

Installing and Configuring Guardium, ODF, and OAV

In this appendix, we will cover the following topics:

- ▶ IBM Infosphere Guardium Database Security
- ▶ Oracle Database Firewall
- ▶ Oracle Audit Vault

IBM Infosphere Guardium Database Security

A Database Activity Monitor (DAM) is a non-intrusive system that implements real time monitoring and alerting for various databases.

Non-intrusive monitoring systems are implemented by the vendor at the network communication and database shared memory level, and therefore a direct connection to the database to query or look for audit information is not required.

Generally these systems may have additional protection and prevention capabilities such as blocking unauthorized access to data or blocking access to data which is violating a defined access policy (for example, do not run a query more than three times in a minute).

Guardium security life cycle

The Guardium life cycle can be summarized as follows:

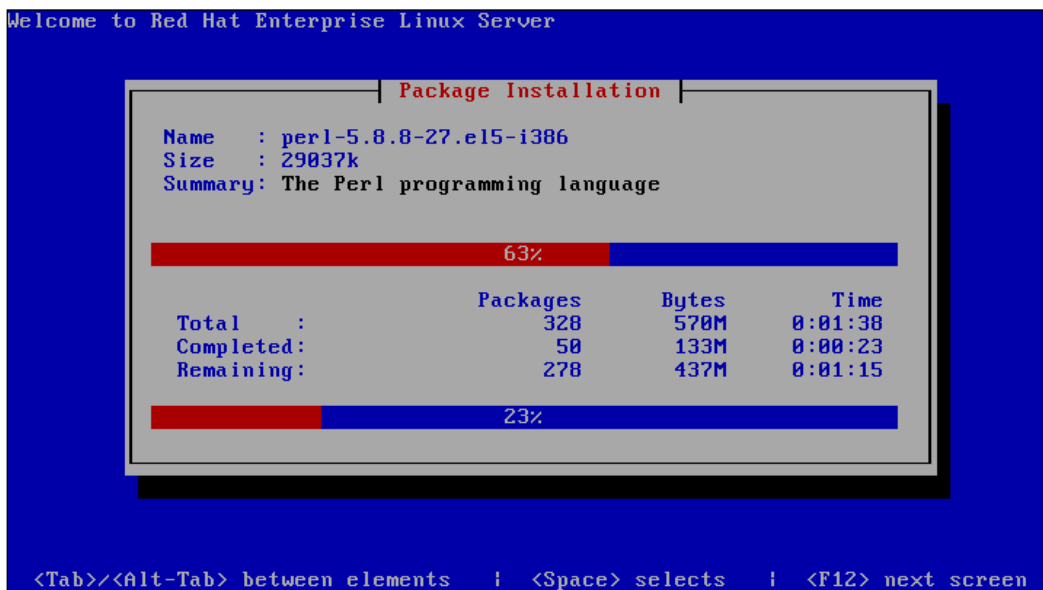
- ▶ **Discover and classify:** Discover all databases, applications, and clients. Discover and classify sensitive data.

- ▶ **Access and harden:** Vulnerability assessment, configuration assessment, behavioral assessment, create baseline, configuration lock-down, and change tracking.
- ▶ **Monitor and enforce:** Non-intrusive monitoring, policy-based actions, anomaly detection, real-time prevention, and granular access controls.
- ▶ **Audit and report:** Centralized governance, compliance reporting, sign-off management, automated escalations, secure audit warehouse, data mining for forensics, and long-term retention.

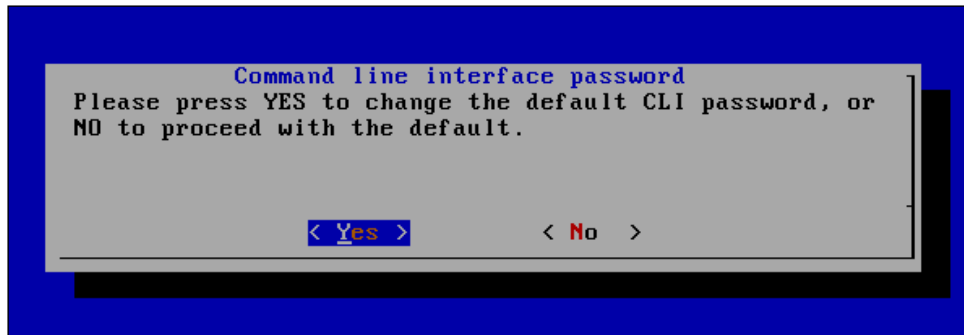
Installation and configuration

Guardium is generally recommended to be installed on a dedicated system or is usually delivered as appliances. The installation kit consists of a customized Linux and application package. It is important to remember to have allocate sufficient free space on the installation drive (about 180 GB free space, or else the installation will fail). The installer will remove and will use all available space.

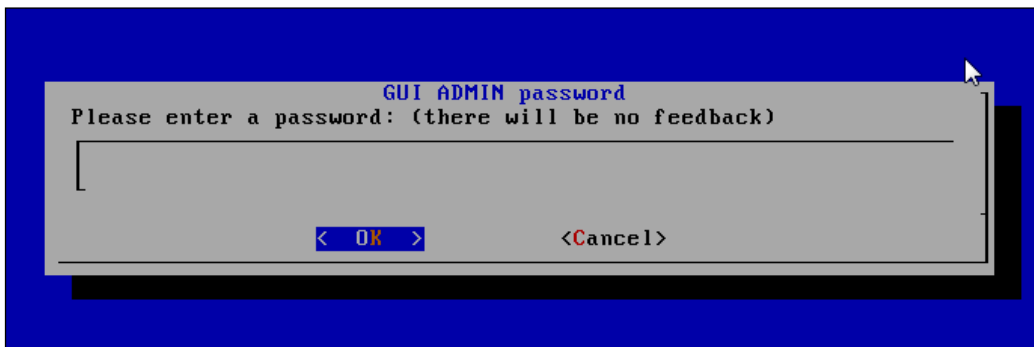
1. Insert the CD or mount the installation kit if you install Guardium on virtual machines. The installation will start by creating the storage layout followed by the Linux and Guardium packages installation, as shown in the following screenshot:



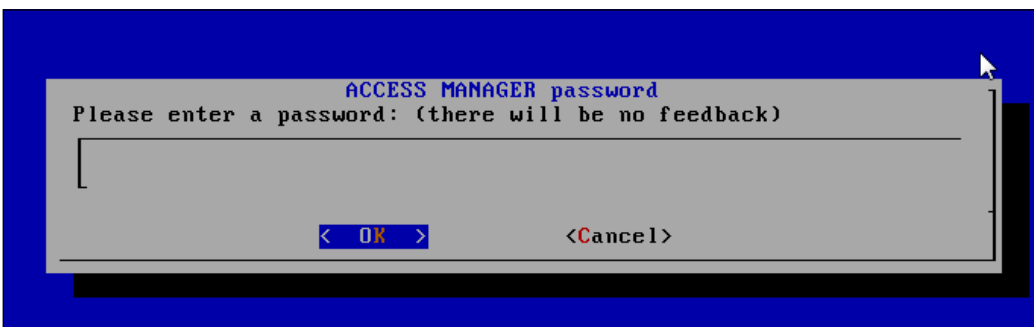
- When installation of the packages is completed, you'll need to introduce temporary user passwords for CLI, ADMIN, and ADMIN_MGM users. These passwords will be changed at the first login. If you do not change the password for the user **CLI** during installation, the default temporary password `guardium` will be used.



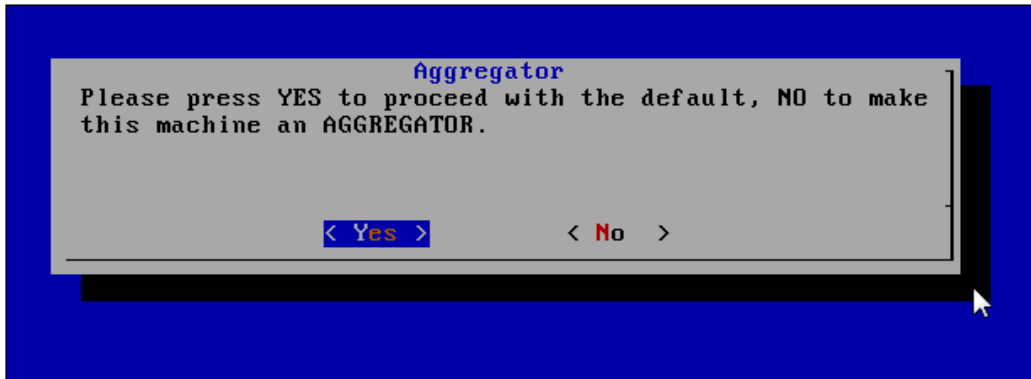
- Enter a temporary password for the user **GUI ADMIN**.



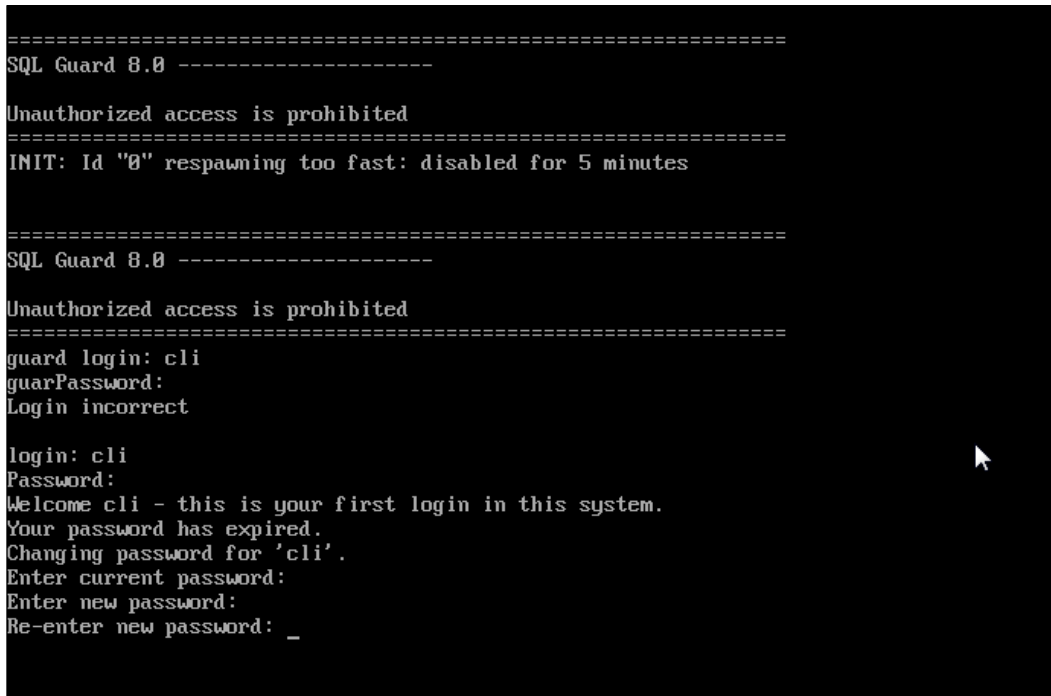
- Enter a temporary password for the user **ACCESS MANAGER**.



- For a standalone installation, chose the default option, Collector, otherwise if you want to run Guardium as an aggregator choose **No**.



- At this moment we have finished installing Guardium. The system will reboot automatically. Next, we will proceed to the network configuration phase. Connect as the **cli** user with the password `guardium`. At this step, it is mandatory to change the password.



7. Configure the IP address and network mask:

```
store network interface ip <your ip>
store network interface mask <your ip's corresponding mask>
```

```
=====
SQL Guard 8.0 -----
Unauthorized access is prohibited
=====
INIT: Id "0" respawning too fast: disabled for 5 minutes
=====
SQL Guard 8.0 -----
Unauthorized access is prohibited
=====
guard login: cli
guarPassword:
Login incorrect

login: cli
Password:
Welcome cli - this is your first login in this system.
Your password has expired.
Changing password for 'cli'.
Enter current password:
Enter new password:
Re-enter new password:
Setting default expiration period to 90 days.
guard.yourcompany.com> store network interface ip 10.241.132.20
This change will take effect after the next reboot.
ok
guard.yourcompany.com> store network interface mask 255.255.255.0
This change will take effect after the next reboot.
ok
guard.yourcompany.com>
```

8. To make these values active, and persistent we have to reboot the system. In the CLI command prompt execute the following command:

```
restart system
```

Deployment and configuration of S-TAP agents

Database and system monitoring is performed by using and deploying agents on each database server host. These agents have the generic name of S-TAP. Installation and configuration of these agents can be made using an interactive installer. After the agents are deployed and are communicating with the Guardium server they can be configured remotely from the administration console local S-TAPs as follows:

- ▶ As the user root starts the S-TAP installer:

```
<db2_shmem_client_position> = <sp>
<db2_shmem_size> = <131072>
<db2bp_path> = <NULL>
<db_exec_file> = </u01/app/oracle/product/11.2.0/dbhome_1/bin/oracle>
<db_install_dir> = </u01/app/oracle/product/11.2.0/dbhome_1>
<db_type> = <ORACLE>
<encryption> = <db>
<exclude_networks> = <192.168.1.115/255.255.255.255,192.168.1.16/255.255.255.255>
<informix_version> = <9>
<intercept_types> = <NULL>
<load_balanced> = <1>
<networks> = <10.241.132.0/255.255.255.0>
<port_range_end> = <4100>
<port_range_start> = <4100>
<real_db_port> = <1521>
<tee_listen_port> = <12344>
<unix_domain_socket_marker> = <NULL>
Parsing DB section DB_HACKDB
IP for 127.0.0.1 is 0100007f
hostname is nodeorcl1
Local IPs
ip = 127.0.0.1
ip = 10.241.132.80
Using infile /usr/local/guardium/guard_stap/guard_tap.ini, backup file /usr/local/guardium/guard_stap/guard_tap.ini.bak
Buffer attached, starting at 16(0x10), next free 16(10)
filter string: <( tcp and ( not ( ( ( net 192.168.1.115 mask 255.255.255.255 ) and ( tcp[0:2] = 4100 or tcp[2:2] = 4100 ) ) or ( ( net 192.168.1.16 mask 255.255.255.255 ) and ( tcp[0:2] = 4100 or tcp[2:2] = 4100 ) ) and ( ( net 10.241.132.0 mask 255.255.255.0 ) ) and port not 16016 ) ) >
SERIALIZATION:-----
10.241.132.80 1 1 1 0 85 18 connect to ip 127.0.0.1 db2 fix_pack.adjustment 20 db2_shmem_client_position 0 db2_shmem_size 131072 db2bp_path NULL db_exec_file /u01/app/oracle/product/11.2.0/dbhome_1 db_type ORACLE encryption 0 informix_version 9 instance_running 1 intercept_types NULL load_balanced 1 port_range_end 4100 port_range_start 4100 real_db_port NULL 1 2 10.241.132.0 255.255.255.0 192.168.1.115 255.255.255.255 192.168.1.16 255.255.255.255
SER-----
Guardium STAP config file OK.

Your configuration has been validated.

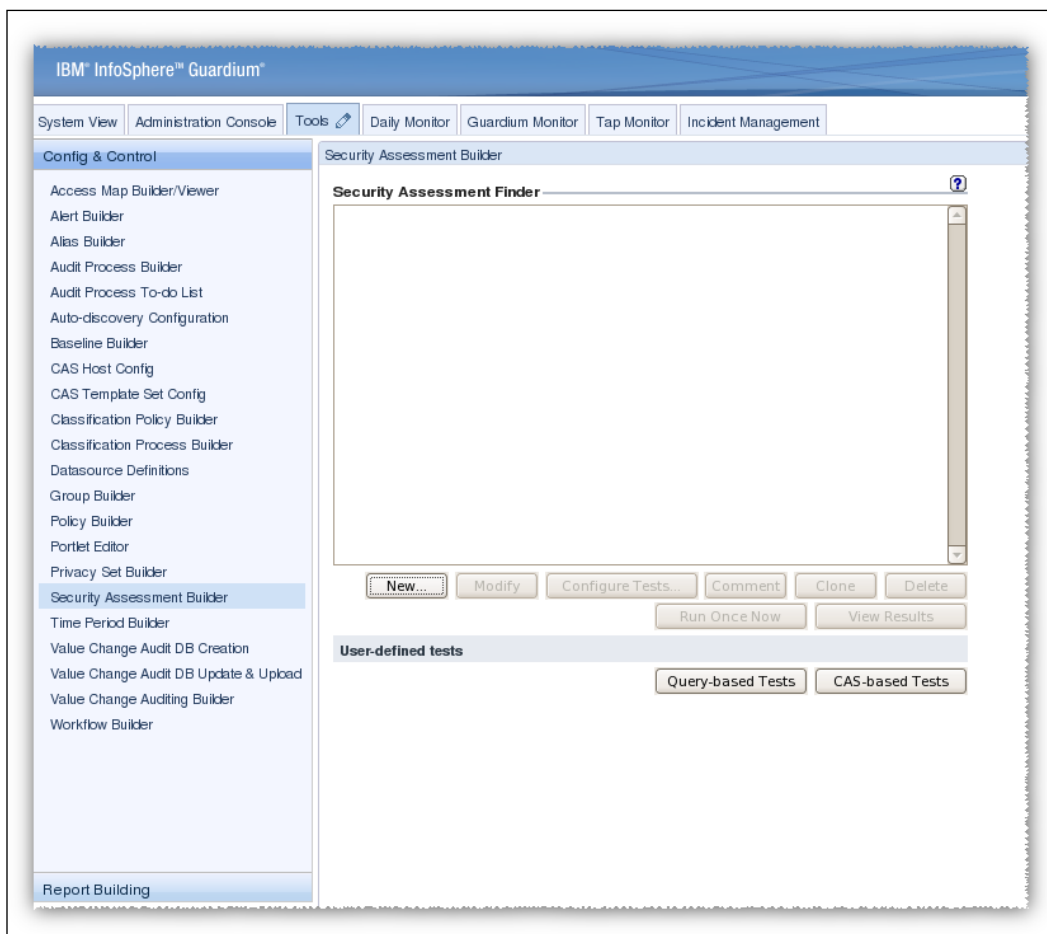
We have been able to provide you with a working copy of L50F, it has run
Starting Ktap module
Searching for modules in /usr/local/guardium/guard_stap/ktap/current/modules-*.tgz
guard_ktap_loader: Module ktap-v82_r33264_1-rh5u3x64m-2.6.18-128.el5-x86_64-SMP.ko selected for kernel 2.6.18-128.el5.
If you later update your kernel to another version, we can
try to load the closest fitting delivered module. This
feature is not enabled by default, but we recommend enabling
it to reduce delays in support.
Do you wish to enable this feature (y/N/h)? y
Extracted module ktap-v82_r33264_1-rh5u3x64m-2.6.18-128.el5-x86_64-SMP.ko from /usr/local/guardium/guard_stap/ktap/current/modules-v82_r33264_1.tgz
KTAP loaded
guard_hnt is using this perl binary: /usr/bin/perl
guard_hnt is using this lsaf binary: /usr/sbin/lsaf
Install finished
[root@nodeorcl1 Shell Installers]#
```

- ▶ Next the installer will perform the installation of all the libraries. The last step is the configuration of the S-TAP init file. S-TAP configuration is contained in a file named `s-tap.init`. The installer will open this file for the purpose of editing with `vi` and this will require to set all the mandatory parameters.
- ▶ To verify and check that the agents are running and communicating with the Guardium server, log in to the Guardium console, check the status of agents. Their status should be green in the console.

Performing a vulnerability assessment

Usually performing a vulnerability assessment is one of the first few steps for securing and defending a database. There are many types of vulnerabilities based on bugs or incorrect configuration.

1. To perform a vulnerability assessment we must connect to the administration console and navigate to **Tools** and in the **Config & Control** panel click on the **Security Assessment Builder** link, as shown in the following screenshot:



2. We must first define the source on which we will run the security assessment. The configuration of a data source is straightforward. In our case, we have the following configuration. In this panel we can test whether we can connect. We have used the system user. If the network data is ready to go, click on the **Apply** button, as shown in the following screenshot:

https://10.241.132.30:8443/datasourceEditAction.do

Datasource Definition

Name: HACKDB_SEC_ASSESSMENT

Database Type: Oracle (DataDirect)

Severity classification: HIGH

Description:

Share Datasource:

Authentication

Save Password:

Login Name: system

Password:

Location

Host Name/IP: 10.241.132.80

Port: 1521

Service Name: HACKDB

Informix Server:

Schema:

Connection Property:

Custom Uri:

CAS

Database Instance Account:

Database Instance Directory:

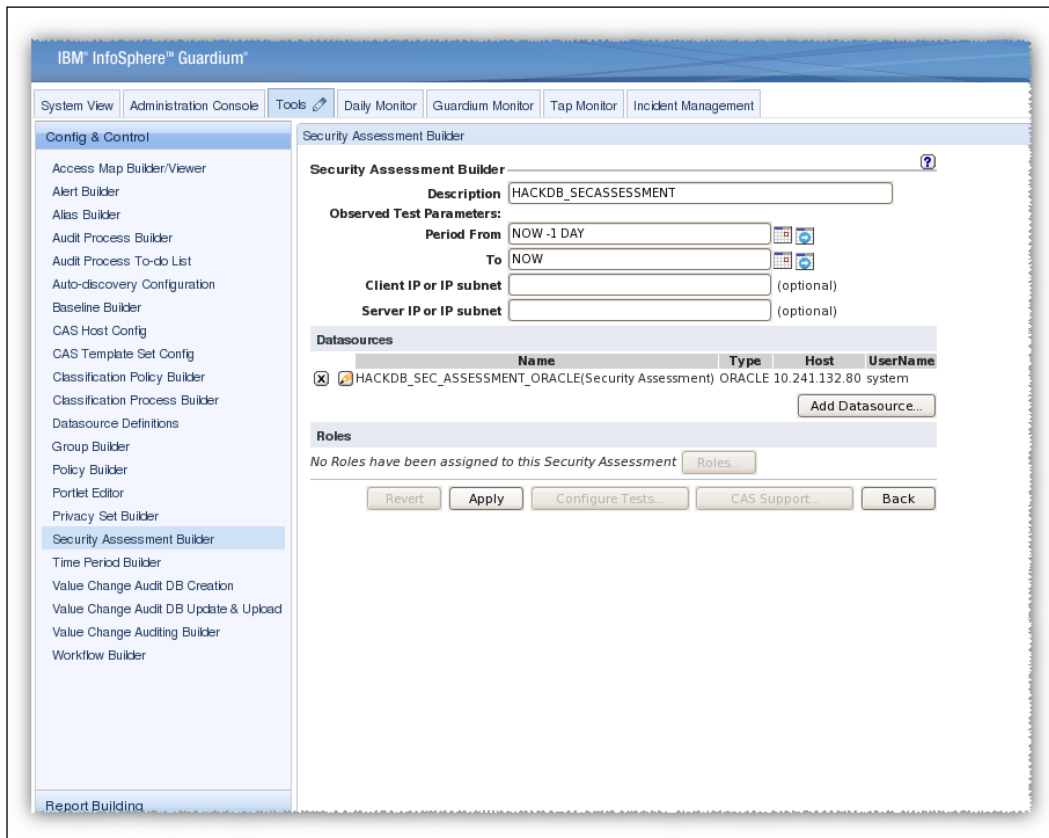
Roles

No roles have been assigned to this datasource. Roles...

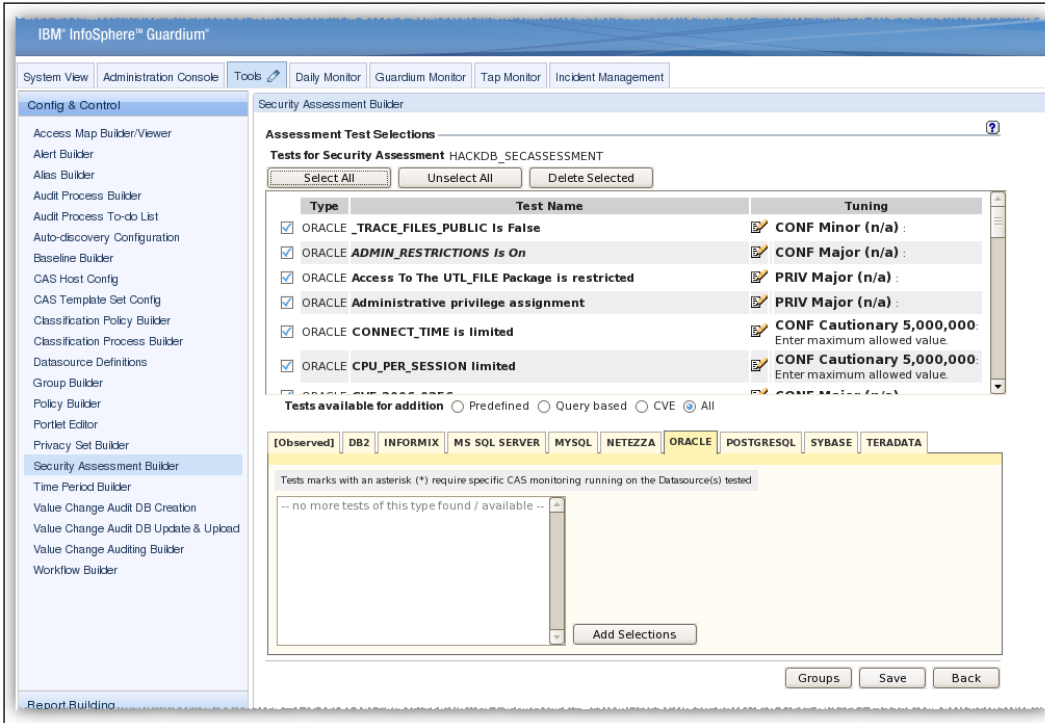
Add Comments Test Connection Apply Back

Done 10.241.132.30:8443

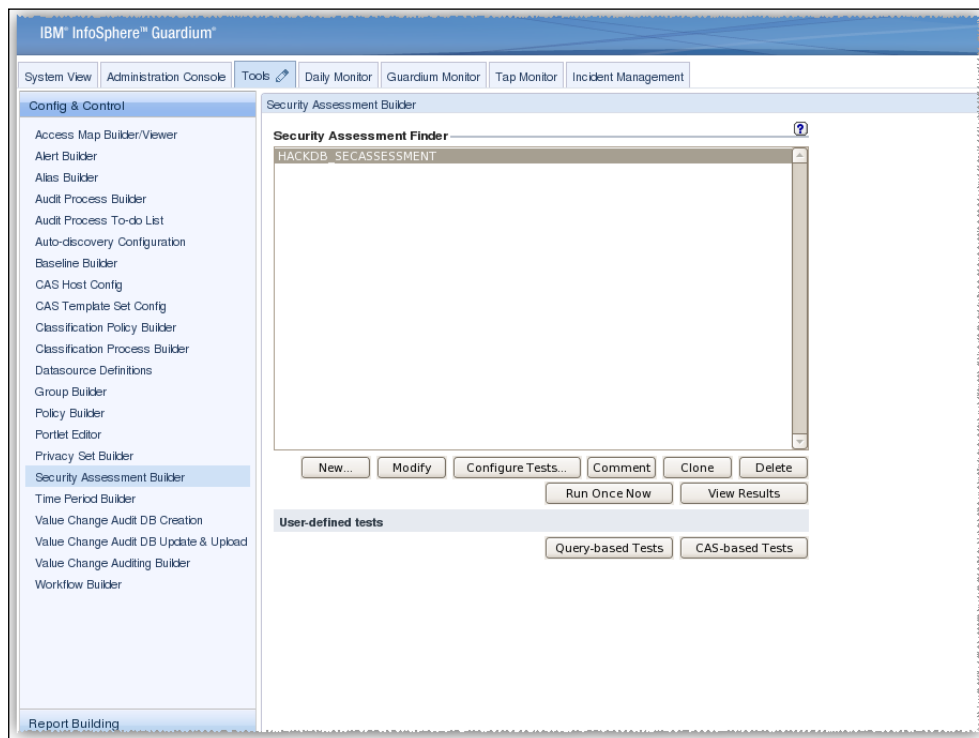
- Our configured data source will appear as the target for the security assessment, then click on the **Apply** button, as shown in the following screenshot:



- Next choose the vulnerabilities to be checked. In the **Test available for addition** option, check the option **All**. Then click on the **ORACLE** tab and select all the vulnerabilities from the list box found on this panel and click on the **Add Selections** button. Click on the **Select All** button and click on **Save**. This is shown in the following screenshot:



- A page will appear with our security assessment defined. Click on the **Run Once Now** button to perform a security assessment.



The result for the security assessment with details and scores is generated as follows:



Perform the necessary correction as the security assessment report advises and repeat the assessment until you have a 100 percent score.

Oracle Database Firewall

Oracle Database Firewall monitors traffic at network level using SQL grammar-based technology. In practice, it dissects the network packet and checks the SQL statements issued by the clients. It is a heterogeneous technology with support for monitoring DB2, MS SQL, MySQL, and Sybase databases. Depending on the traffic registered in a period of time, policies can be defined using a tool called **Oracle Firewall Analyzer**. The definition of policies is largely based on baselines. We may have white lists, black lists, and exceptions. **White listed statements** are a category of statements that may pass from clients to servers without any restriction. **Black listed statements** are a category of statement that may not pass and that are blocked by Oracle Database Firewall. **Exceptions** are a category of statements that can be exempted from a policy.

Policies can also be associated with additional inspection criteria such as the time of day, IP address, and username in order to generate more complex policies when needed.

Traffic can be monitored by interposing an Oracle Database Firewall between a client and a server (in-line monitoring) by using bridged or proxy traffic sources or by using network taps (out of band monitoring). Monitoring can be made either in passive mode, Database Activity Monitor (DAM) mode, reactive mode, or Database Policy Enforcement (DPE) mode. There is also support for remote monitoring using monitoring agents and local connections for statements issued locally using local agents.

Along with monitoring capabilities, Oracle Database Firewall provides real-time alerting and reporting capabilities and provides built-in custom modules to verify compliance with regulatory requirements such as **Sarbanes-Oxley (SOX) Act**, **Payment Card Industry Data Security Standard (PCI DSS)**, and **Health Insurance Portability and Accountability Act (HIPAA)**.

Installation and configuration

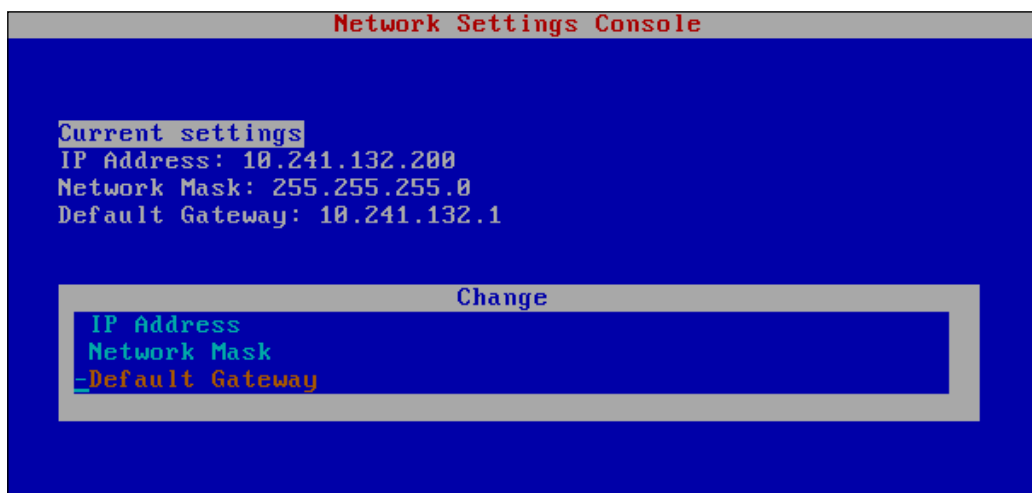
Oracle Database Firewall consists of a custom installation of Oracle Enterprise Linux infrastructure, ODF packages, customized WebLogic application server, and a database which is used as a repository for items such as policies, rules, and exceptions. For a more complex installation where we want to use more than one Database Firewall environment it is recommended to install the Oracle Database Firewall Management server used for centralized management. Depending on the current network configuration in your organization you can decide on the type of monitor configuration to use. We used a configuration with three network cards, one dedicated to ODF management and two for implementing bridged network configuration. In a bridged network configuration all traffic from clients will pass through the network devices configured on ODF machine. We used something similar with the following setup described at this link.

The installation kit for version 5.1 used in our description consists of the following:

- ▶ Oracle Linux Release 5 Update 5 for x86 (32 Bit) - DVD
- ▶ Oracle Database Firewall Management Server 5.1 (ISO)
- ▶ Oracle Database Firewall 5.1 - Disc 1 (ISO)
- ▶ Oracle Database Firewall 5.1 - Disc 2 (ISO)
- ▶ Oracle Database Firewall 5.1 - Disc 3 (ISO)
- ▶ Oracle Database Firewall Utilities 5.1 (ISO)

The installation and configuration steps are as follows:

1. On the dedicated server for DBF, insert Disc 1 and boot from it. If you want to install Oracle Firewall Management server, insert this disc.
2. After the system is booted it will require the disc with **Oracle Enterprise Linux (OEL)**. It will perform the creation of the layout and install the core Linux system.
3. Next Disc 2 and Disc 3 will be required and the installation will continue.
4. Finally Disc 1 will be required again and the installation will finalize with configuration steps.
5. Next we have to configure the IP address for Oracle Database Firewall and the gateway if it is the case:

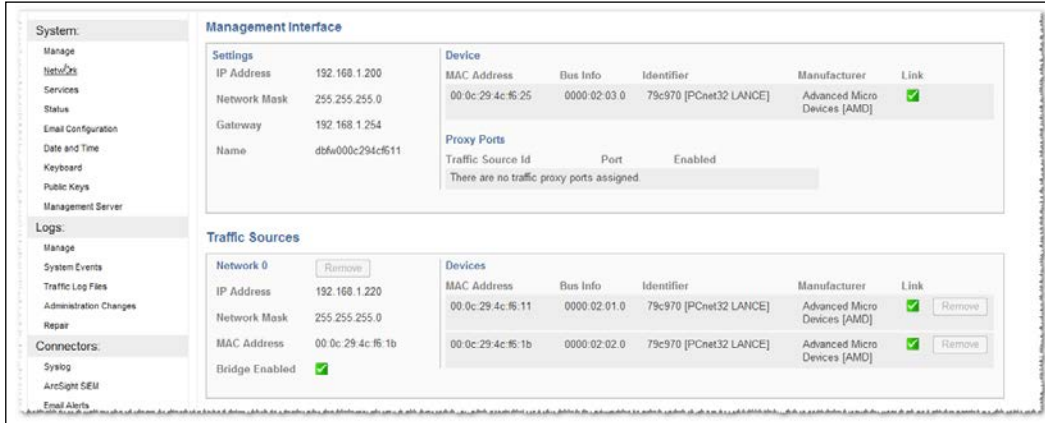


Adding and configuring protection for databases

From the client host, open a browser and type the DBF management host and port number (the default is 80) and log in as an admin user with an admin temporary password. At this step a password change is required.

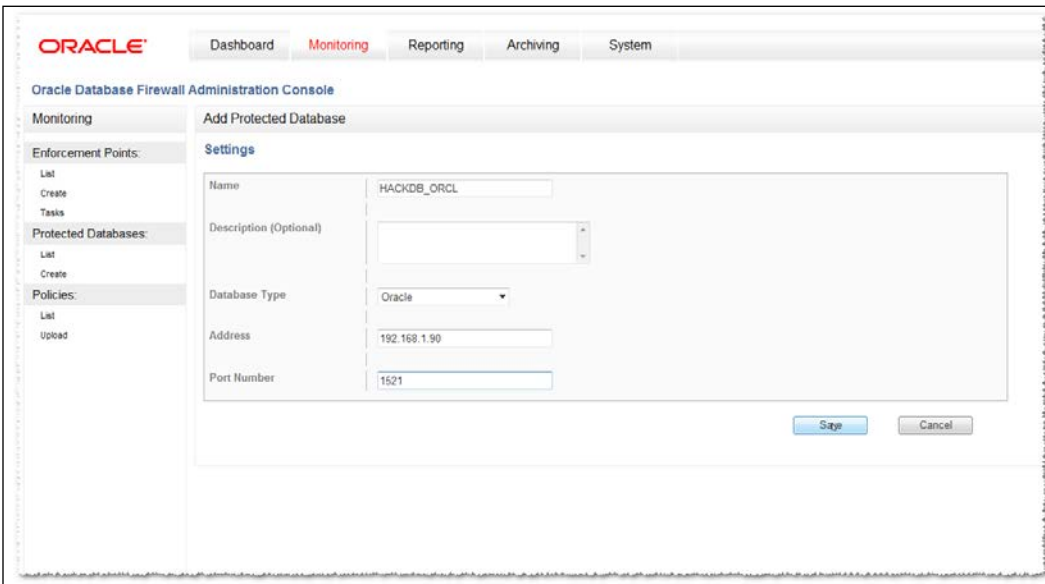
Traffic source configuration

Navigate to the **System** tab. In the left-hand side panel, click on the **Networks** link and add the available network card to **Network 0** and check the **Bridge Enabled** option as follows:



Adding protected database

Navigate to the **Monitoring** tab. In the left-hand side panel, select **Protected Databases** and click on **Create**. Enter **Name** as **HACKDB_ORCL**, the protected database configuration **Database Type** as **Oracle**, add **Address** and **Port Number** and click on the **Save** Button, as shown in the following screenshot:



Creating enforcement point

The enforcement points are the databases to be protected and monitored.

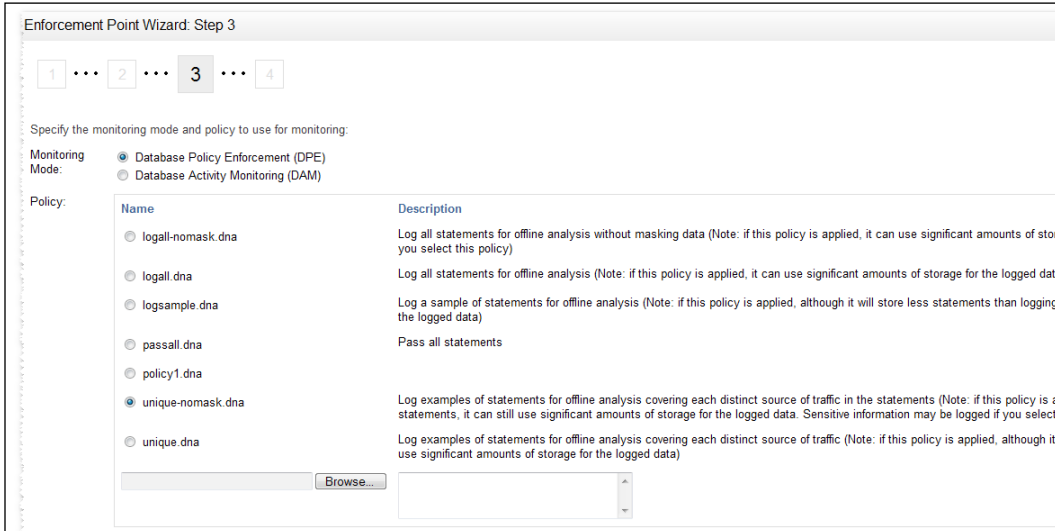
1. Navigate to the **Monitoring** tab. In the **Enforcement Points** panel, click on the **Create** link. Name the enforcement point as **HACKDB_ORCL_ENF**, as shown in the following screenshot, and click on **Next**:

The screenshot shows the Oracle Database Firewall Administration Console. The top navigation bar includes 'Dashboard', 'Monitoring' (selected), 'Reporting', 'Archiving', and 'System'. The main content area is titled 'Enforcement Point Wizard: Step 1'. A progress indicator shows four steps, with '1' selected. The 'Specify enforcement point details:' section contains a 'Name:' field with the value 'HACKDB_ORCL_ENF'. Below this, the 'Use a builtin enforcement point:' section has a radio button selected for 'Monitor locally (80 available)'. A 'Next' button is visible at the bottom.

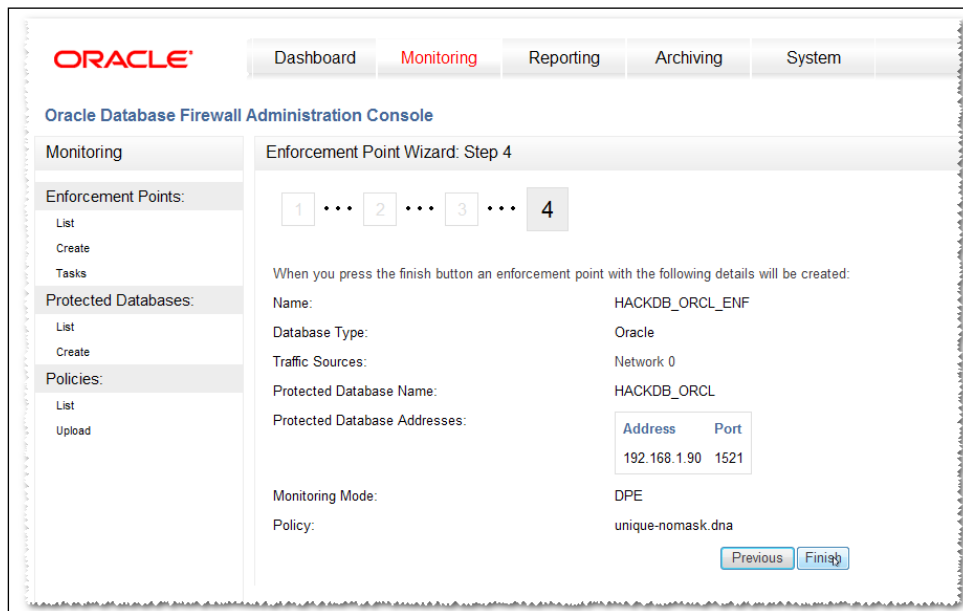
2. Chose the protection point defined before **HACKDB_ORCL** and click on **Next**:

The screenshot shows the Oracle Database Firewall Administration Console. The top navigation bar includes 'Dashboard', 'Monitoring' (selected), 'Reporting', 'Archiving', and 'System'. The main content area is titled 'Enforcement Point Wizard: Step 2'. A progress indicator shows four steps, with '2' selected. The 'Select a protected database:' section has a dropdown menu showing 'Oracle - HACKDB_ORCL'. The 'Specify the details of the protected databases you wish to monitor:' section includes a 'Name:' field, a 'Database Type:' dropdown menu showing 'Oracle', and a table with columns 'Address', 'Port', and 'Resolved Address'. The table is currently empty. 'Previous' and 'Next' buttons are visible at the bottom.

- Choose the monitoring mode – **Database Policy Enhancement (DPE)** and from the available policies select **unique-nomask.dna** and click on **Next**, as shown in the following screenshot:



- Now a summary will be displayed, as we do not have another traffic source defined while **Network 0** is the default one. Finally, click on **Finish**. At this moment our database starts being monitored.



Verify that Oracle Database Firewall monitors the traffic from client host connect to server database server **HACKDB**. To check that our connection is monitored, navigate to the **System** tab. In the **Network Traffic** panel, click on the **Network traffic:** link. Try to issue some statements against the server. From the **Level of details** panel, select **Packet content** and select **Network 0** from **Network**. Issue a statement and click on the **Show Traffic** button. With this, we should see the packet's content as shown in the following screenshot:

The screenshot displays the Administration Console's Network Traffic interface. The 'Level of details' is set to 'Packet content' and the 'Network' is set to 'Network 0'. A 'Show Traffic' button is visible. The main area shows a network traffic capture with hex and ASCII data. An overlaid terminal window shows the execution of an SQL query:

```
C:\Windows\system32\cmd.exe - sqlplus hr/hr@hackdb
Stiles          Stephen
Sullivan        Martha
Sully           Patrick
Taylor          Jonathon
Taylor          Winston
Tobias          Sigal
Tucker          Peter
Tuvault         Oliver

LAST_NAME      FIRST_NAME
-----
Ujman          Jose Manuel
Vargas         Peter
Wishney        Clara
Wollman        Shanta
Walsh          Alana
Weiss          Matthew
Whalen         Jennifer
Zlotkey        Eleni

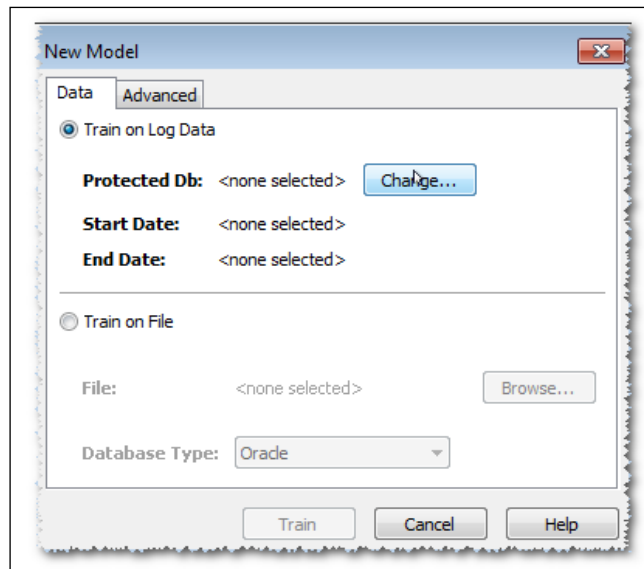
107 rows selected.

SQL> 1
1* select last_name,first_name from employees
SQL>
```

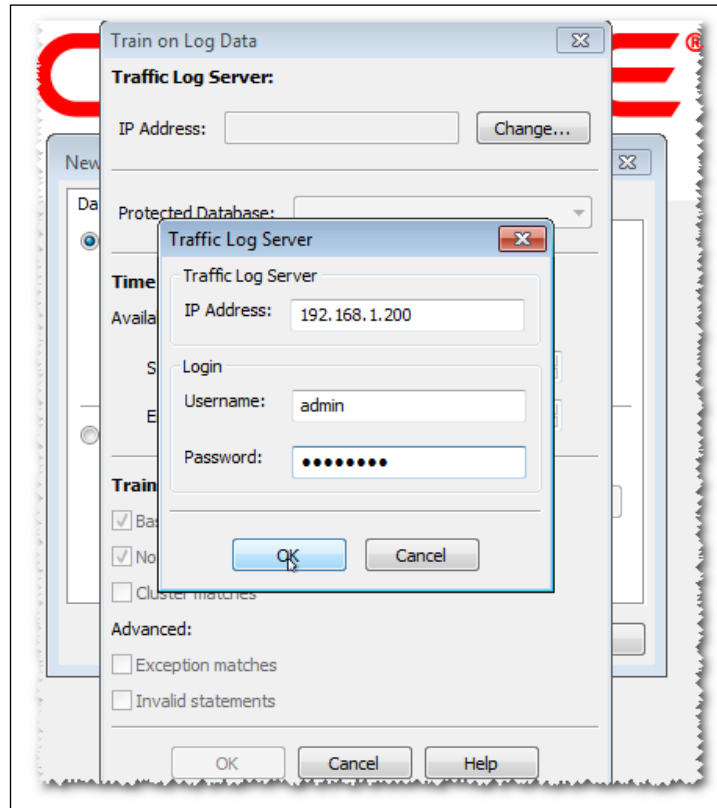
Using Oracle Firewall Analyzer

Oracle Firewall Analyzer is a standalone tool which is designated to create and modify custom policies. The following steps will help you to install and use Oracle Firewall Analyzer.

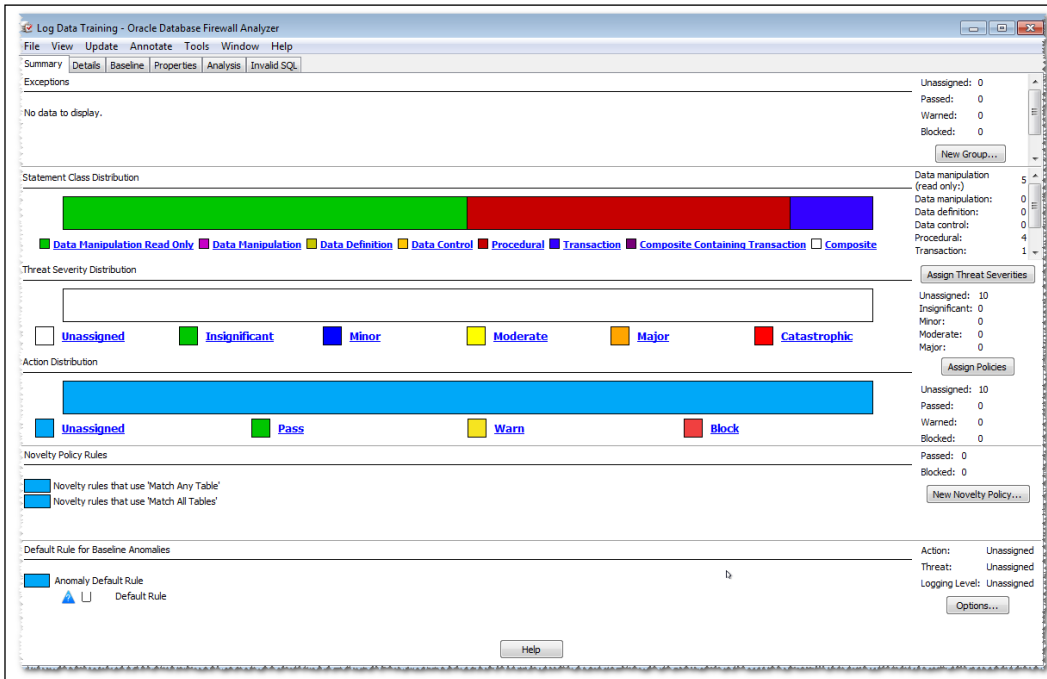
1. On the windows client, install Oracle Firewall Analyzer and launch it. From the **New** menu, click on **New Model** and select **Train on Log Data** and click on the **Change...** button, as shown in the following screenshot:



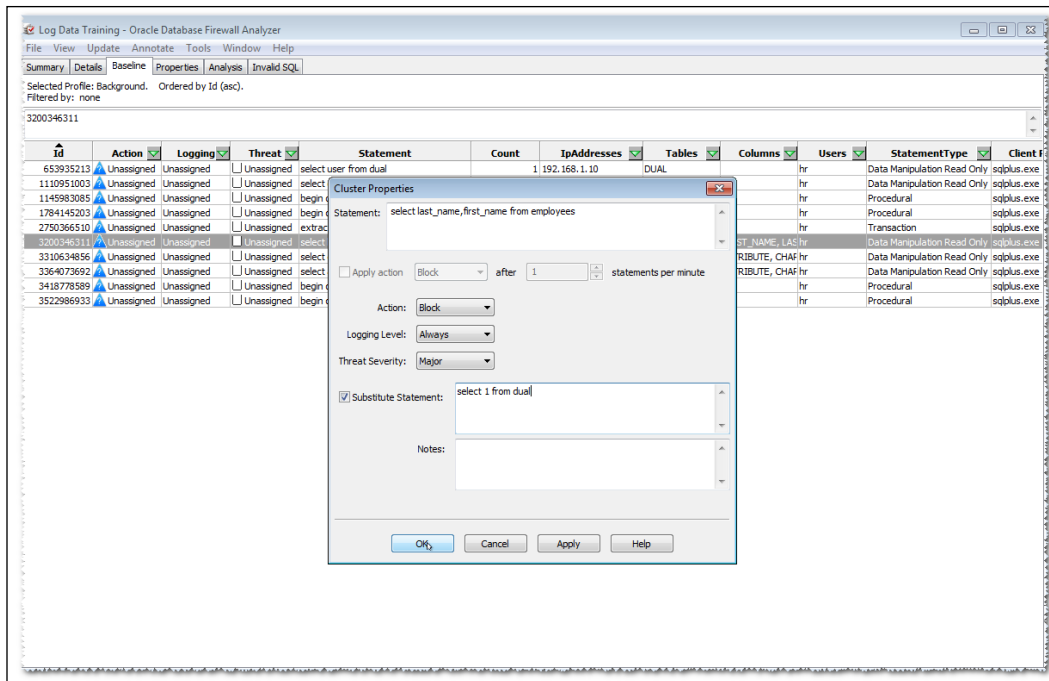
2. Add **Traffic Log Server** which is the same IP used during the installation of Oracle Database Firewall. Next log in with the Oracle Database Firewall credentials as follows and click on **OK**, as shown in the following screenshot:



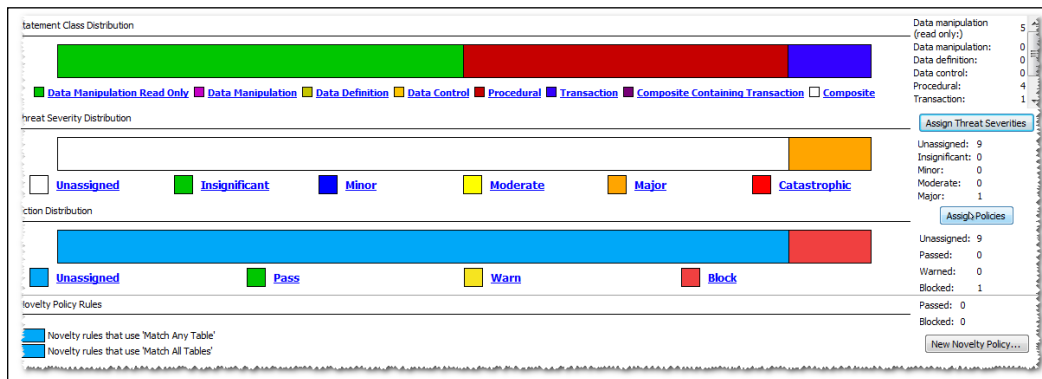
3. After the connection is established, click on the **Train** button. We should then see the captured statements issued in the logging interval, as shown in the following screenshot:



4. Navigate to the **Baseline** tab and click on the **select last_name, first_name from employees** statement. Select **Action** as **Block**, **Logging Level** as **Always**, and **Threat Severity** as **Major**, and change **Substitute Statement** to **select 1 from dual**, and then click on **OK**, as shown in the following screenshot:

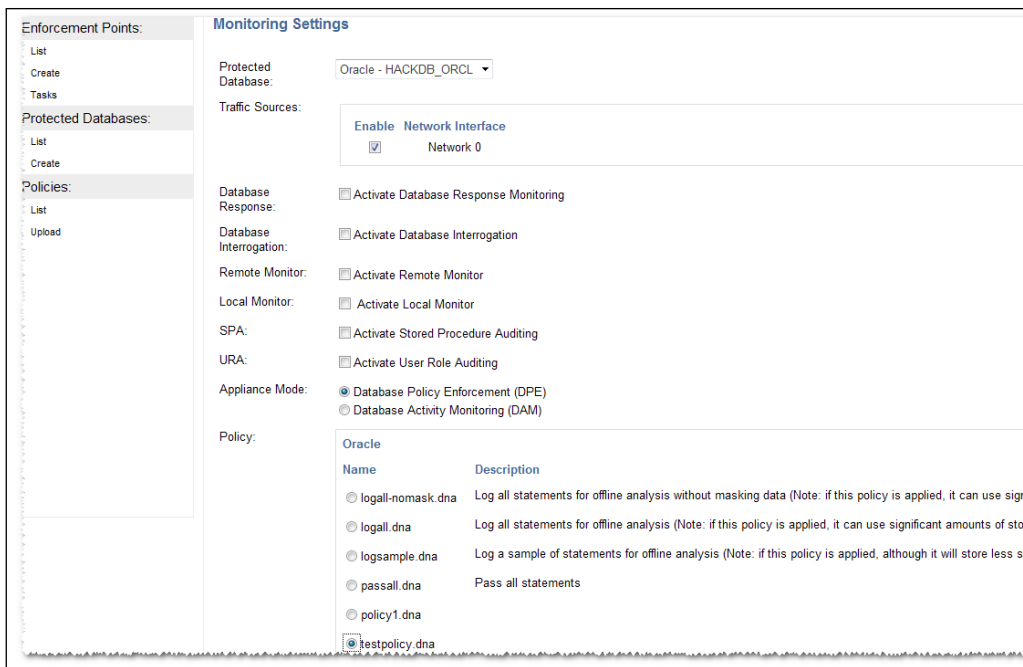


- Next, go back to main menu and click on **Assign Threat Severities**. At this point we are able to save the model and the policy.

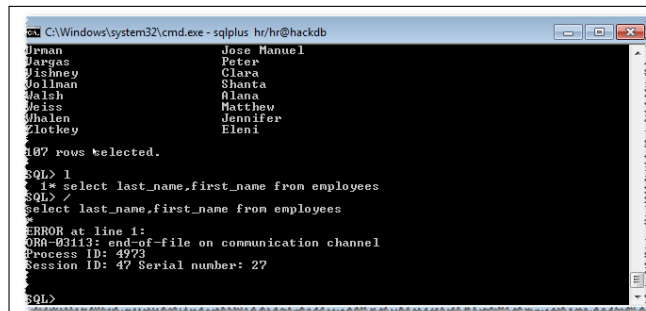


- From the **New** menu, click on **Save the model as test policy** and **Export policy as testpolicy.dna**.

- Next we will proceed to upload the new policy in Oracle Database Firewall. Navigate to the **Monitoring** tab, in the **Policies** panel, and click on the **Upload** link. Select **testpolicy.dna** and click on **Save**.
- Navigate to **Enforcement Points** and click on the **List** link. **HACKDB_ORCL_ENF** will be listed, click on the **Settings** button. Next, from the **Policies** panel, select **testpolicy.dna** and click on **Save** to enforce the defined policies with OFA and ensure that in the **Appliance Mode, Database Policy Enforcement (DPE)** is checked (in DAM mode only monitoring is performed, no reactive measure are applied such as statement blocking). This is shown in the following screenshot:



- Now the enforcement point will be reconfigured and the current monitored connection will be dropped.



10. Reconnect to the HACKDB database and reissue **select last_name, first_name from employees** to verify that the current policy is blocking the statement, shown as follows:

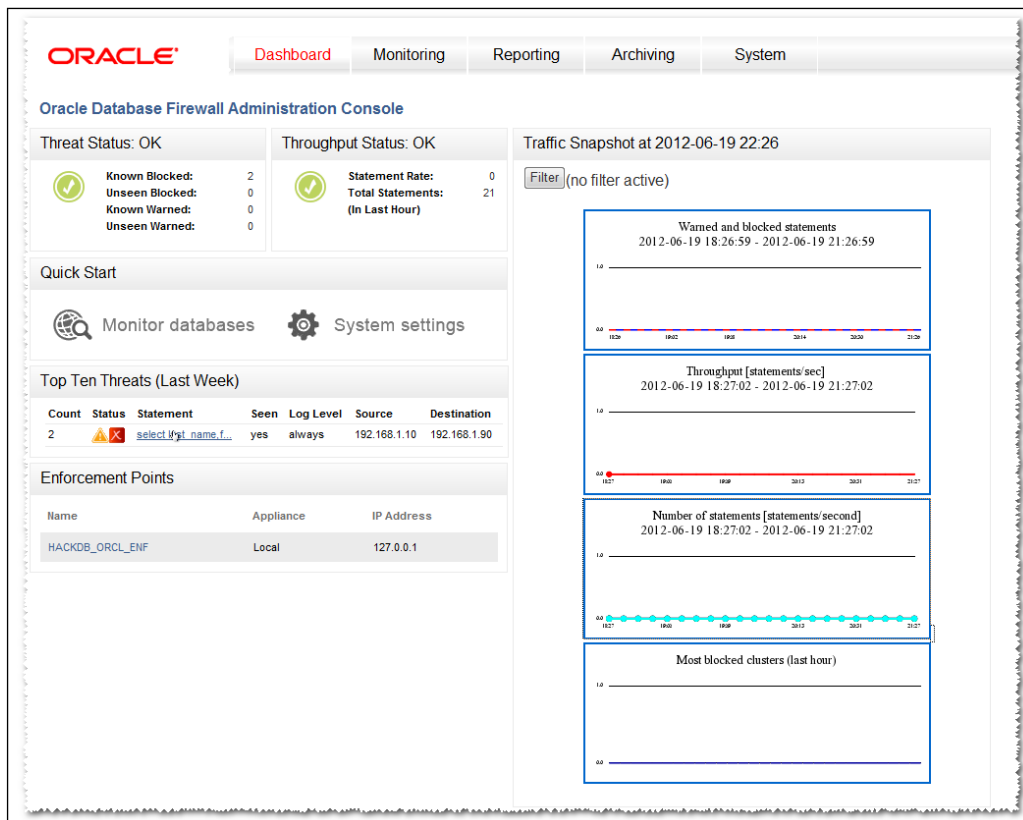
```

C:\Windows\system32\cmd.exe - sqlplus hr/hr@hackdb

SQL> 1
1* select last_name,first_name from employees
SQL> /

-----
1
1
SQL> _
  
```

11. Navigate to the Oracle Database Firewall console's main page. The following screenshot shows the blocked statement listed:



Oracle Audit Vault

Oracle Vault addresses the problem of centralization, separation, and protection of audit trails and provides real-time alerting and reporting capabilities. As we mentioned in *Chapter 8, Tracking and Analysis – Database Auditing*, it is imperative to collect audit information in a location where it cannot be tampered with.

Audit Vault has heterogeneous database support. In addition to Oracle, Audit Vault offers the ability to collect audit data from DB2, MS SQL, and Sybase. It's good to know that Oracle Audit Vault does not implement or alter audit procedures on the database. It is solely based on traditional methods of defining the methods by using the standard audit and fine grained auditing.

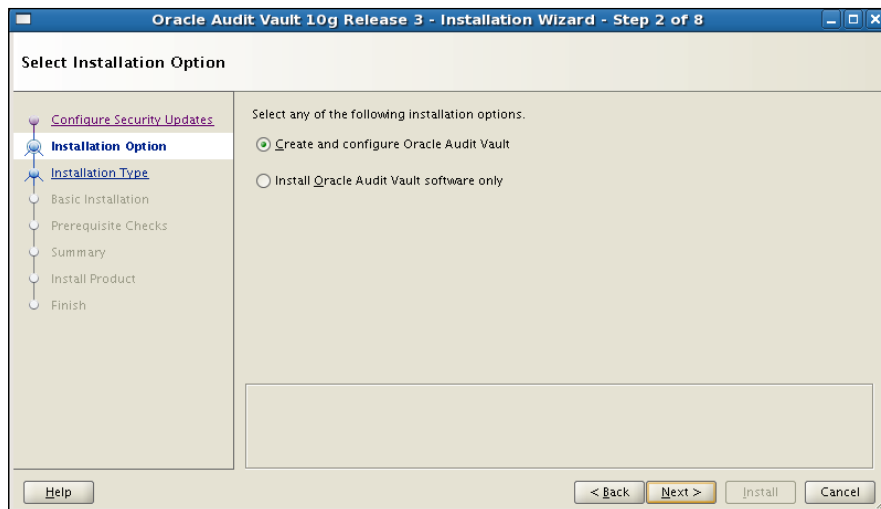
The Audit Vault database used is architecturally designed to be a warehouse. In a large organization that has a need to audit many databases, the audit data volume generated can be significant and the Audit Vault database must be able to handle the volume of data being generated. It is therefore important to install Oracle Audit Vault on a dedicated powerful server. On the other hand AV offers a variety of reporting methods. Since the audit information must be protected, the Audit Vault database is configured with the Oracle Database Vault product.

The Audit Vault environment consists of an audit server, audit agents, and collectors.

Installation and configuration

The following steps will present how to install and perform an initial configuration of AV:

1. Launch the installer and select the **Create and configure Audit Vault** option and click on **Next**, as shown in the following screenshot:



2. Navigate to **Audit Vault Details**, and you will have to configure the audit vault administrator, audit vault auditor, oracle vault owner, and account manager usernames and credentials.

Oracle Audit Vault 10g Release 3 - Installation Wizard - Step 7 of 18

Specify Audit Vault Details

Configure Security Updates

Installation Option

Installation Type

Grid Installation Options

Product Languages

Installation Location

Audit Vault Details

Database Identifiers

Memory Options

Management Options

Database Storage

Backup and Recovery

Schema Passwords

Operating System Groups

Prerequisite Checks

Summary

Install Product

Finish

Enter the Administrator user name and password. You can optionally choose to create a separate Audit Vault Auditor user to provide separation of duties between account and audit management.

Audit Vault Admin: avadmin

Password: ***** Confirm Password: *****

Create a Separate Audit Vault Auditor

Audit Vault Auditor: avaudit

Password: ***** Confirm Password: *****

Enter the Database Vault Owner user name and password. You can optionally choose to create a separate Database Vault Account Manager to provide separation of duties between account and security policy management.

Database Vault Owner: dvwowner

Password: ***** Confirm Password: *****

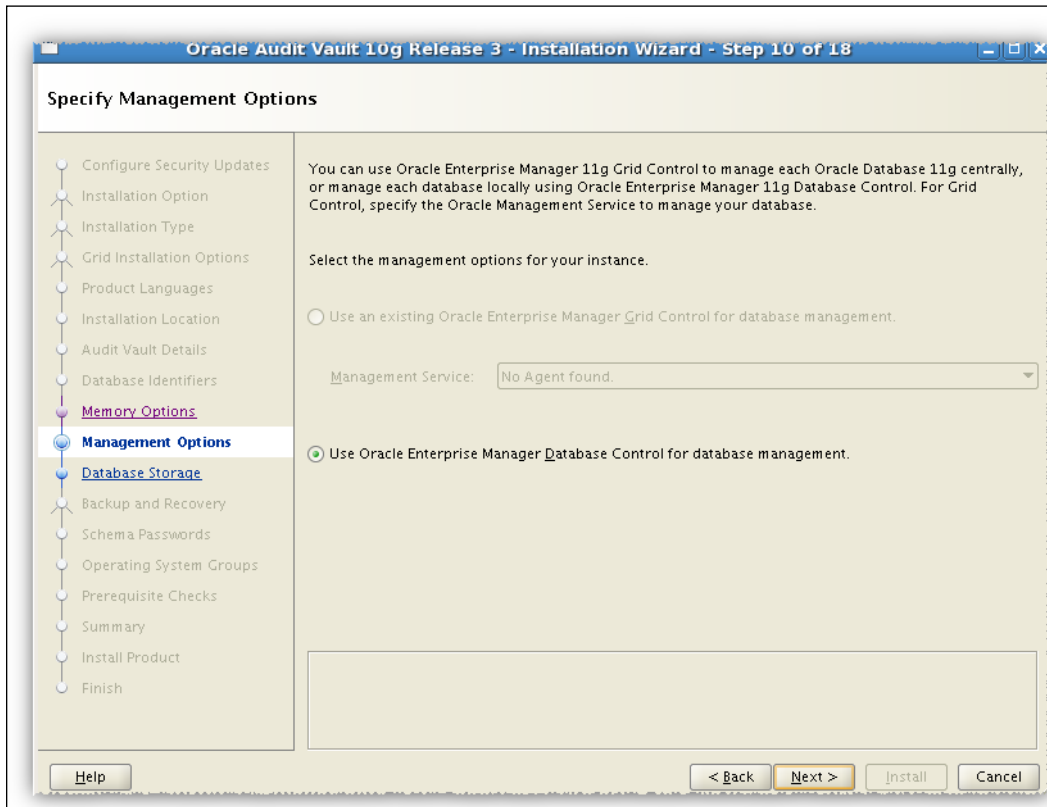
Create a Separate Database Vault Account Manager

Account Manager: dvwacctmgr

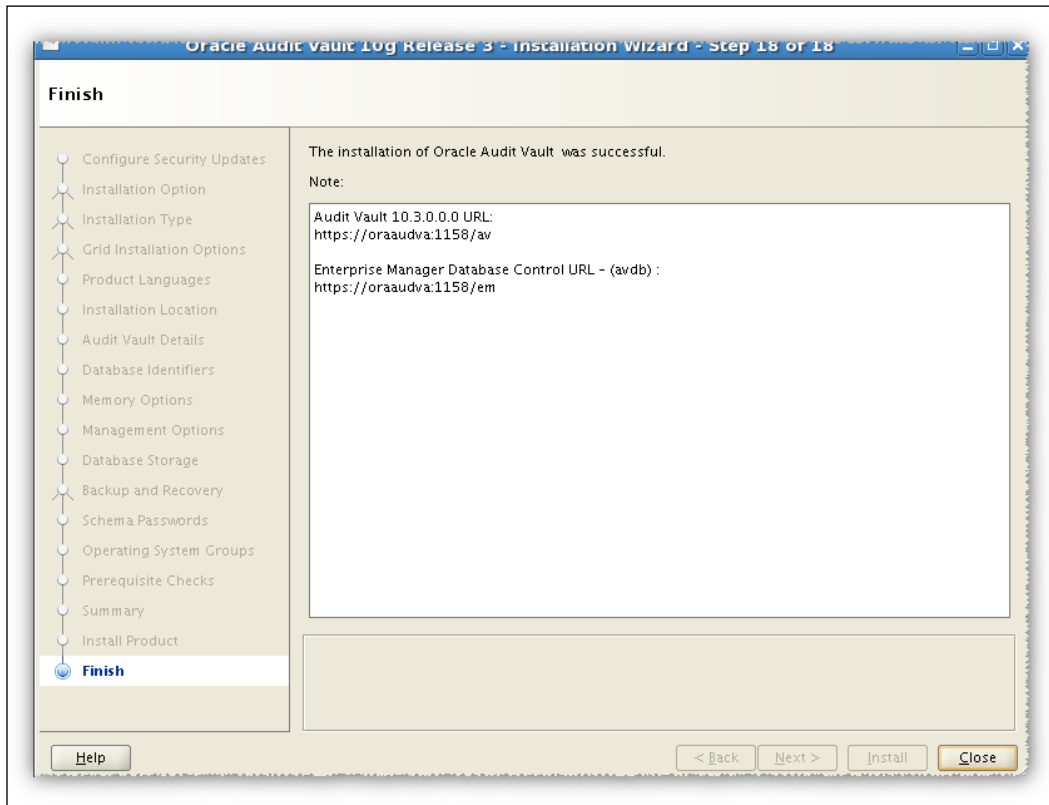
Password: ***** Confirm Password: *****

Help < Back Next > Install Cancel

3. When at **Management Options**, depending on your local configuration, you can select **Use an existing Oracle Enterprise Management Grid Control for database management** or **Use Oracle Enterprise Manager Database Control for database management**.



4. Navigate through all the steps and finalize the installation:



Deploying and configuring agents and collectors

In order to communicate with a Oracle Database Vault repository, a set of collectors, and an Oracle Audit Vault, an agent must be deployed on each host.

1. Copy the agent installation kit on each source database host. Launch the installer and configure the password, the port, and the connection string for the connection to the central repository.

Oracle Audit Vault Agent Installation - Agent Details

Oracle Audit Vault Agent Installation Agent Details

Each Audit Vault Agent is identified by a unique Agent Name. Specify the name of the Agent, the path for the location where the Agent installation will take place, the Agent user name and password, and the connect string for the Audit Vault Server.

Audit Vault Agent Name:

Audit Vault Agent Home:

Agent User Name:

Agent User Password:

Audit Vault Server Connection Information:

Connect String: (Hostname:Port:Service Name)

ORACLE

2. Complete the installation and proceed to configuration.
3. On a server host, define the agent for audit collection as follows:

```
[oracle@oraaudva Disk1]$ avca add_agent -agentname avagnt  
-agenthost nodeorc11
```

```
Enter agent user name: avagnt
```

```
Enter agent user password:
```

```
Re-enter agent user password:
```

```
Agent added successfully.
```

4. On the client database, create a user `av_collector` for audit trail collection and grant collector privileges as follows:

```
SQL> create user av_collector identified by "gY5+TY?z2$5";
```

```
User created.
```

```
SQL>
```

```
SQL> @/u01/app/oracle/product/11.2.0/avagent/av/scripts/streams/  
source/zarsspriv.sql av_collector setup
```

```
Granting privileges to AV_COLLECTOR ... Done.
```

5. On the audit vault host, verify if the source database is enabled for the audit trail collection:

```
[oracle@oraudva Disk1]$ avorcldb verify -src  
nodeorc11:1521:HACKDB -colltype ALL
```

```
Enter Source user name: av_collector
```

```
Enter Source password: gY5+TY?z2$
```

```
source HACKDB verified for OS File Audit Collector collector
```

```
source HACKDB verified for Aud$/FGA_LOG$ Audit Collector collector
```

```
parameter _JOB_QUEUE_INTERVAL is not set; recommended value is 1
```

```
parameter UNDO_RETENTION = 900 is not in recommended value range  
[3600 - ANY_VALUE]
```

```
parameter GLOBAL_NAMES = false is not set to recommended value  
true
```

```
source HACKDB verified for REDO Log Audit Collector collector
```

6. Perform corrections on the source database as instructed by the verification output:

```
SQL> alter system set global_names=true scope=both;
```

```
System altered.
```

```
SQL> alter system set undo_retention=3600 scope=both;
```

```
System altered.
```

```
SQL> alter system set job_queue_interval=1 scope=spfile;
```

```
System altered.
```

7. Perform a verification again as follows:

```
[oracle@oraaudva Disk1]$ avorcldb verify -src
nodeorcl1:1521:HACKDB -colltype ALL
Enter Source user name: av_collector
Enter Source password:
source HACKDB verified for OS File Audit Collector collector
source HACKDB verified for Aud$/FGA_LOG$ Audit Collector collector
source HACKDB verified for REDO Log Audit Collector collector
[oracle@oraaudva Disk1]$
```

8. With this the source database is enabled for collection. Add a source database and a collection agent:

```
[oracle@oraaudva ~]$ avorcldb add_source -src
10.241.132.80:1521:HACKDB -desc HACKDB -srcname HACKDB_SCHM
-agentname avagnt
Enter Source user name: av_collector
Enter Source password:
Adding source...
Source added successfully.
```

```
remember the following information for use in avctl
Source name (srcname): HACKDB_SCHM
Credential stored successfully.
Mapping Source to Agent...
```

9. If we plan to use OS audit trails then add a collector of type OSAUD as follows:

```
[oracle@oraaudva ~]$ avorcldb add_collector -srcname HACKDB_SCHM
-agentname avagnt -colltype OSAUD -orclhome /u01/app/oracle/
product/11.2.0/dbhome_1
source HACKDB_SCHM verified for OS File Audit Collector collector
Adding collector...
Collector added successfully.
```

```
remember the following information for use in avctl
Collector name (collname): OSAUD_Collector
[oracle@oraaudva ~]$
```

10. Add a database audit collector type as follows:

```
[oracle@oraaudva ~]$ avorcldb add_collector -srcname HACKDB_SCHM
-agentname avagnt -colltype DBAUD
source HACKDB_SCHM verified for Aud$/FGA_LOG$ Audit Collector
collector
Adding collector...
Collector added successfully.
```

```
remember the following information for use in avctl
Collector name (collname): DBAUD_Collector
[oracle@oraaudva ~]$
```

11. Add a redo collector as follows:

```
[oracle@nodeorc11 av]$ /u01/app/oracle/product/11.2.0/avagent/bin/
avorcldb setup -srcname HACKDB_SCHM
Enter Source user name: av_collector
Enter Source password:
adding credentials for user av_collector for connection [SRCDB21]
Credential stored successfully.
updated tnsnames.ora with alias [SRCDB21] to source database
verifying SRCDB21 connection using wallet
```

12. Start the agent on the source database, as follows:

```
[oracle@nodeorc11 av]$ /u01/app/oracle/product/11.2.0/avagent/bin/
avc
avca avctl
[oracle@nodeorc11 av]$ /u01/app/oracle/product/11.2.0/avagent/bin/
avctl start_agent -agentname avagnt
Starting agent...
Agent started successfully.
[oracle@nodeorc11 av]$
```

13. Start the collectors as follows:

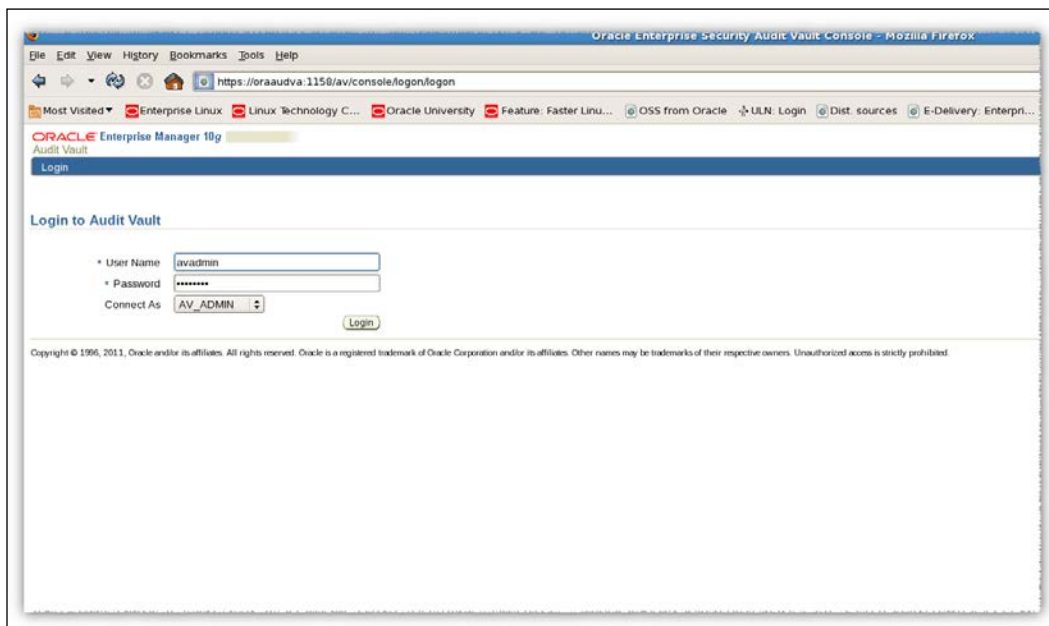
```
[oracle@oraaudva ~]$ avctl start_collector -collname OSAUD_
Collector -srcname HACKDB_SCHM
Starting collector...
Collector started successfully.
```

```
[oracle@oraaudva ~]$ avctl start_collector -collname DBAUD_
Collector -srcname HACKDB_SCHM
Starting collector...
Collector started successfully.
[oracle@oraaudva ~]$
```

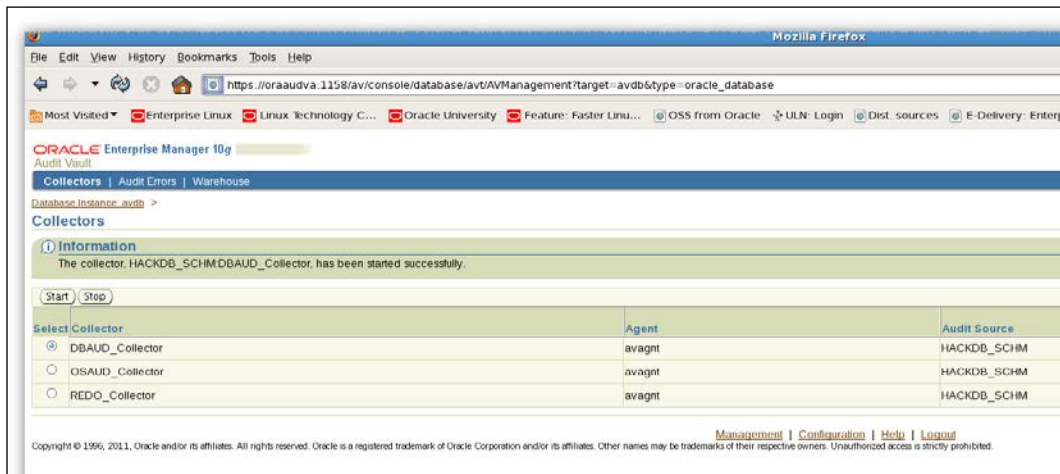
Audit vault administration

In the following section we will summarize the main administrative tasks used with Audit Vault:

Open Audit Vault administration control and log in as the AV administrator (type the password set during installation):



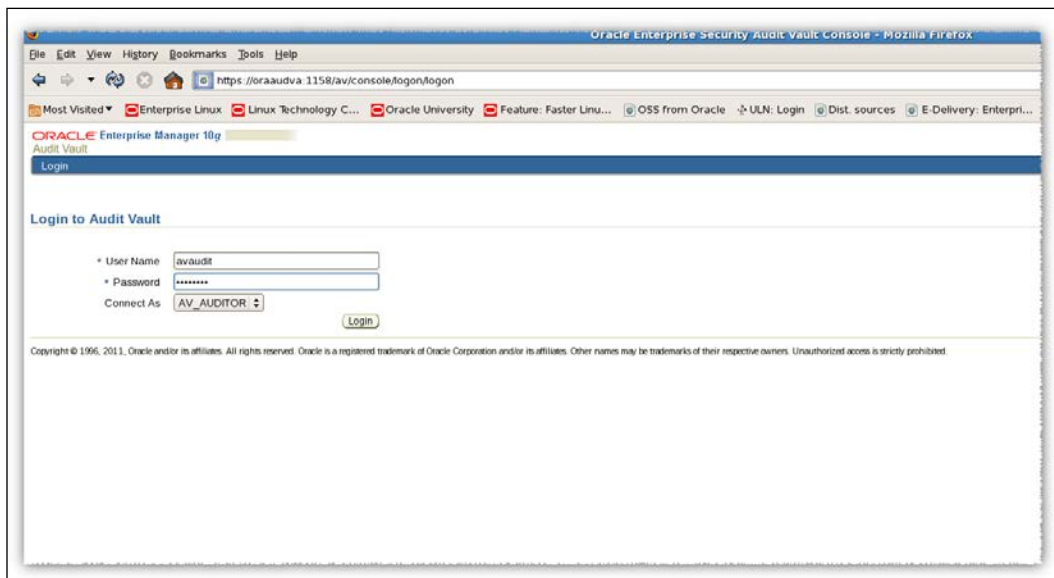
Instead of using the command line prompt, you can manage collectors from the **Management** tab and within that **Collectors**, as shown in the following screenshot:



Creating additional audit policies

Audit Vault has the capability to generate scripts for different audit statements which can be applied later on to the source database:

1. Log in as the Audit owner **avaudit**, as shown in the following screenshot:



- Navigate to the Audit policy tab. The overview page will show you what type of auditing and how many objects or statements are audited:

Audit Settings | Alerts
 Database Instance: avdb > Audit Settings >
HACKDB_SCHM

Overview | Statement | Object | Privilege | FGA | Capture Rule

Save Audit Settings
 You can save your work by clicking on the Save All Audit Settings button below. Please note, saving your work does not automatically apply these settings to the source database.
 [Save All Audit Settings]

Apply Audit Settings
 You can verify that the audit settings can be successfully applied to a given source by clicking on Verify. If the DBA for the source has provided you an account on the source, you can changes to a SQL script that you can give the DBA, who can then apply the settings for you.

Select All | Select None

Select	Audit Settings Type	In Use
<input checked="" type="checkbox"/>	Statement	10
<input checked="" type="checkbox"/>	Object	11
<input checked="" type="checkbox"/>	Privilege	26
<input checked="" type="checkbox"/>	FGA	0
<input checked="" type="checkbox"/>	Capture Rule	0

[Verify] [Export as SQL] * Audit Source User Name [Provision]
 * Audit Source Password

Copy Audit Settings from Another Source
 You can quickly replicate audit settings from one database to the source database to seed it with common audit settings. You either can use settings that are already in use in the data
 Copy Actual (In Use) Needed (Not Yet In Use) Audit Settings
 From [Load]

- Next navigate to **Statements** and click on the **Create** button. The **Create Object Audit** page will open, then check **SELECT** in the statement box, select the **Object Type** as **TABLE**, **Object** as **HR.DEPARTMENTS**, **Statement Execution Condition** as **Both** and **DML Audit Granularity** as **SESSION**, as shown in the following screenshot:

ORACLE Enterprise Manager 10g
 Audit Vault
Audit Settings | Alerts
 Database Instance: avdb > Audit Settings >
Create Object Audit

* Statements:

Object Type:

Object:

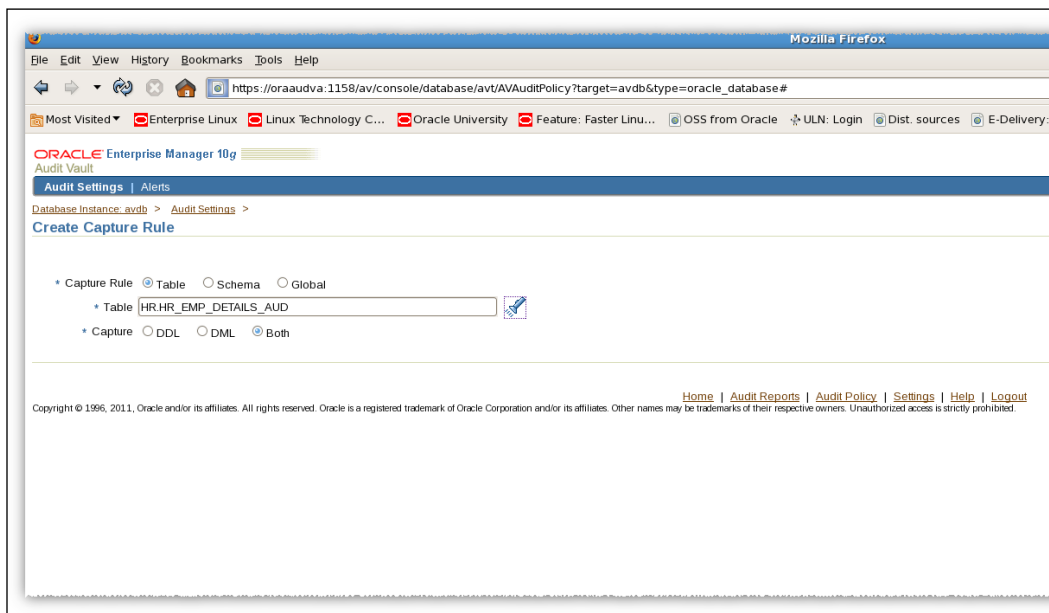
* Statement Execution Condition:

* DML Audit Granularity:

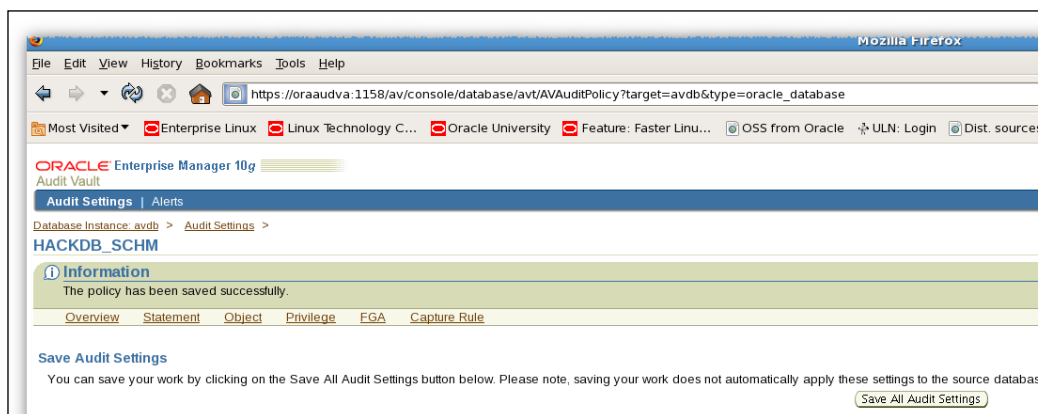
Audit granularity is used for auditing DML statement DDL statements are always audited by access.

Copyright © 1996, 2011, Oracle and/or its affiliates. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners. Unauthorized access is strictly prohibited.

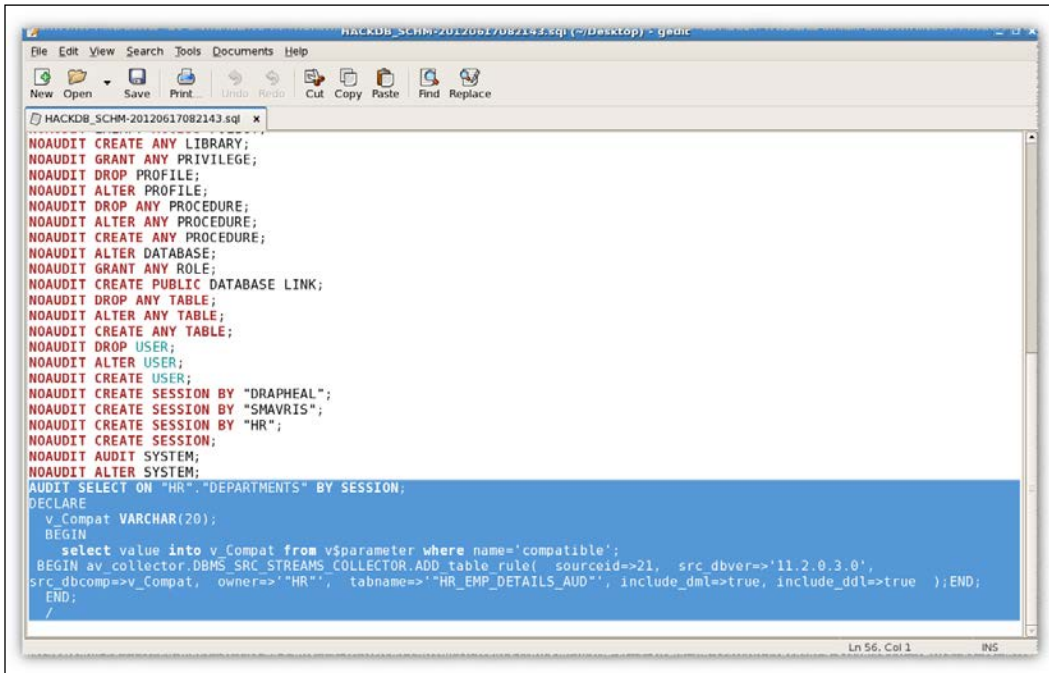
- Next, we will create a capture rule. Navigate to the **Capture Rule** and set **Capture Rule to Table**, set **Table to HR.HR_EMP_DETAIL_AUD**, **Capture to Both**, and then click on **OK**, as shown in the following screenshot:



- To generate the provisioning script, navigate to the **Overview** tab and click on the **Save All Audit Settings** button, as shown in the following screenshot:



6. A script will be generated; select the last two statements and run it on the source database:



```
HACKDB_SCHM-20120617082143.sql (=)Desktop) - gedit
File Edit View Search Tools Documents Help
New Open Save Print Undo Redo Cut Copy Paste Find Replace
HACKDB_SCHM-20120617082143.sql x
NOAUDIT CREATE ANY LIBRARY;
NOAUDIT GRANT ANY PRIVILEGE;
NOAUDIT DROP PROFILE;
NOAUDIT ALTER PROFILE;
NOAUDIT DROP ANY PROCEDURE;
NOAUDIT ALTER ANY PROCEDURE;
NOAUDIT CREATE ANY PROCEDURE;
NOAUDIT ALTER DATABASE;
NOAUDIT GRANT ANY ROLE;
NOAUDIT CREATE PUBLIC DATABASE LINK;
NOAUDIT DROP ANY TABLE;
NOAUDIT ALTER ANY TABLE;
NOAUDIT CREATE ANY TABLE;
NOAUDIT DROP USER;
NOAUDIT ALTER USER;
NOAUDIT CREATE USER;
NOAUDIT CREATE SESSION BY "DRAPHEAL";
NOAUDIT CREATE SESSION BY "SHAVRIS";
NOAUDIT CREATE SESSION BY "HR";
NOAUDIT CREATE SESSION;
NOAUDIT AUDIT SYSTEM;
NOAUDIT ALTER SYSTEM;
AUDIT SELECT ON "HR"."DEPARTMENTS" BY SESSION;
DECLARE
v_Compat VARCHAR(20);
BEGIN
select value into v_Compat from v$parameter where name='compatible';
BEGIN av_collector.DBMS_SRC_STREAMS_COLLECTOR.ADD_table_rule( sourceid=>21, src_dbver=>'11.2.0.3.0',
src_dbcomp=>v_Compat, owner=>'HR", tabname=>'HR_EMP_DETAILS_AUD", include_dm1=>true, include_ddl=>true );END;
END;
/
Ln 56, Col 1 INS
```

Using Audit Vault reports

Audit Vault provides us with the possibility of creating different types of built-in or customized reports, as we will see in the following section.

1. Navigate to the **Audit Reports** tab, as shown in the following screenshot:



2. To summarize the audit data collected, go to **Activity Overview**. This is an unsorted list of audit events that are captured. This is illustrated in the following screenshot:

The screenshot displays the Oracle Enterprise Manager 10g Audit Vault interface. The main section is titled "Activity Overview" and shows a table of audit events. The table has the following columns: Source, Category, Event, User, Target, Host, and Event Time. The events listed include various database operations such as SELECT, LOGON, and ALTER TABLE, performed by users like HR, AV_COLLECTOR, and SMAVRIS. The table is filtered to show events from the last 24 hours.

Source	Category	Event	User	Target	Host	Event Time
HACKDB_SCHM	DATA ACCESS	SELECT	HR	EMPLOYEES	nodeorcl1	6/16/2012 1:48:01 PM
HACKDB_SCHM	DATA ACCESS	SELECT	HR	EMPLOYEES	nodeorcl1	6/16/2012 1:47:39 PM
HACKDB_SCHM	USER SESSION	LOGON	HR		nodeorcl1	6/16/2012 1:47:04 PM
HACKDB_SCHM	DATA ACCESS	SELECT	HR	EMPLOYEES	nodeorcl1	6/16/2012 1:46:17 PM
HACKDB_SCHM	USER SESSION	LOGON	HR		nodeorcl1	6/16/2012 1:46:04 PM
HACKDB_SCHM	USER SESSION	SUPER USER LOGON	/		nodeorcl1	6/16/2012 1:44:39 PM
HACKDB_SCHM	USER SESSION	LOGON	AV_COLLECTOR		nodeorcl1	6/16/2012 1:28:53 PM
HACKDB_SCHM	USER SESSION	LOGON	AV_COLLECTOR		nodeorcl1	6/16/2012 1:28:37 PM
HACKDB_SCHM	USER SESSION	LOGOFF	SMAVRIS		nodeorcl1	6/16/2012 1:23:15 PM
HACKDB_SCHM	OBJECT MANAGEMENT	ALTER TABLE	SMAVRIS	HR_EMP_DETAILS_AUD	nodeorcl1	6/16/2012 1:21:11 PM
HACKDB_SCHM	USER SESSION	LOGON	SMAVRIS		nodeorcl1	6/16/2012 1:20:35 PM
HACKDB_SCHM	USER SESSION	LOGOFF	DRAPHEAL		nodeorcl1	6/16/2012 1:20:35 PM
HACKDB_SCHM	OBJECT MANAGEMENT	ALTER TABLE	DRAPHEAL	HR_EMP_DETAILS_AUD	nodeorcl1	6/16/2012 1:20:19 PM
HACKDB_SCHM	OBJECT MANAGEMENT	ALTER TABLE	DRAPHEAL	HR_EMP_DETAILS_AUD	nodeorcl1	6/16/2012 1:18:54 PM
HACKDB_SCHM	USER SESSION	LOGON	DRAPHEAL		nodeorcl1	6/16/2012 1:18:43 PM

You also have the ability to perform compliance reporting. In Audit Vault you have built-in reports for **Credit Card**, **Financial**, and **Health** compliance reports.

The screenshot displays the Oracle Enterprise Manager 10g Audit Vault interface, specifically the "Compliance Reports" section. The page is divided into three columns of reports: Credit Card, Financial, and Health Care. Each column lists various audit reports. The "Audit Reports" link in the top navigation bar is circled in red.

Defining an alert rule

To create an alerting condition, navigate to the Audit policy tab and click on **Create Alert Rule**. In this example we will create an alert rule named **delete_table**. Select the severity level to be **Warning**, **Audit Source Type** to be **HACKDB_SCH**, **Audit Event Category** to be **DATA ACCESS**. **User** should be set to **HR**, **Table** to **HR.HR_EMP_DETAILS_AUD**, and **Audit Event** to **DELETE**.

The screenshot shows the 'Create Alert Rule' web interface in Mozilla Firefox. The browser address bar shows the URL: `https://oraaudva:5707/av/console/f?p=7700:35:2685972581566074::NO:::`. The page title is 'Create Alert Rule - Mozilla Firefox'. The form contains the following fields and values:

- Alert:** delete_table
- Description:** (empty text area)
- Alert Severity:** Warning
- Audit Source Type:** ORCLDB
- Audit Source:** HACKDB_SCHM
- Audit Event Category:** DATA ACCESS
- Specify additional alert conditions in:** Basic (selected), Advanced
- Basic Alert Condition:**
 - User:** HR
 - Table:** HR.HR_EMP_DETAILS_AUD
 - Audit Event:** DELETE
 - Audit Event Status:** Both (selected), Success, Failure
- Alert Action:** When an alert is raised, take the following actions.
- Notification Action:**
 - Template:** Alert Notification Template
 - Profile:** notifications
 - To:** alert@orcl.com
 - Cc:** (empty)
 - Add to List:** (button)
- Table:** (table with columns: Profile Name, To, Cc, Template Name)
- Trouble Ticket Action:**
 - Template:** No Template

The bottom of the page shows the word 'Done'.