at the bottom right indicates 'Copyright © 2006, 2010 Oracle and/or its affiliates. All Rights Reserved.'

11. The report is generated dynamically. Do not refresh the browser window, DBFW will automatically refresh until the report is created.



The screenshot shows the Oracle Database Firewall Administration Console interface. The main menu on the left includes 'Reporting', 'Traffic Log', and 'Reports'. A central dialog box displays the message 'Report creation in progress'. The status bar at the bottom right indicates 'Connecting to dbfwall.oracle.com...'.

12. You can view the PDF report embedded in the browser, or download it to be distributed/viewed offline. Please take a moment to review the report. Notice that it is showing a summary of the access attempts and is also categorized by DML, DDL etc.

The screenshot shows the Oracle Database Firewall Administration Console interface. The main title bar reads "Oracle Database Firewall Administration Console - Report: Access traffic analysis report by user - Mozilla Firefox". The URL in the address bar is https://dbfrewall.oracle.com/dashboard/report/display_report?22/use_defaults=true. The left sidebar has sections for Dashboard, Reporting, System, Traffic Log, View, Search Log, Log Search Results, Reports, Main Group, List, Related, and Scheduled Reports. The Reporting section is selected. The main content area displays a report titled "Access traffic analysis report" for user "SOE". It includes a note about the report being created using only traffic that has been summarized so far. The report shows a chart titled "Number of statements for each access type" with two data series: "data manipulation read" (blue line) and "data manipulation write" (orange line). The chart shows a sharp decline from over 6000 statements to around 2000 statements. The legend indicates the blue line represents "data manipulation read" and the orange line represents "data manipulation write". The bottom status bar shows "Log Data Training - Oracle Database Firewall" and "Oracle Database Firewall Admin".

Scroll down to read the remainder of the report.

The screenshot shows the Oracle Database Firewall Administration Console interface. The title bar reads "Oracle Database Firewall Administration Console - Report: Access traffic analysis report by user - Mozilla Firefox". The address bar shows the URL: "https://dbfirewall.oracle.com/dashboard/report/display_report?222;use_defaults=true".

The left sidebar includes navigation links: "Most Visited", "Oracle Database Firewalls", "Main Group", "List", "Retained", and "Scheduled Reports".

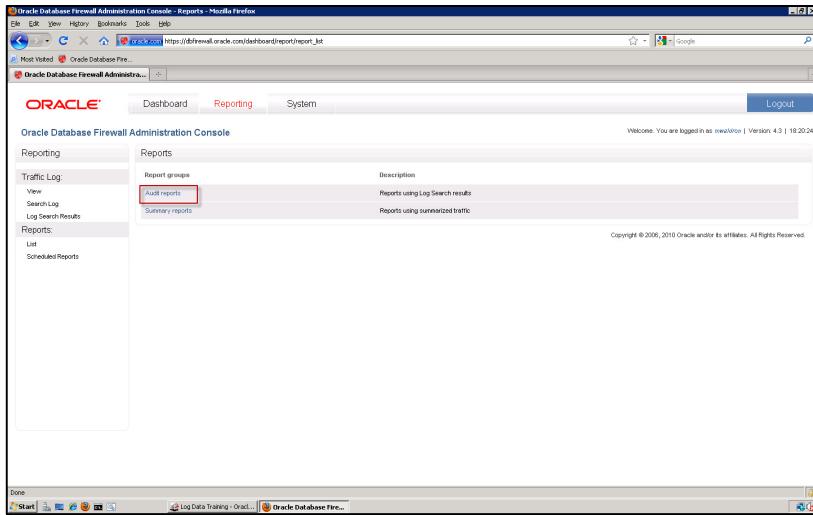
The main content area displays a table of access traffic analysis. The columns are: cluster, sample statement, client IP, %, and count. The table contains several rows of log entries, such as:

cluster	sample statement	client IP	%	count
Statement type: data manipulation	682501107 insert into t_order_items(ORDER_ID,LINE_ITEM_ID, PRODUCT_ID, UNIT_PRICE, QUANTITY) values (0 ,0 ,0 ,0 ,0)	67.202.22.15	9.38%	1,718
	2147483647 update inventory set quantity_on_hand = 0 where product_id = 0 and warehouse_id = 0	67.202.22.15	9.38%	1,718
-1391135981 insert into t_order_items(ORDER_ID, ORDER_DATE, CUSTOMER_ID) values (0 ,0 ,0)	67.202.22.15	2.70%	495	
2147483647 update orders set order_manager = 0, order_status = 0 , where order_id = 0	67.202.22.15	2.70%	495	
2147483647 update /*+ index/orders, order_pk */ orders set order_status = 0 where order_id = 0	67.202.22.15	1.70%	312	
7767619059 insert into customers (customer_id, cust_first_name, cust_last_name, cust_street_address, cust_town, credit_limit, cust_email, account_mgr_id) values (0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0)	67.202.22.15	1.59%	292	
Statement type: data manipulation	2147483647 select product_id, product_name, product_description, category_id, weight, class, supplier_id, product_status, list_price, min_price, catalog_url from product_information where category_id = 0	67.202.22.15	17.62%	3,228

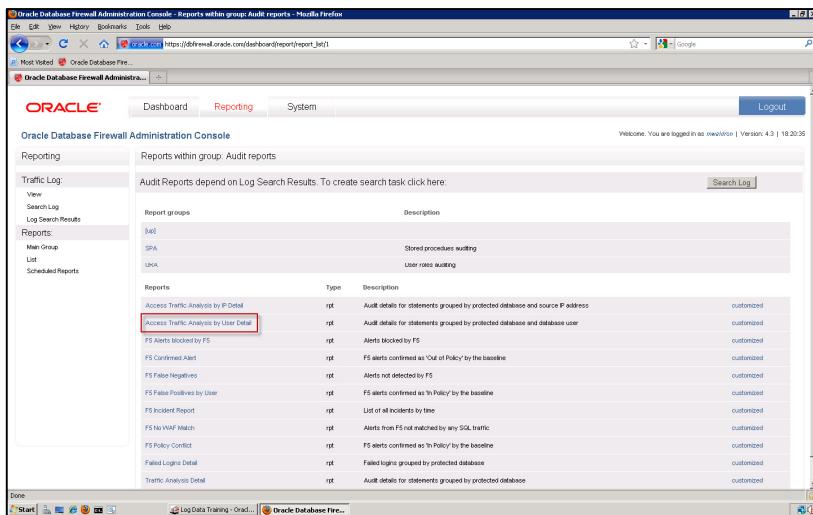
A footer message at the bottom right says "printed on 15/09/2010 at 17:38".

Once you are finished with this report, click on the '**Main Group**' in the '**Reports**' section. This will return you to the top level reporting group.

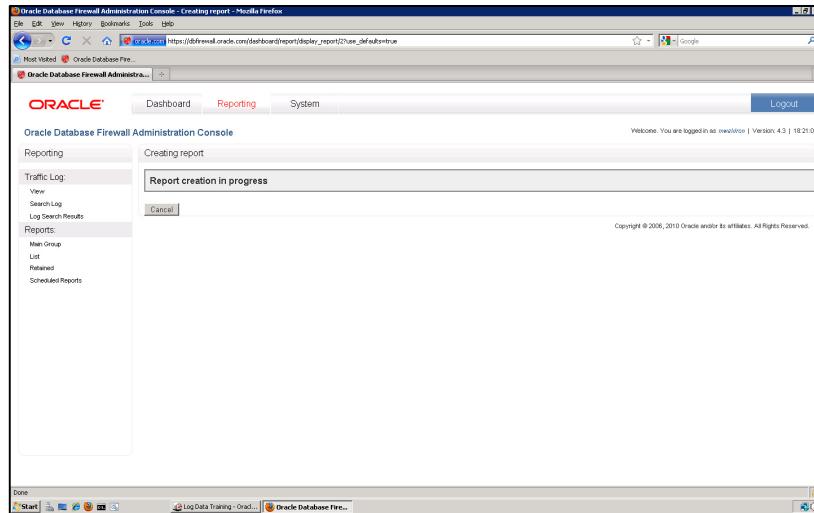
13. Click on the ‘Audit Reports’ link.



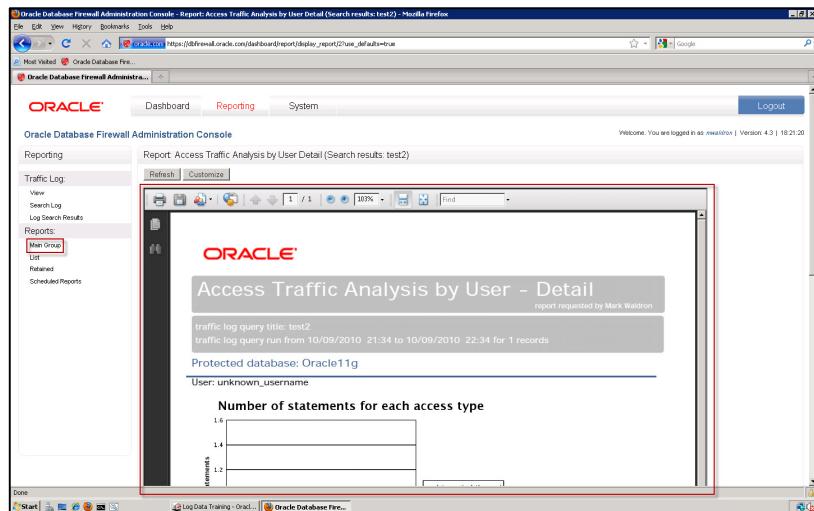
14. Click on the ‘Access Traffic Analysis by User Detail’ link. This report will show us monitored data grouped by database and database user.



The report will take a few moments to dynamically create.

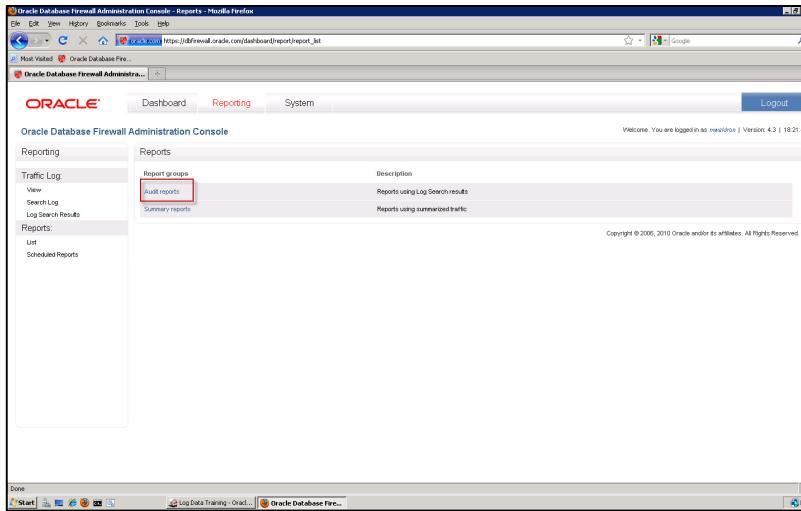


15. Please take some time to review the report.

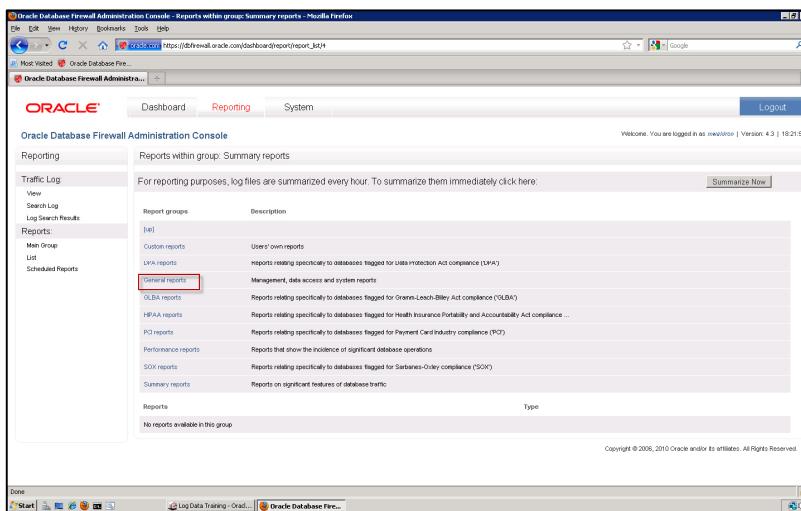


Once you are finished with this report, click on the '**Main Group**' in the '**Reports**' section. This will return you to the top level reporting group.

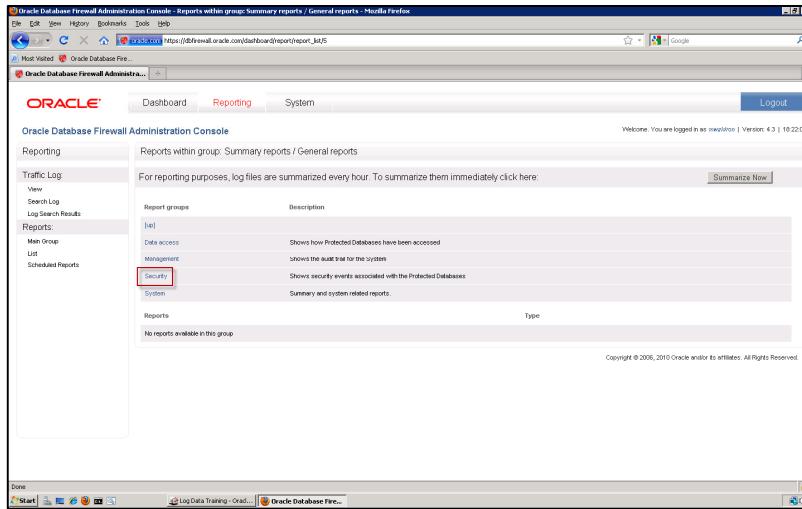
16. Click on the ‘Audit Reports’ link.



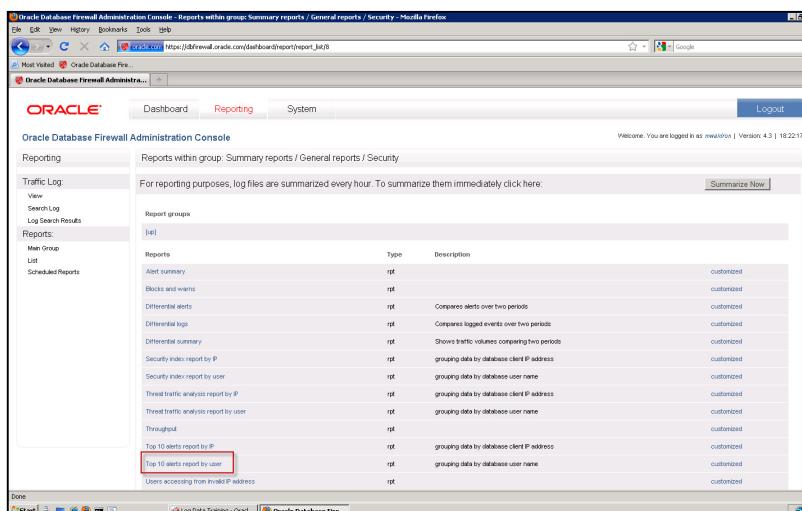
17. Click on the ‘General Reports’ link.

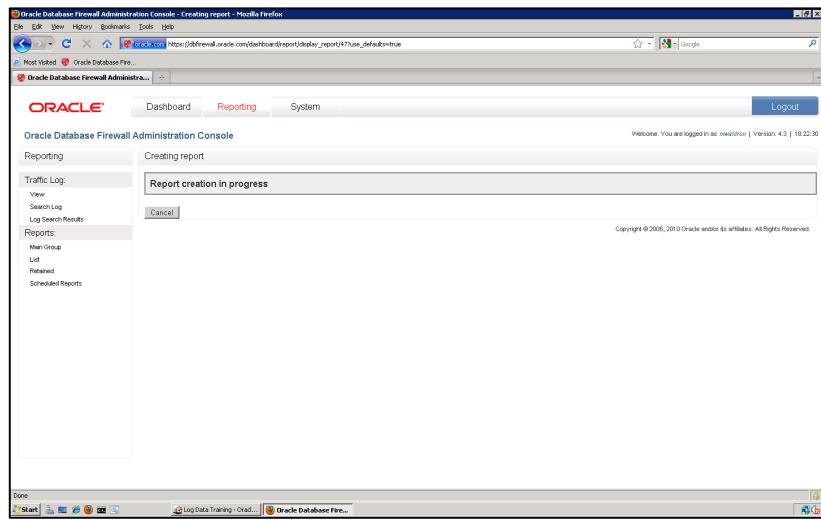


18. Click on the ‘Security’ link.

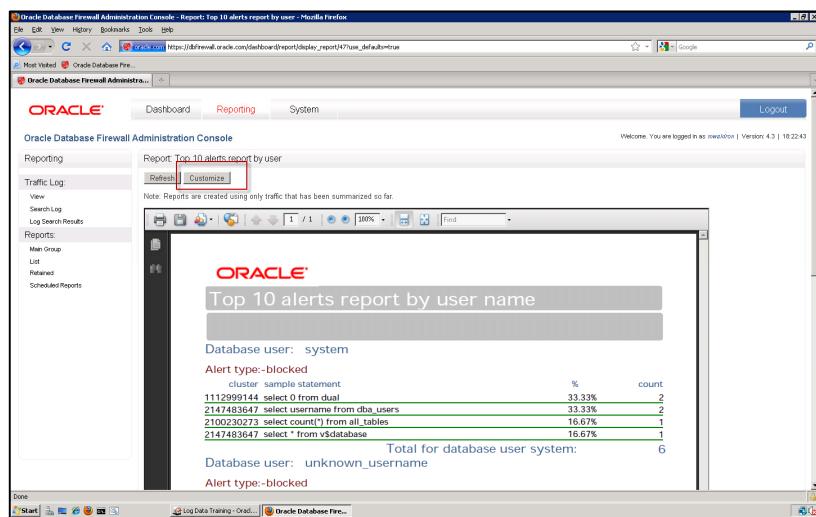


19. Click on the ‘Top 10 alerts report by user’ link.





20. Finally, let's quickly review the customization functionality for a report.
Click on the 'Customize' button at the top of the screen.



Database user: system
Alert type: blocked

cluster sample statement % count

1112999144 select 0 from dual 33.33% 2

2147483647 select username from dba.users 33.33% 2

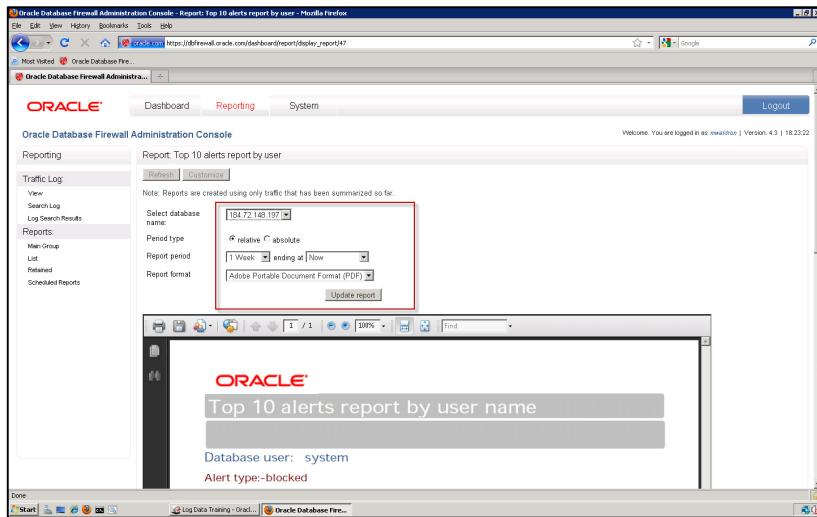
2100230273 select count(*) from all_tables 16.67% 1

2147483647 select * from v\$database 16.67% 1

Total for database user system: 6

Database user: unknown_username
Alert type: blocked

21. Notice that you can specify which database you are viewing data for, what time period the report is for and what output format the report will be produced in.



D. Summary

You accomplished the following in this lab exercise:

1. Created a Read-only Report User
2. Generated and review available reports
3. Gained experience with report groups and hierarchies

LAB EXERCISE 04 – ORACLE DATABASE FIREWALL – USING WHITELISTS TO PREVENT SQL INJECTION ATTACKS

INTRODUCTION

In this Lab, we will review some of the techniques used to build and maintain an Application white list of acceptable SQL queries. We will use as our existing white list containing the statements in ‘Oracle Workload’. Will we add a new application, called ‘My HR’, and add this as a new white list under its own profile.

The User Acceptance Testing environment is often the best place to update a white list to support a new application, or to support new functionality for an existing application. When a UAT environment is available, the Database Firewall policies can be left in strict blocking mode in production during the update process. When updating a policy based on activity observed directly in the production environment, it will be necessary to tailor the policy to allow unseen statements. This can be done by the following three ways:

1. Alter the existing policy to allow (Pass or Warn) all Unseen statements
2. Add the new application user/ip address to the Exception policy
3. Change the Enforcement Point from DPE (policy enforcement) mode to DAM (activity monitoring).

In our example, we will take the ‘**Lab2_block_with_dba_exception.dna**’ policy and alter it to allow all activity by the new application user ‘demoapps’. (In a test environment, we could simply change the Anomaly Default Rule to pass rather than block).

A. Lab Scenarios and Objectives

In this lab exercise you will accomplish the following:

1. *Allow Unseen Activity for a New Application to develop the whitelist*
2. *Update the Whitelist Policy with new activity*

B. Setup and Preparation

- Completion of LAB EXERCISE 01 – ORACLE DATABASE FIREWALL ENFORCEMENT POINTS TO MONITOR AND PROTECT DATABASES
- Completion LAB EXERCISE 02 – ORACLE DATABASE FIREWALL – USE THE TRAFFIC ANALYZER TO CONFIGURE POLICIES AND BLOCK UNAUTHORIZED TRAFFIC

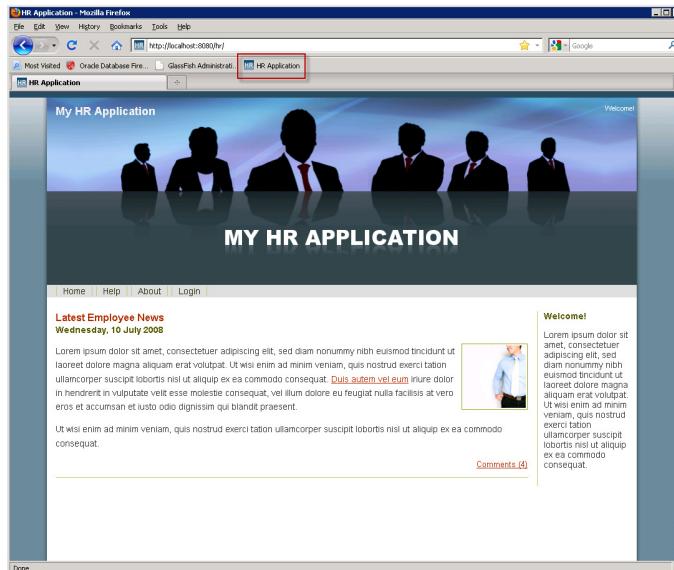
1. On the Windows Desktop, find the icon **Shortcut to startserv.bat** and click on it to start the web-based application infrastructure.



When prompted with the following dialog box, click on the **Open** button.



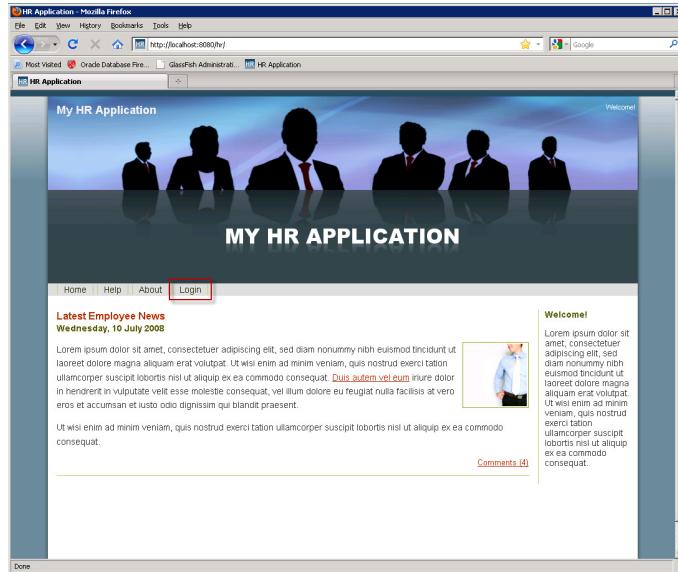
2. After a couple of minutes and there is no more activity in the DOS command window, minimize the DOS window, open up the browser and click on the shortcut to open the web-based HR Application. You can alternatively access the application by the URL:
<http://localhost:8080/hr>.



The environment is properly setup for you to begin the lab.

C. USING WHITELISTS TO PREVENT SQL INJECTION ATTACKS

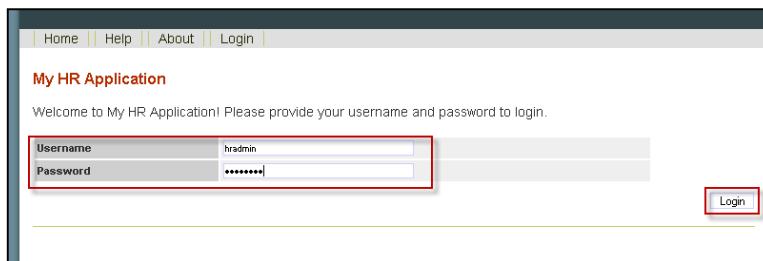
1. Return to the browser and click on the **Login** button (located in the middle of the screen) to attempt and login to the application.



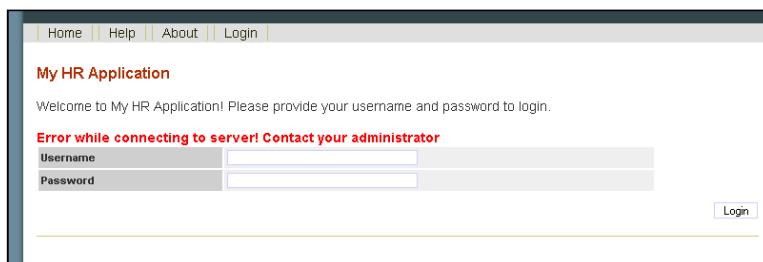
Use the following credentials and click on the **Login** button.

Username: **hadmin**

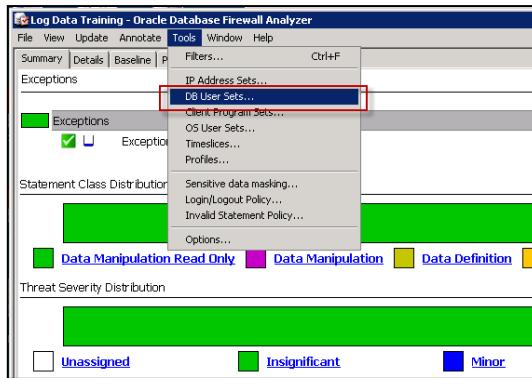
Password: **abcd1234**



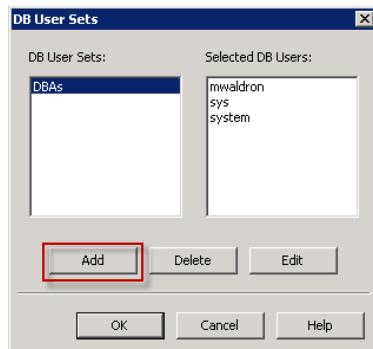
2. You will notice the error, '**Error while connecting to server! Contact your administrator**'. Since we have an active policy in enforcement from our previous lab exercise, all of the application traffic and attempts for the application user to connect to the database is considered 'Unseen' traffic—thus, it is being blocked as expected.



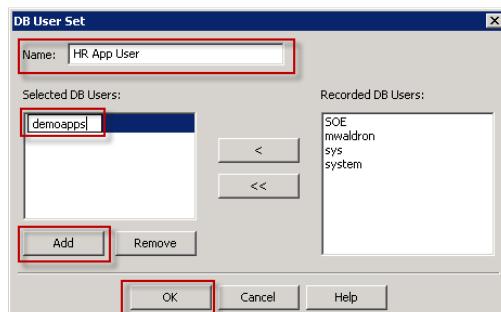
- We will now refine our policy to make an exception allowing SQL traffic from the HR application—the **DEMOAPPS** user. Return to the DBFW Analyzer. Once there click on the ‘Tools’ menu. Then click on the ‘DB User Sets’ link.



- In the same way we defined users in a previous exercise to specify users as our administrators; we will allow the HR application’s user **DEMOAPPS**. In the **DB User Sets** dialog, click on the **Add** button.



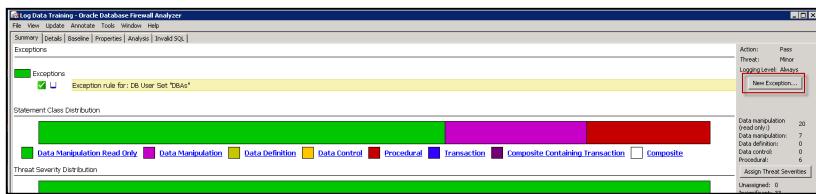
Provide the descriptive name, ‘**HR App User**’, and add the ‘**demoapps**’ user. Since we have not collected any traffic with the **DEMOAPPS** user, the user will not be available in the Recorded DB Users list.



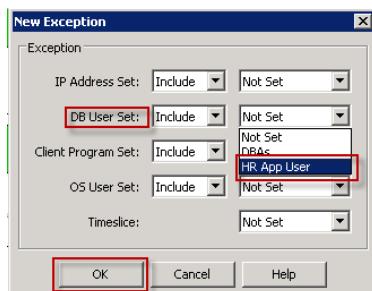
Once the **HR App User** has been added, click on the **OK** button to continue.



5. Create a new exception for the DB User Set named HR App User. Click on the **New Exception** button highlighted.



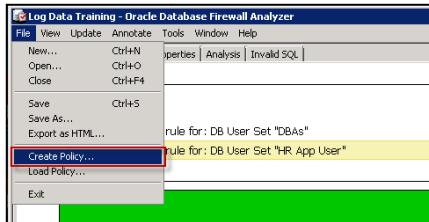
Choose to Include the **HR App User** in the New Exception dialog and click on the **OK** button.



Review the newly added Exception.

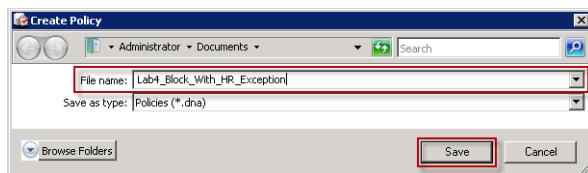


6. We can now save a new baseline and implement it in DBFW. Click on the ‘File’ menu, followed by ‘Create Policy’.



Enter the following file name: **Lab4_Block_With_HR_Exception**

Click ‘Save’.

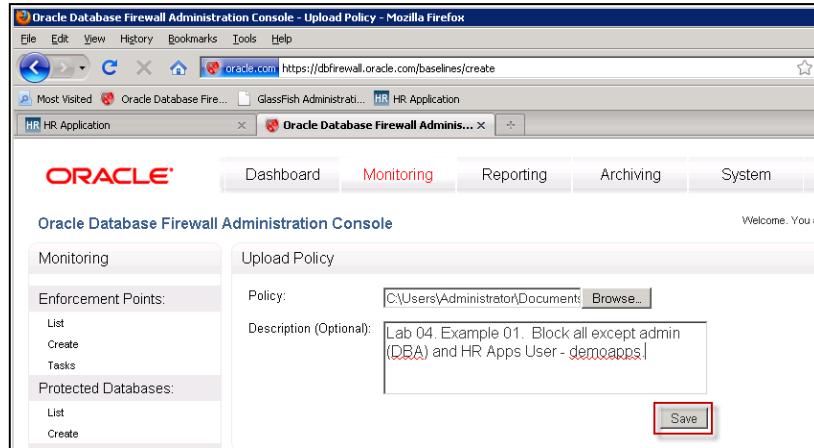


7. Open the DBFW Administration Console. Logout of Reporting User ‘**mwaldron**’ and login as the administrator ‘**kzeng**’ (password ‘oracle1’). Click on the ‘Monitoring’ tab. Then click on the ‘List’ link in the ‘Policies’ section. You will see that the previous uploaded baseline policy is in place. Click on the ‘Upload’ link in the ‘Policies’ section.

Policy	Description
log&mask.dna	Log all statements for offline analysis without masking data (Note: if this policy is applied, it can use significant amounts of storage for the logged data. Sensitive information may be logged if you select this policy)
log.dna	Log all statements for offline analysis (Note: if this policy is applied, it can use significant amounts of storage for the logged data)
logexample.dna	Log a sample of statements for offline analysis (Note: if this policy is applied, although it will store less statements than logging all statements, it can still use significant amounts of storage for the logged data)
passall.dna	Pass all statements
unique-&mask.dna	Log examples of statements for offline analysis covering each distinct source of traffic without masking data (Note: if this policy is applied, although it will store less statements than logging all statements, it can still use significant amounts of storage for the logged data. Sensitive information may be logged if you select this policy)
unique.dna	Log examples of statements for offline analysis covering each distinct source of traffic (Note: if this policy is applied, although it will store less statements than logging all statements, it can still use significant amounts of storage for the logged data)

Policy	Created	Database Type	Description	Edit	Delete
lab2_bloc_unseen.dna	2011-04-16	Oracle	Lab 02. Example 01: Block All Unseen Statements	Edit	Delete
lab2_bloc_with_dba_exception.dna	2011-04-16	Oracle	Lab 02. Example 02: Block all except admin (DBA)	Edit	Delete

Select the **Lab4_Block_With_HR_Exception.dna** file from the Administrator Documents folder. Then enter a meaningful description, we've entered '**Lab 04. Example 01. Block all except admin (DBA) and HR Apps User - demoapps**'. Click 'Save'.



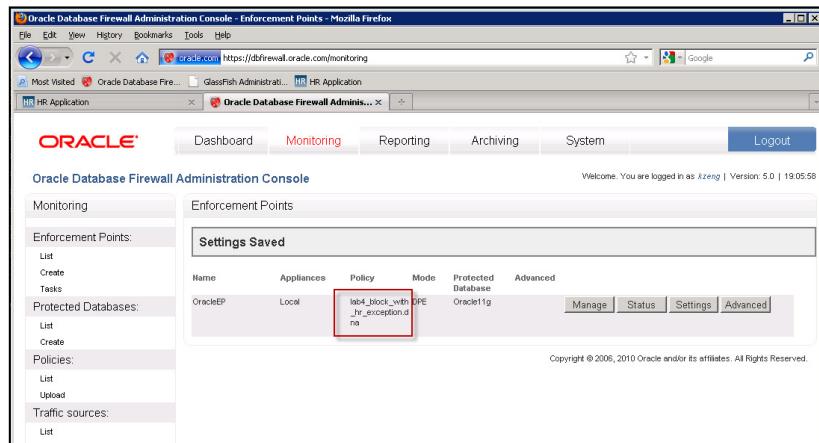
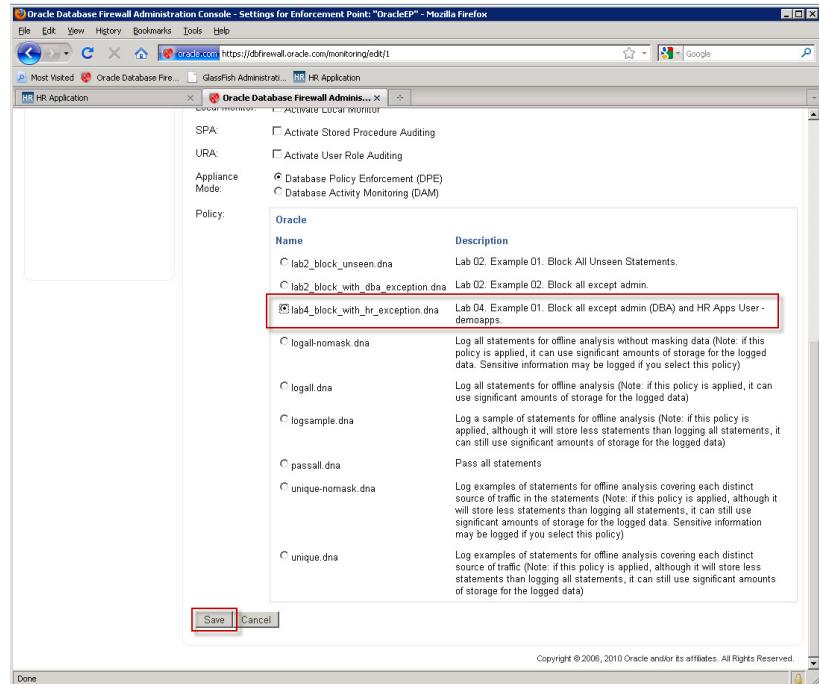
You will see that the new policy is uploaded and available.

Uploaded Policies:				
Policy	Created	Database Type	Description	
lab2_block_unseen.dna	2011-02-09	Oracle	Lab 02. Example 01. Block All Unseen Statements.	Edit Delete
lab2_block_with_db_exception.dna	2011-02-09	Oracle	Lab 02. Example 02. Block all except admin.	Edit Delete
lab4_block_with_hr_exception.dna	2011-02-09	Oracle	Lab 04. Example 01. Block all except admin (DBA) and HR Apps User - demoapps.	Edit Delete

8. Navigate to the '**Monitoring**' tab, then to the '**List**' link in the '**Enforcement Points**' section. Click on the '**Settings**' button. We will be enforcing our new baseline policy by configuring our '**OracleEP**' to use it.



Select the new '**Lab4_Block_With_HR_Exception.dna**' baseline policy radio button from the list. Then, click '**Save**'.

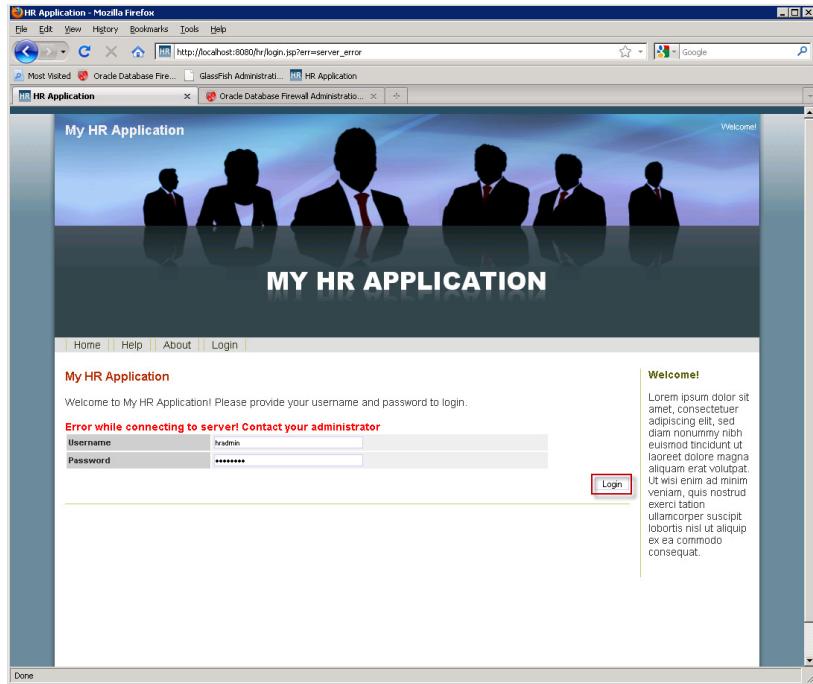


9. Return to the browser and attempt to login to the HR Application.

Use the following credentials and click on the **Login** button.

Username: **hadmin**

Password: **abcd1234**



Notice that the change to our enforced policy now allows the application to work properly. Now that the **DEMOAPPS** user can connect successfully to the database, the application can do its work.



Even if you type in the password incorrectly, you will see that there is a different error message generated this time.

The screenshot shows a web browser window with the title 'My HR Application'. At the top, there are links for Home, Help, About, and Login. Below the title, a message reads: 'Welcome to My HR Application! Please provide your username and password to login.' A red error message 'Could not log user! Please verify username and password' is displayed above the login form. The form has two input fields: 'Username' containing 'mmalfoy' and 'Password' containing '*****'. To the right of the password field is a 'Login' button.

10. We will now perform some ‘standard’ activities in the HR application which we will then add to our existing whitelist of permitted activities. Make a note of the time—as you will want to train on logged data based on the timing of this current session.

Logout of the current user and login using the username **mmalfoy** and password **oracle**

The screenshot shows a web browser window with the title 'My HR Application'. At the top, there are links for Home, Help, About, and Login. Below the title, a message reads: 'Welcome to My HR Application! Please provide your username and password to login.' The 'Username' field contains 'mmalfoy' and the 'Password' field contains '*****'. Both fields are highlighted with a red border. To the right of the password field is a 'Login' button, which is also highlighted with a red border.

(If the login fails, check your HOSTS file to ensure cloud.oracle.com is set to the database firewall IP address)

Click on **Search Employees** link on the right under the Employees section.

The screenshot shows a web browser window with the title 'My HR Application'. At the top, there are links for Home, Help, About, Logout, and a search bar. Below the title, a section titled 'Latest Employee News' displays the date 'Wednesday, 10 July 2008' and a paragraph of placeholder text. On the right side, there is a sidebar with a profile picture of a man and a list of links under the heading 'Employees': 'Search Employees' (highlighted in yellow), 'New Employee', 'Absence And Attendance', 'Timesheets', and 'Vacation'. The 'Search Employees' link is enclosed in a red box.

Enter “%” in the Department field and click on the **Search** button.

The screenshot shows a search interface for employees. In the 'Department' field, the value '%Engineering%' is entered and highlighted with a red box. The 'Search' button is also highlighted with a red box. The results table shows several employees, all of whom belong to the 'Engineering' department, as indicated by the green status icon in the 'Organization' column.

HR ID	Full Name	Emp Type	Position	Manager	Cost Center	Department	Organization
5	Borst, Hugo	Full-Time				Engineering	Xellerate Users
2	Jansen, Henk	Full-Time	End-User			Engineering	Xellerate Users
4	Karelse, Karel	Full-Time				Engineering	Xellerate Users
6	Koelewijn, Frans	Full-Time	DBA	Jansen, Henk	101	Engineering	Xellerate Users
129	Opedijk, Jeen	Full-Time	Project Manager	Jansen, Henk	101	Engineering	Xellerate Users
47	Stok, Frank	Full-Time				Engineering	Xellerate Users
146	krabe, martin	Full-Time	DBA	Jansen, Henk	101	Engineering	Xellerate Users
168	kraas, frank	Full-Time	Administrator I	Jansen, Henk	101	Engineering	Xellerate Users
144	van Koelen, Frans	Full-Time	DBA	Jansen, Henk	101	Engineering	Xellerate Users

Notice that all records returned are for the Engineering department, since the application enforces row level security.

Click on the full name hyperlink [Borst, Hugo](#) to see his full details.

The screenshot shows the employee profile for 'Borst, Hugo'. The 'Full Name' field is highlighted with a red box. The 'Search Employees' link on the right is also highlighted with a red box.

HR ID	Full Name	Emp Type	Position	Manager	Cost Center	Department	Organization
5	Borst, Hugo	Full-Time				Engineering	Xellerate Users

Return to the Search page by clicking on the Search Employees link on the right.

The screenshot shows the main application homepage. The 'Search Employees' link in the top right corner is highlighted with a red box.

Search for all employees named Frank by entering the name in the First Name field and hitting enter.

The screenshot shows a search interface with fields for HR ID, Employee Type, First Name, Department, Active status, Position, Last Name, and Organization. The 'First Name' field contains 'Frank' and is highlighted with a red box. The 'Search' button is also highlighted with a red box. Below the search form is a table titled 'Search Result' showing two entries:

HR ID	Full Name	Emp Type	Position	Manager	Cost Center	Department	Organization
47	Stok, Frank	Full-Time				Engineering	Xellerate Users
166	kras, frank	Full-Time	Administrator I	Jansen, Henk	101	Engineering	Xellerate Users

Search for everyone in one department by entering 'engineering' in the Department field and hit Enter.

The screenshot shows a search interface with fields for HR ID, Employee Type, First Name, Department, Active status, Position, Last Name, and Organization. The 'Department' field contains 'engineering' and is highlighted with a red box. The 'Search' button is also highlighted with a red box. Below the search form is a table titled 'Search Result' showing seven entries:

HR ID	Full Name	Emp Type	Position	Manager	Cost Center	Department	Organization
5	Borst, Hugo	Full-Time				Engineering	Xellerate Users
2	Jansen, Henk	Full-Time	End-User			Engineering	Xellerate Users
4	Karelse, Karel	Full-Time				Engineering	Xellerate Users
6	Koolewijin, Frans	Full-Time	DBA	Jansen, Henk	101	Engineering	Xellerate Users
129	Opedijk, Jeen	Full-Time	Project Manager	Jansen, Henk	101	Engineering	Xellerate Users
47	Stok, Frank	Full-Time				Engineering	Xellerate Users

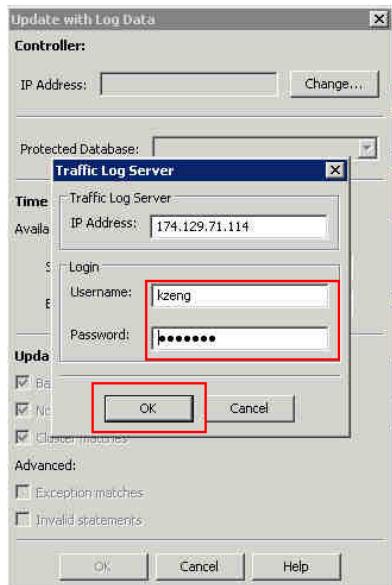
Click on Logout near the top left.

We have now created traffic logs of 'normal' and expected behavior in this application surrounding employee information searches. We will now go into the Database Firewall Analyzer to analyze those logs to update our whitelist.

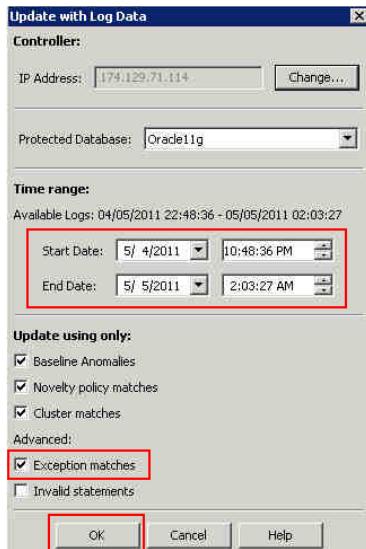
11. Switch to your analyzer window, click on the Update menu and choose Update with Log Data.



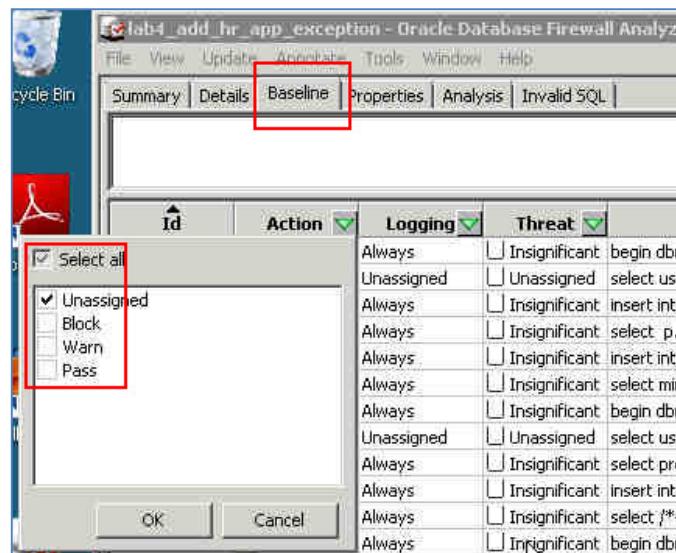
12. Enter the username and password you created on the Database Firewall web GUI. Your database firewall IP address should already appear correctly.



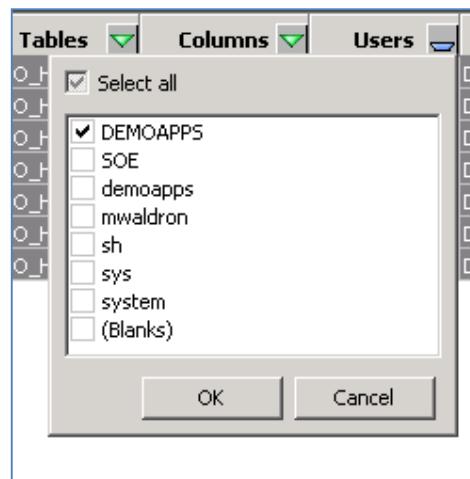
13. We configured our policy to log the HR App traffic as an Exception. We therefore want to update our policy using Exception matches. To do this, ensure that ONLY the Exception matches box is ticked under the Update using only menu. Also verify that the start/end times include the period we were updating our traffic logs.



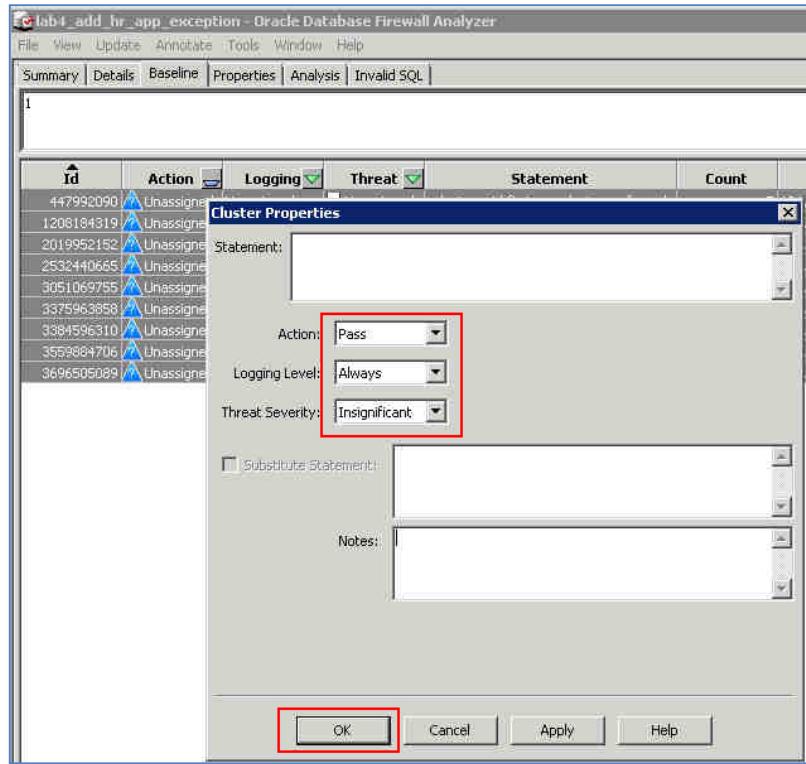
14. Click on the **Baseline** tab. To see only new clusters, filter for only 'unassigned' statements by clicking on the triangle next to Action, unticking Select all, and ticking **Unassigned**. Click **OK**.



15. Filter to see only 'unassigned' for the Users DEMOAPPS.



16. To simplify the policy update, use Ctrl-A to select all visible clusters, right click and select **Properties**. Set the policy to be Pass, Log Always, and Insignificant. Click 'OK'.

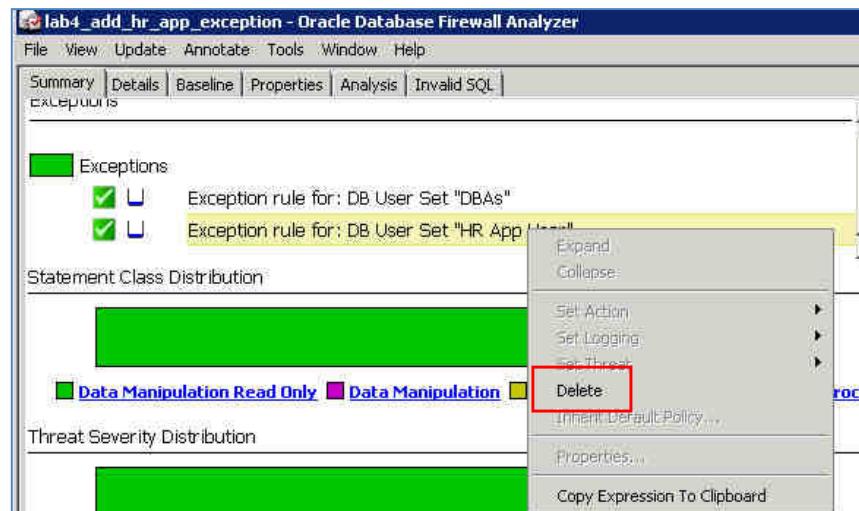


17. We have now incrementally added and defined more expected and 'acceptable' traffic to our baseline.

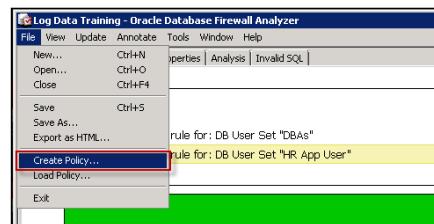
ID	Action	Logging	Threat	Statement	Count	IP Addresses
447992090	Pass	Always	Insignificant	select userid,firstname,lastname from dba...	7	184.73.24.206
1208184319	Pass	Always	Insignificant	select username from dba_users	2	184.73.24.206
2019952152	Pass	Always	Insignificant	select a.userid, a.firstname, a.lastname,	2	184.73.24.206
2532440665	Pass	Always	Insignificant	select a.userid, a.firstname, a.lastname,	2	184.73.24.206
3051069755	Pass	Always	Insignificant	select roleid from demo_hr_roles where l	1	184.73.24.206
3375963858	Pass	Always	Insignificant	select a.userid, a.firstname, a.lastname,	1	184.73.24.206
3384596310	Pass	Always	Insignificant	select * from v\$database	1	184.73.24.206
3559884706	Pass	Always	Insignificant	select a.userid, a.firstname, a.lastname,	1	184.73.24.206
3696505089	Pass	Always	Insignificant	select name from dual	1	184.73.24.206

18. Our current exception policy allows all activity by the HR App User—DEMOAPPS. This policy is far too broad. Now that we have defined all acceptable transaction behavior, we can remove the exception for that user.

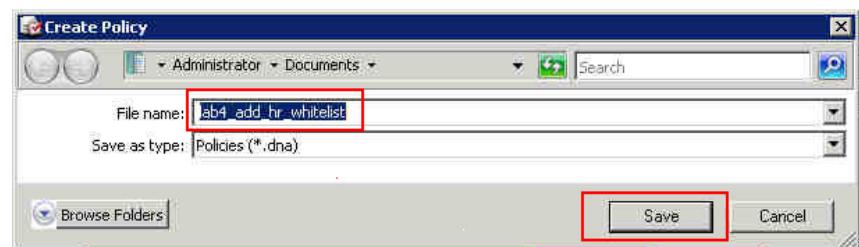
Return to the summary screen and right-click on the exception rule for the HR App User, and select delete. Confirm the deletion.



19. Create a new baseline, including our new transactions by selecting ‘File’, then ‘Create Policy’.



20. Enter the following file name: **Lab4_Block_With_HR_Whitelist**



21. Before we implement this newly created policy based upon ‘expected’ traffic after the training process, let’s first demonstrate what we’re looking to prevent—vulnerabilities from SQL Injection attacks. Return to the HR Application by clicking on the shortcut in Firefox, and log in as ‘mmalfoy/oracle’.

22. As we did previously, to demonstrate that the ‘mmalfoy’ user can only view employee records in the Engineering department, click on ‘Search Employees’ link, and enter ‘%’ in the Department field. Note that only employees in Engineering are listed. Then, search for all employees with the first name ‘Frank’. Only two employees will be listed.

HR ID	Full Name	Emp Type	Position	Manager	Cost Center	Department	Organization
47	Stolk, Frank	Full-Time				Engineering	Xellerate Users
166	Kras, Frank	Full-Time	Administrator I	Jansen, Henk	101	Engineering	Xellerate Users

23. Now we will demonstrate the vulnerability of this poorly written application against a SQL Injection attack. Remember, many applications have been written (and not updated due to cost or effort) and continue to be written with the necessary coding standards to avoid (but not fully eliminate) these vulnerabilities.

Change the First Name search condition from just ‘Frank’ to the following:

Frank’ OR 1=1 --

(Note – there is no space between the two dashes)

The right quote ends the Firstname field, ‘**OR 1=1**’ enters a condition that will be true for every record in the table, and the double-dash ends the statement, turning any further conditions into a comment that will be ignored.

Since we have broken the application security by injecting our own SQL text into the application, employees from all departments are now listed, including the Project Director, and even Larry Ellison (in his role as DBA!).

Search Employee							
HR ID		Active	— Choose a value —				
Employee Type	— Choose a value —		Position				
First Name	Frank' or 1=1—		Last Name				
Department			Organization				
Search Result							
HR ID	Full Name	Emp Type	Position	Manager	Cost Center	Department	Organization
244	<u>Y..</u>	Full-Time	Administrator I	<u>Jansen, Henk</u>	101	Corporate	Xellerate Users
245	<u>x..</u>	Full-Time	Administrator I	<u>Jansen, Henk</u>	101	Corporate	Xellerate Users
190	<u>ellison, larry</u>	Full-Time	DBA	<u>Jansen, Henk</u>	101	Corporate	Xellerate Users
188	<u>janssen, jim</u>	Full-Time	DBA	<u>Jansen, Henk</u>	101	Corporate	Xellerate Users
186	<u>kaptijn, joop</u>	Full-Time	DBA	<u>Jansen, Henk</u>	101	Corporate	Xellerate Users
166	<u>kras, frank</u>	Full-Time	Administrator I	<u>Jansen, Henk</u>	101	Engineering	Xellerate Users
152	<u>krabe, patrick</u>	Full-Time	DBA	<u>Jansen, Henk</u>	101	Corporate	Xellerate Users
148	<u>krabe, henk</u>	Full-Time	DBA	<u>Jansen, Henk</u>	101	Corporate	Xellerate Users
126	<u>Schonis, Peter</u>	Full-Time	DBA	<u>Jansen, Henk</u>	101	Corporate	Xellerate Users
65	<u>Forde, John</u>	Full-Time	DBA	<u>Jansen, Henk</u>	101	Corporate	Xellerate Users
24	<u>Kennis, Ferrie</u>	Full-Time	Documentation Clerk	<u>Jansen, Henk</u>	101	Sales	Xellerate Users
6	<u>Koolewijn, Frans</u>	Full-Time	DBA	<u>Jansen, Henk</u>	101	Engineering	Xellerate Users
205	<u>Hoersma, Klaas</u>	Full-Time	DBA	<u>Jansen, Henk</u>	101	Corporate	Xellerate Users
150	<u>krabe, klaas</u>	Full-Time	DBA	<u>Jansen, Henk</u>	101	Corporate	Xellerate Users
146	<u>krabe, martin</u>	Full-Time	DBA	<u>Jansen, Henk</u>	101	Engineering	Xellerate Users
127	<u>Veek, Klaas</u>	Full-Time	DBA	<u>Jansen, Henk</u>	101	Corporate	Xellerate Users
108	<u>Franken, Klaas</u>	Full-Time	DBA	<u>Jansen, Henk</u>	101	Corporate	Xellerate Users
107	<u>Degraaf, Bas</u>	Full-Time	DBA	<u>Jansen, Henk</u>	101	Corporate	Xellerate Users

Log out of the HR Application.

24. We will now implement our whitelist for the HR application to prevent this kind of attack—and the thousands of SQL Injection variations possible. Open the DBFW Administration Console. Ensure that you are still logged in as the administrator ‘kzeng’ (password ‘oracle1’). Click on the ‘Monitoring’ tab. Then click on the ‘List’ link in the ‘Policies’ section. You will see that the previous uploaded baseline policy is in place. Click on the ‘Upload’ link in the ‘Policies’ section.

Policy	Created	Database Type	Description
lab2_block_unseen.dna	2011-04-16	Oracle	Lab 02, Example 01. Block All Unseen Statements
lab2_block_with_dbadmin_exception.dna	2011-04-16	Oracle	Lab 02, Example 02. Block all except admin (DBA)

25. Select the **Lab4_Block_With_HR_Whitelist.dna** file from the Administrator Documents folder. Then enter a meaningful description such as ‘Add HR Whitelist’. Click ‘Save’.

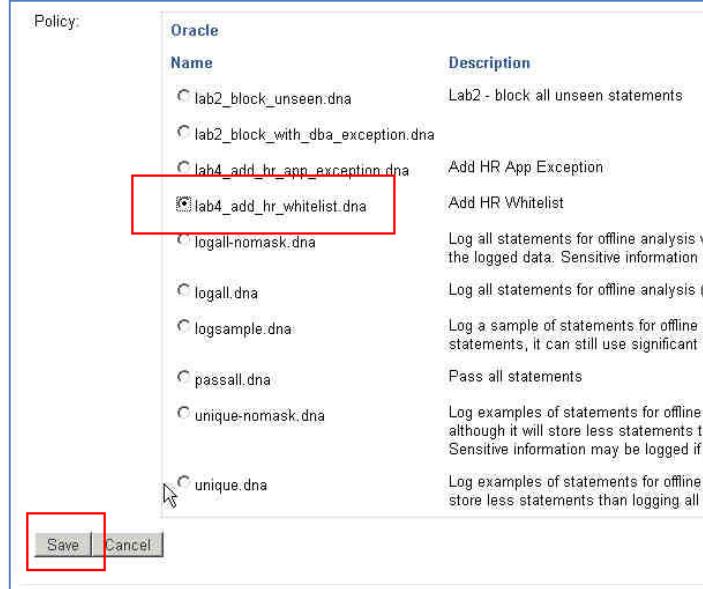
You will see that the new policy is uploaded and available.

The screenshot shows the Oracle Database Firewall Administration Console interface. On the left, there's a sidebar with 'Protected Databases' and 'Policies' sections. Under 'Policies', there are four entries: 'legal.dna', 'logsample.dna', 'passall.dna', and two entries under 'unique-' (with 'nomask.dna' and 'unique.dna'). Below this is a section titled 'Uploaded Policies' which lists four policies: 'lab2_block_unseen.dna', 'lab2_block_with_dbe_exception.dna', 'lab4_add_hr_app_exception.dna', and 'lab4_add_hr_whitelist.dna'. The last entry, 'lab4_add_hr_whitelist.dna', is highlighted with a red border.

26. Navigate to the ‘Monitoring’ tab, then to the ‘List’ link in the ‘Enforcement Points’ section. Click on the ‘Settings’ button. We will be enforcing our new baseline policy by configuring our ‘OracleEP’ to use it.

This screenshot shows the Oracle Database Firewall Administration Console with the 'Monitoring' tab selected. In the left sidebar, 'Enforcement Points' is expanded, and the 'List' link is highlighted with a red box. In the main content area, there's a table titled 'Enforcement Points' with one row for 'OracleEP'. The 'Settings' button in the table header is also highlighted with a red box.

27. Select the new '**Lab4_Block_With_HR_Whitelist.dna**' baseline policy radio button from the list. Then, click '**Save**'.



You will see that you have enabled your new whitelist policy.

The screenshot shows the Oracle Database Firewall Administration Console. The top navigation bar includes 'Dashboard', 'Monitoring', 'Reporting', 'Archiving', 'System', and 'Logout'. The main area is titled 'Oracle Database Firewall Administration Console' and shows 'Welcome. You are logged in as kzeng | Version: 5.0'. On the left, a sidebar has 'Monitoring' selected. In the center, under 'Enforcement Points', there is a table with one row. The row shows 'Name' as 'OracleEP', 'Appliances' as 'Local', 'Policy' as 'lab4_add_hr_w DPE whitelist.dna' (highlighted with a red box), 'Mode' as 'DPE', 'Protected Database' as 'Oracle11g', and 'Advanced' settings. Below the table are 'Manage', 'Status', and 'Settings' buttons.

28. Return to the HR Application, logging in as **mmalfoy/oracle**, and selecting '**Employee Search**' again. Attempt to execute the search which previously resulted in the exposure of all records in the employees table, by entering

Frank' OR 1=1 --
(Again – there is no space between the two dashes)

Since this query is not part of our whitelist, the entire statement fails.

This demonstrates how Oracle Database Firewall can protect against SQL injection attacks without tedious, time consuming application development, validation, changes and testing.

D. Summary

You accomplished the following in this lab exercise:

1. Used exceptions to classify new traffic patterns
2. Incrementally modify your baseline based on that traffic
3. Updated your white list to prevent SQL Injection attacks

LAB CONFIGURATION – ENTERPRISE MANAGER 12c DATA MASKING

OVERVIEW

For these lab exercises, the following infrastructure components need to be started and available for your use.

- **Database: Database DB06**
- **Here is a summary of the users.**
 - MASKING_ADMIN – Data Masking Administrator

We will step through the simplified version of starting the necessary infrastructure components using the desktop.

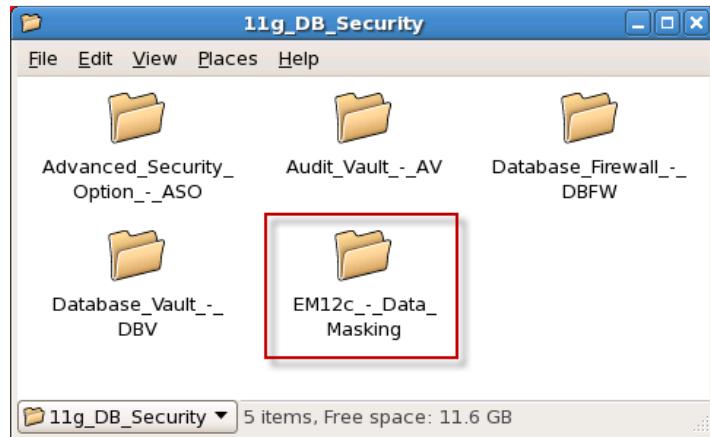
1. On the desktop, navigate to the **Labs** folder, double-click and open the contents.



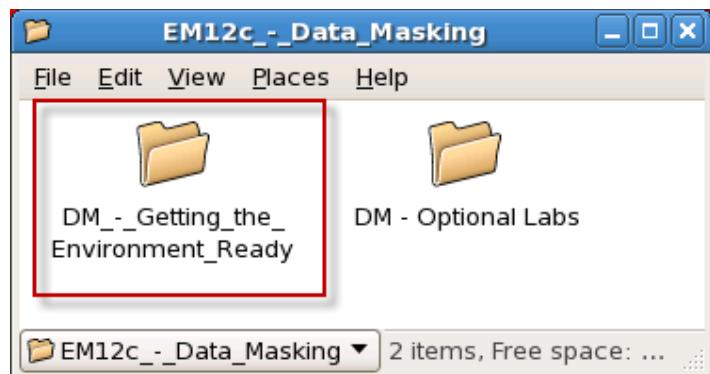
2. Select the folder, **11g_DB_Security**.



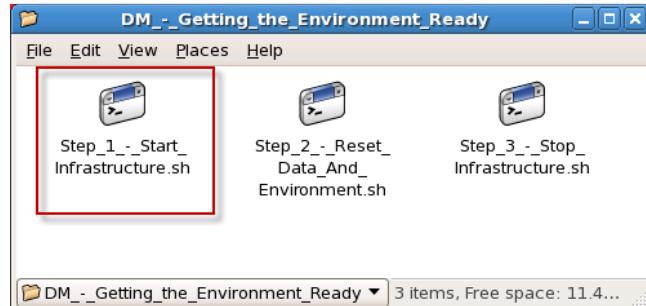
3. Select the folder, **EM12c_-_Data_Masking**.



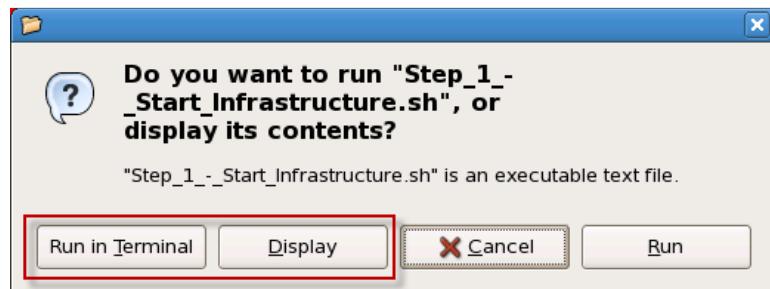
4. Within the **EM12c_-_Data_Masking**, you can access all of the Lab folders. Select '**DM_-_Getting_the_Environment_Ready**'.



5. Select '**Step_1_Start_Infrastructure.sh**'. This script will start the database and initialize the environment used in this lab.



When you double click on the script, the OS will prompt you to specify what you want to do with the shell script.



You can select either '**Run in Terminal**' to execute the script, or '**Display**' if you want to see more detail on what is actually being executed. You obviously must run each script to execute the labs, but feel free to display them if you wish to see what is being executed.

You will notice that output files will be saved in the folder after the script executes. You may review this output, as well.

Unless otherwise indicated, the windows will close when the script has completed. Please wait for each script to complete before executing the next script.

6. Select '**Step_2_-_Reset_Data_And_Environment.sh**'. This script will reset the data in the image so you can successfully run through these exercises. At any time after you've masked data, you can return back to this script to reset the data.
7. You are ready move forward with the Data Masking labs. Enjoy!!

LAB EXERCISE 00 - ENTERPRISE MANAGER 12c

DATA MASKING OVERVIEW

INTRODUCTION

Oracle Data Masking pack for Enterprise Manager, part of Oracle's comprehensive portfolio of database security solutions, helps organizations comply with data privacy and protection mandates such as Sarbanes-Oxley, Payment Card Industry (PCI) Data Security Standard (DSS), Health Insurance Portability and Accountability Act (HIPAA), as well as numerous laws that restrict the use of actual customer data. With Oracle Data Masking, sensitive information such as credit card or social security numbers can be replaced with realistic values, allowing production data to be safely used for development, testing, or sharing with out-source or off-shore partners for other nonproduction purposes. Oracle Data Masking uses a library of templates and format rules, consistently transforming data in order to maintain referential integrity for applications.

Data masking (also known as data scrambling and data anonymization,) is the process of replacing sensitive information copied from production databases to test or non-production databases with realistic, but scrubbed, data based on masking rules. Data masking is ideal for virtually any situation when confidential or regulated data needs to be shared with other non-production users; for instance, internal users such as application developers, or external business partners, like offshore testing companies or suppliers and customers. These non-production users need to access some of the original data, but do not need to see every column of every table, especially when the information is protected by government regulations.

Data masking allows organizations to generate realistic and fully functional data with similar characteristics as the original data to replace sensitive or confidential information. This contrasts with encryption or Virtual Private Database, which simply hides data, and the original data can be retrieved with the appropriate access or key. With data masking, the original sensitive data cannot be retrieved or accessed. Names, addresses, phone numbers, and credit card details are examples of data that require protection of the information content from inappropriate visibility. Live production database environments contain valuable and confidential data — access to this information is tightly controlled. However, each production system usually has replicated development copies, and the controls on such test environments are less stringent. This greatly increases the risks that the data might be used inappropriately. Data masking can modify sensitive database records so that they remain usable, but contain no confidential or personally identifiable information. Yet, the masked test data resembles the original in appearance to ensure the integrity of the application.

A. Lab Scenarios and Objectives

In our fictitious company, CashBankTrust is currently evaluating masking and deidentification requirements and challenges within their database environment (Test, Development, and/or Production).

Now that CashBankTrust is protecting their production data using the techniques described in the Advanced Security, Database Vault, Audit Vault and Database Firewall labs, it is imperative to maintain that same level of confidentiality and protection even when providing realistic test data to outside application developers and analysts. The

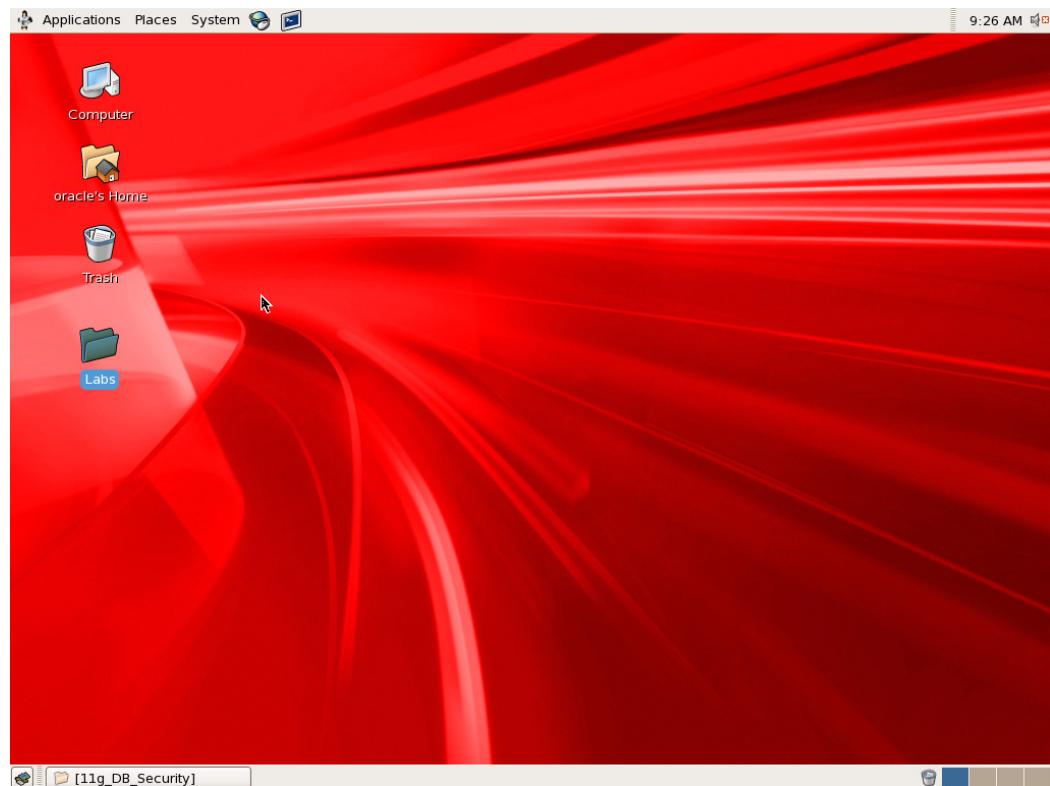
Data Masking Pack (an add-on to Oracle Enterprise Manager) allows masking of data, utilizing a variety of flexible masking options, while preserving referential integrity and the normal, measurable actions of applications.

The Data Masking labs that you will complete will demonstrate solutions specifically to the identified challenges below.

Product	Identified Challenges
Enterprise Manager – Data Masking	Personally Identifiable and sensitive data is being shared with parties that do not have a business need-to-know in development and testing groups.
	The use of operational databases containing personal information or any other sensitive information is being used for testing purposes. All identified sensitive details and content should be removed or modified beyond recognition before use.
	There is no established, documented procedure and enforcement of data cleansing standards in masking and cleansing of sensitive production data before distribution to development and QA environments.
	The steps and process necessary to provide development and QA environments with properly masked data are very time consuming, manual and inconsistent.

Let's get started.

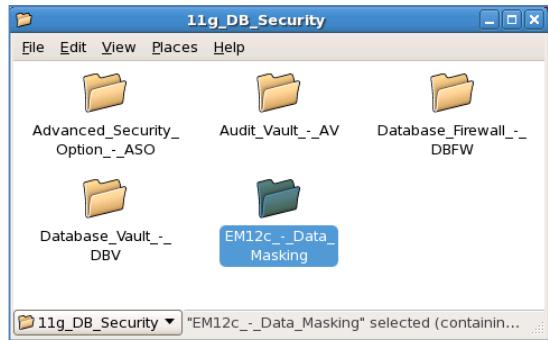
1. On the desktop, navigate to the **Labs** folder, double-click and open the contents.



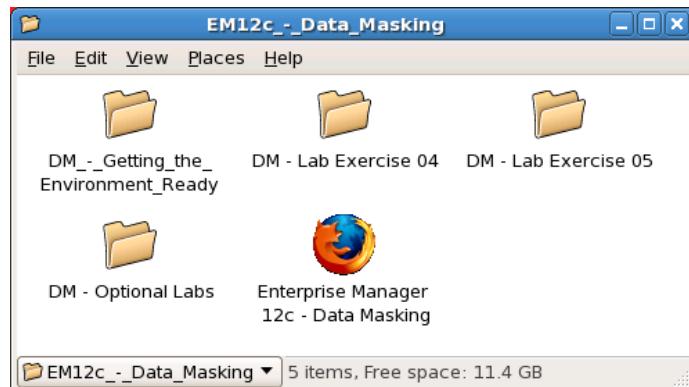
2. Select the folder, **11g_DB_Security**.



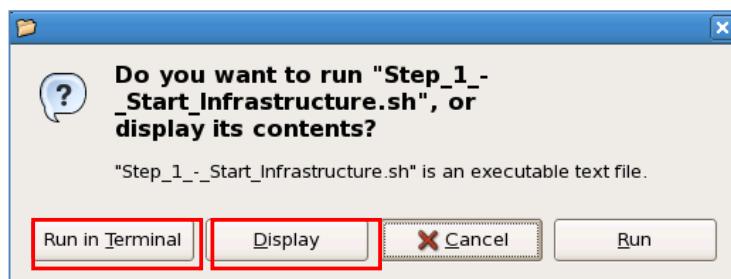
3. Select the folder, **EM12c_-_Data_Masking**.



4. Within the **EM12c_-_Data_Masking** folder, you can access all the Lab folders.



Throughout these exercises, when you double click on the script, the OS will prompt you to specify what you want to do with the shell script.



You can select either '**Run in Terminal**' to execute the script, or '**Display**' if you want to see more detail on what is actually being executed. You obviously must run each script to execute the labs, but feel free to display them if you wish to see what is being executed.

You will notice that output files will be saved in the folder after the script executes. You may review this output, as well.

Unless otherwise indicated, the windows will close when the script has completed. Please wait for each script to complete before executing the next script.

You are ready move forward with the Data Masking labs. Enjoy!!

LAB EXERCISE 01 – CREATING A DATA MODEL

Identified Challenge – Manual, Inconsistent and non-Standardized Test Data Management Operations

With the growth in the number of database applications, enterprises are faced with the challenge of provisioning non-production environments for application development and testing purposes. They cannot afford to incur the storage expenses of provisioning the same production data in their non-production databases, nor do they have the tools or the application knowledge to shrink production data to a right-sized development environment.

Faced with these challenges, organizations end up incurring high storage costs or end up reducing the productivity of their application development and testing staff by manually creating improper data sets that ultimately impact production application quality.

There is no consistent, single source of truth for schema metadata.

INTRODUCTION

Creating referentially intact data subsets of production data for modern enterprise applications is a daunting task to any organization even with highly skilled DBAs. These enterprise applications are incredibly complex spanning multiple schemas containing thousands of tables governed by myriad of business rules. The reason for the difficulty lies in the large and often complex data models that govern the relationships between the columns of the tables that sometimes span across different schemas. Oracle Test Data Management automatically discovers these relationships and can store them within an entity called the Application Data Model.

There are also pre-defined drivers to capture the data relationships for Oracle Applications such as Oracle Fusion Applications and Oracle E-Business Suite Applications directly from the application meta-data tables. What was a monumental task requiring DBAs to sift through the application code or meta-data to uncover these relationships can now be accomplished in minutes.

We've introduced a new capability, the Application Data Model, that allows businesses to catalog and storage application metadata such as sensitive data, primary key-foreign key relationships across enterprise databases in a single repository.

In this lab we will create an Application Data Model.

A. Overview

Application Data Models and Data Masking

In the context of Data Masking, the Application Data Model must first be created. We choose Applications and Schemas. Then, a job is run that will automatically discover Referential Relationships.

In this section, you will create an Application Data Model and see that primary key and foreign key relationships are captured here, even when the relationships span across schemas.

During this lab you will:

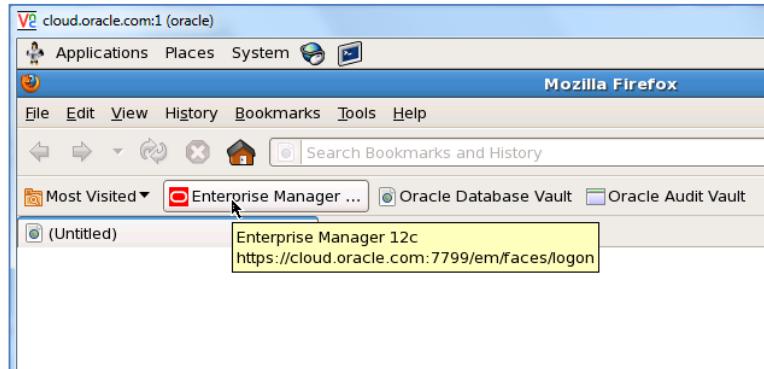
1. *Learn how to create an Application Data Model.*
2. *Drill into the Referential Relationships and see the primary key and foreign key relationships captured.*
3. *See how Referential Relationships not included in the Data Dictionary (typically those enforced in Application code) can be added manually.*

B. Setup & Preparation

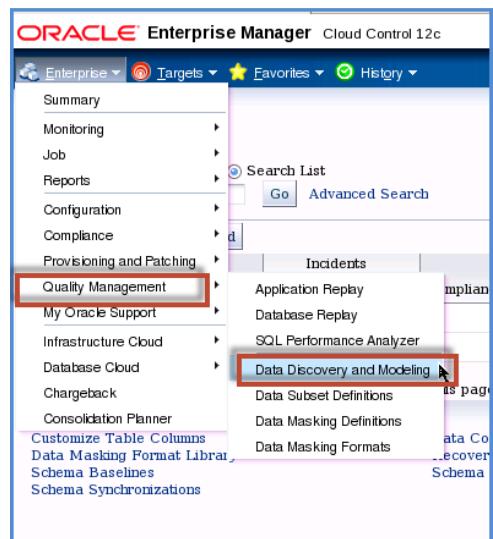
- You should have already completed **Step 1 – Start Infrastructure** and **Step 2 – Reset Data and Environment** before using this lab.

C. Creating an Application Data Model

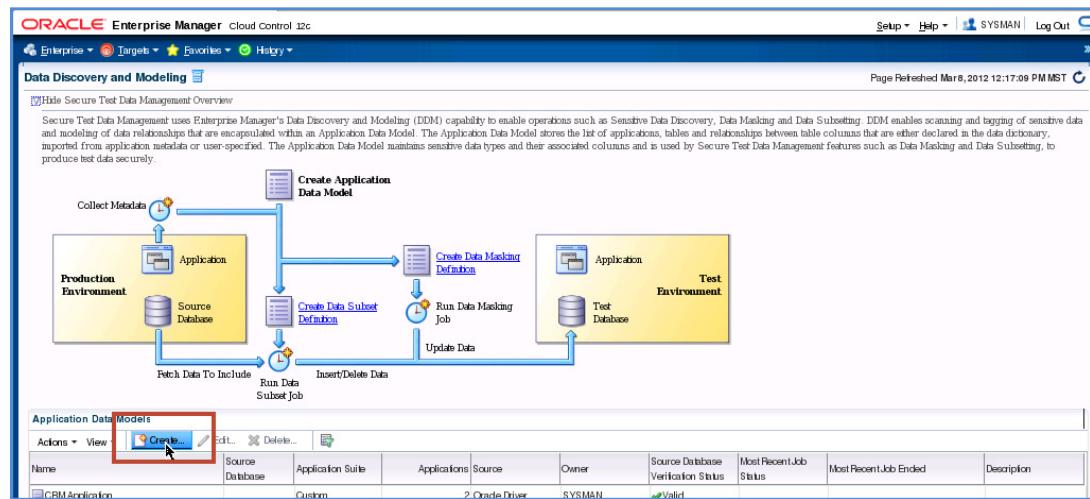
1. If you have not already done so, login to Enterprise Manager 12c as **sysman/oracle123** at the URL
<https://cloud.oracle.com:7799/em/faces/logon>

A screenshot of the Oracle Enterprise Manager 12c Login page. It features a "User Name" field containing "sysman" and a "Password" field with masked input. A "Login" button is at the bottom.

2. Navigate to the **Data Discovery and Modeling** page from the **Quality Management** submenu by selecting the menu **Enterprise** → sub-menu **Quality Management** → sub-menu **Data Discovery and Modeling**.



- Briefly review the **Secure Test Data Management** diagram to familiarize yourself with the process. In this exercise, we will Create Application Data Model from a Production Environment → Create a Data Masking Definition → Run a Data Masking Job into a Test Database within the Test Environment. We will keep the first example simple, but know you have the flexibility in this process to subset the data before masking. Let's begin by creating an Application Data Model henceforth referred to as ADM as the first step to masking data. Click on the '**Create**' button.



- We will create a new ADM called Employee Data on the DB06 database. Notice the options to create an ADM for packaged Oracle applications such as Oracle E-Business Suite and Oracle Fusion Applications. For this example, we will create a custom ADM to support our custom application. Choose the option type, '**Custom Application Suite**' and checkbox the option, '**Create One Application For Each Schema**' (default). Fill in the Name field with the entry '**Employee Data**' and Description field with the entry '**Employee Data in the HR and OE Schemas**'. Click the **Spyglass** icon to select a database.

Application Data Model Properties: General

* Name: Employee and Customer Data Model

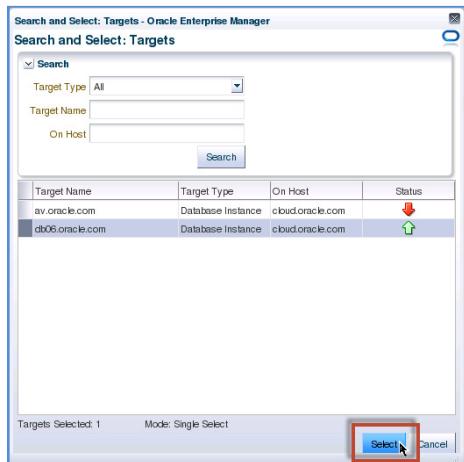
Description: Model of Sensitive data in HR and OE Schemas

* Source Database

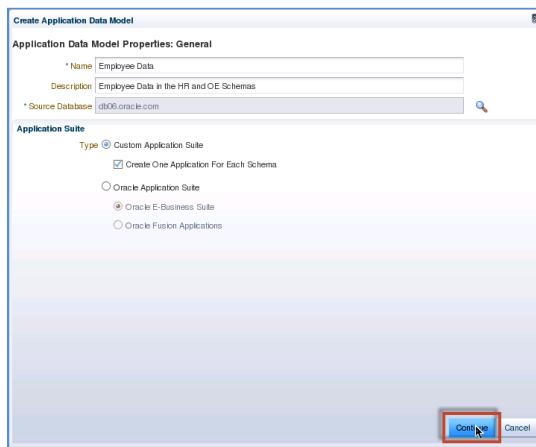
Type: Custom Application Suite Create One Application For Each Schema

Oracle Application Suite Oracle E-Business Suite Oracle Fusion Applications

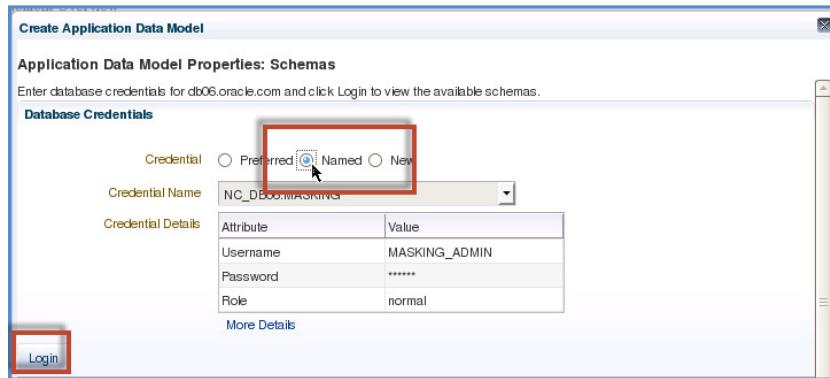
Select the **db06.oracle.com** target by highlighting the row and then click 'Select' button.



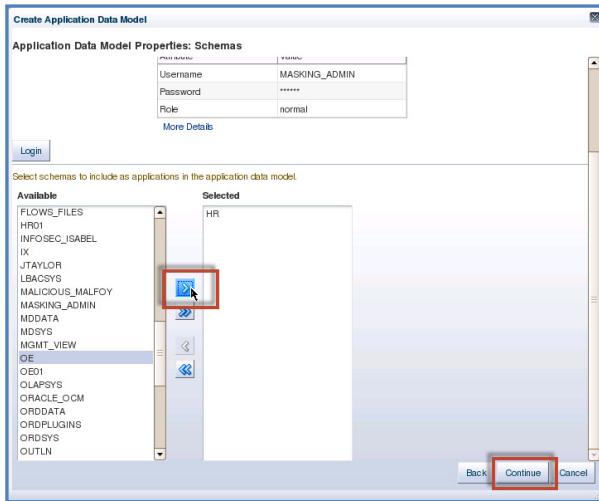
Click the 'Continue' button to continue creating the ADM.



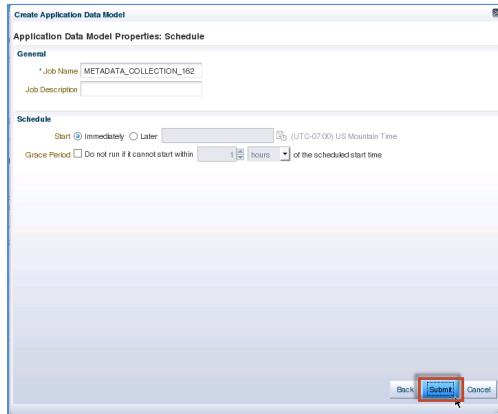
5. Now we need to login to db06 database. We will do using a Named Credential by selecting the ‘Named’ radio button and clicking on the ‘Login’ button. We have already created a user called **MASKING_ADMIN** in the database. Since this database is protected using Oracle Database Vault to restrict unnecessary access to the database, we needed to provide the necessary roles and privileges to carry out the functional task of data masking. To review these, see the script ‘**Used_to_Create_Masking_Admin_user.sql**’ in the Data Masking Readiness folder.



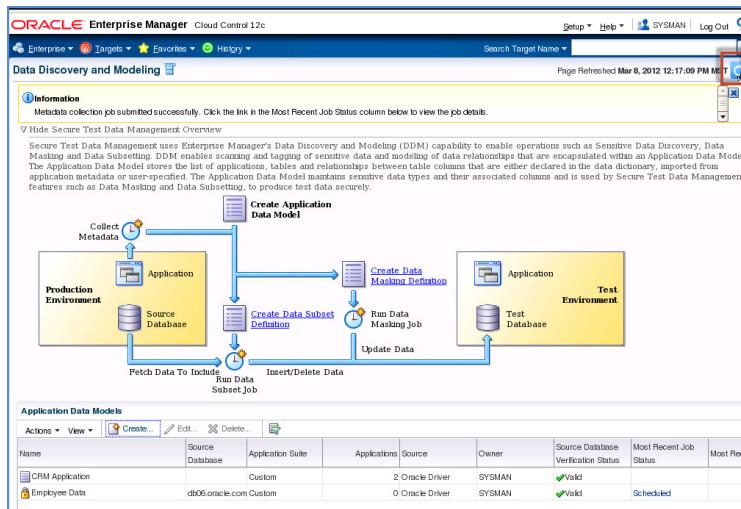
6. Select the **HR** and **OE** schema for the application data model. Select HR, then OE (shown below) or use **<Ctrl><Enter>** to select both simultaneously and then click the ‘Continue’ button.



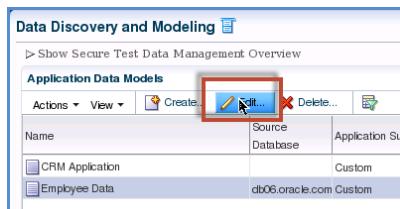
This action will create a job that will create the ADM for HR and OE schemas. Click the '**Submit**' button to schedule the job.



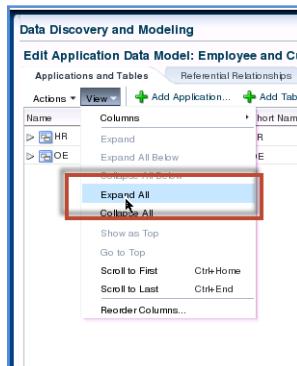
7. The job to collect the ADM has been submitted. Once the job completes, the HR OE ADM will no longer be in a locked, uneditable status. Check the status by refreshing this page. Move forward when the Most Recent Job Status of the Employee Data Model has '**Succeeded**'.



8. Review the information that was collected in the ADM as a part of the discovery. Highlight the '**Employee Data**' Application Data Model and click the '**Edit**' button. You may be asked for the database credentials. If so, select the '**Named**' radio button, choose the default credential using the **MASKING_ADMIN** username and click on the '**Continue**' button.



9. In the **Edit Application Data Model: Employee Data** screen, notice the applications for HR and OE have been created based on their respective HR and OE schema. Under the menu **View**, select the sub-menu **Expand All** to see the full list of tables associated with these applications. Review all the tables associated with the HR schema and with the OE schema grouped under their respective applications. We can see that the source of this information is from the data dictionary.



10. Now, let's view the referential relationships captured in the ADM. Click on the tab, '**Referential Relationships**'. Expand the entire list of applications (Menu **View** → Submenu **Expand All**) to examine the referential relationships under each application.

Name	Short Name	Schema	Table Type	Source
HR	HR	HR	Transaction Data	Dictionary
COUNTRIES		HR	Transaction Data	Dictionary
DEPARTMENTS		HR	Transaction Data	Dictionary
EMPLOYEES		HR	Transaction Data	Dictionary
JOBS		HR	Transaction Data	Dictionary
JOB_HISTORY		HR	Transaction Data	Dictionary
LOCATIONS		HR	Transaction Data	Dictionary
MANAGERS		HR	Transaction Data	Dictionary
MASK_ADDRESSES		HR	Transaction Data	Dictionary

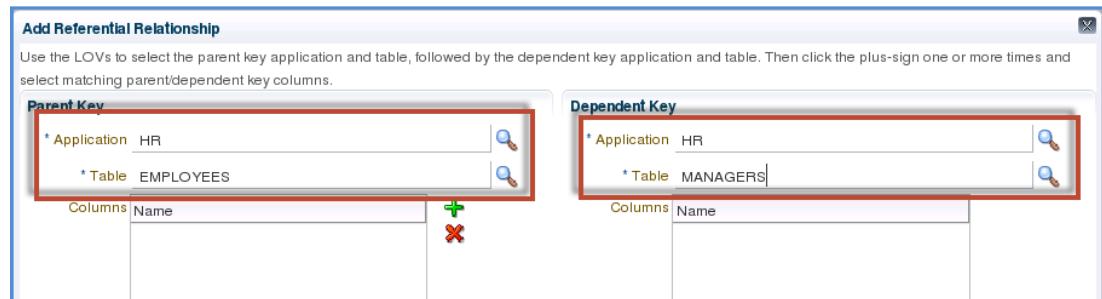
11. Notice the extensive set of primary key and foreign key relationships captured here, even when the relationships span across schema. For example, observe that **ACCOUNT_MGR_ID** in the **OE.CUSTOMERS** table and **SALES_REP_ID** in the **OE.ORDERS** table are foreign keys to the **EMPLOYEE_ID** column in the **HR.EMPLOYEES** table. Note that if we have to mask a column that is a primary key, we can automatically identify its foreign keys because we have captured it in the ADM.

Application	Table	Columns	Key Type	Source	Comment
HR	COUNTRIES	COUNTRY_ID	Parent	Dictionary	
HR	LOCATIONS	COUNTRY_ID	Dependent	Dictionary	
HR	DEPARTMENTS	DEPARTMENT_ID	Parent	Dictionary	
HR	EMPLOYEES	DEPARTMENT_ID	Dependent	Dictionary	
HR	JOB_HISTORY	DEPARTMENT_ID	Dependent	Dictionary	
HR	EMPLOYEES	EMPLOYEE_ID	Parent	Dictionary	
HR	DEPARTMENTS	MANAGER_ID	Dependent	Dictionary	
HR	EMPLOYEES	MANAGER_ID	Dependent	Dictionary	
OE	CUSTOMERS	ACCOUNT_MGR_ID	Dependent	Dictionary	
OE	ORDERS	SALES_REP_ID	Dependent	Dictionary	

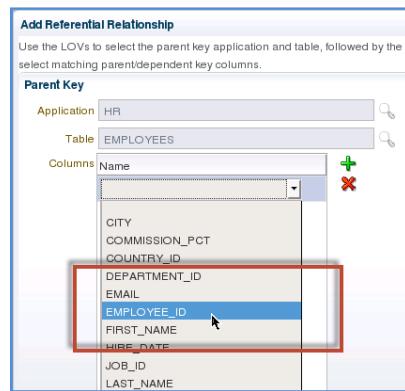
Now, in addition to the ones captured in the data dictionary, we can also capture these relationships that are maintained in the application, either in the meta-data or in the application logic. Applications, such as Peoplesoft, do not store the primary key-foreign key relationships in the database in order to be database independent; they are enforced in the application. In those cases, the Data Masking Pack provides administrators with the ability to register these relationships so that the columns in the related tables, e.g. **EMPLOYEE_ID**, **MGR_ID**, are masked identically using the same masking rules.

12. If the database manages the referential relationships, the ADM will automatically capture these. However, if these are managed by the application, you will need to define these manually. If it is necessary to define a Referential Relationships, click on the **Add** button provide the details. In our case, there is an additional table named **MANAGERS** that is part of the HR application, but all of its constraints are enforced by the application and NOT in the database. The **MANAGERS** table uses **EMPLOYEE_ID**, but the relationship is not registered in the database as a foreign key constraint. Therefore, we must add a Dependent column on the **EMPLOYEE_ID** column. Click **Add**.

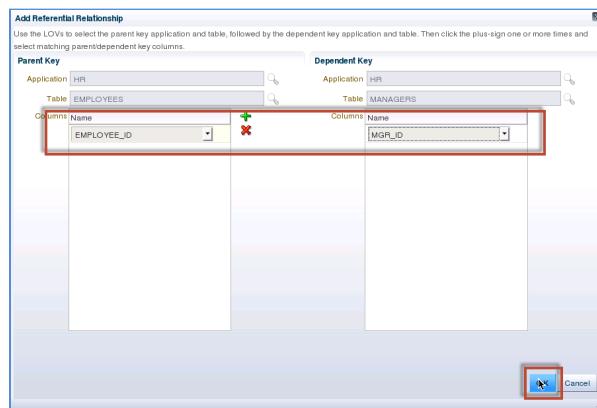
Use the spyglasses to select the Parent and Dependent Applications and Tables. You may need to expand the size of the window to see everything.



Then push the green 'plus' sign and select **EMPLOYEE_ID** as the Parent Key.



Select **MGR_ID** as the Dependent Key and then click the **OK** button to complete the process to exit.



D. Summary

In this lab, you:

1. Learned how to create an Application Data Model.
2. Drilled into the Referential Relationships and saw the primary key and foreign key relationships captured.
3. Saw how Referential Relationships not included in the Data Dictionary (typically those enforced in Application code) can be added manually.

LAB EXERCISE 02 – IDENTIFYING SENSITIVE DATA

Identified Challenge – Difficulty of Locating Sensitive Data

There is no established, documented procedure and enforcement of data cleansing standards in masking and cleansing of sensitive production data before distribution to development and QA environments.

The steps and process necessary to provide development and QA environments with properly masked data are very time consuming, manual and inconsistent.

INTRODUCTION

Prior to masking sensitive data, organizations must ensure that all sensitive data has been identified.

Sensitive Data Discovery

Today's enterprise applications have very complex database schema often containing hundreds or thousands of database objects. Administrators have a daunting and time consuming job of identifying all tables and columns containing confidential or sensitive. Fortunately, Oracle Data Masking Pack makes this task easy through the built-in search function that allows the information security administrator to query the entire data dictionary to identify potential tables and columns containing sensitive data.

This functionality may be utilized by our customers who license any of our Database Security products. For example, this capability may assist Audit Vault users with identifying candidates for Fine Grained Audit policies.

A. Overview

After creating the Application Data Model that catalogues all the Referential Relationships, the next step in the masking process is to identify Sensitive Columns. Only columns identified and defined during this phase will be candidates for Data Masking.

During this lab you will:

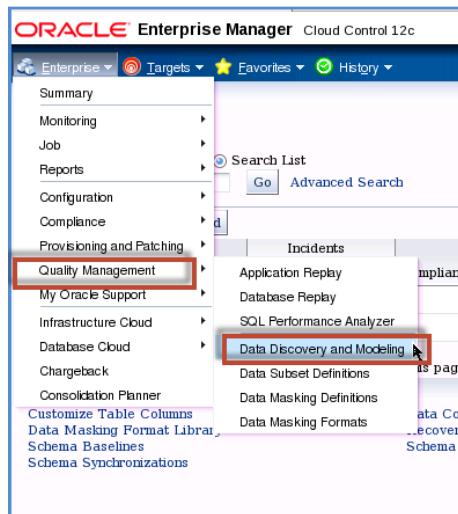
1. *Identify and use pre-defined Sensitive Column Types.*
2. *Create new Sensitive Column Types and apply them to certain columns.*
3. *Manually identify a Sensitive Column*

B. Setup & Preparation

- None

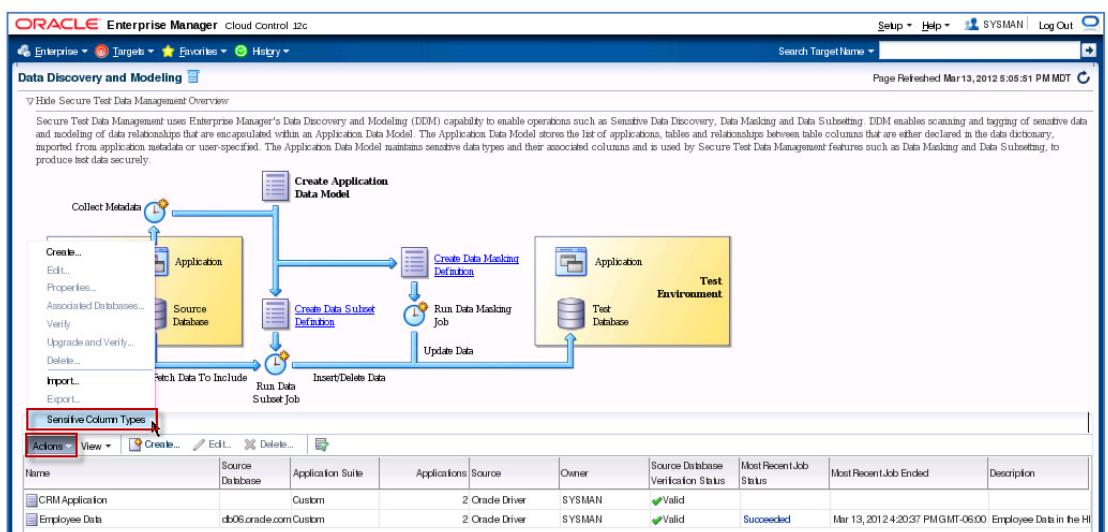
C. Identifying Sensitive Data

1. Navigate to the Data Discovery and Modeling page from the Quality Management submenu (**Enterprise → Quality Management → Data Discovery and Modeling**).



Finding Sensitive Data Using Pre-defined Templates

2. Let us look at the list of Sensitive Column Types. Click on the menu 'Actions' and select the sub-menu 'Sensitive Column Types'.



- Review the Sensitive Column Discovery Templates that are shipped by default with the Data Masking Pack. As an example, review the **EMAIL_ID** template by hovering over the name '**EMAIL_ID**'. When using this Sensitive Column Type will 1) search for '**EMAIL**' or '**MAIL**' in the Column Name, 2) search for '**EMAIL**' or '**MAIL**' in the Column Comment and 3) apply a regular expression pattern match to all of the Column Data if the user (i.e. MASKING_ADMIN) has access to the data. This process uses Oracle Regular Expressions which is compatible with the IEEE Portable Operating System Interface (POSIX) regular expression standard and to the Unicode Regular Expression Guidelines of the Unicode Consortium. In this case, the **Search Type** has been set as an '**Or**' condition, so if any of the conditions listed above are met, it will result in a match.

Name	Column Name	Column Comment	Column Data	Search Type
EMAIL_ID	EMAIL_';MAIL*	EMAIL_';MAIL*	^[a-zA-Z0-9_%.+]+@[a-zA-Z0-9.]+\.[a-zA-Z]{2,4}\$	Or
CREDIT_CARD				
EMAIL				
IP_ADDRESS				
ISBN_10				
ISBN_13				
NATIONAL_INSURANCE_NUMBER				
PHONE_NUMBER				
SOCIAL_INSURANCE_NUMBER				
SOCIAL_SECURITY_NUMBER				
UNDEFINED				
UNIVERSAL_PRODUCT_CODE				

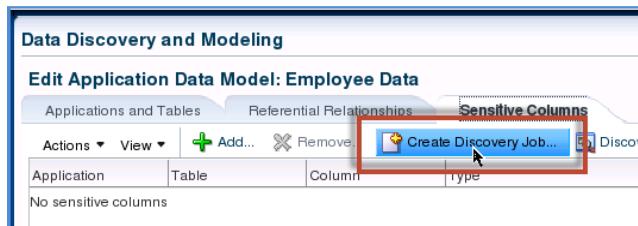
- We will first use one of the Sensitive Column Types and then return to create a custom one. Navigate back to the Data Discovery and Modeling page from the Quality Management submenu (Enterprise → Quality Management → Data Discovery and Modeling). Then, select the **Employee Data ADM** and click on the '**Edit**' button. You may be asked for the database credentials. If so, select the 'Named' radio button, choose the default credential using the **MASKING_ADMIN** username and click on the '**Continue**' button.

Name	source	Application Suite
CRM Application	Database	Custom
Employee Data	db06.oracle.com	Custom

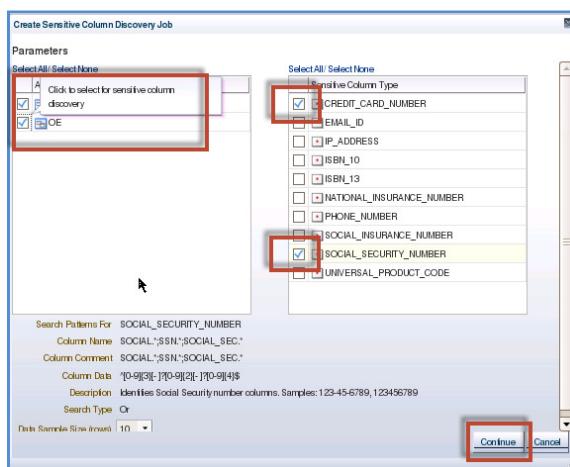
5. On the **Edit Application Data Model: Employee** screen, click on the ‘Sensitive Columns’ tab.



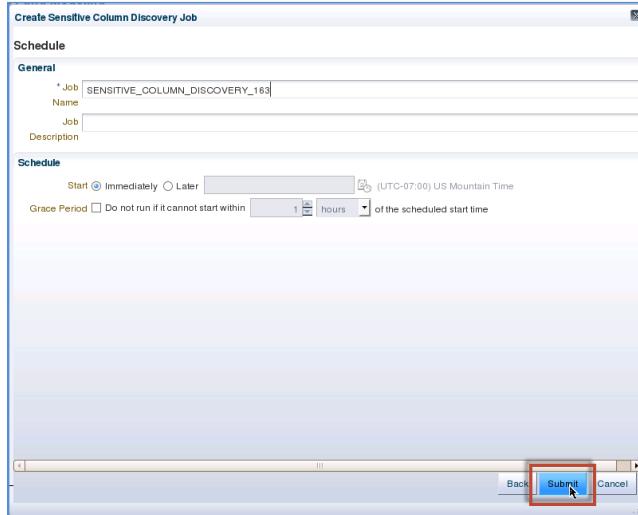
6. Currently, there are no sensitive columns discovered. We will search for sensitive columns. Click on the option to ‘Create Discovery Job...’.



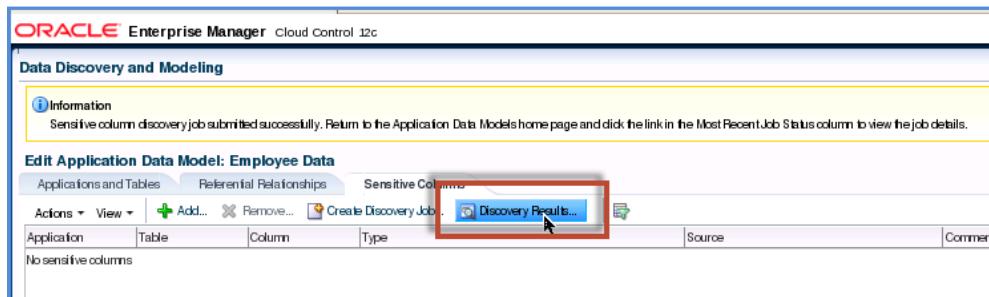
7. Provide the parameters for the sensitive columns discovery job. Choose both the **HR** and the **OE** Applications and choose both the **CREDIT_CARD_NUMBER** and **SOCIAL_SECURITY_NUMBER** Sensitive Column Types. Click on the ‘Continue’ button to perform the search.



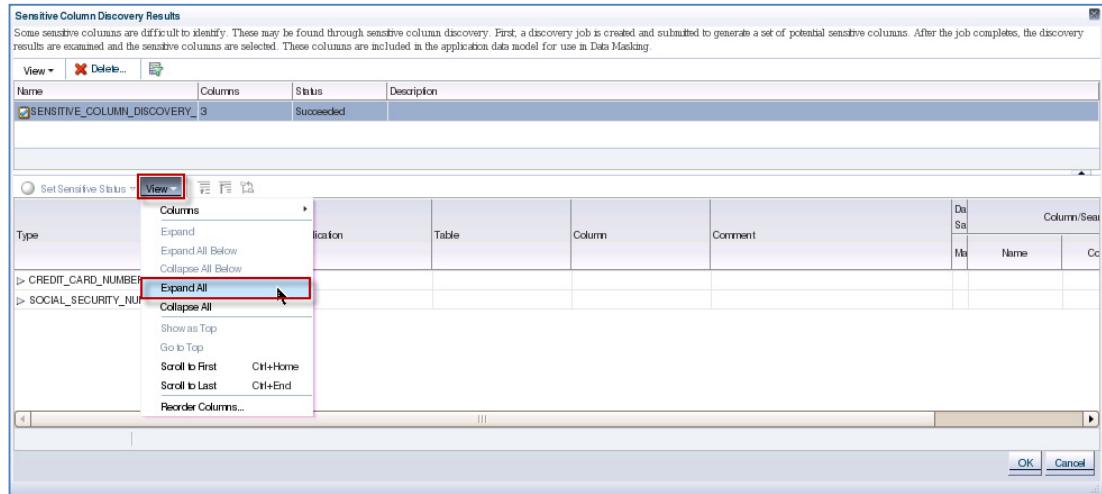
5. After clicking the '**Continue**' button, provide the Job information and click on the '**Submit**' button to run the job based upon the options selected.



6. If this were a more complex job with a larger dataset, we might want to track the progress of the job as suggested at the top of the screen. However, in our case, we can just drill into the results. Click on the option, '**Discover Results....**'.



- Click on 'View' then 'Expand All' to review the Sensitive Column Discovery Results.



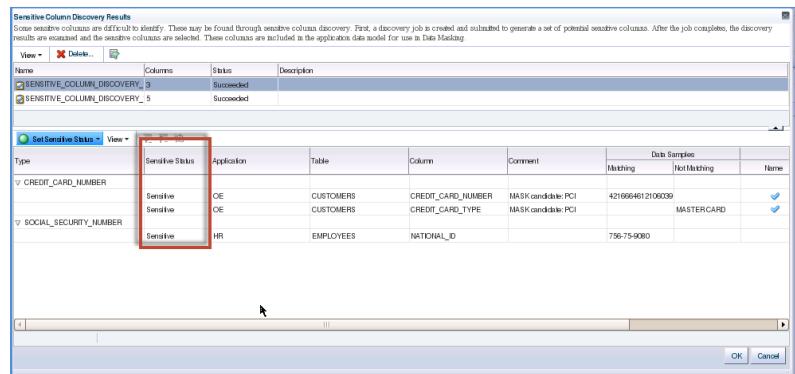
Let's spend a little time on the resulting screen. Here are a few things to notice:

- Both **CREDIT_CARD_NUMBER** and **CREDIT_CARD_TYPE** were selected based on the name of the column, but only the **CREDIT_CARD_NUMBER** contains matching data.
- Both **CREDIT_CARD_NUMBER** and **CREDIT_CARD_TYPE** contain a column named '**Comment**'. When the table was being built, CashBankTrust Security personnel made sure the DBAs identified these columns explicitly as being candidates for Data Masking. We will be using this information in the next section.
- Observe that the type **SOCIAL_SECURITY_NUMBER** type was found, neither in the Column Name or Comment, but found as data in the **NATIONAL_ID** Column of the **EMPLOYEES** table.

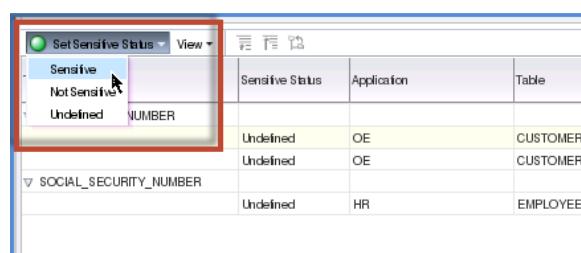
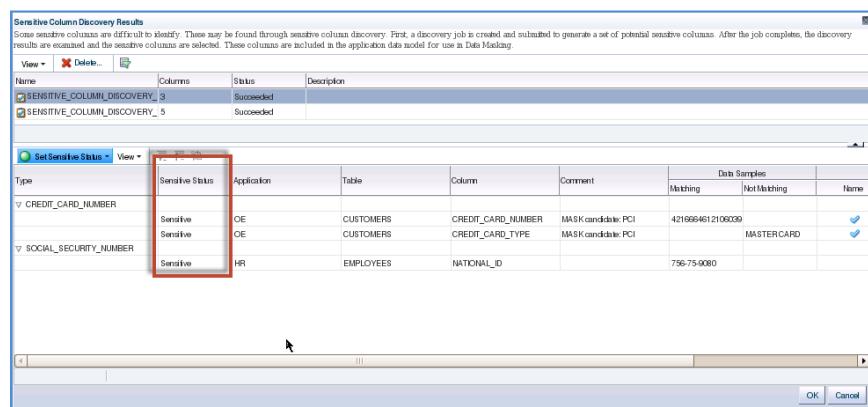
If we scroll to the right, we can see exactly how Sensitive Column Discovery made the automatic decisions.

Column	Comment	Data Samples		Column/Search Criteria Match		
		Matching	Not Matching	Name	Comment	Data (%)
CREDIT_CARD_NUMBER	MASK candidate: PCI	4216664612106039		✓		100
CREDIT_CARD_TYPE	MASK candidate: PCI		MASTERCARD	✓		0
NATIONAL_ID		756-75-9080				100

8. Notice that the Sensitive Status of these columns is currently set to 'Undefined'. Set the sensitive status of all columns to 'Sensitive'. Select each identified sensitive column entry and click on 'Set Sensitive Status' menu item and then pick 'Sensitive' sub menu item. Upon successful completion, you should see all of the 'Undefined' labels toggle to 'Sensitive'.



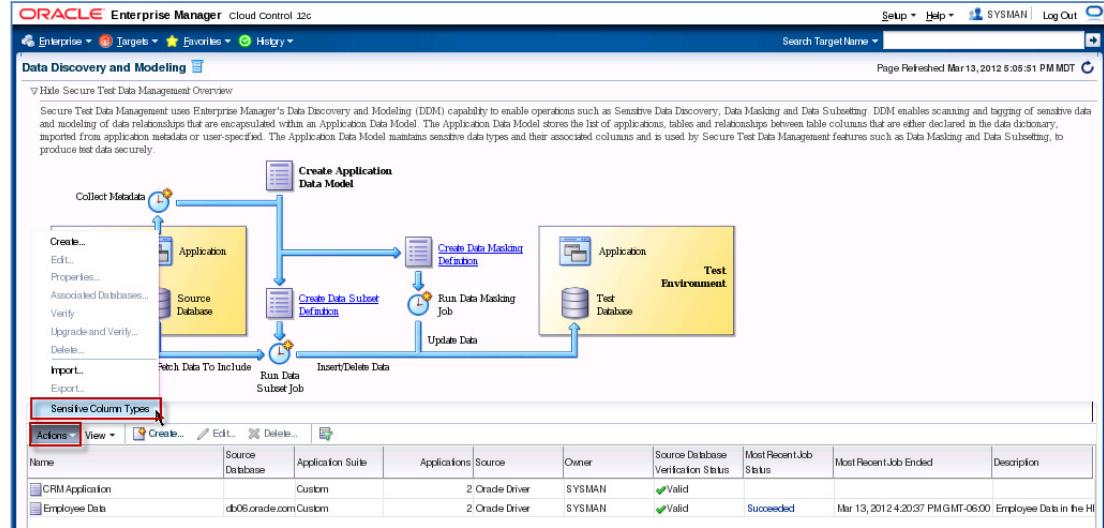
Your screen will look like this.



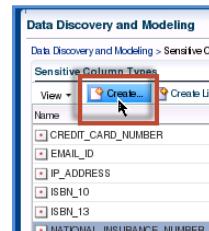
Click on the 'OK' button and 'Save and Return' button to continue.

Finding Sensitive Data Using a Customized Templates

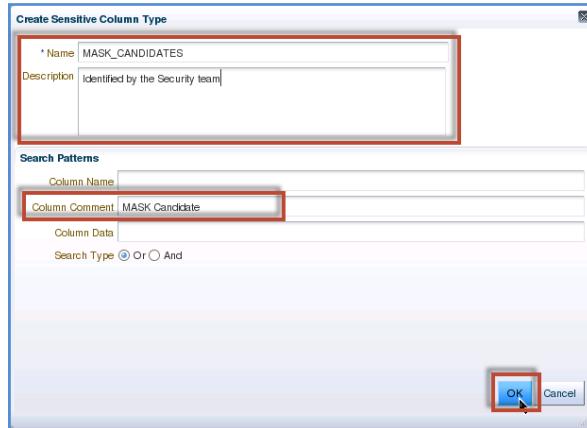
9. We will now create a customized template and search for sensitive information. Navigate to the Data Discovery and Modeling page from the Quality Management submenu (**Enterprise → Quality Management → Data Discovery and Modeling**). From there, open up the list of Column Definitions by choosing **Actions → Sensitive Column Types**.



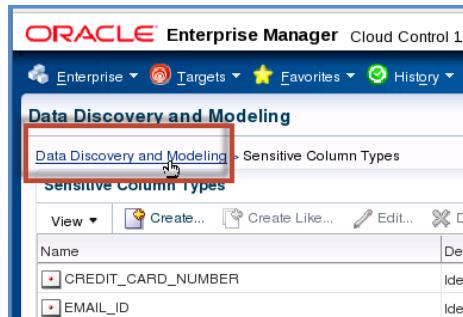
10. As we reviewed in the last section, CashBankTrust has already done some manual work around identifying sensitive data by adding some comments. These sensitive columns are identified with the text "**MASK candidate**". We will now create a custom Sensitive Data Type. Click the 'Create' icon to start the process.



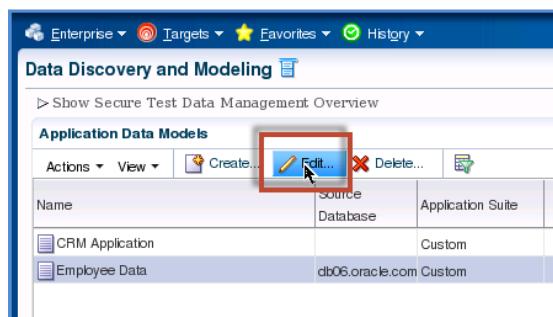
11. Fill in the **Name** field with the entry, “**MASK_CANDIDATES**”, the **Description** field with “**Identified by the Security Team**” and the Column Comment field **“MASK Candidate”**. **“MASK Candidate”** is our string search pattern for our new Sensitive Column Type. Click the **OK** button to accept and continue.



12. We will now search for sensitive data based upon our custom Sensitive Column Type. Click on the breadcrumb ‘**Data Discovery and Modeling**’ to return to our list of Application Data Models.



13. Select the **Employee Data** ADM by highlighting the row and click on the **Edit...** icon. We are going to search our Employee Data Model for sensitive columns looking for our custom Sensitive Column Type.



Click on the **Sensitive Columns** tab.

The screenshot shows the Oracle Enterprise Manager Cloud Control 12c interface. The main title bar reads 'ORACLE Enterprise Manager Cloud Control 12c'. Below it, the page title is 'Data Discovery and Modeling'. A sub-header 'Edit Application Data Model: Employee Data' is displayed. There are three tabs at the top: 'Applications and Tables' (selected), 'Referential Relationships', and 'Sensitive Columns'. The 'Sensitive Columns' tab has a red box drawn around it. Below the tabs is a toolbar with buttons for 'Actions', 'View', 'Add Application...', 'Add Table...', 'Remove...', and other options. A table below the toolbar lists applications: HR and OE. The table columns are 'Name', 'Short Name', 'Schema', 'Table Type', 'Source', and 'Comment'. Both entries have 'Dictionary' listed under 'Source'.

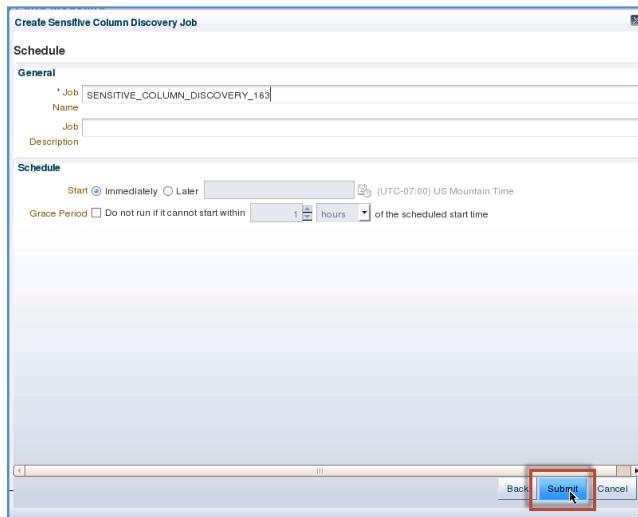
14. We will search for sensitive columns. Click on the option to 'Create Discovery Job....'.

This screenshot shows the same interface as the previous one, but with a red box highlighting the 'Create Discovery Job...' button. This button is located in the toolbar area under the 'Sensitive Columns' tab.

15. Provide the parameters for the sensitive columns discovery job. Choose both the **HR** and the **OE** Applications and choose the **CREDIT_CARD_NUMBER**, **SOCIAL_SECURITY_NUMBER**, and **MASK_CANDIDATES** Sensitive Column Types. Remember, **MASK_CANDIDATES** was our custom Sensitive Data Type. Click on the 'Continue' button to perform the search.

This screenshot shows the 'Create Sensitive Column Discovery Job' dialog box. On the left, under 'Parameters', there is a section for 'Select All/ Select None' with checkboxes for 'Application'. Two applications are checked: 'HR' and 'OE', which are also highlighted with a red box. On the right, under 'Sensitive Column Type', several checkboxes are listed, each with a small orange square icon. Three checkboxes are checked and highlighted with a red box: 'CREDIT_CARD_NUMBER', 'MASK_CANDIDATES', and 'SOCIAL_SECURITY_NUMBER'. Below these sections, there are fields for 'Search Patterns For', 'Column Name', 'Column Comment', 'Column Data', 'Description', 'Search Type', and 'Data Sample Size (Rows)'. At the bottom right of the dialog box, there are 'Continue' and 'Cancel' buttons, with 'Continue' also highlighted with a red box.

16. On the Create Sensitive Column Discovery Job screen, provide the relevant Job information (keeping defaults) and click on the 'Submit' button to run the job based upon the options selected.



17. Click on the option, 'Discover Results....' .

Edit Application Data Model: Employee Data					
Applications and Tables		Referential Relationships		Sensitive Columns	
Actions		View		Add... Remove... Create Discovery Job... Discovery Results...	
Application	Table	Column	Type	Source	Comment
HR	EMPLOYEES	NATIONAL_ID	SOCIAL_SECURITY_NUMBER	Sensitive Column Discovery	
OE	CUSTOMERS	CREDIT_CARD_NUM	CREDIT_CARD_NUMBER	Sensitive Column Discovery	MASK candidate: PCI
OE	CUSTOMERS	CREDIT_CARD_TYF	CREDIT_CARD_NUMBER	Sensitive Column Discovery	MASK candidate: PCI

18. You'll quickly notice there are added results in this search—specifically the type of **MASK_CANDIDATES** along with CREDIT_CARD_NUMBER and SOCIAL_SECURITY_NUMBER. This is what we expect. Click on 'View' then 'Expand All' to review the Sensitive Column Discovery Results.

Sensitive Column Discovery Results

Some sensitive columns are difficult to identify. These may be found through sensitive results are examined and the sensitive columns are selected. These columns are incl

Name	Columns	Status
SENSITIVE_COLUMN_DISCOVERY_11		Succeeded
SENSITIVE_COLUMN_DISCOVERY_3		Succeeded

Type

Columns

View

Set Sensitive Status

Expand All

Expand All Below

Collapse All Below

CREDIT_CARD_NUMBER

MASK_CANDIDATES

SOCIAL_SECURITY_NUMBER

Reorder Columns...

Notice that the columns that have been discovered containing the “**MASK candidate**” string defined in our custom Sensitive Data Type.

Type	Sensitive Status	Application	Table	Column	Comment	Da	Ca	Column
Type	Sensitive Status	Application	Table	Column	Comment	Da	Ca	Column
MASK_CANDIDATES	Undefined	HR	EMPLOYEES	EMAIL	MASK candidate: HR Privacy Policy			
MASK_CANDIDATES	Undefined	HR	EMPLOYEES	EMPLOYEE_ID	MASK candidate: HR Benefits Policy			
MASK_CANDIDATES	Undefined	HR	EMPLOYEES	FIRST_NAME	MASK candidate: HR Privacy Policy			
MASK_CANDIDATES	Undefined	HR	EMPLOYEES	LAST_NAME	MASK candidate: HR Privacy Policy			
MASK_CANDIDATES	Undefined	HR	EMPLOYEES	SALARY	MASK candidate: HR Compensation Policy			
MASK_CANDIDATES	Undefined	HR	MANAGERS	MGR_ID	MASK candidate: HR Benefits Policy			

The previous search results and columns have remained identified as Sensitive. Set the status of these newly discovered columns to '**Sensitive**'. Note: You can use <Shift><Enter> to select several of the columns at once.

Sensitive Column Discovery Results					
Some sensitive columns are difficult to identify. These may be found through sensitive column discovery. First, a discovery job is created and submitted to generate a set of potential sensitive columns. After the results are examined and the sensitive columns are selected. These columns are included in the application data model for use in Data Masking.					
Set Sensitive Status		View			
Sensitive	Not Sensitive	Sensitive Status	Application	Table	Column
Sensitive	Not Sensitive	Undefined			Comment
Sensitive	Not Sensitive	Sensitive	OE	CUSTOMERS	CREDIT_CARD_TYPE
MASK_CANDIDATES		Sensitive	HR	EMPLOYEES	EMAIL
MASK_CANDIDATES		Sensitive	HR	EMPLOYEES	EMPLOYEE_ID
MASK_CANDIDATES		Sensitive	HR	EMPLOYEES	FIRST_NAME
MASK_CANDIDATES		Sensitive	HR	EMPLOYEES	LAST_NAME
MASK_CANDIDATES		Sensitive	HR	EMPLOYEES	SALARY
MASK_CANDIDATES		Sensitive	HR	MANAGERS	MGR_ID
MASK_CANDIDATES		Sensitive	OE	CUSTOMERS	CUST_FIRST_NAME
MASK_CANDIDATES		Sensitive	OE	CUSTOMERS	CUST_LAST_NAME

Once the Sensitive Status is set to '**Sensitive**', click on the '**OK**' button to continue.

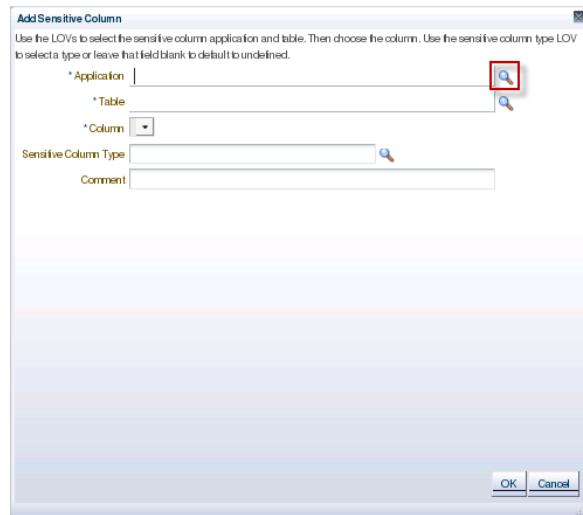
19. You will be brought back to the **Edit Application Data Model: Employee Data** screen. We will go through the process of manually identifying a Sensitive Column. Click on the '**Add**' button to add Sensitive Columns manually.

We will be adding the column **PHONE_NUMBER** from the **HR.EMPLOYEES** table.

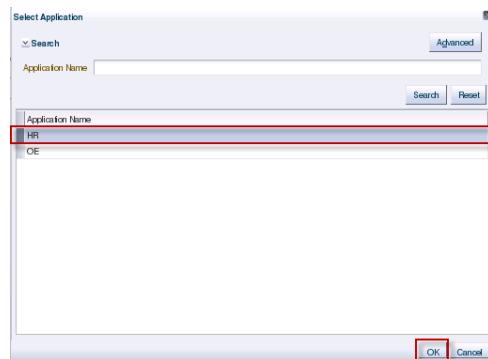
Application: HR
Table: EMPLOYEES
Column: PHONE_NUMBER
Sensitive Column Type: PHONE_NUMBER

Edit Application Data Model: Employee Data					
Actions		Sensitive Columns		Import Content Save and Return	
Application	Table	Column	Type	Source	Comment
HR	DEPARTMENTS	MANAGER_ID	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: HR Privacy Policy
HR	EMPLOYEES	EMAIL	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: HR Benefits Policy
HR	EMPLOYEES	EMPLOYEE_ID	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: HR Privacy Policy
HR	EMPLOYEES	FIRST_NAME	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: HR Privacy Policy
HR	EMPLOYEES	LAST_NAME	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: HR Privacy Policy
HR	EMPLOYEES	MGR_ID	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: HR Privacy Policy
HR	EMPLOYEES	NATIONAL_ID	SOCIAL_SECURITY_NUMBER	Sensitive Column Discovery	
HR	EMPLOYEES	SALARY	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: HR Compensation Policy
HR	JOB_HISTORY	EMPLOYEE_ID	MASK_CANDIDATES	Sensitive Column Discovery	
HR	MANAGERS	MGR_ID	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: HR Benefits Policy
OE	CUSTOMERS	ACCOUNT_MGR_ID	MASK_CANDIDATES	Sensitive Column Discovery	
OE	CUSTOMERS	CREDIT_CARD_NIN	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: PCI
OE	CUSTOMERS	CREDIT_CARD_TYF	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: PCI
OE	CUSTOMERS	CUST_FIRST_NAME	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: Consumer PII
OE	CUSTOMERS	CUST_LAST_NAME	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: Consumer PII
OE	ORDERS	SALES REP_ID	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: Consumer PII

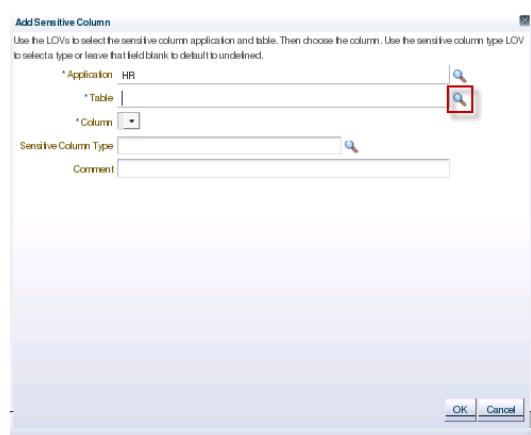
In the **Add Sensitive Column** dialog, click on the 'search' icon to specify the desired Application.



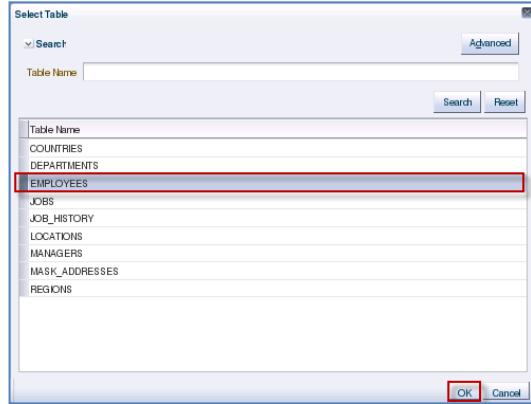
Highlight the **HR** Application Name and click on the **OK** button to continue.



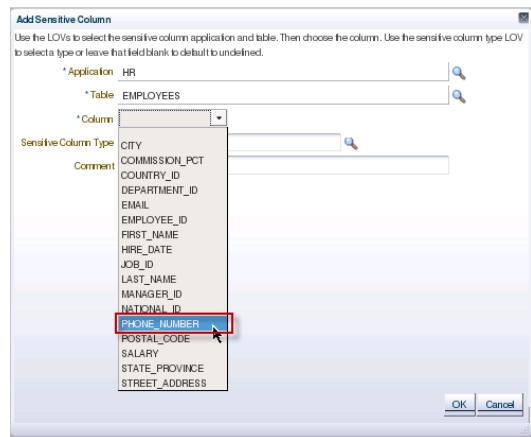
In the **Add Sensitive Column** dialog, click on the 'search' icon to specify the desired **Table**.



Highlight the **EMPLOYEES** Table Name and click on the **OK** button to continue.



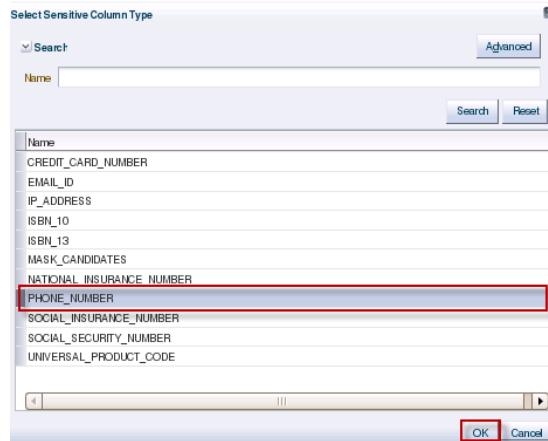
Choose the Column **PHONE_NUMBER** from the drop-down list box.



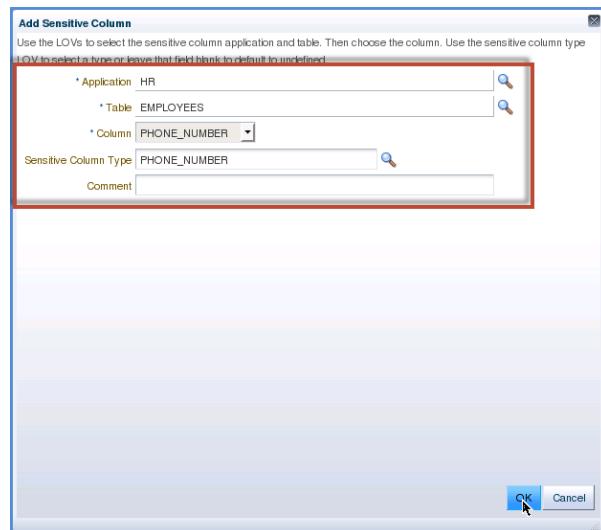
In the **Add Sensitive Column** dialog, click on the 'search' icon to specify the desired **Sensitive Column Type**.



In the **Select Sensitive Column Type** dialog, highlight the Name **PHONE_NUMBER** and click on the **OK** button.



Review and click on the **OK** button to complete the form. Return to the Application Data Model.



Click **OK** and return to the Application Data Model. Notice that the user defined, custom Type of **PHONE_NUMBER** is now part of the list of Sensitive Columns. Finally, click **Save and Return**.

Data Discovery and Modeling

Edit Application Data Model: Employee Data

Actions View + Add... Remove... Create Discovery Job... Discovery Results...

Import Content Save and Return

Application	Table	Column	Type	Source	Comment
HR	DEPARTMENTS	MANAGER_ID	MASK_CANDIDATES	Sensitive Column Discovery	
HR	EMPLOYEES	EMAIL	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: HR Privacy Policy
HR	EMPLOYEES	EMPLOYEE_ID	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: HR Benefits Policy
HR	EMPLOYEES	FIRST_NAME	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: HR Privacy Policy
HR	EMPLOYEES	LAST_NAME	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: HR Privacy Policy
HR	EMPLOYEES	MANAGER_ID	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: HR Privacy Policy
HR	EMPLOYEES	NATIONAL_ID	MASK_CANDIDATES	Sensitive Column Discovery	
HR	EMPLOYEES	PHONE_NUMBER	PHONE_NUMBER	User Defined	
HR	EMPLOYEES	SALARY	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: HR Compensation Policy
HR	JOB_HISTORY	EMPLOYEE_ID	MASK_CANDIDATES	Sensitive Column Discovery	
HR	MANAGERS	MGR_ID	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: HR Benefits Policy
OE	CUSTOMERS	ACCOUNT_MGR_ID	MASK_CANDIDATES	Sensitive Column Discovery	
OE	CUSTOMERS	CREDIT_CARD_NUM	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: PCI
OE	CUSTOMERS	CREDIT_CARD_TYPE	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: PCI
OE	CUSTOMERS	CUST_FIRST_NAME	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: Consumer PII
OE	CUSTOMERS	CUST_LAST_NAME	MASK_CANDIDATES	Sensitive Column Discovery	MASK candidate: Consumer PII
OE	ORDERS	SALES REP_ID	MASK_CANDIDATES	Sensitive Column Discovery	

E. Summary

In this lab, you:

1. Identified and used pre-defined Sensitive Column Types.
2. Create two new Sensitive Column Types and applied them to certain columns.
3. Manually identified a Sensitive Column

LAB EXERCISE 03 – CREATING, EXPORTING & IMPORTING DATA MASKING FORMATS

Identified Challenge – Manual, Inconsistent and non-Standardized Data Masking Operations

There is no established, documented procedure and enforcement of data cleansing standards in masking and cleansing of sensitive production data before distribution to development and QA environments.

The steps and process necessary to provide development and QA environments with properly masked data are very time consuming, manual and inconsistent.

Introduction

In addition to the need to effectively and efficiently mask sensitive data, organizations must ensure that particular standards have been achieved when masking data.

Centralized Library of Reusable Masking Formats

Oracle Data Masking provides a centralized library of out-of-the-box mask formats for common types of sensitive data, such as credit card numbers, phone numbers, national identifiers (social security number for US, national insurance number for UK). By leveraging the Format Library in Oracle Data Masking, enterprises can apply data privacy rules to sensitive data across enterprise-wide databases from a single source and thus ensure consistent compliance with regulations. Enterprises can also extend this library with their own mask formats to meet their specific data privacy and application requirements.

You can edit existing formats defined in the Format Library for re-use. Ideally you should first make a duplicate of the format you wish to edit, and modify the duplicate instead of the original.

Masking Formats are Portable

Masking Formats may be Exported to XML files for safekeeping and transport from one environment to another. You can copy a previously exported data masking format saved as an XML file to the current Enterprise Manager repository, then import and re-use it.

A. Overview

In addition to the Masking Format Library and the ready-to-use masking formats provided, you may need to create your own customized masking formats.

During this lab you will:

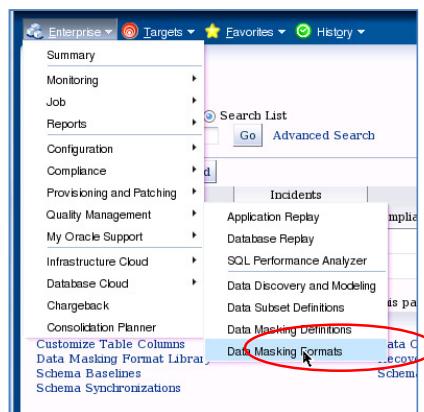
1. *Create a custom masking format.*
2. *Export the Masking Format Library for future use.*

B. Setup & Preparation

- None

C. Creating, Exporting & Importing Data Masking Formats

1. Open the Format Library by clicking the menu **Enterprise → Quality Management → Data Masking Formats**.



2. Here you can see the Masking Formats that are shipped with the Data Masking Pack. Select the Format named '**Visa Credit Card Number**' and click on the **Edit** button.

The screenshot shows the 'Format Library' page. At the top, there's a search bar and buttons for 'View', 'Create Lib', 'Edit', and 'Delete'. A red box highlights the 'Edit' button. Below is a table listing masking formats:

Select	Format	Data Type	Sensitive Column Type	Sample	Description	Owner
<input type="radio"/>	American Express Credit Card Number	Character	CREDIT_CARD_NUMBER	3774201052133792	~10 billion unique American Express credit card numbers	SYSMAN
<input type="radio"/>	Discover Card Credit Card Number	Character	CREDIT_CARD_NUMBER	6011624411510291	~10 billion unique Discover Card credit card numbers	SYSMAN
<input type="radio"/>	MasterCard Credit Card Number	Character	CREDIT_CARD_NUMBER	5254382922029263	~10 billion unique MasterCard credit card numbers	SYSMAN
<input checked="" type="radio"/>	Visa Credit Card Number	Character	CREDIT_CARD_NUMBER	4539574068046115	~10 billion unique Visa credit card numbers	SYSMAN
<input type="radio"/>	Generic Credit Card Number	Character	CREDIT_CARD_NUMBER	6011388953187419	~10 billion unique generic credit card numbers	SYSMAN
<input type="radio"/>	Generic Credit Card Number Formatted	Character	CREDIT_CARD_NUMBER	2100-6661-0733-3358	~10 billion unique generic credit card numbers	SYSMAN

- In the '**Visa Credit Card Number**' Format, notice how Random digits are first generated and then a SQL Function containing the logic to generate a valid Visa number is called as a post- processing function. Click on the **OK** or **Cancel** button to return.

The screenshot shows the 'Format Library' interface in Oracle Enterprise Manager. The title bar says 'Format Library: View Format: Visa Credit Card Number'. The main content area displays the following details:

- Name:** Visa Credit Card Number
- Description:** ~10 billion unique Visa credit card numbers
- Sensitive Column Type:** CREDIT_CARD_NUMBER

Format Entries:

Type	Description
Random Digits	Digits Length Range: 10 - 10

Post Processing Function: DBSNMP.DM_FMTLIB.MGMT_DM_GEN_VC
The function can either be a standalone function (Example: scott.masking_func) or a f...

Sample Masked Data:

- 4797443810358847
- 4929465980421221
- 4916264726268037
- 4916416327162106
- 4904836222431088

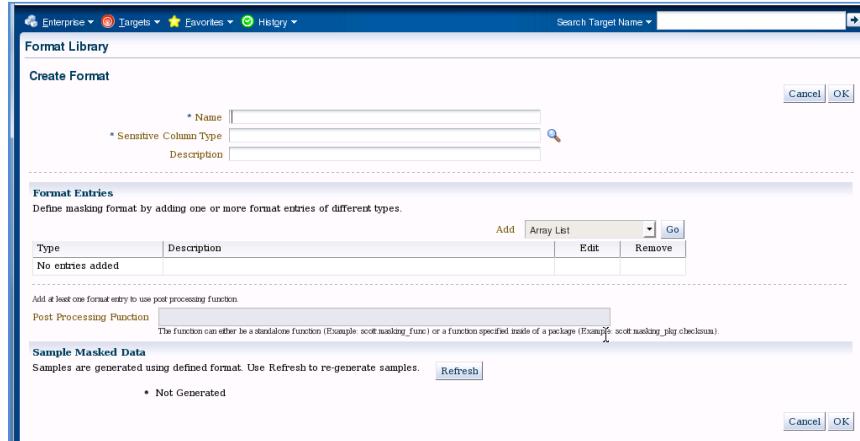
- In addition to the Masking Formats provided out-of-the-box, you have the flexibility to create custom Masking Formats as well. We will step through that process. From the **Format Library** screen, click on the **Create** button to begin creating a custom Masking Format.

The screenshot shows the 'Format Library' interface in Oracle Enterprise Manager. The title bar says 'Format Library - Oracle Enterprise Manager'. The main content area displays a list of available masking formats:

Select	Format	Data Type	Sensitive Column Type	Sample	Description	Owner
<input checked="" type="radio"/>	American Express Credit Card Number	Character	CREDIT_CARD_NUMBER	3473930980043384	>10 billion unique American Express credit card numbers	SYSMAN
<input type="radio"/>	Discover Card Credit Card Number	Character	CREDIT_CARD_NUMBER	6011724971484323	>10 billion unique Discover Card credit card numbers	SYSMAN
<input type="radio"/>	MasterCard Credit Card Number	Character	CREDIT_CARD_NUMBER	522291091034679	>10 billion unique MasterCard credit card numbers	SYSMAN
<input type="radio"/>	Visa Credit Card Number	Character	CREDIT_CARD_NUMBER	4556513973093430	>10 billion unique Visa credit card numbers	SYSMAN
<input type="radio"/>	Generic Credit Card Number	Character	CREDIT_CARD_NUMBER	3458500169434750	>10 billion unique generic credit card numbers	SYSMAN
<input type="radio"/>	Generic Credit Card Number Formatted	Character	CREDIT_CARD_NUMBER	60115832-6388-2283	>10 billion unique generic credit card numbers	SYSMAN
<input type="radio"/>	National Insurance Number Formatted	Character	NATIONAL_INSURANCE_NUMBER	TZ 13 05 42 A	Generates unique UK National Insurance Numbers	SYSMAN
<input type="radio"/>	Social Insurance Number	Character	SOCIAL_INSURANCE_NUMBER	998808745	>1 billion unique Canadian Social Insurance Numbers	SYSMAN
<input type="radio"/>	Social Insurance Number Formatted	Character	SOCIAL_INSURANCE_NUMBER	411-841-844	>1 billion unique Canadian Social Insurance Numbers	SYSMAN
<input type="radio"/>	Social Security Number	Character	SOCIAL_SECURITY_NUMBER	196885502	>718 million unique US Social Security Numbers	SYSMAN

The 'Create' button in the top right corner is circled in red.

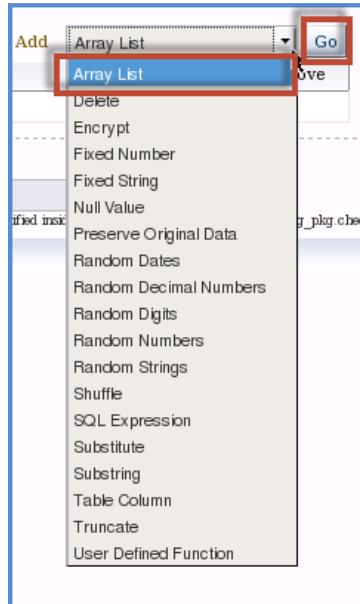
- From the **Create Format** dialog, we will configure our custom Masking Format.



Use the information below to create custom format.

Name:	Colors
Sensitive Column Type:	UNDEFINED
Description:	Colors of the rainbow

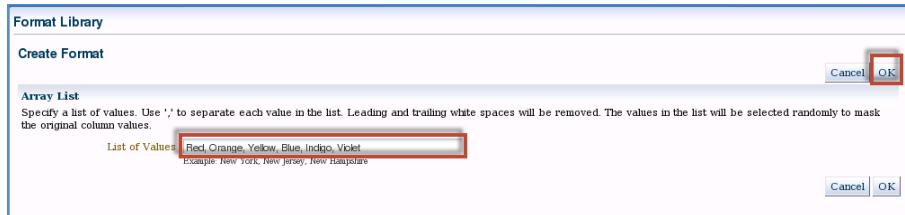
- Type '**Colors**' in the Name field. Select '**UNDEFINED**' using the LOV (List of Values available via spyglass icon) and '**Colors of the rainbow**' in the Description Field. Before adding a field type, view the number of different options which you can mask data. Choose **ArrayList** and click the **Go** button.



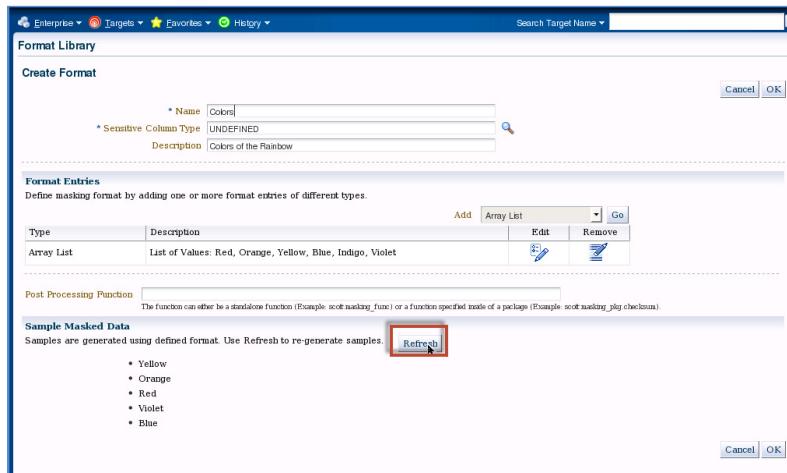
- Define the **List of Values** for the **Colors** Format and click on the **OK** button when finished.

The values include:

- Red, Orange, Yellow, Green, Blue, Indigo, Violet



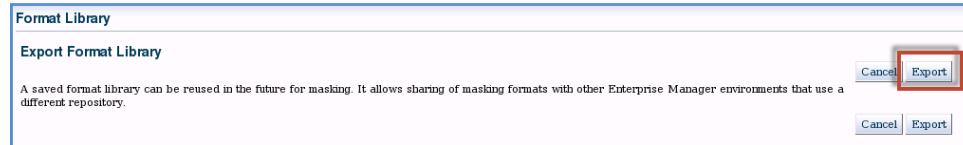
- You can see samples of the masked data in the Sample Masked Data Section. Click on the **Refresh** button to see a random sample from the defined Array List. This screen allows you to edit any values of the Masking Format. Click the **OK** button when you are satisfied with the entries.



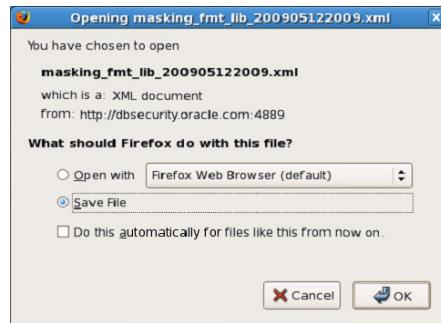
- Return to the Format Library screen and click on the **Export** button to begin the process of exporting the entire library.

Select	Format	Data Type	Sensitive Column Type	Sample	Description	Owner
<input checked="" type="radio"/>	Colors	Source Type	UNDEFINED	Orange	Colors of the Rainbow	SYSMAN
<input type="radio"/>	American Express Credit Card Number	Character	CREDIT_CARD_NUMBER	3481905191941899	>10 billion unique American Express credit card numbers	SYSMAN
<input type="radio"/>	Discover Card Credit Card Number	Character	CREDIT_CARD_NUMBER	6011261572471128	>10 billion unique Discover Card credit card numbers	SYSMAN
<input type="radio"/>	MasterCard Credit Card Number	Character	CREDIT_CARD_NUMBER	5241797530701423	>10 billion unique MasterCard credit card numbers	SYSMAN
<input type="radio"/>	Via Credit Card Number	Character	CREDIT_CARD_NUMBER	4556977435593172	>10 billion unique Visa credit card numbers	SYSMAN
<input type="radio"/>	Generic Credit Card Number	Character	CREDIT_CARD_NUMBER	3744626642541707	>10 billion unique generic credit card numbers	SYSMAN

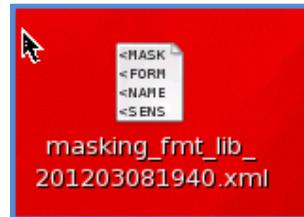
10. As the dialog states, exporting a format mask can be saved and re-used in the future for masking. This mask can be shared and/or imported into another Format Library in another Enterprise Manager environment. Click on the **Export** button to continue.



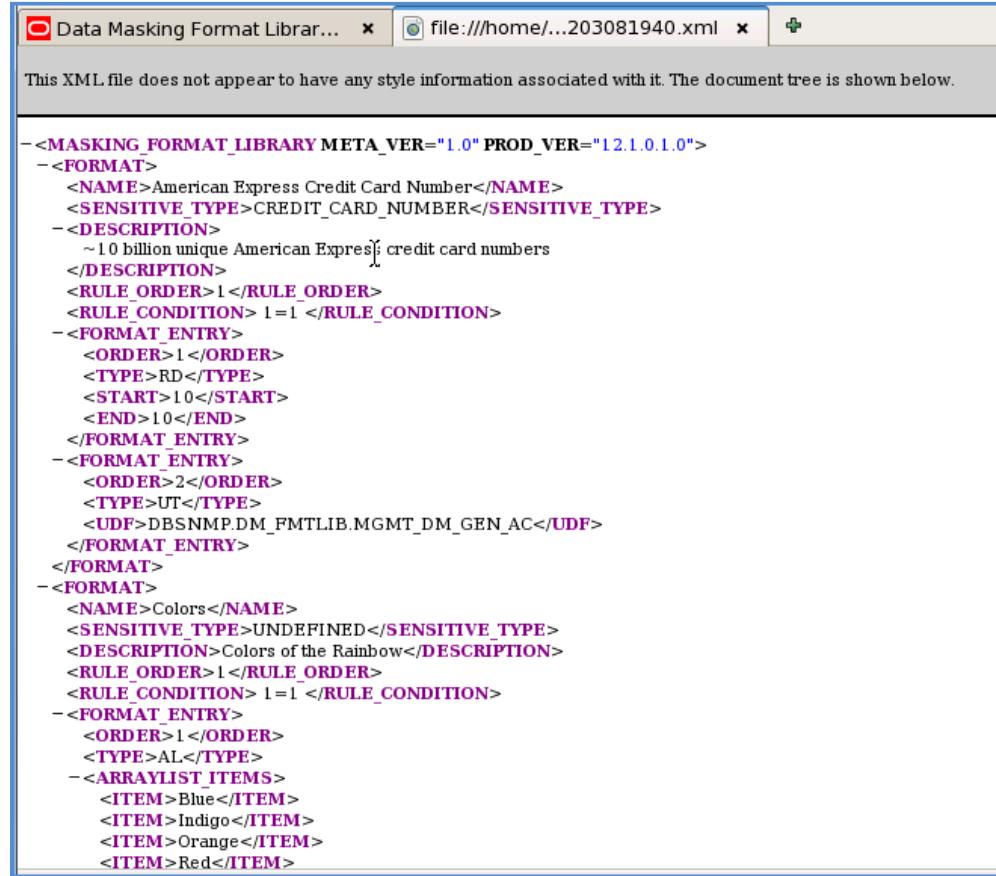
11. You will be prompted by this dialog to save the file. Save this file to the default location on the Desktop.



12. Navigate to the Desktop and double-click on the newly created XML document. Note: Your filename will be different than what has been captured here.



13. Review the XML document and the information that has been captured in the document.



The screenshot shows a software interface for managing masking formats. The title bar says "Data Masking Format Librar..." and "file:///home/...203081940.xml". A message at the top states: "This XML file does not appear to have any style information associated with it. The document tree is shown below." Below this, the XML code is displayed:

```
<MASKING_FORMAT_LIBRARY META_VER="1.0" PROD_VER="12.1.0.1.0">
- <FORMAT>
  <NAME>American Express Credit Card Number</NAME>
  <SENSITIVE_TYPE>CREDIT_CARD_NUMBER</SENSITIVE_TYPE>
  - <DESCRIPTION>
    ~10 billion unique American Express credit card numbers
  </DESCRIPTION>
  <RULE_ORDER>1 </RULE_ORDER>
  <RULE_CONDITION> 1=1 </RULE_CONDITION>
  - <FORMAT_ENTRY>
    <ORDER>1 </ORDER>
    <TYPE>RD</TYPE>
    <START>10</START>
    <END>10</END>
  </FORMAT_ENTRY>
  - <FORMAT_ENTRY>
    <ORDER>2 </ORDER>
    <TYPE>UT</TYPE>
    <UDF>DBSNMP.DM_FMTLIB.MGMT_DM_GEN_AC</UDF>
  </FORMAT_ENTRY>
</FORMAT>
- <FORMAT>
  <NAME>Colors</NAME>
  <SENSITIVE_TYPE>UNDEFINED</SENSITIVE_TYPE>
  <DESCRIPTION>Colors of the Rainbow</DESCRIPTION>
  <RULE_ORDER>1 </RULE_ORDER>
  <RULE_CONDITION> 1=1 </RULE_CONDITION>
  - <FORMAT_ENTRY>
    <ORDER>1 </ORDER>
    <TYPE>AL</TYPE>
    - <ARRAYLIST_ITEMS>
      <ITEM>Blue</ITEM>
      <ITEM>Indigo</ITEM>
      <ITEM>Orange</ITEM>
      <ITEM>Red</ITEM>
```

D. Summary

In this lab, you:

1. Created a custom masking format.
2. Exported the Masking Format Library for future use.

LAB EXERCISE 04 – MASKING SENSITIVE APPLICATION DATA

Identified Challenge – Exposure of Sensitive Application Data in Test, Q&A and Development Systems.

Personally Identifiable and sensitive data is being shared with parties that do not have a business-need-to-know in development and testing groups.

The use of operational databases containing personal information or any other sensitive information is being used for testing purposes. All identified sensitive details and content should be removed or modified beyond recognition before use.

INTRODUCTION

It is imperative to maintain the same level of confidentiality and protection of data when providing realistic test data to outside application developer and analysts. The Data Masking Pack (an add-on to Oracle Enterprise Manager) allows masking of data, utilizing a variety of flexible masking options, while preserving referential integrity and the normal, measurable action of applications.

A. Overview

This lab focuses on the **EMPLOYEES** and related tables, with the goal of protecting PII (Personally Identifiable Information) from outside developers who work on their HR Application.

A data masking definition is the association of tables and columns in a set of schema with masking format. The Data Masking definition contains a list of sensitive columns in the application tables, e.g. employee social security numbers, and its corresponding association with data masking formats, e.g. a fake social security number generator. In this example, we will choose the sensitive columns in our associated tables and then associate them with our masking formats.

During this lab you will:

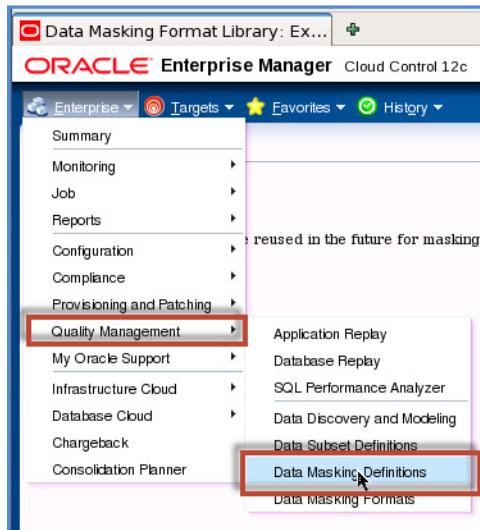
1. *Create a Masking Definition to mask the HR.EMPLOYEE data*
2. *Format columns using the Format Library and Masking Primitive*
3. *Generate data masking scripts*
4. *Execute the data masking script*
5. *Compared the pre-masked data vs. the post-masked data.*

B. Setup & Preparation

1. You should have already completed **Lab Exercise 01 – Creating an Application Data Model** and **Lab Exercise 02 – Identifying Sensitive Columns** before using this lab.

C. Masking Sensitive Application Data

1. Navigate to the Data Masking Definitions by selecting the menu **Enterprise → Quality Management → Data Masking Definitions**.



2. From the **Data Masking Definitions** Dialog, we will create a new definition. Click on the **Create** button to begin the process of masking data.

A screenshot of the 'Data Masking Definitions' dialog box. At the top, there is a brief description of what data masking is. Below it is a search bar with 'Masking Definition' and a 'Go' button. To the right of the search bar is an 'Import' button and a 'Create' button, which is highlighted with a red box. Underneath the search bar is a table header with columns: 'Select', 'Masking Definition', 'Application Data Model', 'Description', 'Columns', 'Status', 'Most Recent Job Ended', and 'SQL Performance Analyzer Task'. The table body below the header shows one row with the status 'No definitions'. At the bottom of the dialog, there is a section titled 'Format Library' with a brief description and a 'Format Library' link.

3. From the **Create Masking Definition** screen, fill in the **Name**, **Application Data Model** and **Description** field with the provided values below. Then click on the **Add** button.

Name: EMPLOYEE_DATA_MASK

Application Data Model: Employee Data

Reference Database: <Will be filled in for you>

Description: Mask Employee Data

3. At the **Database Login** screen, accept the Named Credential, and then click the **Login** button.

4. We are going to search the **EMPLOYEES** table in the **HR** Schema for Sensitive Columns that we have previously included in the Application Data Model. Type in the following values and click on the **Search** button.

Schema: HR

Table Name: EMPLOYEE

5. Select the column for **EMPLOYEE_ID** and click the **Add** button.

The screenshot shows the Oracle Database Masking Definition interface. In the search bar, 'Schema: HR' and 'Table Name: EMPLOYEES' are specified. A red box highlights the 'EMPLOYEE_ID' column under the 'Column Name' column. The 'Sensitive Column Type' is 'MASK_CANDIDATES'. The 'Data Type' is 'NUMBER'. The 'Comment' is 'MASK candidate: HR Benefits Policy'. Below the table, a message states: 'Add one or more columns for masking. Foreign key columns will be added automatically. You can define masking format at once for all selected columns if they have the same data type.' At the bottom right, there are 'Cancel', 'Define Format And Add' (highlighted with a red box), and 'Add' buttons.

6. Notice how all associated foreign key columns (6) were added automatically to this Masking Definition due to our earlier work.

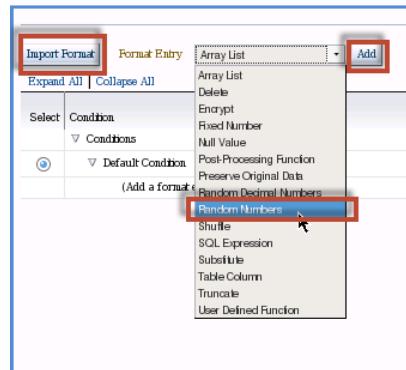
A yellow callout box contains the following message: 'Foreign key columns and/or dependent columns were added and will be masked the same way as the parent column. HR_EMPLOYEES.EMPLOYEE_ID -> HR.DEPARTMENTS.MANAGER_ID; HR.EMPLOYEES.MANAGER_ID; HR.JOB_HISTORY.EMPLOYEE_ID; HR.MANAGERS.MGR_ID; OE.CUSTOMERS.ACCOUNT_MGR_ID; OE.ORDERS.SALES REP_ID'.

The screenshot shows the 'Foreign Key Columns' section of the tool. It lists six foreign key columns: MANAGER_ID, EMPLOYEE_ID, EMPLOYEE_ID, EMPLOYEE_ID, MGR_ID, and EMPLOYEE_ID. The 'Format' column for the first row (MANAGER_ID) has a red box around it, indicating it is the target for formatting.

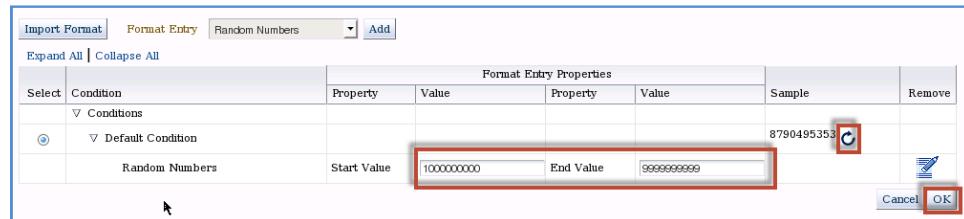
7. The next step is to format the **EMPLOYEE_ID** column. Continue by clicking on the icon.

The screenshot shows the 'Format' column for the 'EMPLOYEE_ID' column being targeted for formatting, indicated by a red box around the pencil icon.

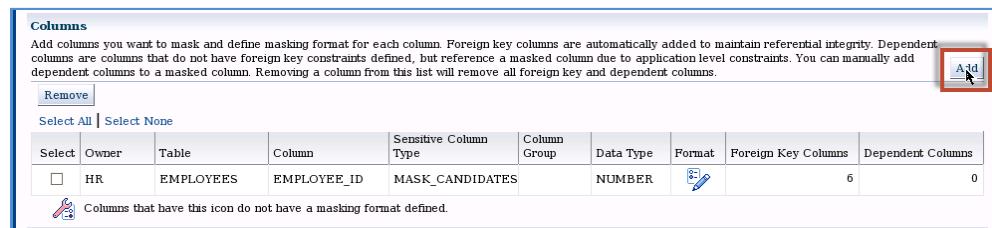
8. As previously discussed, there are many different options to format the column of data to ensure the quality of the data masking. If you were to use an existing format from the Format Library, you would click on the Import Format button. In this particular example, we are going to select **Random Numbers** from the drop down list box and click on the **Add** button.



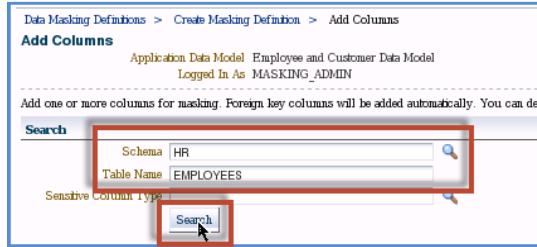
9. Enter 1000000000 for the Start Value and 9999999999 for the End Value. Click on the Sample icon to view sample data and continue by clicking the **OK** button.



10. The next step is to add additional columns in the **EMPLOYEES** table to include in this masking operation. Click the **Add** button to continue.



11. Set **HR** as the Schema and **EMPLOYEES** as the Table Name and click on the **Search** button to query for appropriate columns.



12. Add 4 columns in **HR.EMPLOYEES** for masking (**FIRST_NAME**, **LAST_NAME**, **PHONE_NUMBER**, **SALARY**). Select these 4 columns and click on the **Add** button.

Select	Owner	Table Name	Column Name	Sensitive Column Type	Data Type	Comment
<input type="checkbox"/>	HR	EMPLOYEES	EMAIL	MASK_CANDIDATES	VARCHAR2(100)	MASK candidate: HR Privacy
<input type="checkbox"/>	HR	EMPLOYEES	EMPLOYEE_ID	MASK_CANDIDATES	NUMBER	MASK candidate: HR Benefit Policy
<input checked="" type="checkbox"/>	HR	EMPLOYEES	FIRST_NAME	MASK_CANDIDATES	VARCHAR2(20)	MASK candidate: HR Privacy
<input checked="" type="checkbox"/>	HR	EMPLOYEES	LAST_NAME	MASK_CANDIDATES	VARCHAR2(25)	MASK candidate: HR Privacy
<input type="checkbox"/>	HR	EMPLOYEES	MANAGER_ID	MASK_CANDIDATES	NUMBER	Manager id of the employee; same domain as manager_id in departments table. Foreign key employee_id column of employee table. (useful for reflexive join CONNECT BY query)
<input type="checkbox"/>	HR	EMPLOYEES	NATIONAL_ID	SOCIAL_SECURITY_NUMBER	VARCHAR2(100)	
<input checked="" type="checkbox"/>	HR	EMPLOYEES	PHONE_NUMBER	PHONE_NUMBER	VARCHAR2(20)	Phone number of the employee; includes country code and area
<input checked="" type="checkbox"/>	HR	EMPLOYEES	SALARY	MASK_CANDIDATES	NUMBER(8,2)	MASK candidate: HR Compensation Policy

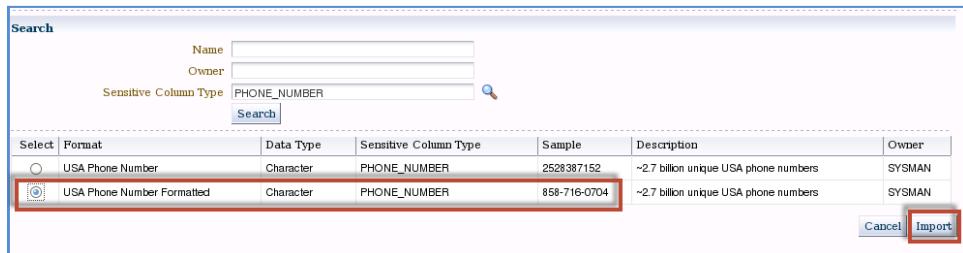
13. Now that we've added 4 more columns to mask, we need to define a masking format for each column. Click on the icon to define a masking format for the column **PHONE_NUMBER**.



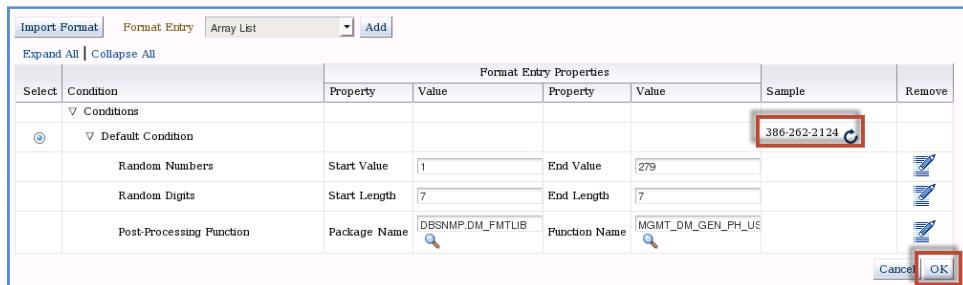
14. For the column **PHONE_NUMBER**, click on the **Import Format** button.



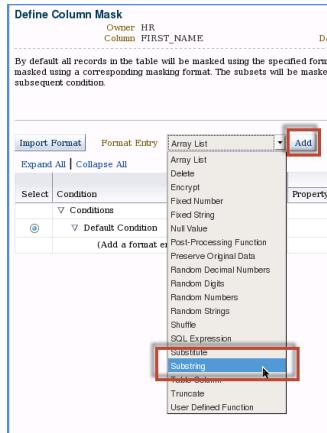
15. From the **Import Format** dialog, select **USA Phone Number Formatted** and click on the **Import** button.



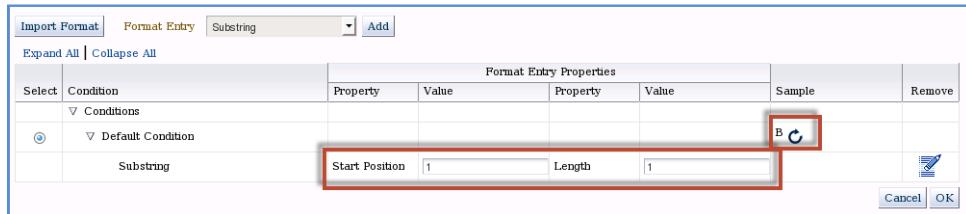
16. Review the Default Condition for the format masking for the **PHONE_NUMBER** column. Notice how Random Digits are generated for the Area Code and Phone number and then passed to an Oracle-supplied SQL Function for further processing. Click on the icon to review sample data from this format mask. Click on the **OK** button to continue.



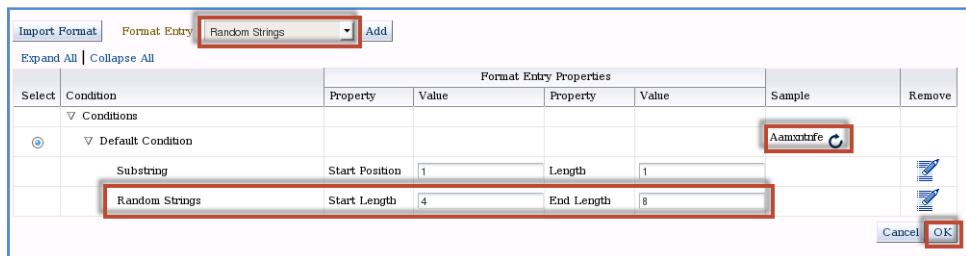
17. Continue by clicking on the icon to define a masking format for the column **FIRST_NAME**. We will construct the First Name as the first initial of unmasked user followed by a Random String of 4 – 8 characters. First, select Substring from the menu and click **Add**.



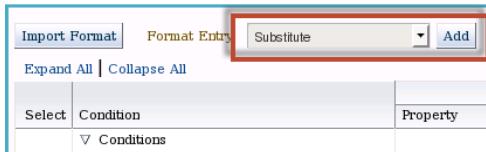
18. Select the first initial by defining a substring starting at Position 1 of Length 1. See the initial sample data generated.



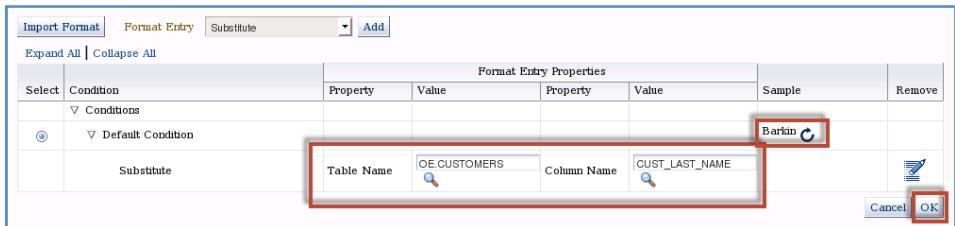
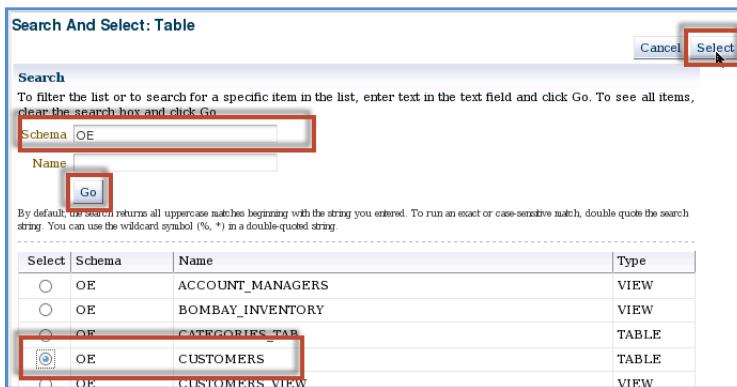
19. Now we'll generate the rest of the name. Select **Random Strings** from the dropdown and click **Add**. Then, fill in a Start Length of 4 and an End Length of 8. View several examples of the Sample data and then click **OK**.



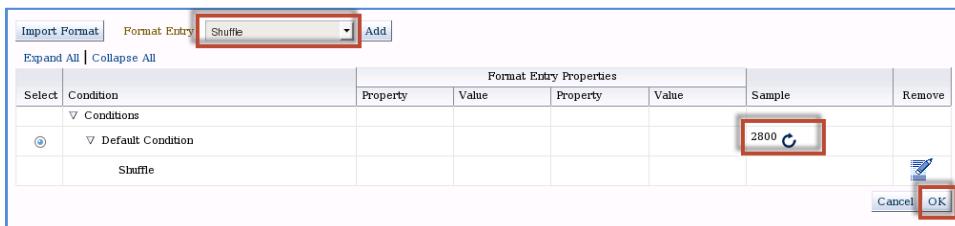
20. For the column **LAST_NAME**, we will pull in random last names from our **CUSTOMERS** table. Start by clicking on the icon. From the dropdown, pick **Substitute** and then **Add**.



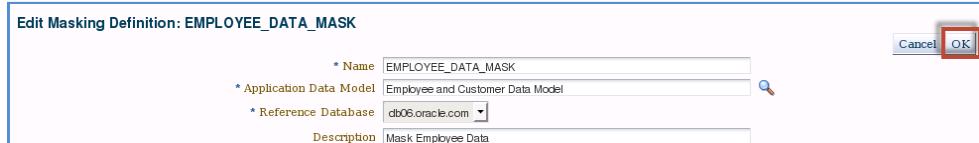
21. Click the icon to select the **OE.CUSTOMERS** table and then add the **CUST_LAST_NAME** column. Generate sample data several times to see the different last names. Then click **OK**.



22. Continue by clicking on the icon to define a masking format for the column **SALARY**. For this column, we will randomly Shuffle the original column data within the table. Select **Shuffle** from the dropdown list box and then click on the **Add** button. Review the Default Condition for the format masking for the **SALARY** column. Click on the icon to review sample data from this format mask. Click on the **OK** button to continue.

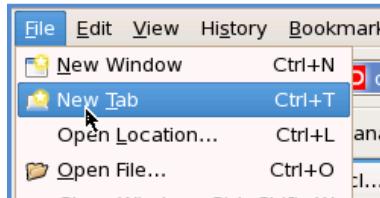


23. Click on the **OK** button to complete the creation of a Masking Definition for the **EMPLOYEES** table.



24. Review that you have now successfully created a Data Masking Definition.

25. Before we generate the Script to mask data, let's first query the existing unmasked data to compare the results after we mask the data. In the browser, select **File -> New Tab**.



26. In the new tab, click on the shortcut to go to **Enterprise Manager**.

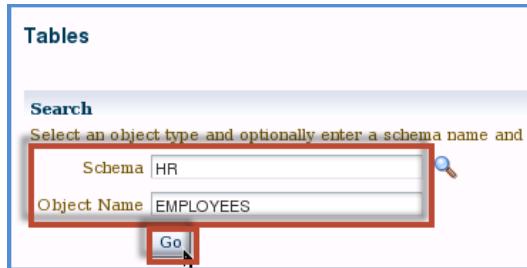


27. Navigate **EM** and select **Targets -> Databases -> db06.oracle.com**.

28. From the dropdown, select **Schema, Database Objects, Tables**.



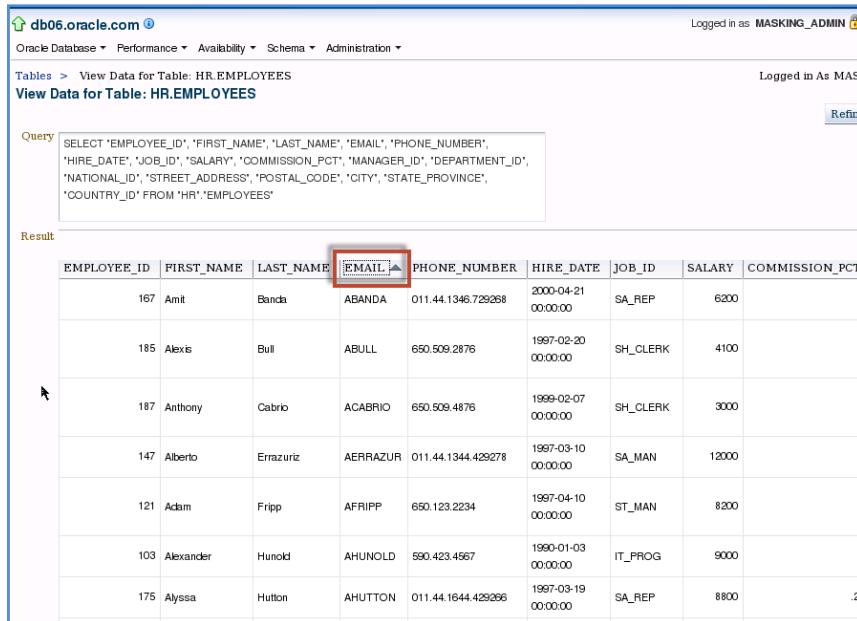
29. For the table search, enter **HR** for the Schema and **EMPLOYEES** for the Object name. Click **Go**.



30. Select **View Data** from the drop-down list box and click on the **GO** button.



31. Sort the data by EMAIL (by clicking on the column header). Leave this tab open so you can later reference the data as it was before the data masking operation is executed.



db06.oracle.com

Oracle Database Performance Schema Administration

Tables > View Data for Table: HR.EMPLOYEES

View Data for Table: HR.EMPLOYEES

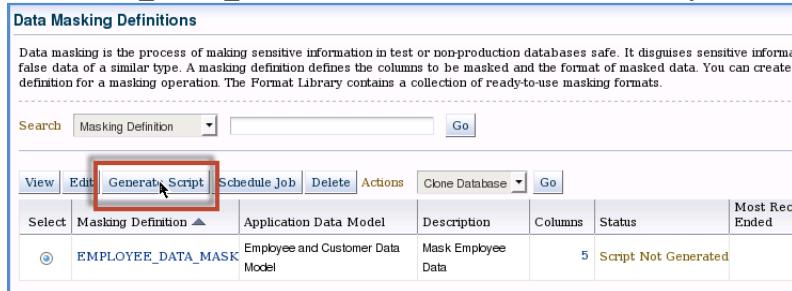
Query:

```
SELECT 'EMPLOYEE_ID', 'FIRST_NAME', 'LAST_NAME', 'EMAIL', 'PHONE_NUMBER',
'HIRE_DATE', 'JOB_ID', 'SALARY', 'COMMISSION_PCT', 'MANAGER_ID', 'DEPARTMENT_ID',
'NATIONAL_ID', 'STREET_ADDRESS', 'POSTAL_CODE', 'CITY', 'STATE_PROVINCE',
'COUNTRY_ID' FROM 'HR'.EMPLOYEES
```

Result:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT
167	Amit	Banda	ABANDA	011.44.1346.729268	2000-04-21 00:00:00	SA_REP	6200	
185	Alexis	Bull	ABULL	650.509.2876	1997-02-20 00:00:00	SH_CLERK	4100	
187	Anthony	Cabrio	ACABRIO	650.509.4876	1999-02-07 00:00:00	SH_CLERK	3000	
147	Alberto	Errazuriz	AERRAZUR	011.44.1344.429278	1997-03-10 00:00:00	SA_MAN	12000	
121	Adam	Fripp	AFRIPP	650.123.2234	1997-04-10 00:00:00	ST_MAN	8200	
103	Alexander	Hunold	AHUNOLD	590.423.4567	1990-01-03 00:00:00	IT_PROG	9000	
175	Alyssa	Hutton	AHUTTON	011.44.1644.429266	1997-03-19 00:00:00	SA_REP	8800	

32. Navigate back to the first browser tab. The next step is to select the **EMPLOYEE_DATA_MASK** and click on the **Generate Script** button.



Data Masking Definitions

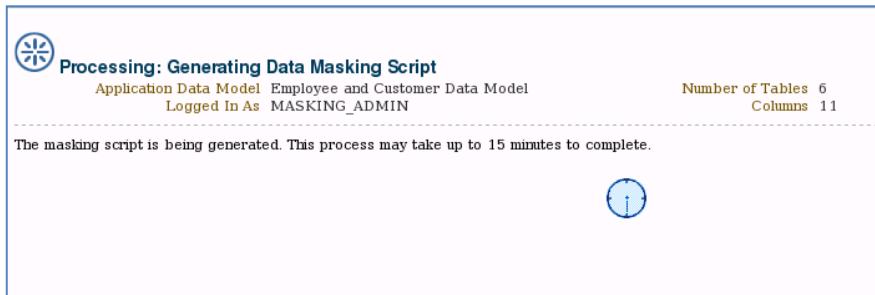
Data masking is the process of making sensitive information in test or non-production databases safe. It disguises sensitive information with false data of a similar type. A masking definition defines the columns to be masked and the format of masked data. You can create a masking definition for a masking operation. The Format Library contains a collection of ready-to-use masking formats.

Search Masking Definition Go

View Edit **Generate Script** Schedule Job Delete Actions Clone Database Go

Select	Masking Definition	Application Data Model	Description	Columns	Status	Most Recent
<input checked="" type="radio"/>	EMPLOYEE_DATA_MASK	Employee and Customer Data Model	Mask Employee Data	5	Script Not Generated	

33. After clicking on the **Generate Script** button, the data masking script will be generated.



34. You will be forwarded to the **Script Generation Results** page. There are a number of areas to explore. All of the highlighted buttons and actions can also be accessed on the Data Masking Definitions screen.

The screenshot shows the Oracle Database Script Generation Results page. At the top, it says "Data masking script generation completed successfully." Below that, it displays "Script Generation Results: EMPLOYEE_DATA_MASK" and "Number of Tables: 6 Columns: 11". Under "Script Options", there are buttons for "Clone And Mask" and "Schedule Job", both of which are highlighted with red boxes. The "Script" section contains a "View" button and a "Full Script" button, with "Full Script" being selected. The generated PL/SQL script is displayed below:

```
-- Target database: db06.oracle.com
-- Script generated at: 09-MAR-2012 10:58
COMMIT
ALTER SESSION ENABLE PARALLEL DML
DROP TABLE 'MGMT_DM_TT_150' PURGE
declare
adj number:=0;
num number:=0;
begin
select length(count(*)) into adj from (select distinct 'PHONE_NUMBER' from 'HR'.'EMPLOYEES');
num := adj;
adj := greatest(adj - 3, 0);
execute immediate 'create table MGMT_DM_TT_150
(org_val null, new_val null, delete_val null) NOLOGGING PARALLEL as
select CAST(null AS VARCHAR2(20)) org_val, CAST(null AS VARCHAR2(20)) new_val, CAST(0 AS NUMBER) delete_val from dual union all
select s.org_val,

```

Mask Validation and Execution

Oracle Data Masking Pack performs a series of validation steps to ensure that the data masking process proceeds to a successful completion without errors. One of the checks that it performs is validating the masking formats. This is a necessary step in the data masking process to ensure that the chosen masking formats meet the database and application integrity requirements. These requirements may include generating unique values for the column being masked because of uniqueness constraints or generating values that meet the column length or type requirements.

Upon successful completion of the validation check, Oracle Data Masking Pack generates the PL/SQL-based masking script that is transferred to the target database for execution. Oracle Data Masking Pack uses a highly efficient and robust mechanism to create masked data. Oracle Data Masking Pack performs bulk operations to rapidly replace the table containing sensitive data with an identical table containing masked data while retaining the original database constraints, referential integrity and associated access structures, such as INDEXes and PARTITIONs, and access permissions, such as GRANTS. Unlike masking processes that are traditionally slow because they perform table updates, Oracle Data Masking Pack takes advantage of the built-in optimizations in the database to disable database logging and execute in parallel to quickly create a masked replacement for the original table. The original table containing

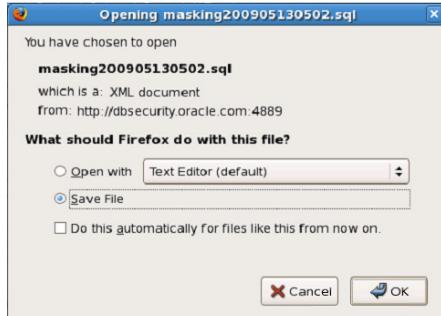
sensitive data is dropped from the database completely and is no longer accessible.

35. Scroll down to the bottom of the page and expand the **Impact Report** section. The Impact Report will provide a summary of the script generation and important details about the objects and resources necessary to complete the job successfully. If there are any issues here, they should be corrected before moving forward.

The screenshot shows the 'Impact Report' section of a database cloning tool. It includes a 'Script Generation Summary' table with columns for Object Name, Object Type, Message Severity, Message Type, and Message. The summary table provides details on free space in tablespaces EXAMPLE, USERS, HR, and OE. A note at the bottom states: 'The following table provides information about the objects and resources examined during script generation and lists details of any warnings or errors detected.'

Object Name	Object Type	Message Severity	Message Type	Message
EXAMPLE	TABLESPACE	INFORMATION	Plan	Sufficient free space in Tablespace EXAMPLE. Starting Freespace with automatic extension: 968MB. Ending Freespace: 967MB. Lowest Freespace: 967MB.
USERS	TABLESPACE	INFORMATION	Plan	Sufficient free space in Tablespace USERS. Starting Freespace with automatic extension: 33550MB. Ending Freespace: 33550MB. Lowest Freespace: 33550MB.
HR	USER	INFORMATION	Plan	Sufficient tablespace quota for User HR.
OE	USER	INFORMATION	Plan	Sufficient tablespace quota for User OE.

36. Scroll back up the page and click on the **Save Full Script** button. Take note of the file name of the .sql file to review in detail later. This script could be taken and executed on other targets.



37. Click on the **Clone Mask** button under the **Script Option** section. Review the number of supported options to clone the database and create a staging environment for the script to be executed and data to be masked.

The screenshot shows the 'Clone Database: Source Type' dialog box. It asks 'Specify the type of source database backup that will be used for the cloning operation.' with four options: 'Online Backup' (selected), 'Use Recovery Manager (RMAN) to copy database files' (with a note about staging areas), 'Copy database files via staging areas' (with a note about requiring staging areas on both source and destination hosts), and 'Existing Backup'. Below this is a 'TIP' note: 'A snapshot standby database is an alternative for running repetitive testing scenarios. Create Snapshot Standby Database.' On the right side, there is an 'Overview' panel with sections for cloning options (using RMAN or existing backups) and a note about cloning database files via staging areas. At the bottom are 'Cancel' and 'Continue' buttons.

38. Click on the browser's back button to return to the previous screen and click on the **Schedule Job** button to immediately schedule and run the masking operation. Select and accept the default Named Credentials (EXA1_ORACLE_SSH) for both the Host and the Database (NC_DB06.MASKING). You will also be prompted for a Seed for the Substitute Mask Format. Use something simple like "data". Click on the **Submit** button to execute the job.

Schedule Data Masking Job: EMPLOYEE_DATA_MASK

Application Data Model: Employee and Customer Data Model
Logged In As: MASKING_ADMIN
Number of Tables: 6 Columns: 11

* Job Name: MASKING_JOB_171
Job Description:
* Reference Database: db06.oracle.com
* Script File Location: /u01/app/oracle/product/11.2.0/dbhome_1/sec/dbs/
* Script File Name: masking171.sql

Encryption Seed
A seed is required for masking definitions that use the Substitute format. The seed can be any text string.
* Seed: ...
* Confirm Seed: ...

Host Credentials
Enter credentials to login to the database host.
Credential: Preferred Named New
Credential Name: EXA1_ORACLE_SSH
Attribute: UserName: oracle
 Password: *****
 More Details

Database Credentials
Enter credentials to login to the reference database.
Credential: Preferred Named New
Credential Name: NC_DB06.MASKING
Attribute: Username: MASKING_ADMIN
 Value

39. Once you submit the job, you will be forwarded to a confirmation page that the job was submitted successfully. Click on the **Go** button to refresh the status of the job.

Warning
db06.oracle.com is a Database Vault enabled database. Ensure you have privileges to do masking operations.

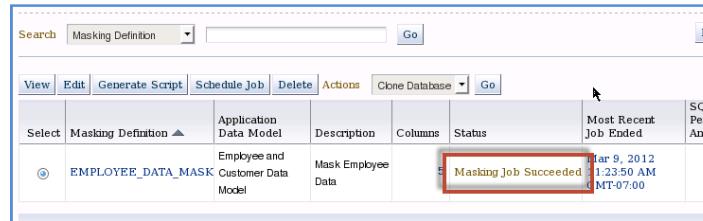
Data Masking Definitions
Data masking is the process of making sensitive information in test or non-production databases safe. It disguises sensitive false data of a similar type. A masking definition defines the columns to be masked and the format of masked data. You can definition for a masking operation. The Format Library contains a collection of ready-to-use masking formats.

Search: Masking Definition Go

View	Edit	Generate Script	Schedule Job	Delete	Actions	Clone Database	Go
Select	Masking Definition	Application Data Model	Description	Columns	Status	M	E
<input checked="" type="radio"/>	EMPLOYEE_DATA_MASK	Employee and Customer Data Model	Mask Employee Data	5	Masking Job Scheduled		

Format Library
A masking format defines the format of masked data. You can create a new masking format and reuse it later when creati

40. Once the job successfully completes, follow the provided steps again to create a new tab and query the masked data for a before and after comparison.



41. Toggle between the two browser tabs and review the data before the masking job and after the successful masking operation of the 5 columns defined. Sort the new data by EMAIL to explicitly see the differences.

Masked Data:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	CO
2976384007	Aaabqgek	Russell	ABANDA	618-156-8007	2000-04-21 00:00:00	SA_REP	6100	
149073033	Aaaaaaxdv	Moranis	ABULL	616-752-3054	1997-02-20 00:00:00	SH_CLERK	7800	
6372072104	Aaaaadkdp	Hershey	ACABRIO	305-854-9030	1999-02-07 00:00:00	SH_CLERK	9000	
1773202020	Aaaaaauch	Sanders	AERRAZUR	916-513-7037	1997-03-10 00:00:00	SA_MAN	17000	
2949114016	Aaaaaaccb	Koira	AFRIPP	213-876-6098	1997-04-10 00:00:00	ST_MAN	8600	

Unmasked Data:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	CO
167	Amit	Banda	ABANDA	011.44.1346.729268	2000-04-21 00:00:00	SA_REP	6200	
185	Alexis	Bull	ABULL	650.509.2876	1997-02-20 00:00:00	SH_CLERK	4100	
187	Anthony	Cabrio	ACABRIO	650.509.4876	1999-02-07 00:00:00	SH_CLERK	3000	
147	Alberto	Errazuriz	AERRAZUR	011.44.1344.429278	1997-03-10 00:00:00	SA_MAN	12000	
121	Adam	Fripp	AFRIPP	650.123.2234	1997-04-10 00:00:00	ST_MAN	8200	

D. Additional Steps

1. Explore the generated .sql data masking script to better understand the operations during the execution of the masking function.

E. Summary

In this lab, you:

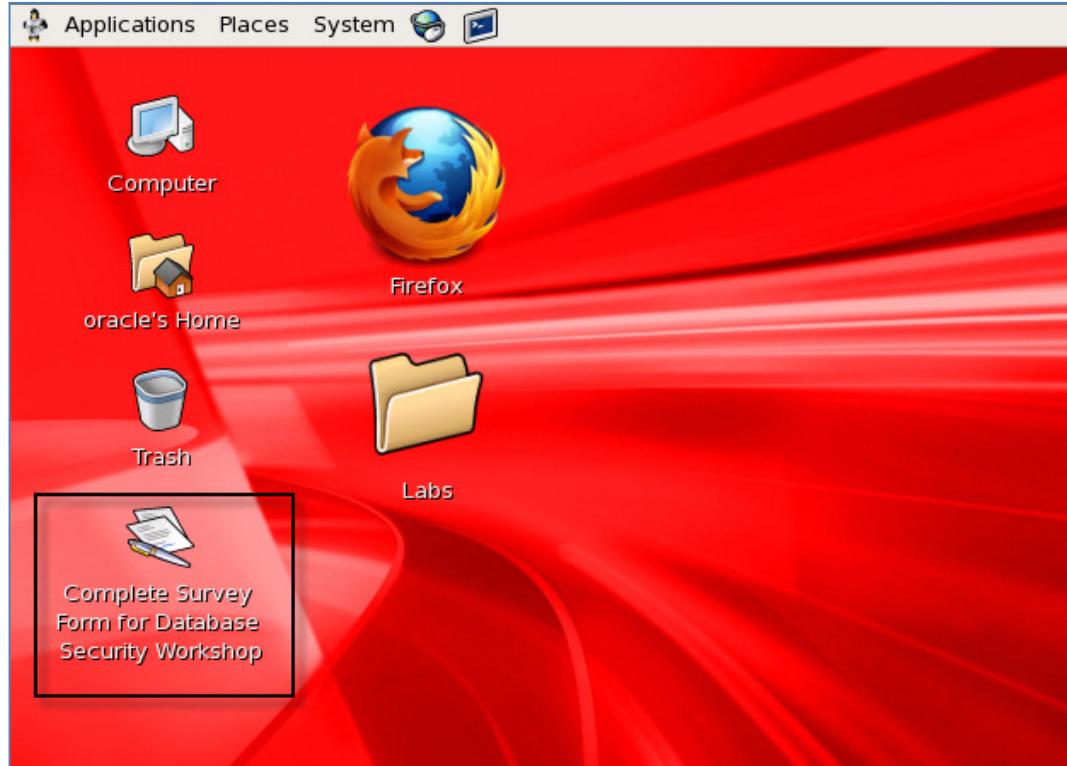
1. Created a Masking Definition to mask the HR.EMPLOYEE data
2. Formatted columns using the Format Library and Masking Primitives
3. Generated data masking scripts
4. Executed the data masking script
5. Compared the pre-masked data vs. the post-masked data.

PLEASE COMPLETE THE FEEDBACK SURVEY

Thank you!! For taking the time to participate in the 11g Database Security Hands-on Workshop. We hope you found the experience valuable.

If you could please quickly provide us your feedback, we would greatly appreciate it.

1. Go to the Desktop and Click on the icon, Workshop Survey Online.



2. If the icon is not on the Desktop, please go to the URL:
<https://www.surveymonkey.com/s/oracledatabasesecurity>
3. Complete the Survey.

Thank You again for joining us!!!!