



Audit Vault and Database Firewall 20 Cookbook



A practical approach to understanding AVDF20

April 2020 | Version 20.01
Copyright © 2021, Oracle and/or its affiliates

1. PURPOSE

Oracle Audit Vault and Database Firewall (AVDF) is a complete Database Activity Monitoring (DAM) solution that combines native audit data with network-based SQL traffic capture. AVDF includes an enterprise quality audit data warehouse, host-based audit data collection agents, powerful reporting and analysis tools, alert framework, audit dashboard, and a multi-stage Database Firewall. The Database Firewall uses a sophisticated grammar analysis engine to inspect SQL statements before they reach the database and determines with high accuracy whether to allow, log, alert, substitute, or block the incoming SQL.

This cookbook is intended to help you get started with AVDF in a couple of simple steps. Sample schema, users, audit policies, and workload are also provided for beginners and those who want to explore updated the look and feel of AVDF20.

2. INTENDED AUDIENCE

If you are responsible for designing, implementing, maintaining, or operating security controls for an Oracle Database this paper is intended for you.

3. DISCLAIMER

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

4. TABLE OF CONTENTS

1.	Purpose	1
2.	Intended Audience	1
3.	Disclaimer	1
4.	Table of contents	2
5.	Introduction	3
1.	REGISTRATION OF HOSTS AND AGENTS	4
1.1.	Configurations	4
2.	TARGET REGISTRATION AND CONFIGURATION OF AUDIT TRAIL COLLECTION	5
2.1.	Configurations	5
3.	CONFIGURING SENSITIVE DATA REPORTS	8
3.1.	Configurations	8
3.2.	Verification of successful import	9
3.3.	Tracking activities on sensitive data	10
4.	AUDIT POLICY PROVISIONING	12
4.1.	Configurations	13
4.2.	Tracking audit events	14
5.	PROACTIVE MONITORING WITH ALERT CONFIGURATION	15
5.1.	Configurations	15
5.2.	Tracking alerts	19
6.	NETWORK MONITORING CONFIGURATION	20
6.1.	Pairing Database Firewall Server with Audit Vault Server	20
6.2.	Preparing Database Firewall server	21
6.3.	Configuration of Network monitoring for the pluggable database instance	23
7.	DATABASE FIREWALL POLICY CONFIGURATION	24
7.1.	Train the firewall to understand the permitted SQL traffic	25
7.2.	Create a user-defined firewall policy to inspect network activity of HR Database	26
7.3.	Tracking network activity using reports	32
7.3.	Create a user-defined firewall policy to detect data exfiltration attempts from HR Database	34
7.4.	Tracking potential data exfiltration attempts over the network using reports	38
8.	<OPTIONAL>REDO LOG COLLECTION WITH GOLDEN GATE	38
8.1.	Installation and Configuration of GoldenGate Microservice	39
8.2.	Preparation of Oracle Database target for transaction log collection	41
8.3.	Configuring Integrated Extract process in GoldenGate	42
8.4.	Configuring transaction log collection in Audit Vault console	43
9.	<OPTIONAL>NETWORK MONITORING WITH HOST MONITOR	45
	SUMMARY	50

5. INTRODUCTION

The cookbook flows are built on storyline matching a typical real-world scenario where we have sensitive information in some of the tables in the database, and we use database auditing and network monitoring to monitor the database activity. We configure HCM schema, and populate tables with employee data as shown here:

HCM Schema:

EMPLOYEES	EMP_EXTENDED	SUPPLEMENTAL_DATA	DEPARTMENTS	COUNTRIES	LOCATIONS	REGIONS	JOB_HISTORY	JOBS
"EMPLOYEE_ID" NUMBER(6,0), "FIRST_NAME" VARCHAR2(20 BYTE), "LAST_NAME" VARCHAR2(25 BYTE), "EMAIL" VARCHAR2(25 BYTE), "PHONE_NUMBER" VARCHAR2(20 BYTE), "HIRE_DATE" DATE, "JOB_ID" VARCHAR2(10 BYTE), "SALARY" NUMBER(8,2), "COMMISSION_PCT" NUMBER(2,2), "MANAGER_ID" NUMBER(6,0), "DEPARTMENT_ID" NUMBER(4,0)	"EMPLOYEE_ID" NUMBER(6,0), "TAXPAYERID" VARCHAR2(15), "PAYMENTACCOUNTNO" VARCHAR2(20)	PERSON_ID NUMBER, USERNAME VARCHAR2(50), TAXPAYER_ID VARCHAR2(20), LAST_INS_CLAIM VARCHAR2(2000), BONUS_AMOUNT NUMBER	"DEPARTMENT_ID" NUMBER(4,0), "DEPARTMENT_NAME" VARCHAR2(30 BYTE), "MANAGER_ID" NUMBER(6,0), "LOCATION_ID" NUMBER(4,0)	"COUNTRY_ID" CHAR(2 BYTE), "COUNTRY_NAME" VARCHAR2(40 BYTE), "REGION_ID" NUMBER,	"LOCATION_ID" NUMBER(4,0), "STREET_ADDRESS" VARCHAR2(40 BYTE), "POSTAL_CODE" VARCHAR2(12 BYTE), "CITY" VARCHAR2(30 BYTE), "STATE_PROVINCE" VARCHAR2(25 BYTE), "COUNTRY_ID" CHAR(2 BYTE),	"REGION_ID" NUMBER, "REGION_NAME" VARCHAR2(25 BYTE),	"EMPLOYEE_ID" NUMBER(6,0), "DATE_OF_HIRE" DATE CONSTRAINT, "DATE_OF_TERMINATION" DATE CONSTRAINT, "JOB_ID" VARCHAR2(10 BYTE) CONSTRAINT , "DEPARTMENT_ID" NUMBER(4,0).	"JOB_ID" VARCHAR2(10), "JOB_TITLE" VARCHAR2(35), "MIN_SALARY" NUMBER(6,0), "MAX_SALARY" NUMBER(6,0).

There are three different types of users that we configure in our sample scripts who are accessing the sensitive tables, and their trusted path is tagged with defined values in the user application context:

1. Regular employees through the Employee application traffic (trusted path = EMPLOYEE_USER)
2. HR Users through Employee application traffic (trusted path = HR_USER, HR_MANAGER)
3. Database Administrators

Employee	HR Users	Database Administrators
sophie@example.com lucas@example.com jron@example.com henrywil@example.com samkirk@example.com	hr_tim@example.com hr_joe@example.com hr_jim@example.com hr_lan@example.com hr_ann@example.com (HR Manager)	dba_debra@example.com dba_harvey@example.com dba_charles@example.com secadmin_steve@example.com (Security Admin DBA)

The cookbook assumes that following pre-requisites are done prior to beginning with the rest of the flows:

1. Oracle Audit Vault and Database Firewall binaries are downloaded from [Oracle Software Delivery Cloud](#), and Audit Vault server, and Database Firewall server is installed on separate machines.
2. You have logged into Audit Vault server URL and created AVADMIN and AVAUDITOR users.
3. You have identified an Oracle pluggable database instance to be used as a target and configured it with HCM schema, users, and data. Please follow the details below.
 - a. Create the schema, users and different artifacts to be used for testing the scenarios



hcm_data_script.sql

- i. Open the script
 - ii. Execute the script against the target pluggable database instance
- b. Run the workload for capturing audit events



hcm_workload_script.sql

- i. Open the script

- ii. Execute the script against the target instance.

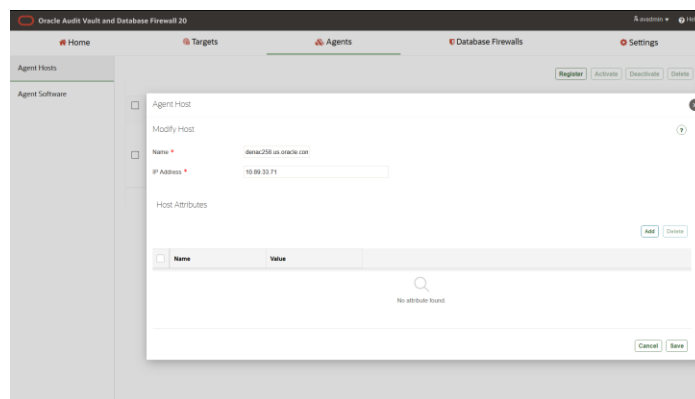
The rest of the cookbook details the flows to get started.

1. REGISTRATION OF HOSTS AND AGENTS

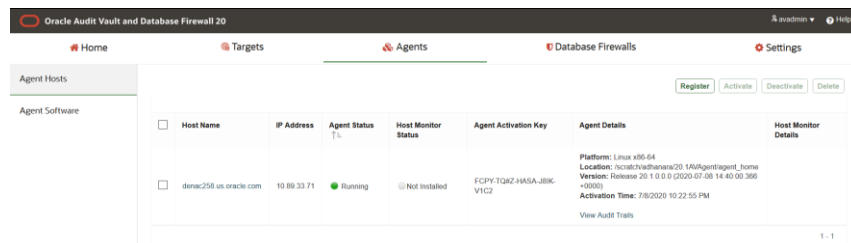
Goal: To register host and agent on the database server target.

1.1. Configurations

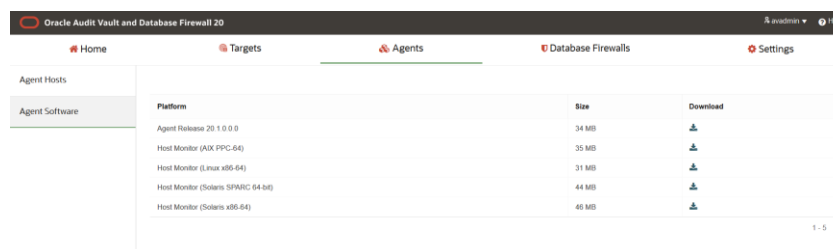
- a. Login into **AVS server** as **AVADMIN**
- b. Navigate to **“Agents”** menu
- c. Click **“Agent Hosts”** in the left navigation menu
- d. Click **[Register]** to open the popup as shown here
- e. Enter a unique name for the host (usually the fully qualified hostname) and IP Address



- f. Click **[Save]** in the popup to see the agent host in **“Activated”** status. This generates a unique agent activation key
- g. Copy the **“Agent Activation Key”**



- h. Click **“Agent Software”** in the left navigation menu to download the agent binary



- i. Download the **“Agent Release 20.1.0.0.0”** binary to the local and transfer it to the database target instance

- j. Ensure the JAVA_HOME environment variable is set and the java executable is in your path

```
echo $JAVA_HOME
which java
```

- k. Ensure that your host server has a supported Java Runtime Environment (JRE) or JDK installed – version 1.8.0_45 or higher, or 11.0.3

```
java -version
```

- l. Unzip the jar to an agent home using the following command where AGENT_HOME_DIRECTORY is the directory where you want the agent to be installed. The directory will be created if it does not exist

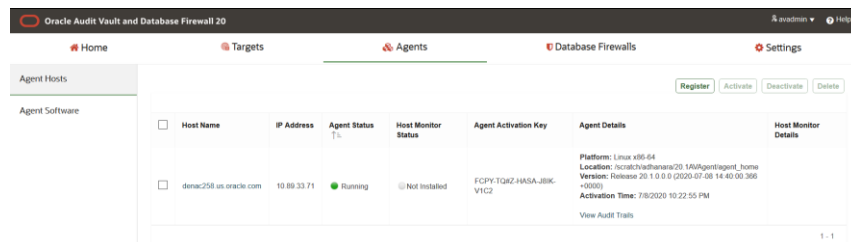
```
java -jar agent.jar -d <AGENT_HOME_DIRECTORY>
```

- o. Go to the agent home/bin and execute the following command. Enter the “**Agent Activation Key**” when prompted. Ensure the agent is running with “**agentctl status**” command.

```
agentctl start -k
```

```
[adhanara@denac258 bin]$ ./agentctl start -k
Enter Activation Key:
Agent started successfully.
[adhanara@denac258 bin]$ ./agentctl status
Agent is running.
[adhanara@denac258 bin]$
```

- o. Login to **AVS server** and refresh the page to ensure the Agent is running successfully.



Oracle Audit Vault and Database Firewall 20							
Agents							
Agent Hosts							
Agent Software							
Register Activate Deactivate Delete							
<input type="checkbox"/>	Host Name	IP Address	Agent Status	Host Monitor Status	Agent Activation Key	Agent Details	Host Monitor Details
<input type="checkbox"/>	denac258.us.oracle.com	10.89.33.71	Running	Not installed	FCPY-TQAZ-HASA-JBKC-V1C2	Platform: Linux x86_64 Location: /usr/local/auditvault/20.1.0.0/agent_home Version: Release 20.1.0.0 (2020-07-08 14:40:00:368 +0000) Activation Time: 7/8/2020 10:22:55 PM View Audit Trails	
1 - 1							

Ensure the Agent status is “Running”.

2. TARGET REGISTRATION AND CONFIGURATION OF AUDIT TRAIL COLLECTION

Goal: To register the Oracle pluggable database instance as a target in AVDF and configure the audit trail to start collecting audit events.

2.1. Configurations

2.1.1. Configure a user in the pluggable database. For the agent to be able to collect audit records it requires connecting as a user that is privileged to collect audit data. To create the user and grant it appropriate privileges:

- a. Connect to the pluggable database instance and create the “**avdfuser**” user.
- b. Execute the script that can be found in the agent home to grant the required privileges to the user

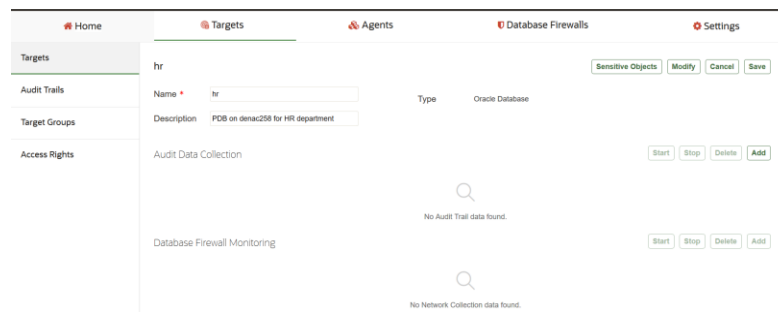
```
@<agent_home>/av/plugins/com.oracle.av.plugin.oracle/config/oracle_user_setup.sql avdfuser
SETUP
SQL> alter session set container=orclpdb;
Session altered.
SQL> create user avdfuser identified by ████████
User created.
SQL> @/scratch/adhanara/20.1AVAgent/agent_home/av/plugins/com.oracle.av.plugin.oracle/config/oracle_user_setup.sql avdfuser SETUP
Session altered.
Granting privileges to "AVDFUSER" ... 'AVDFUSER'
Done.
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.5.0.0.0
```

2.1.2. Register the database server instance as the target in Audit Vault Server

- a. Login to **AVS server** as **AVADMIN**
- b. Open “**Targets**” menu and click **[Register]** to enter the following registration details in the “**Audit Connection Details**” tab for the pluggable database instance which we name as “**hr**”
 - Name: **hr**
 - Type: **Oracle Database**
 - IP Address: **<DB server IP Address where instance is running>** - *Note: If the server is in Oracle Cloud this should be the private IP address*
 - Port: **<port where database service is running>**
 - Service Name: **<database service name>**
 - Protocol: **TCP**
 - Username: **avdfuser**
 - Password: **<password>**

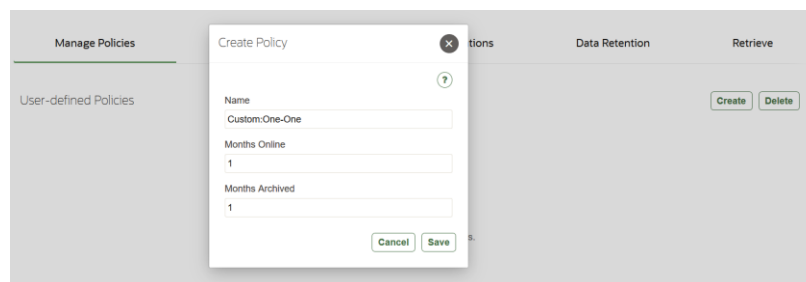
The screenshot shows the 'Targets' configuration page in the Audit Vault Server. The 'Name' field is set to 'hr' and the 'Type' is 'Oracle Database'. The 'Description' is 'PDB on demac258 for HR department'. The 'Audit Connection Details' tab is active, showing fields for Host Name / IP Address (demac258.us.oracle.com), Port (1522), Service Name (orclpdb.us.oracle.com), Protocol (TCP), User Name (avdf_user), and Password (masked).

- c. Click **[Save]** and target home page for “**hr**” is displayed as shown here

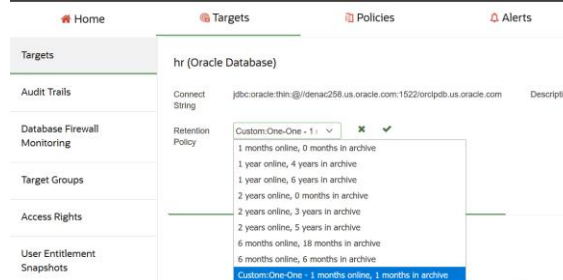


2.1.3. Configure an archiving policy for “hr” target

- As AVADMIN, open “**Settings**” menu and click [**Archiving**] in the left menu
- Navigate to “**Manage Policies**” tab
- Click [**Create**] to create a new data retention policy and enter details below:
 - Name (**Custom:One-One**)
 - Months Online (**1**)
 - Months Archived(**1**)

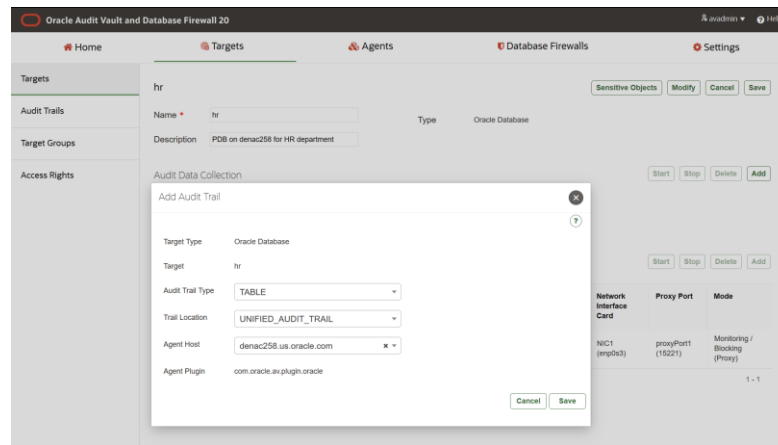


- Click [**Save**]
- Login to **AVS server** as **AVAUDITOR**
- Open “**Targets**” menu, drilldown into the target and select the retention policy “**Custom:One-One - 1 months online, 1 months in archive**” in the dropdown
- Click the checkmark (✓) next to the dropdown to save the configuration

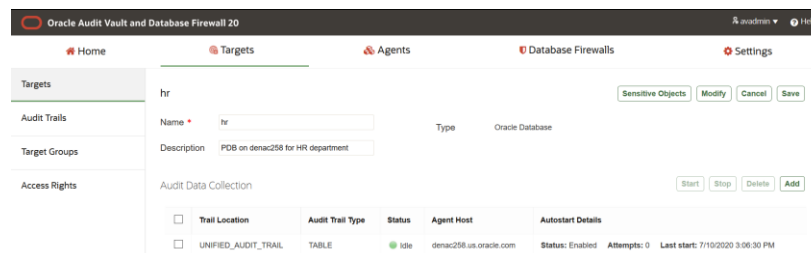


2.1.4. Configure the audit trails for collecting audit records from “hr” target

- Login to **AVS server** as **AVADMIN**
- Open “**Targets**” menu and drilldown into target “**hr**”
- Click [**Add**] in the “Audit Data Collection” region to open a popup to add audit trail for Unified Audit Trail



- d. Enter the following details in the dropdowns within “**Add Audit Trail**” popup as shown
 - i. Audit Trail type: `table`
 - ii. Trail Location: `UNIFIED_AUDIT_TRAIL`
 - iii. Agent Host: `<DB server where agent is installed>`
- e. Click [**Save**] to display the audit trail configured in the “**hr**” target home page



- 2.1.5. Enable audit trail cleanup in the pluggable database instance. This will enable the audit records purge from “hr” target as they are stored in the Audit Vault server repository.



`oracle_audit_trail_cleanup.sql`

- a. Open the script and execute.

3. CONFIGURING SENSITIVE DATA REPORTS

Goal: Use a proactive approach to monitor activities on sensitive data and alert if any un-authorized activities (data modifications) takes place on sensitive data.

3.1. Configurations

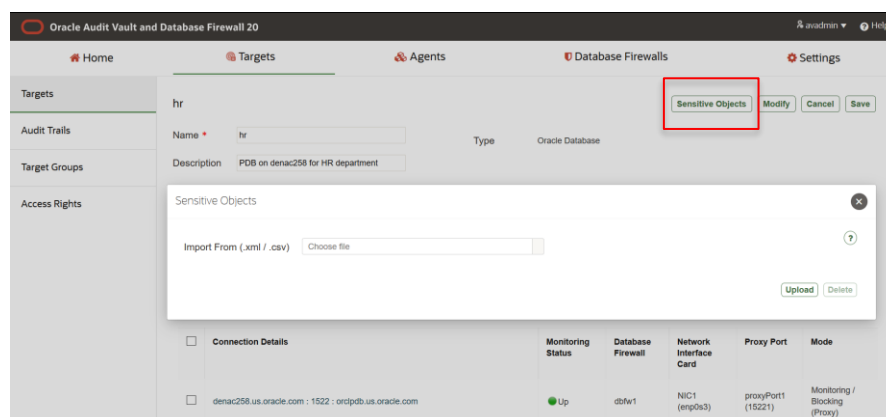
- a. Consider granting AV_SENSITIVE_ROLE to AVADMIN user

- i. Login into **AVS server** appliance as support, su to root, and then become the Oracle user using '**su - oracle**'
- ii. Execute the following command as oracle as shown in the screenshot:

```
python /usr/local/dbfw/bin/av_sensitive_role grant avadmin
```

```
[root@avs080027a6cb6b ~]# su - oracle
Last login: Thu Jul  9 12:19:35 UTC 2020
The Oracle base has been set to /var/lib/oracle
[oracle@avs080027a6cb6b ~]# python /usr/local/dbfw/bin/av_sensitive_role grant avadmin
Role granted successfully
[oracle@avs080027a6cb6b ~]# _
```

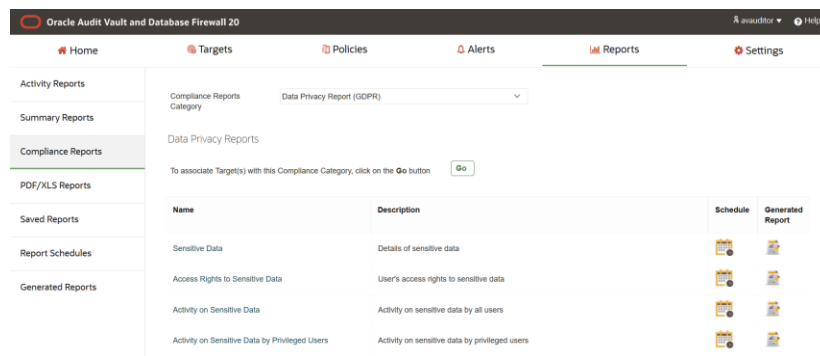
- b. Login to **the Audit Vault server console** as AVADMIN
- c. Copy the sensitive data discovery results which have already been staged for you. Refer **out_discover.csv** in the archived folder.
Note: This CSV file is the output of a DBSAT discovery run against a database with the HCM demo schema installed.
- d. Go to the '**Targets**' home page
- e. Drilldown into the registered '**hr**' target
- f. Click **[Sensitive Objects]** in the '**hr**' home page



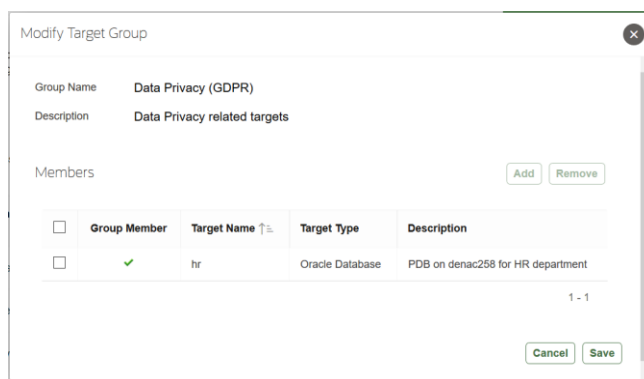
- g. Click **[Choose file]** to browse for the file from the local and upload the file by clicking **[Upload]**
- h. Click **[Save]** in the target home page

3.2. Verification of successful import

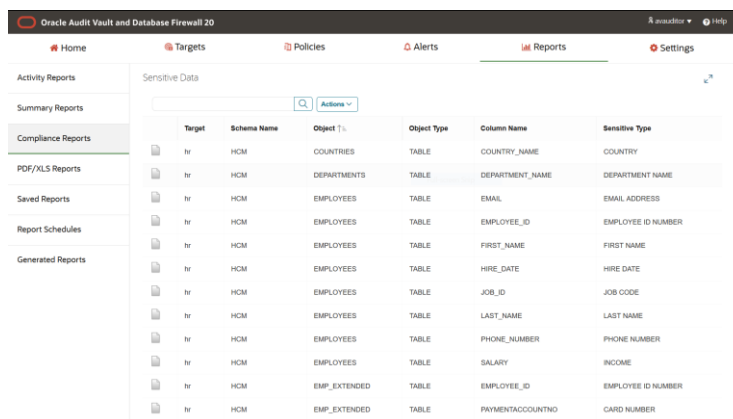
- a. Login to AV console as **AVAUDITOR**
- b. Go to '**Reports**' home page
- c. Click '**Compliance Reports**' in the left navigation menu
- d. Click '**Go**' to associate the target:



- In the popup, select the checkbox for the 'hr' target
- Click **[Add]**
- A green checkmark (✓) is displayed against the record.



- Click **[Save]** to close the popup.
- Click **'Sensitive Data'** in the Data Privacy Reports to see if sensitive data is shown:



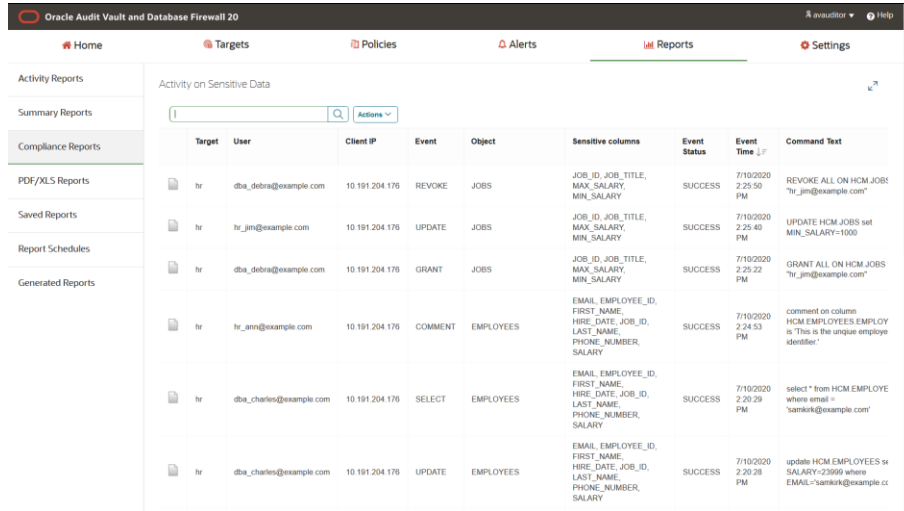
With this complete, Audit Vault is now aware of sensitive data residing in the protected database.

3.3. Tracking activities on sensitive data

- Navigate to the following report in 'Reports' home page
- Go to **'Compliance Reports' > 'Activity on Sensitive Data'**
- Observe the activities of users on the sensitive data in the report.

Note: If you have not run the workload script as part of [Introduction](#) pre-requisites, execute the


hcm_workload_script.
 script **sql** to see the data.



Target	User	Client IP	Event	Object	Sensitive columns	Event Status	Event Time	Command Text
hr	dba_debra@example.com	10.191.204.176	REVOKE	JOBS	JOB_ID, JOB_TITLE, MAX_SALARY, MIN_SALARY	SUCCESS	7/10/2020 2:25:50 PM	REVOKE ALL ON HCM JOBS 'hr_jm@example.com'
hr	hr_jm@example.com	10.191.204.176	UPDATE	JOBS	JOB_ID, JOB_TITLE, MAX_SALARY, MIN_SALARY	SUCCESS	7/10/2020 2:25:40 PM	UPDATE HCM JOBS set MIN_SALARY=1000
hr	dba_debra@example.com	10.191.204.176	GRANT	JOBS	JOB_ID, JOB_TITLE, MAX_SALARY, MIN_SALARY	SUCCESS	7/10/2020 2:25:22 PM	GRANT ALL ON HCM JOBS 'hr_jm@example.com'
hr	hr_arn@example.com	10.191.204.176	COMMENT	EMPLOYEES	EMAIL, EMPLOYEE_ID, FIRST_NAME, HIRE_DATE, JOB_ID, LAST_NAME, PHONE_NUMBER, SALARY	SUCCESS	7/10/2020 2:24:53 PM	comment on columns HCM EMPLOYEES EMPLOY is 'This is the unique employee identifier'
hr	dba_charles@example.com	10.191.204.176	SELECT	EMPLOYEES	EMAIL, EMPLOYEE_ID, FIRST_NAME, HIRE_DATE, JOB_ID, LAST_NAME, PHONE_NUMBER, SALARY	SUCCESS	7/10/2020 2:20:29 PM	select * from HCM EMPLOYE where email = 'charles@example.com'
hr	dba_charles@example.com	10.191.204.176	UPDATE	EMPLOYEES	EMAIL, EMPLOYEE_ID, FIRST_NAME, HIRE_DATE, JOB_ID, LAST_NAME, PHONE_NUMBER, SALARY	SUCCESS	7/10/2020 2:20:28 PM	update HCM EMPLOYEES set SALARY=23999 where EMAIL='samir@example.com'

d. To track activities on sensitive data by privileged users:

1. Navigate to the following report in **'Reports'** home page
2. Go to **'Compliance Reports' > 'Activity on Sensitive Data by Privileged Users'**.
3. **The report has no data!**

This is because the Entitlements have not been fetched from the target **'hr'** into Audit Vault. There is no way that Audit Vault can know automatically who my privileged users are in the target **'hr'** until entitlements are fetched.

e. To fetch the entitlements

1. Login into **the** pluggable database target
2. Execute the following to grant the required entitlement retrieval privileges to **'avdfuser'** as shown in the screenshot:

```
@/<agent_home>/av/plugins/com.oracle.av.plugin.oracle/config/oracle_user_setup.sql avdfuser ENTITLEMENT
```

```
SQL> alter session set container=orclpdb;
Session altered.

SQL> @/scratch/adhanara/20.1AVAgent/agent_home/av/plugins/com.oracle.av.plugin.oracle/config/oracle_user_setup.sql avdfuser ENTITLEMENT
Session altered.

Granting privileges to 'AVDFUSER' ... Done.
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.5.0.0.0
```

3. Login to AV console as **AVAUDITOR**
4. To fetch the entitlements, go to **'Targets'** menu
5. Drilldown to **'hr'**
6. Go to **'User Entitlements'** region in **'Audit Data Collection'** tab
7. Enable checkbox **[Retrieve Immediately]**

8. Click [Save]

A job 'User Entitlement' is submitted.

9. Monitor the job status by going to 'Settings' page

10. Click 'Jobs' in the left navigation menu.

Job Type	Status	Last Updated	Started At	Created By	Message
User Entitlement	Completed	7/10/2020 3:40:12 PM	7/10/2020 3:39:36 PM	AUDITOR	

11. Navigate to the following report in 'Reports' home page

12. Go to 'Compliance Reports' > 'Activity on Sensitive Data by Privileged Users'.

13. Observe the activities of privileged users in the report. Now the data appears.

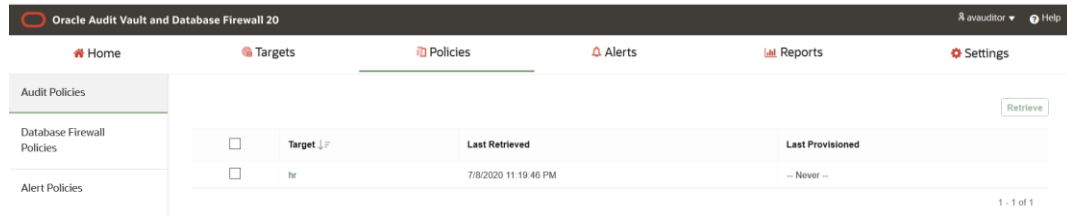
Target	User	Client IP	Event	Object	Sensitive columns	Event Status	Event Time	Command Text
hr	dba_debra@example.com	10.191.204.176	REVOKE	JOBS	JOB_ID, JOB_TITLE, MAX_SALARY, MIN_SALARY	SUCCESS	7/10/2020 2:25:50 PM	REVOKE ALL ON HCM JOBS from 'hr_jm@example.com'
hr	dba_debra@example.com	10.191.204.176	GRANT	JOBS	JOB_ID, JOB_TITLE, MAX_SALARY, MIN_SALARY	SUCCESS	7/10/2020 2:25:22 PM	GRANT ALL ON HCM JOBS to 'hr_jm@example.com'
hr	dba_charles@example.com	10.191.204.176	SELECT	EMPLOYEES	EMAIL, EMPLOYEE_ID, FIRST_NAME, HIRE_DATE, JOB_ID, LAST_NAME, PHONE_NUMBER, SALARY	SUCCESS	7/10/2020 2:20:29 PM	select * from HCM EMPLOYEES where email = 'samkirk@example.com'
hr	dba_charles@example.com	10.191.204.176	UPDATE	EMPLOYEES	EMAIL, EMPLOYEE_ID, FIRST_NAME, HIRE_DATE, JOB_ID, LAST_NAME, PHONE_NUMBER, SALARY	SUCCESS	7/10/2020 2:20:28 PM	update HCM EMPLOYEES set SALARY=23999 where EMAIL='samkirk@example.com'
hr	dba_charles@example.com	10.191.204.176	SELECT	EMPLOYEES	EMAIL, EMPLOYEE_ID, FIRST_NAME, HIRE_DATE, JOB_ID, LAST_NAME, PHONE_NUMBER, SALARY	SUCCESS	7/10/2020 2:20:28 PM	select * from HCM EMPLOYEES where email = 'henrywil@example.com'
hr	dba_charles@example.com	10.191.204.176	UPDATE	EMPLOYEES	EMAIL, EMPLOYEE_ID, FIRST_NAME, HIRE_DATE, JOB_ID, LAST_NAME, PHONE_NUMBER, SALARY	SUCCESS	7/10/2020 2:20:27 PM	update HCM EMPLOYEES set SALARY=99999 where email = 'henrywil@example.com'

4. AUDIT POLICY PROVISIONING

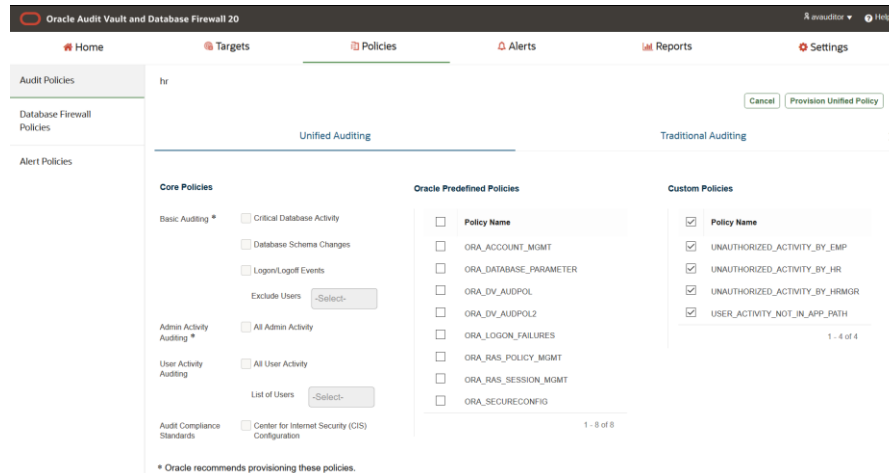
Goal: To retrieve audit policies already provisioned in the database target. Provision additional audit policies for some of the common security configurations / compliance needs.

4.1. Configurations

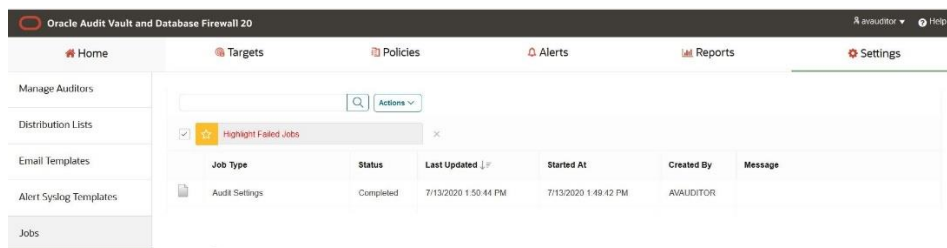
- As **AVAUDITOR**, go to “Policies” menu
- Click [**Audit Policies**] in the left navigation menu



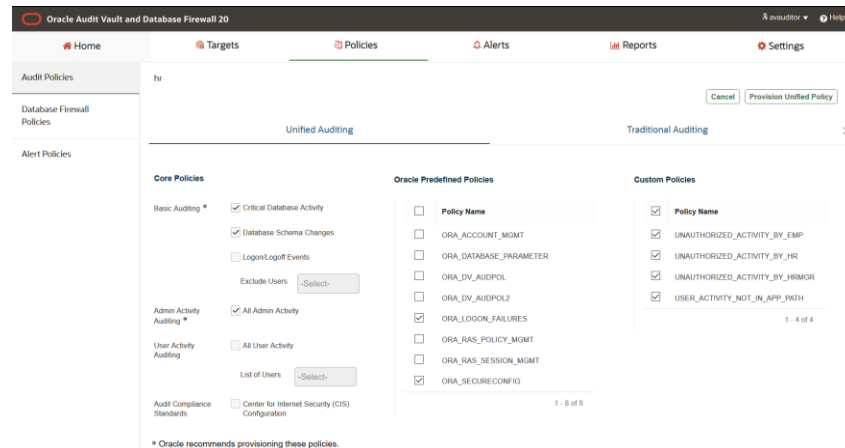
- Drilldown into ‘hr’ target to navigate to **Audit Policies** page as shown below



- Check the following core recommended policies
 - Critical Database Activity
 - Database Schema Changes
 - All Admin Activity
- Check the following predefined policies of Oracle Database
 - ORA_SECURECONFIG
 - ORA_LOGON_FAILURES
- You will observe that a few custom policies are already provisioned. They were created and enabled as part of the data script execution
- Click [**Provision Unified Policy**]
- Monitor the job “**Unified Audit Policy**” for completion status in Settings page as shown here



- i. Once the job is completed successfully, navigate back to the **Audit Policies** page, and observe that the provisioned policies are enabled



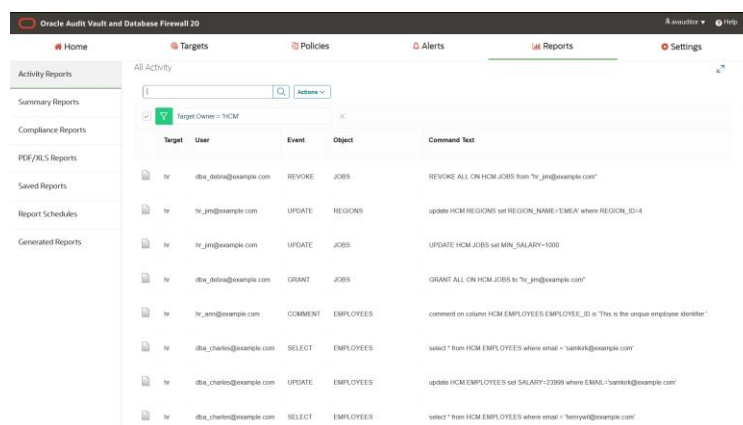
- j. To simulate audit events run the hcm_workload script again to generate new audit records



**hcm_workload_script.
sql**

4.2. Tracking audit events

- a. As AVAUDITOR, navigate to the **“Reports”** menu to see the **“All Activity”** reports. To see activity on the HCM schema, apply a filter on **‘Target Owner’**.



- b. Navigate to **“All Activity by Privileged Users”** report to see the activities of privileged users. Filter for any INSERTS, DELETES, and UPDATES on HCM schema.

Target	User	Event	Object	Event Status	Command Text	Event Time
hr	dba_charles@example.com	UPDATE	EMPLOYEES	SUCCESS	update HCM EMPLOYEES set SALARY=23999 where EMAIL='hcmk@example.com'	7/13/2020 4:15:10 PM
hr	dba_charles@example.com	UPDATE	EMPLOYEES	SUCCESS	update HCM EMPLOYEES set SALARY=99999 where email = 'hcmk@example.com'	7/13/2020 4:15:09 PM

Notice that DBA Charles has modified the salary of two employees, which he is not authorized to. This audit event is captured because we have provisioned "Admin Activity Auditing" audit policy which tracks actions of privileged users in the system.

- c. Navigate to "Failed Login Events" report to see the failed login audit events captured

Target	User	Client IP	Event	Object	Event Status	Event Time
hr	secadmin_stevens@example.com	10.195.180.249	LOGIN		FAILURE	7/13/2020 4:21:41 PM
hr	secadmin_stevens@example.com	10.195.180.249	LOGIN		FAILURE	7/13/2020 4:21:37 PM
hr	secadmin_stevens@example.com	10.195.180.249	LOGIN		FAILURE	7/13/2020 4:21:33 PM
hr	secadmin_stevens@example.com	10.195.180.249	LOGIN		FAILURE	7/13/2020 4:21:28 PM
hr	secadmin_stevens@example.com	10.195.180.249	LOGIN		FAILURE	7/13/2020 4:21:24 PM

Note: These audit events are capture since we have provisioned ORA_LOGON_FAILURES predefined policy. To see the audit policy that generated the corresponding audit event, enable the extension column in these reports, and search for the keyword "UNIFIED_AUDIT_POLICIES=".

5. PROACTIVE MONITORING WITH ALERT CONFIGURATION

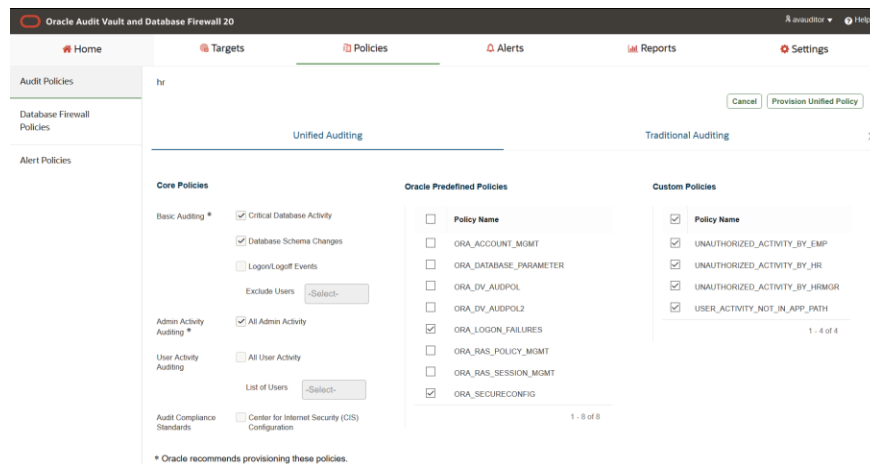
Goal: To create alert policies for some of the most common security-relevant activities. We will create alert policies for

- Login failures - these are frequently an indication of an attack in progress
- DBA activity on sensitive tables in HCM schema. DBAs should not normally access application sensitive data

5.1. Configurations

5.1.1. Ensure the audit policies discussed in the previous sections are provisioned

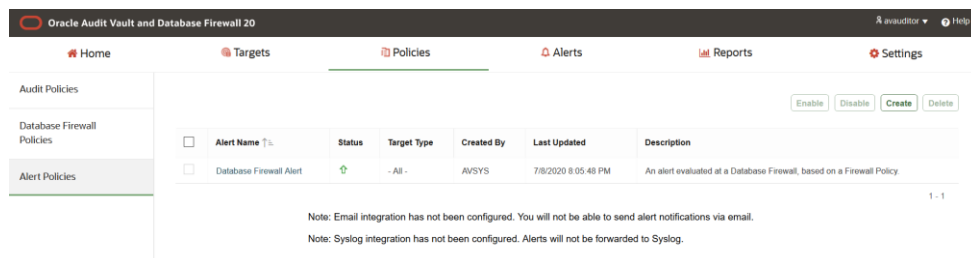
- Go to "**Policies**" menu as **AVAUDITOR**
- Click [**Audit Policies**] in the left navigation menu
- Drilldown into 'hr' target to navigate to **Audit Policies** page
- Ensure the policies are provisioned as shown below:



Note: We are leveraging audit policies `ORA_LOGON_FAILURES` and the 'Admin Activity' auditing policy to generate audit events that we will configure alerts on.

5.1.2. Examine the alert policies in the system

- Login to AV console as **AVAUDITOR**
- Go to **"Policies"** menu
- Click **[Alert Policies]** in the left navigation menu to see the alert policies in the system
- Database Firewall alert is a pre-seeded alert policy for network events. It is configured to generate alerts if the SQL traffic matches any firewall policy rule condition and corresponding rule action is block/alert.



Note that email and syslog connectors are not yet configured for proactive alert notifications.

5.1.3. Configure an alert policy to alert on consecutive failed login attempts

- Click **[Create]** in the Alert policies home page
- Name the alert policy as 'Oracle Failed Logon' and description as 'Alert when there are 5 failed logon attempts within 1 minute'
- Enter Group By (`USER_NAME`), Threshold (`5`), Duration (`1`), and Severity (`Warning`)
- Copy the following condition clause and paste into the Condition field:

```
upper(:EVENT_STATUS)='FAILURE' and upper(:EVENT_NAME)='LOGON'
```

- In the notification section,
 - Select 'Alert Notification template' from the dropdown.

- ii. Enter your email address
- iii. Click **[Add to List]** and ensure the notification list table is populated with the above entry

f. Click **[Save]**.

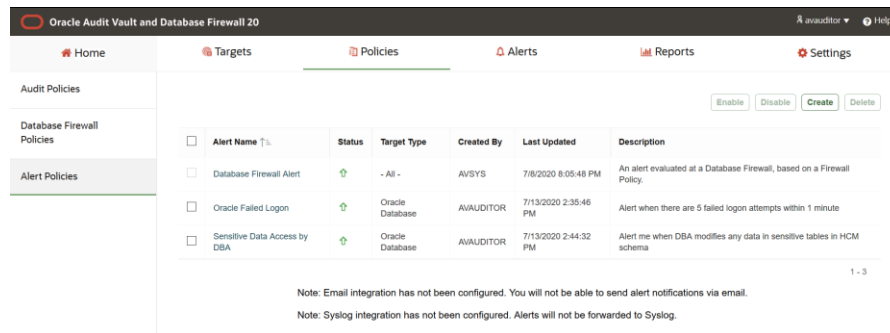
5.1.4. Configure an alert policy to alert on DBA activity on sensitive tables in HCM schema

- a. Click **[Create]** in the Alert policies home page
- b. Name the alert policy '**Sensitive Data Access by DBA**' and set the description as '**Alert me when DBA modifies any data in sensitive tables in HCM schema**'
- c. Enter Group By (**USER_NAME**), Threshold (**1**), Duration (**0**), and Severity (**Critical**).
- d. Copy the following condition clause and paste into the Condition field:

```
upper(:TARGET_OWNER)='HCM' and upper(:TARGET_OBJECT) IN
('EMPLOYEES','JOB_HISTORY','JOBS','SUPPLEMENTAL_DATA','EMP_EXTENDED') and
upper(:EVENT_NAME) NOT IN ('SELECT') and upper(:USER_NAME) like 'DBA_'
```

- e. In the notification section,
 - i. Select 'Alert Notification template' from the dropdown.
 - ii. Enter your email address
 - iii. Click **[Add to List]** and ensure the notification list table is populated with the above entry
- f. Click **[Save]**.

5.1.5. Three alert policies should be shown in the alert policies home page as given below:

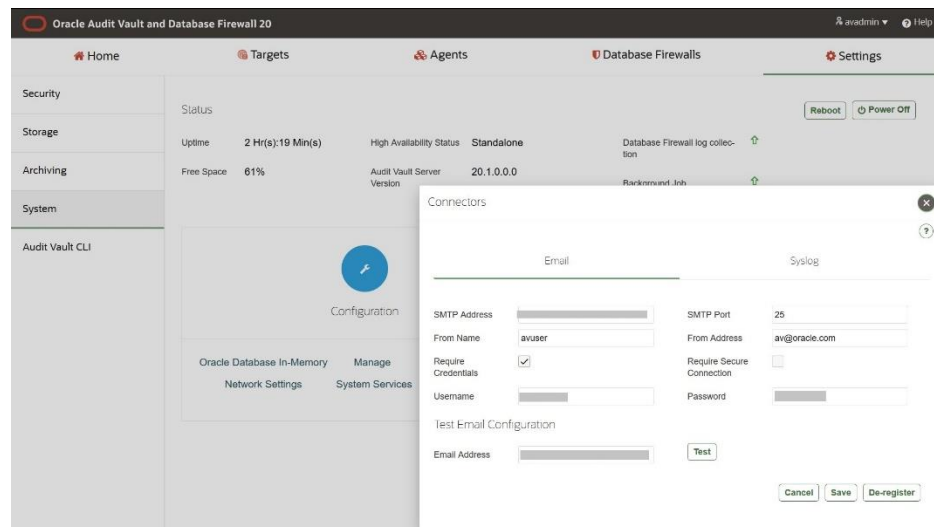


<input type="checkbox"/>	Alert Name ↑%	Status	Target Type	Created By	Last Updated	Description
<input type="checkbox"/>	Database Firewall Alert	↑	- All -	AVSYS	7/8/2020 8:05:48 PM	An alert evaluated at a Database Firewall, based on a Firewall Policy.
<input type="checkbox"/>	Oracle Failed Logon	↑	Oracle Database	AVAUDITOR	7/13/2020 2:35:46 PM	Alert when there are 5 failed logon attempts within 1 minute
<input type="checkbox"/>	Sensitive Data Access by DBA	↑	Oracle Database	AVAUDITOR	7/13/2020 2:44:32 PM	Alert me when DBA modifies any data in sensitive tables in HCM schema

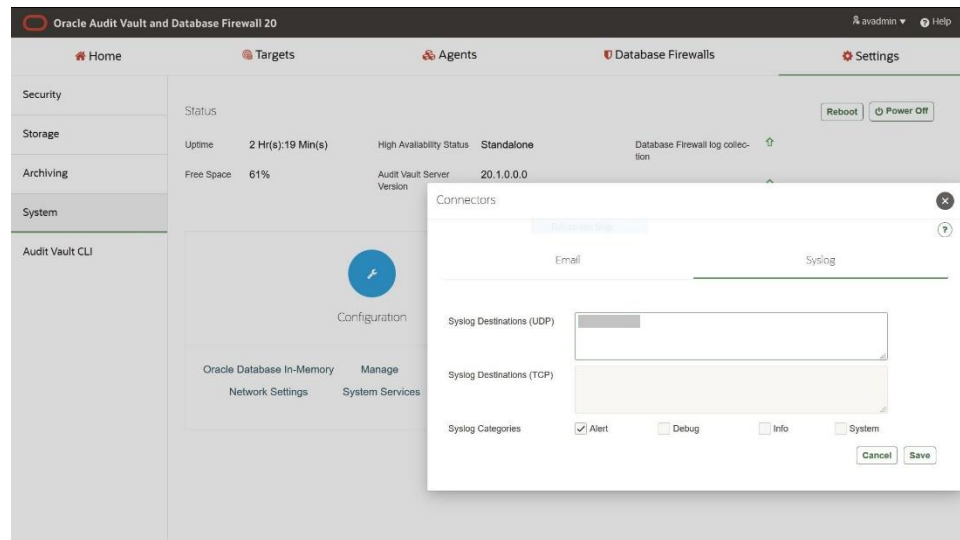
Note: Email integration has not been configured. You will not be able to send alert notifications via email.
Note: Syslog integration has not been configured. Alerts will not be forwarded to Syslog.

5.1.6. Configure email and syslog connector for proactive monitoring of alerts

- As AVADMIN, navigate to Settings menu and click **[System]** in left navigation menu
- Click **[Connectors]** in the Configuration section to open the popup to enter the SMTP server details- including address, port, username and password
- Click **[Register]** in the popup and optionally test the configuration.
- Click **[Save]**.

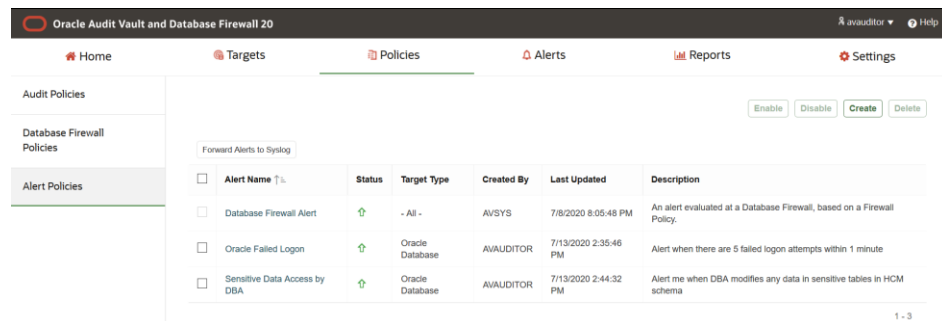


- Click **[Connectors]** in the Configuration section again to open the popup to enter the syslog server details.
- iv. Ensure the rsyslog service on the syslog server host is configured on a UDP or TCP port. For instance, to see if rsyslog is listening on UDP port, use "netstat -tnlpu | grep rsyslog" to check the same.
- f. Click **[Save]**.



5.1.7. Enable forwarding of alerts to syslog

- As **AVAUDITOR**, navigate to Alert policies home page
- Click the button [**Forward Alerts to Syslog**] to enable forwarding of alerts to the syslog server



**hcm_workload_script.
sql**

To simulate alerts, run the workload script again

5.2. Tracking alerts

As AVAUDITOR, navigate to “Alerts” menu to see the alerts raised:

Oracle Audit Vault and Database Firewall 20							
<div> Home Targets Policies Alerts Reports Settings </div>							
Alerts							
<div> <div>Manage Alert Status</div> <div> <input type="text"/> <div>Actions</div> </div> <div> <div>Set Alert Status</div> <div>Closed</div> <div>Apply</div> </div> <div> <div>Schedule Report</div> <div>Generated Report</div> <div>Notify</div> </div> </div>							
<div> <input checked="" type="checkbox"/> Alert Time is in the last 24 hours </div>							
<input type="checkbox"/>	Target	Alert Policy Name	Alert Status	Alert Severity	User	Alert Time	Event Time
<input type="checkbox"/>	hr	Oracle Failed Logon	New	Warning	secadmin_steve@example.com	7/13/2020 4:22:08 PM	7/13/2020 4:21:24 PM
<input type="checkbox"/>	hr	Sensitive Data Access by DBA	New	Critical	dba_debra@example.com	7/13/2020 4:20:40 PM	7/13/2020 4:20:28 PM
<input type="checkbox"/>	hr	Sensitive Data Access by DBA	New	Critical	dba_debra@example.com	7/13/2020 4:20:09 PM	7/13/2020 4:20:00 PM
<input type="checkbox"/>	hr	Sensitive Data Access by DBA	New	Critical	dba_charles@example.com	7/13/2020 4:15:23 PM	7/13/2020 4:15:10 PM

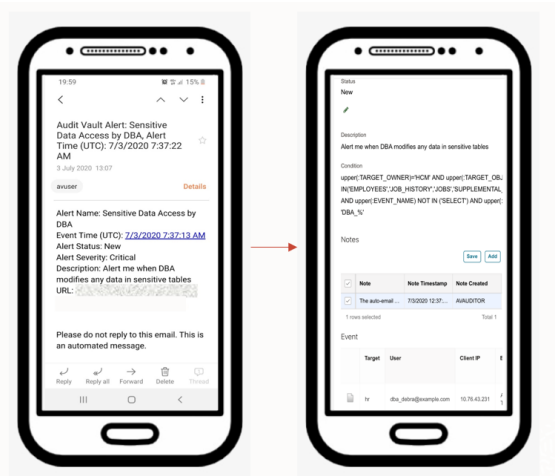
With email configured, you will receive an email alert like this:

From avuser <av@oracle.com> ☆
Subject: Audit Vault Alert: Sensitive Data Access by DBA, Alert Time (UTC): 7/13/2020 10:50:40 AM
To: [redacted] ★

Alert Name: Sensitive Data Access by DBA
Event Time (UTC): 7/13/2020 10:50:28 AM
Alert Status: New
Alert Severity: Critical
Description: Alert me when DBA modifies any data in sensitive tables in HCM schema
URL: https://10.89.33.136/console/f?p=7700:33::NO::P33_ALERT_ID:7

Please do not reply to this email. This is an automated message.

You can also view the email alerts on a mobile device, and drilldown to see relevant alert details as shown here:



With syslog configured, examine /var/log/messages in the syslog server to see the below messages:

```
Jul 13 10:45:23 av5080027a6cb6b oracle: [AVDFAlert@111 name="Sensitive Data Access by DBA" severity="Critical" url="https://10.89.33.136/console/f?p=7700:33::NO::P33_ALERT_ID:4" time="2020-07-13T10:45:23.32309Z" target="hr" user="dba_charles@example.com" desc="Alert me when DBA modifies any data in sensitive tables in HCM schema"]
```

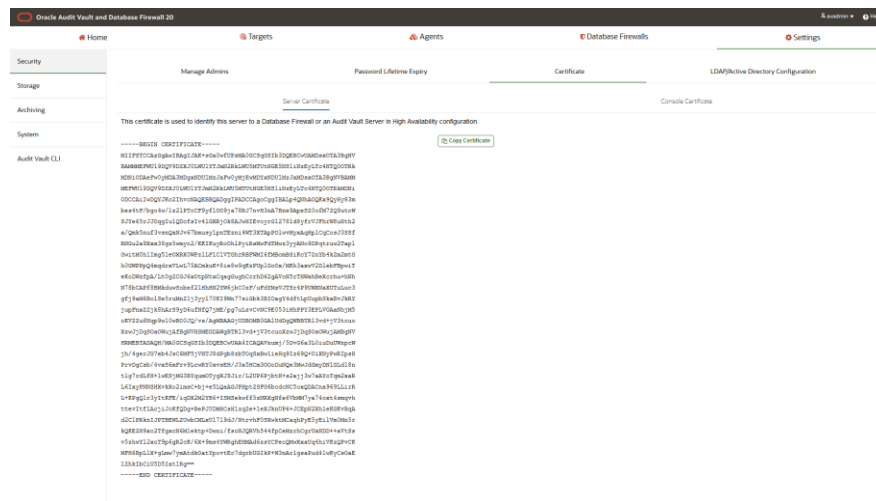
6. NETWORK MONITORING CONFIGURATION

Goal: To register Database Firewall instance to the Audit Vault server, configure network settings on the Database Firewall, and start monitoring the database target for network activity.

6.1. Pairing Database Firewall Server with Audit Vault Server

- Login into **AVS server** console as **AVADMIN**

- b. Go to '**Settings**' home page
- c. Click '**Security**' in left navigation menu.
- d. Navigate to '**Certificate**' tab
- e. Click **[Copy]** in the '**Server Certificate**' tab as shown here



- f. Copy to a local file '**ca.crt**' and using SCP, transfer it to /tmp folder in the Database Firewall server instance
- g. Login to the Database Firewall server instance as root.
- h. Execute the following command

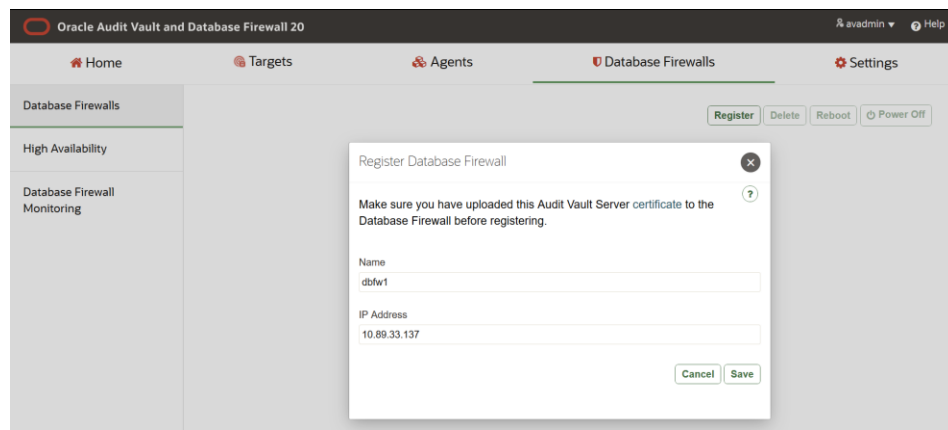
```
/opt/avdf/config-utils/bin/config-avs set avs=primary address=<IP address of AVS server> certificate=/tmp/ca.crt
```

```
[root@dbfw080027e4b5ac ~]# /opt/avdf/config-utils/bin/config-avs set avs=primary address=10.89.33.6 certificate=/tmp/key.txt
Notice: Success. Settings saved.
[root@dbfw080027e4b5ac ~]#
```

6.2. Preparing Database Firewall server

6.2.1. Register Database Firewall server with Audit Vault server

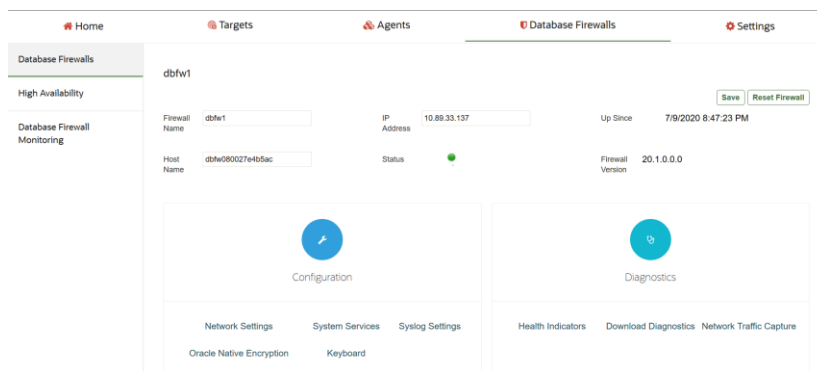
- a. Login to AV console as AVADMIN
- b. Navigate to 'Database Firewalls' menu
- c. Click **[Register]** to register the Database Firewall instance.
- d. Enter Name and IP Address of Firewall instance where it is installed.
- e. Click **[Save]** to see the registered firewall in "**Database Firewalls**" home page



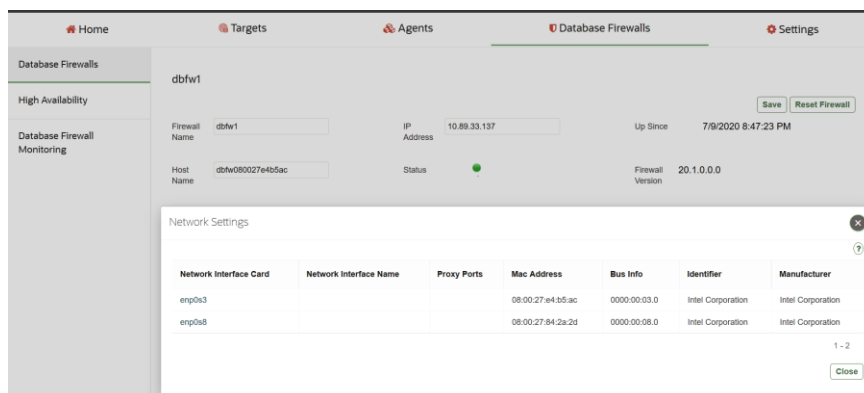
- f. On clicking Save, the Database Firewall is registered

6.2.2. Configuring network settings for registered instance

- Drilldown into the Firewall instance **you just registered** and configure network settings
- Go to '**Network Settings**' link in the Configuration region:



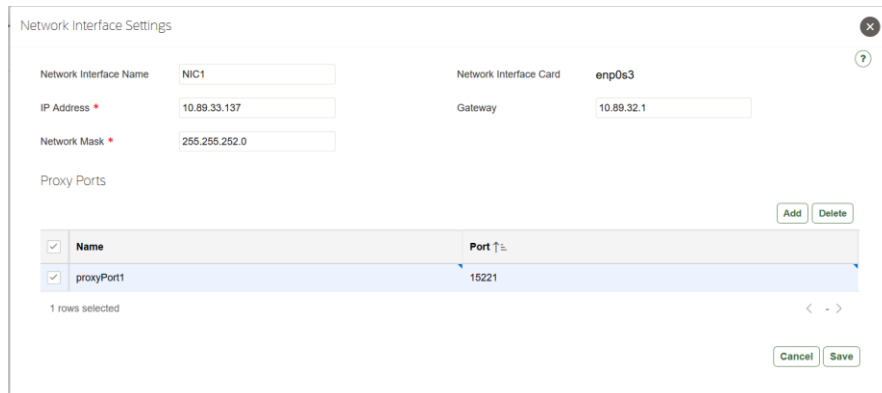
- c. In the 'Network Settings' popup that opens,



- Click **[enp0s3]** to open the dialog for configuring NIC interfaces in firewall.
- In the dialog box for configuring Firewall NIC interfaces,
 - Enter name(**NIC1**)
 - Click **[Add]**
 - Enter Port Name(**ProxyPort1**), Port(**15221**)

4. Click **[Save]**

This step opens the proxy port on **enp0s3** to receive client traffic.



The image shows a 'Network Interface Settings' dialog box. It contains fields for 'Network Interface Name' (NIC1), 'Network Interface Card' (enp0s3), 'IP Address' (10.89.33.137), 'Gateway' (10.89.32.1), and 'Network Mask' (255.255.252.0). Below these is a 'Proxy Ports' section with an 'Add' button and a table. The table has columns 'Name' and 'Port'. One row is selected with 'proxyPort1' and '15221'. At the bottom are 'Cancel' and 'Save' buttons.

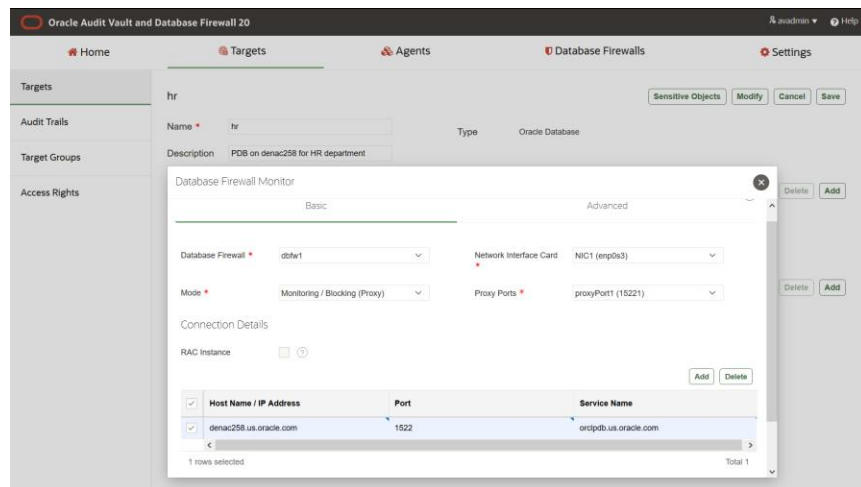
Name	Port
proxyPort1	15221

f. Once saved, close the popup.

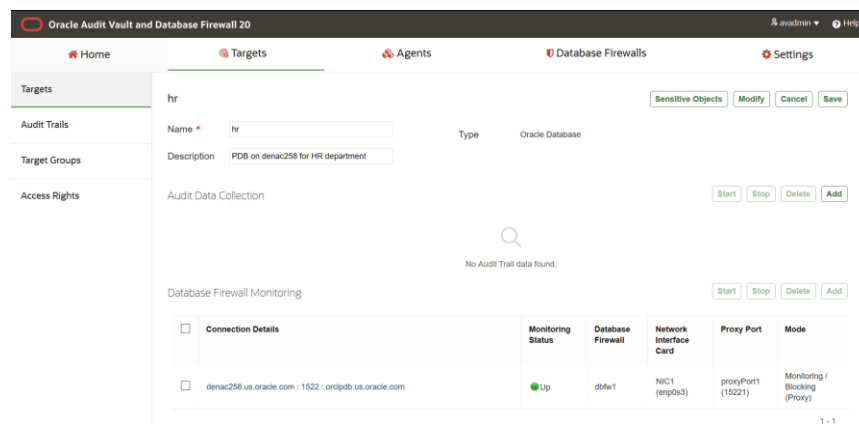
6.3. Configuration of Network monitoring for the pluggable database instance

6.3.1. Configuring Firewall monitoring on 'hr' target

- Navigate to **'Targets'** home page,
- Drilldown into target **'hr'**
- Click **[Add]** button in the Database Firewall monitoring region.
- In the popup,
 - Select Database Firewall(**dbfw1**),
 - Traffic Source(**enp0s3**),
 - Mode(Monitoring/Blocking)
 - Proxy Port (**15221**)
 - Enter the database target identified by the hostname /IP Address, with port and service name in the Target details region
- Click **[Save]**

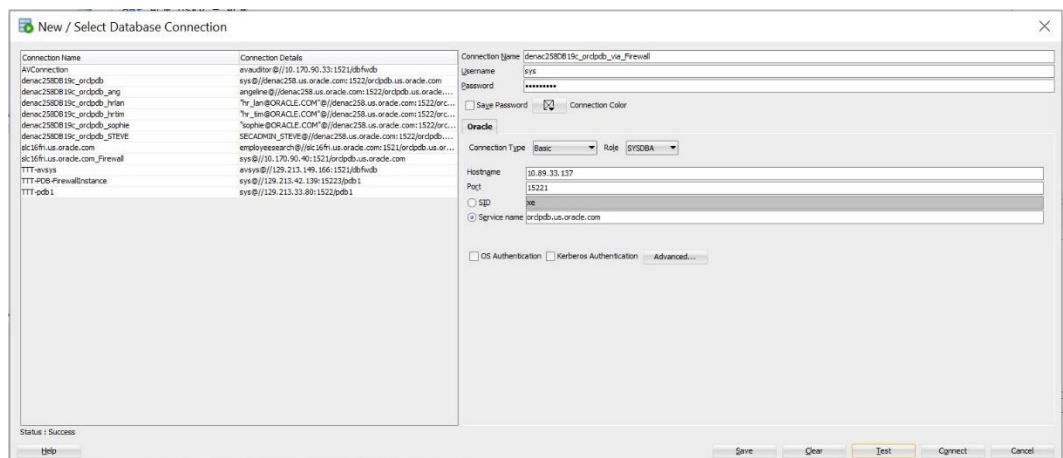


- f. Refresh the page and ensure the monitoring point is started as shown here.



Note: If you are using a FQDN name for the target, ensure the DNS system service for the Database Firewall is configured.

6.3.2. Test the proxy by connecting to the IP address and port of the Database Firewall:



Now the clients can connect to the database through the firewall!

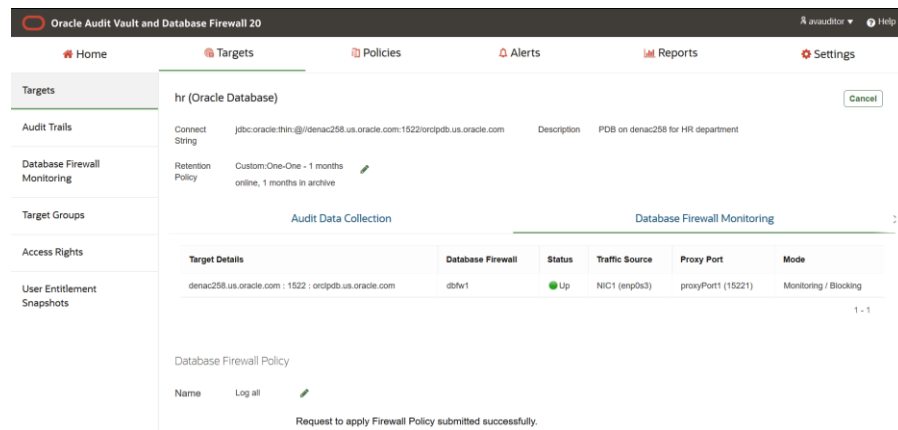
7. DATABASE FIREWALL POLICY CONFIGURATION

Goal: You will create a firewall policy that allows the employee application SQL traffic from an employee authorized connection while blocking a connection from an unauthorized IP address. You will be alerted on data modification attempts on sensitive data that are not in the allow-list of application SQL traffic.

7.1. Train the firewall to understand the permitted SQL traffic

7.1.1. Setting the Database Firewall in “Log all” policy mode for training :

- Login to AV console as **AVAUDITOR**,
- Go to ‘**Targets**’ menu,
- Drilldown into “**hr**”
- Click on ‘**Database Firewall Monitoring**’ sub-tab as shown below



- In the ‘Database Firewall Policy’ region,
 - Click the pencil icon to edit
 - Select the firewall policy ‘**Log all**’ from the dropdown
 - Click the checkmark (✓) to save the setting.

7.1.2. Capture the normal expected application traffic

- For typical employee application SQL traffic:

Compile the employee.java and then execute it in your environment giving args[0] as firewall IP address, args[1] as proxy port, args[2] as the database service name, and args[3] as sys password supplied in the prior data loading script

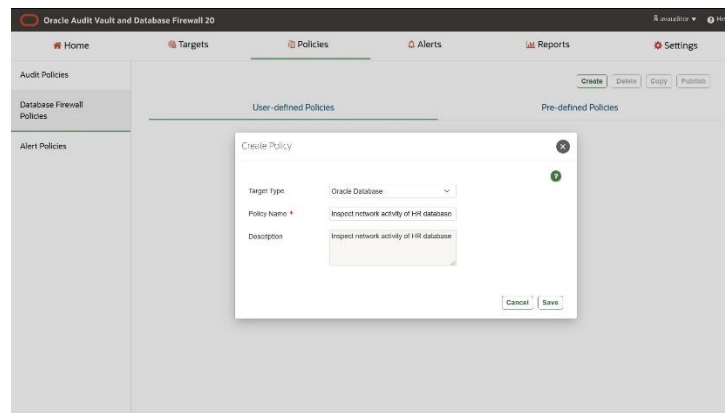


```
D:\DBSecurity\AVDF\UITesting\AVDF20Customers>java -classpath D:\software\OJDBC\ojdbc6.jar;. employee 10.89.33.137 15221 orclpdb.us.oracle.com
Connection String entered is ::jdbc:oracle:thin:@//10.89.33.137:15221/orclpdb.us.oracle.com
Training script for employee application SQL traffic
Fetching the employee record :: Sophie
Completed updating the employee record :: SJAIN
Successful update processed for :: Sophie
Firewall training script successful
```

7.2. Create a user-defined firewall policy to inspect network activity of HR Database

7.2.1. Create a user-defined Database Firewall policy

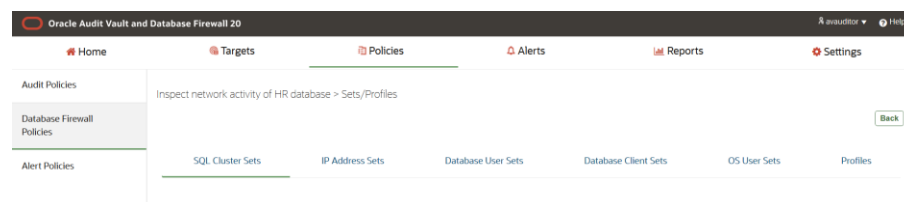
- a. Log into the AV console as **AVAUDITOR**
- b. Navigate to the '**Policies**' menu
- c. Select '**Database Firewall Policies**' in the left navigation menu: 'User-defined Policies' tab shows empty list.
- d. Click [**Create**] button



- e. In the popup that appears, fill in the appropriate information
 1. Database Type: **Oracle Database**
 2. Policy Name: **Inspect network activity of HR database**
 3. Description: **Inspect network activity of HR database**
 4. Click [**Save**]:
 5. Proceed to creating the Database Firewall policy rules.

7.2.2. Create a firewall policy rule to block unauthorized IP address connections.

- a. In the Database Firewall "**Inspect network activity of HR database**" home page,
 1. Click [**Sets/Profiles**] to see the page shown here



- b. Create '**IP Address Sets**' to create allowed IP address connections to the database.
 1. Click [**Add**]

- Enter Name/Description: **Allowed corporate IP address**
- In '**Sets Values**' field, enter your allowed IP address values(new line separated)
- Click **[Save]**

Inspect network activity of HR database > Sets/Profiles

Back

SQL Cluster Sets	IP Address Sets	Database User Sets	Database Client Sets	OS User Sets	Profiles
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Name	Values	Description			
Allowed corporate IP address	10.191.240.74	Allowed corporate IP address			

1 - 1 of 1

c. Create a **"Session Context Rule"** to block unauthorized IP address connections

1. Enter Rule Name: **Block unauthorized IP address**
2. Select 'IP Address Set' to exclude '**Allowed corporate IP address**' from the dropdown. Any IP address which is NOT IN < **Allowed corporate IP address** > will be blocked.
3. In the Action region, enter the following and click **[Save]**.

Actions: **Block**

Logging Level: **Always**

Threat Severity: **Critical**

4. Session context rule appears as shown here

Rules

Session Context	SQL Statement	Database Objects	Default
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Rule Name	Action	Logging Level	Threat Severity
Block unauthorized IP address	Block	Always	Critical
Block SQL traffic if it is not from allowed IP address list			

Add Delete

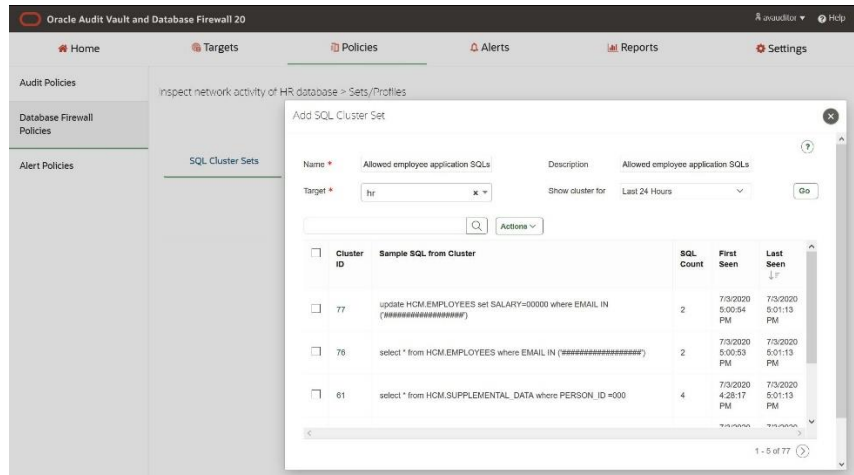
1 - 1

7.2.3. Create a firewall policy rule to allow trained application SQL traffic from employees

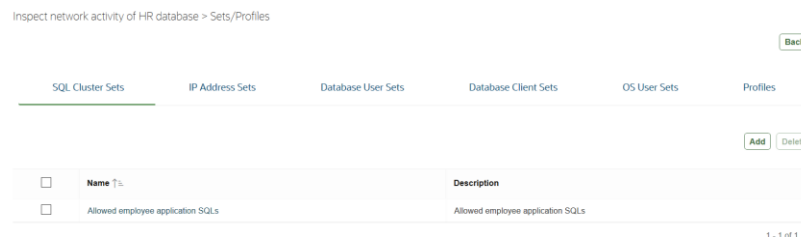
- a. Click **[Sets/Profiles]** and create **'SQL Cluster Sets'** to create a set of allowed employee application SQL traffic (i.e. firewall has been trained for this traffic using the *logall* policy).

1. Click **[Add]**

- Enter Name: **Allowed employee application SQLs**
- Select **'hr'** in the target dropdown
- Click **[Go]** to get the clusters generated by the firewall from the trained SQL traffic.
- Select all the SQL clusters and click **[Save]**



- The SQL cluster set appears as below:



- b. Create **'Database User Sets'** to create a set for database users that represents the shared application service account **EMPLOYEE_APPUSER**.

2. Click **[Add]**

- Enter Name: **Shared application service account for employees,**
- In **'Sets Values'** field, enter: **EMPLOYEE_APPUSER**
- Click **[Save]** to see the below entry

Back

SQL Cluster Sets

IP Address Sets

Database User Sets

Database Client Sets

OS User Sets

Profiles

Add Delete

<input type="checkbox"/>	Name ↑	Values	Description
<input type="checkbox"/>	Shared application service account for employees	EMPLOYEE_APPUSER	Shared application service account for employees

1 - 1 of 1

c. Create **'Database Client Set'** to indicate the allowed DB Client program for employees

3. Click **[Add]**

- Enter Name (**Allowed client for employees**),
- In the **'Set Values'** column, enter: **JDBC Thin Client**
- Click **[Save]** to see the created Database Client set as shown here

Back

SQL Cluster Sets

IP Address Sets

Database User Sets

Database Client Sets

OS User Sets

Profiles

Add Delete

<input type="checkbox"/>	Name ↑	Values	Description
<input type="checkbox"/>	Allowed client for employees	JDBC Thin Client	Allowed client for employees

1 - 1 of 1

d. Create a **'Profile'** in the last tab 'Profiles' to identify authorized users leveraging an approved connection pattern

- Enter Name: **Employee profile**
- Select the dropdowns for IP Address Set, DB User Set and DB Client Set

Back

SQL Cluster Sets

IP Address Sets

Database User Sets

Database Client Sets

OS User Sets

Profiles

Modify Profile

☐ Name
☐ Employee profile

Name: Employee profile

IP Address Set: Allowed corporate IP address

OS User Set: -- Not Set --

Description: Employee profile

DB User Set: Shared application service accou

DB Client Set: Allowed client for employees

Cancel Save

- Click **[Save]**
- Click **[Back]** to go back to the Database Firewall policy home page
"Inspect network activity of HR database"

e. Create a **"SQL Statement Rule"** to forward the SQL activities that are coming from authorized connections if the SQL queries are on among the allowed application SQL traffic.

- Threat Severity: Minimal

5. SQL Statement Rule appears as shown here

Rules

Session Context

SQL Statement

Database Objects

Default

Add

Delete

<input type="checkbox"/>	Rule Name	Profile Name	Cluster Sets	Action	Logging Level	Threat Severity	Description
<input type="checkbox"/>	Allowed SQL traffic from employees	Employee profile	Allowed employee application SQLs	Pass	One-Across-Sessions	Minimal	Allowed SQL traffic from employees

- a. Create a **“Database Objects Rule”** to monitor any modification attempts on sensitive data (apart from what has been allowed as application SQL traffic in prior rule).

- i. Enter Rule Name: **Modification attempts on sensitive tables in HCM schema**
- ii. Select Statement Classes **‘Data Manipulation’** from the dropdown.
- iii. Toggle Selected tables to **“Any”**.
- iv. Select the following tables in the HCM schema and Click **[Add]**
 - HCM.EMPLOYEES
 - HCM.EMP_EXTENDED
 - HCM.EMP_EXTENDED
 - HCM.SUPPLEMENTAL_DATA
- v. In the Action region, enter the following and click **[Save]**.

Actions: **Alert**

Logging Level: **One-per-Session**

Threat Severity: **Major**

Database Objects

Database Tables

Selected Tables

ANY ?

Remove

Add

<input type="checkbox"/>	Table Name	Target Name	In Policy
<input type="checkbox"/>	HCM.EMPLOYEES	hr	✓
<input type="checkbox"/>	HCM.EMP_EXTENDED	hr	✓
<input type="checkbox"/>	HCM.JOBS	hr	✓
<input type="checkbox"/>	HCM.SUPPLEMENTAL_DATA	hr	✓
<input type="checkbox"/>	HCM.DEPARTMENTS	hr	✗

1 - 5

Action

Action **Alert** ▼

Logging Level **One-Per-Set** ▼

Threat Severity **Major** ▼

Cancel

Save

vi. Database Object Rules appear as shown here

Rules

Session Context

SQL Statement

Database Objects

Default

:

Add

Delete

<input type="checkbox"/>	Rule Name	Rule Type	Action	Logging Level	Threat Severity	Description
<input type="checkbox"/>	Modification attempts on sensitive tables in HCM schema	Any	Alert	One-Per-Session	Major	Modification attempts on sensitive tables in HCM schema

1 - 1

7.2.5. Create a “Default Rule” to allow UNKNOWN SQL traffic with Minor Threat Severity

- In the Action region, enter the following and click [**Save**].

Actions: **Pass**

Logging Level: **One-Per-Session**

Threat Severity: **Minor**

Saved Default Rule appears as shown here

Rules			
Session Context	SQL Statement	Database Objects	Default
Rule Name	Action	Logging Level	Threat Severity
Default Rule	Pass	One-Per-Session	Minor

1 - 1

7.2.6. Publish and deploy the user-defined policy

- Click **[Save]** in the Database Firewall policy home page.
- Click **“Database Firewall Policies”** in the left navigation menu to see the list of user-defined policies
- Select the policy **“Inspect network activity of HR database”** and Click **[Publish]**
- Go to **“Targets”** home page and drilldown into **“hr”** target
- Navigate to the **“Database Firewall Monitoring”** tab and change the Firewall policy to **“Inspect network activity of HR database”**.
- Click **checkmark**.
- Firewall policy **“Inspect network activity of HR database”** is applied !!!

7.2.7. Simulation of network activity on the Firewall

- To run the authorized employee application SQL traffic:
Execute `employee.java` in your environment giving `args[0]` as firewall IP address, `args[1]` as proxy port, `args[2]` as the database service name, and `args[3]` as sys password supplied in the prior data loading script



```
D:\DBSecurity\AVDF\UITesting\AVDF20Customers>java -classpath D:\software\OJDBC\ojdbc6.jar;. employee 10.89.33.137 15221 orclpdb.us.oracle.com
Connection String entered is ::jdbc:oracle:thin:@//10.89.33.137:15221/orclpdb.us.oracle.com
Training script for employee application SQL traffic
Fetching the employee record :: Sophie
Completed updating the employee record :: SJAIN
Successful update processed for :: Sophie
Firewall training script successful
```

- To mimic DBA activity over the network and see how it is processed by Firewall:
Run `dba_over_network.sql` using an SQL connection through the firewall IP address



7.3. Tracking network activity using reports

- Go to **“Reports”** home page
- Click **[Activity Reports]** in the left navigation menu

- c. Click **[Database Firewall Monitored Activity]** report
- d. Select columns Policy Name, Log Cause, and Action Taken from Actions dropdown
- e. Filter by column Log Cause which represents the different firewall policy rules to see the below network activity.
 1. Session Context rule (Log cause = exception) blocks the SQL traffic to the database if the authorized IP address connection does not happen on the allowed corporate IP address (10.191.240.74)

Database Firewall Monitored Activity

<input type="text"/> <input type="button" value="Go"/> <input type="button" value="Actions"/>										
<input checked="" type="checkbox"/> Log Cause = 'exception'										
	Target	Client IP	User	Client Program	Command Text	Threat Severity	Action Taken	Policy Name	Log Cause	Event Time
	hr	10.166.180.249	EMPLOYEE_APPUSER	JDBC Thin Client	select * from HCM.EMPLOYEES where employee_id=000	catastrophic	block	Inspect network activity of HR database	exception	7/13/2020 9:54:39 PM

1 - 1 of 1

2. SQL Statement rule (Log cause =cluster) allows employee application SQL traffic (within SQL cluster set) from authorized employee connections (represented as the Employee profile).

Database Firewall Monitored Activity

<input type="text"/> <input type="button" value="Go"/> <input type="button" value="Actions"/>										
<input checked="" type="checkbox"/> Log Cause = 'cluster'										
	Target	Client IP	User	Client Program	Command Text	Threat Severity	Action Taken	Policy Name	Log Cause	Event Time
	hr	10.191.240.74	EMPLOYEE_APPUSER	JDBC Thin Client	select * from HCM.EMPLOYEES where EMAIL IN ('#####')	insignificant	pass	Inspect network activity of HR database	cluster	7/14/2020 3:49:15 PM
	hr	10.191.240.74	EMPLOYEE_APPUSER	JDBC Thin Client	select * from HCM.SUPPLEMENTAL_DATA where PERSON_ID=000	insignificant	pass	Inspect network activity of HR database	cluster	7/14/2020 3:49:15 PM

3. Database Object rule (Log cause =novelty) alerts on database modification attempts on sensitive tables in HCM schema which is not coming from Employee profile and application SQL traffic. The SQL traffic is forwarded to the database:

Database Firewall Monitored Activity

<input type="text"/> <input type="button" value="Go"/> <input type="button" value="Actions"/>										
<input checked="" type="checkbox"/> Log Cause = 'novelty'										
	Target	Client IP	User	Client Program	Command Text	Threat Severity	Action Taken	Policy Name	Log Cause	Event Time
	hr	10.191.240.74	HCM	SQL Developer	update HCM.EMPLOYEES set SALARY=00000 where EMAIL='#####'	major	warn	Inspect network activity of HR database	novelty	7/14/2020 3:52:31 PM
	hr	10.191.240.74	"dba_charles@example.com"	SQL Developer	update HCM.EMPLOYEES set SALARY=00000 where email = "#####"	major	warn	Inspect network activity of HR database	novelty	7/14/2020 3:50:31 PM

4. Default rule (Log cause = unseen) helps track unknown SQL traffic while allowing the SQL traffic to be forwarded to the database

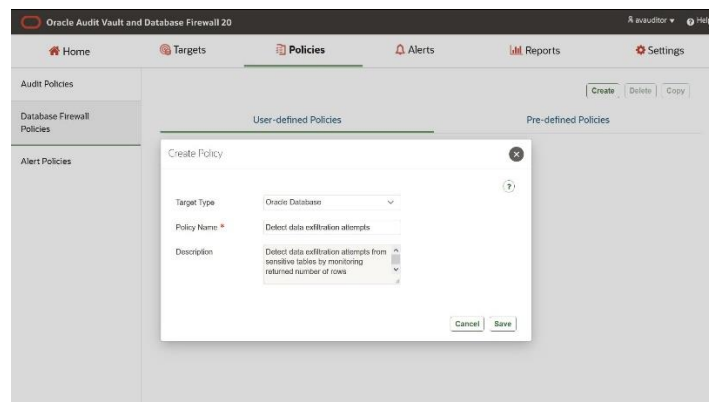
Database Firewall Monitored Activity

<input checked="" type="checkbox"/>	<input type="text" value="Log Cause = 'unseen'"/>	<input type="button" value="Go"/>	<input type="button" value="Actions"/>							
	Target	Client IP	User	Client Program	Command Text	Threat Severity	Action Taken	Policy Name	Log Cause	Event Time
	hr	10.191.240.74	"dba_charles@example.com"	SQL Developer	select null as "MaliciousWork8" from dual	minor	pass	Inspect network activity of HR database	unseen	7/14/2020 3:52:16 PM
	hr	10.191.240.74	"dba_charles@example.com"	SQL Developer	truncate table audsys.aud\$unified	minor	pass	Inspect network activity of HR database	unseen	7/14/2020 3:50:35 PM
	hr	10.191.240.74	"dba_charles@example.com"	SQL Developer	delete from unified_audit_trail	minor	pass	Inspect network activity of HR database	unseen	7/14/2020 3:50:34 PM

7.3. Create a user-defined firewall policy to detect data exfiltration attempts from HR Database

7.3.1. Create a user-defined Database Firewall policy

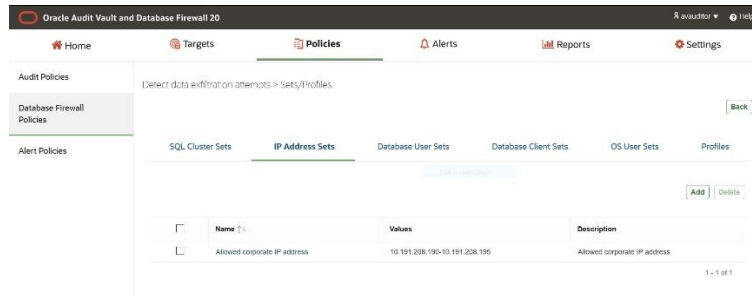
- Log into the AV console as **AVAUDITOR**
- Navigate to the **'Policies'** menu
- Select **'Database Firewall Policies'** in the left navigation menu: 'User-defined Policies' tab shows empty list.
- Click **[Create]** button



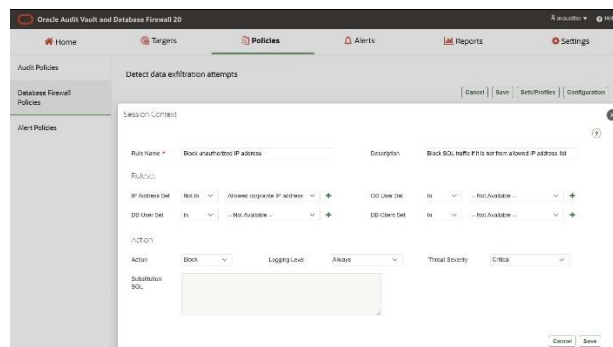
- In the popup that appears, fill in the appropriate information
 - Database Type: **Oracle Database**
 - Policy Name: **Detect data exfiltration attempts**
 - Description: **Detect data exfiltration attempts from sensitive tables by monitoring returned number of rows**
 - Click **[Save]**:
 - Proceed to creating the Database Firewall policy rules.

7.3.2. Create a firewall policy rule to block unauthorized IP address connections.

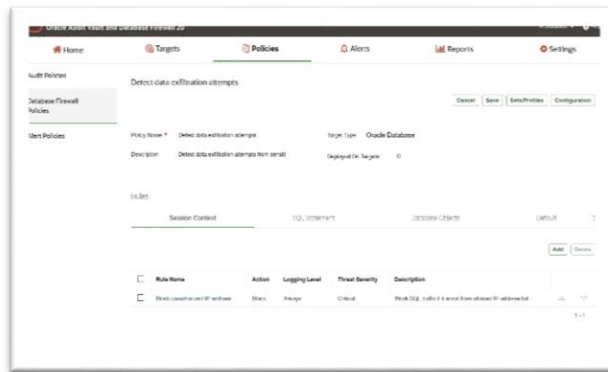
- a. In the Database Firewall “**Detect data exfiltration attempts**” home page, click [**Sets/Profiles**].
- a. Create ‘**IP Address Sets**’ to create allowed IP address connections to the database.
- b. Click [**Add**]
 - Enter Name/Description: **Allowed corporate IP address**
 - In ‘**Sets Values**’ field, enter your allowed IP address values(new line separated /range)
 - Click [**Save**]



- b. Create a “**Session Context Rule**” to block unauthorized IP address connections
 - i. Enter Rule Name: **Block unauthorized IP address**
 - ii. Select ‘IP Address Set’ to exclude ‘**Allowed corporate IP address**’ from the dropdown. Any IP address which is NOT IN < **Allowed corporate IP address** > will be blocked.
 - iii. In the Action region, enter the following and click [**Save**].
Actions: **Block**
Logging Level: **Always**
Threat Severity: **Critical**



- iv. Session context rule appears as shown here



7.3.3. Create a firewall policy rule to monitor and alert on data exfiltration attempts on sensitive table in HCM schema

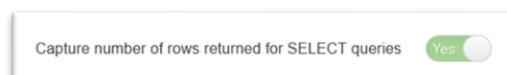
b. Create a **“Database Objects Rule”** to monitor data exfiltration attempts on sensitive table in HCM schema

- Enter Rule Name: **Monitoring sensitive employee data exfiltration attempt**
- Select Statement Classes **‘Data Manipulation Readonly’**
- Toggle Selected tables to **“Any”**.
- Select the following tables in the HCM schema
 - HCM.EMPLOYEES
- In the Action region, enter the following and click **[Save]**.

Actions: **Pass**

Logging Level: **Always**

Threat Severity: **Moderate**
- Toggle the option to capture the return number of rows for SELECT queries to enable Database Firewall to capture the row count for all incoming Select queries on HCM.EMPLOYEES table



vii. The rule details appears as shown here

Database Objects

Rule Name: Monitoring sensitive employee data exfiltration attempt

Description: Monitoring sensitive employee data exfiltration attempt

Statement Classes: Available

Database Tables:

Selected Tables: HCM_EMPLOYEE, HCM_EMP_EXTENDER, HCM_JOB, HCM_JOB_HISTORY, HCM_LOCATION, HCM_SUPPLEMENTAL_DATA, HLS_DATABASE_PARAMETER

Action: Pass

Logging Level: Always

Threat Severity: Moderate

c. Database Object Rules appear as shown here

Rules

Session Context SQL Statement Database Objects Default

Add Delete

Rule Name	Rule Type	Action	Logging Level	Threat Severity	Description
Monitoring sensitive employee data exfiltration attempt	Any	Pass	Always	Moderate	Monitoring sensitive employee data exfiltration attempt

1 - 1

7.3.4. Create a “Default Rule” to allow UNKNOWN SQL traffic with Minor Threat Severity

b. In the Action region, enter the following and click **[Save]**.

Actions: **Pass**

Logging Level: **One-Across-Session**

Threat Severity: **Minimal**

Saved Default Rule appears as shown here

Rules

Session Context SQL Statement Database Objects Default

Rule Name	Action	Logging Level	Threat Severity
Default Rule	Pass	One-Across-Sessions	Minimal

7.3.5. Publish and deploy the user-defined policy

h. Click **[Save]** in the Database Firewall policy home page which publishes the policy also.

- i. Go to “Targets” home page and drilldown into “hr” target
- j. Navigate to the “Database Firewall Monitoring” tab and change the Firewall policy to “Detect data exfiltration attempts”.
- k. Click **checkmark**.
- l. Firewall policy “Detect data exfiltration attempts” is applied !!!

7.3.6. Simulation of network activity on the Firewall

To mimic DBA exfiltration activity over the network and see how it is processed by Firewall:

Run dba_over_network.sql using an SQL connection through the firewall IP address



7.4. Tracking potential data exfiltration attempts over the network using reports

- f. Go to “Reports” home page
- g. Click [Activity Reports] in the left navigation menu
- h. Click [Database Firewall Monitored Activity] report
- i. Select columns-Row count, Policy Name and Log Cause from Actions dropdown
- j. The report should appear as shown below displaying the row count captured for all Select operations on HCM.EMPLOYEES table. Policy name should indicate the firewall policy name “Detect data exfiltration attempts”.

Oracle Audit Vault and Database Firewall 20							
Database Firewall Monitored Activity							
Event Time	Target	User	Command Text	Command Class	Row Count	Policy Name	Log Cause
1/22/2021 2:40:40 PM	hr	"dba_charles@example.com"	select * from HCM.EMPLOYEES where email = 'XXXXXXXXXXXXXXXXXXXX'	SELECT	1	Detect data exfiltration attempts	database objects
1/22/2021 2:40:38 PM	hr	"dba_charles@example.com"	select e * from (select e * , avg(salary) over (partition by department_id as avgsalary from HCM.EMPLOYEES e) e where e.salary < e.avgsalary	SELECT	72	Detect data exfiltration attempts	database objects
1/22/2021 2:40:37 PM	hr	"dba_charles@example.com"	select e * from (select e * , avg(salary) over (partition by department_id as avgsalary from HCM.EMPLOYEES e) e where e.salary < e.avgsalary	SELECT	42	Detect data exfiltration attempts	database objects
1/22/2021 2:40:36 PM	hr	"dba_charles@example.com"	SELECT DEPARTMENT_ID, AVG(SALARY) FROM HCM.EMPLOYEES GROUP BY (DEPARTMENT_ID)	SELECT	12	Detect data exfiltration attempts	database objects
1/22/2021 2:40:35 PM	hr	"dba_charles@example.com"	select employee_id, first_name, last_name, salary, commission_pct, commission_pct(salary(000'000)) from HCM.EMPLOYEES where COMMISSION_PCT is not null and commission_pct(salary(000'000)) < 0	SELECT	10	Detect data exfiltration attempts	database objects
1/22/2021 2:40:34 PM	hr	"dba_charles@example.com"	SELECT e email as EMPLOYEE, e salary as EMP_SALARY, m email as MANAGER, m salary as MGR_SALARY FROM HCM.EMPLOYEES e, HCM.EMPLOYEES m WHERE e.manager_id = m.employee_id(+)	SELECT	117	Detect data exfiltration attempts	database objects
1/22/2021 2:40:34 PM	hr	"dba_charles@example.com"	SELECT e email as EMPLOYEE, e salary as EMP_SALARY, m email as MANAGER, m salary as MGR_SALARY FROM HCM.EMPLOYEES e, HCM.EMPLOYEES m WHERE e.manager_id = m.employee_id(+)	SELECT	117	Detect data exfiltration attempts	database objects
1/22/2021 2:40:32 PM	hr	"dba_charles@example.com"	select e EMPLOYEE_ID, e EMAIL, e SALARY, d DEPARTMENT_NAME from HCM.EMPLOYEES e FULL OUTER JOIN HCM.DEPARTMENTS d ON e.department_id=d.department_id order by d.department_name	SELECT	133	Detect data exfiltration attempts	database objects

8. <OPTIONAL>REDO LOG COLLECTION WITH GOLDEN GATE

Goal: You will configure redo log collection on the pluggable database instance target using GoldenGate microservices and see the data modification changes in the AVDF report

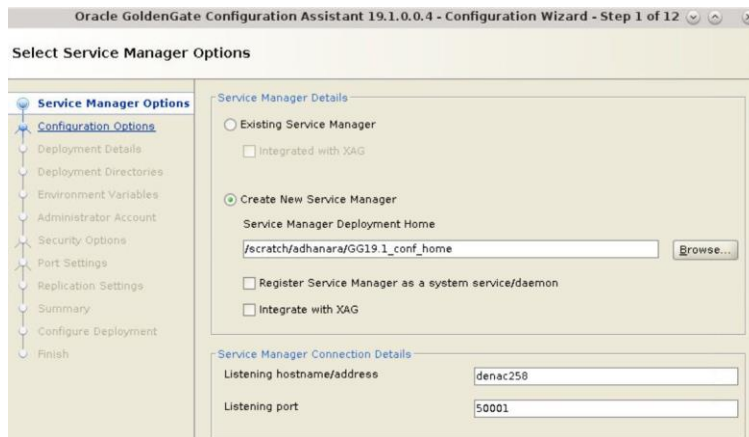
8.1. Installation and Configuration of GoldenGate Microservice

8.1.1. Installation of Oracle GoldenGate 19.1.0.0.4 Microservices

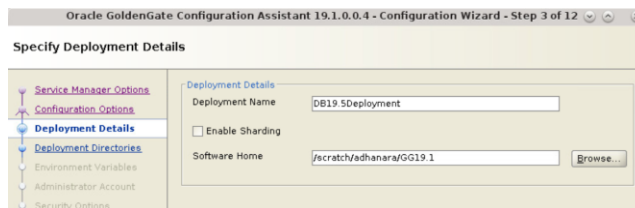
- Download zip from [OTN](#) (min version 19.1.0.0.4)
- Follow the install steps in GG doc- [Section 1.2 Performing an Interactive Installation with OUI](#) to install on the server where database instance is configured
- Unzip and execute fbo_ggs_Linux_x64_services_shophome/Disk1/runInstaller, and give the OGG_HOME (/scratch/adhanara/GG19.1) to install GoldenGate

8.1.2. Configuring service manager, and deployment configuration

- Refer in parallel, the instructions within GG doc - [Section 2.1 How to Add Secure or Non-Secure Deployments](#)
- Execute \$OGG_HOME/bin/oggca.sh to give OGG_CONF_HOME (/scratch/adhanara/GG19.1_conf_home) and port.



- Add a new deployment for the database target



- Select the deployment directory for the target database instance:



- Set the environment variables pointing to pluggable database instance

Oracle GoldenGate Configuration Assistant 19.1.0.0.4 - Configuration Wizard - Step 5 of 12

Specify Environment Variables

Environment Variables

Name	Value
OGG_HOME	/scratch/adhanara/GG19.1
ORACLE_HOME	/scratch/adhanara/db195
LD_LIBRARY_PATH	\${ORACLE_HOME}/lib
TNS_ADMIN	/scratch/adhanara/db195/network/admin
ORACLE_SID	orcl

Add Remove

- f. Create an administrator account for logging into Service Manager

Oracle GoldenGate Configuration Assistant 19.1.0.0.4 - Configuration Wizard - Step 6 of 12

Specify Administrator Account

Administrator account

Username: ggadmin

Password:

Confirm Password:

☒ Enable strong password policy in the new deployment.

- g. Select non-secure deployment options

Oracle GoldenGate Configuration Assistant 19.1.0.0.4 - Configuration Wizard - Step 7 of 12

Specify Security Options

☐ SSL / TLS security

☐ This non-secure deployment will be used to send trail data to a secure deployment

- h. Configure ports which are available for other servers

Oracle GoldenGate Configuration Assistant 19.1.0.0.4 - Configuration Wizard - Step 8 of 12

Specify Port Settings

Service Manager Details

Listening hostname/address: denac258 Listening port: 50001

Servers

☒ Enable Administration Server Administration Server port: 50002

☒ Enable Distribution Server Distribution Server port: 50003

☒ Enable Receiver Server Receiver Server port: 50004

Monitoring

☐ Enable Monitoring ☐ XAG Critical

Metrics Server port: 50005

Metrics Server UDP port (data): 50006

Metrics Server DataStore type: BDB

Metrics Server DataStore home:

- i. Enter replication settings default schema –ggadmin

Oracle GoldenGate Configuration Assistant 19.1.0.0.4 - Configuration Wizard - Step 9 of 12

Specify OGG Replication Settings

Replication Options

Default Schema: ggadmin

- j. Review the summary page and finish deployment
- k. Ensure the following URLs are accessible
- Service Manager URL (<http://localhost:50001>)
 - Administration Server URL (<http://localhost:50002>)

8.2. Preparation of Oracle Database target for transaction log collection

8.2.1. Configure a database user in the target for GoldenGate administration with required privileges

- a. In parallel, refer the instructions within GG docs - [3.1.1.1 Granting the Appropriate User Privileges](#), and [75.3.1 GRANT_ADMIN_PRIVILEGE Procedure](#)
- b. As sysdba in CDB, execute the following to create a new CDB user privileged for GG administration
 1. create user c##gguser identified by c##gguser container=all;
 2. grant create session, resource, alter system to c##gguser container=all;
 3. exec
dbms_goldengate_auth.grant_admin_privilege('c##gguser','*',TRUE,TRUE,NULL,NULL,
NULL,'ALL');
 4. exec dbms_goldengate_auth.grant_admin_privilege('c##gguser',container=>'all');
 5. grant unlimited tablespace to c##gguser;

```
SQL> create user c##gguser identified by c##gguser container=all;
User created.
SQL> grant create session, resource, alter system to c##gguser container=all;
Grant succeeded.
SQL> exec dbms_goldengate_auth.grant_admin_privilege('c##gguser','*',TRUE,TRUE,NULL,NULL,NULL,'ALL');
PL/SQL procedure successfully completed.
SQL> exec dbms_goldengate_auth.grant_admin_privilege('c##gguser',container=>'all');
PL/SQL procedure successfully completed.
SQL> grant unlimited tablespace to c##gguser;
Grant succeeded.
```

8.2.2. Prepare the database for GoldenGate replication

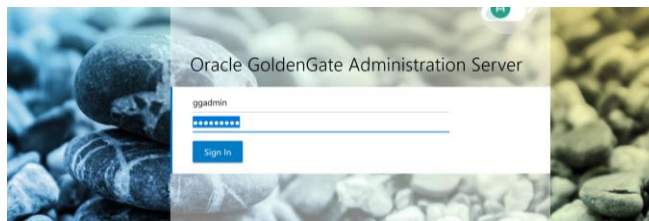
- a. In parallel, refer the instructions within GG docs- [1.1 Preparing the Database](#)
- b. As sysdba in CDB, execute the following to enable GoldenGate replication
 1. alter system set enable_goldengate_replication=true scope=spfile;
 2. shutdown immediate
 3. startup mount
 4. alter database archivelog;
 5. alter database open;
 6. alter pluggable database all open /*Applicable only for multitenant database*/;
 7. select name,log_mode from v\$database;
 8. alter database force logging;
 9. alter database add supplemental log data;
 10. select force_logging, supplemental_log_data_min from v\$database;

11. show parameter compatible; /* If version is prior to 12.2.0.1.0 */ alter system set compatible = '12.2.0.1.0' scope=spfile;
12. alter system set streams_pool_size=1250M scope=spfile; /* Recommended to configure this parameter as per recommendation in [2.5 Managing Server Resources](#) for more than 1 Extract */
13. shutdown immediate;
14. startup;
15. alter pluggable database all open /*Applicable only for multitenant database*/;

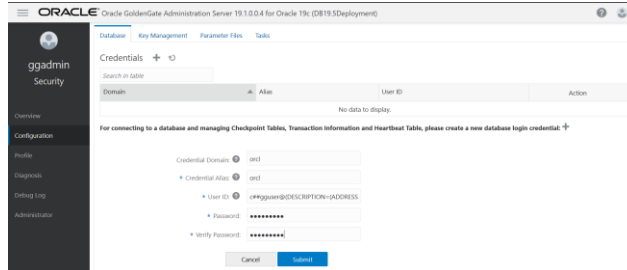
8.3. Configuring Integrated Extract process in GoldenGate

8.3.1. Configuring credentials in GoldenGate Administration server

- a. Log in to the Oracle GoldenGate Administration Server as configured in Step 8.1.2.k



- b. Create a credential with the database user created in step 8.2.1, and CDB service.

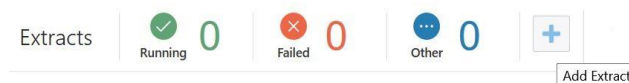


Note: User ID is of the format :

[c##gguser@\(DESCRIPTION=\(ADDRESS=\(PROTOCOL=tcp\)\(HOST=xxxx.us.oracle.com\)\(PORT=1522\)\)\(CONNECT_DATA=\(SERVICE_NAME=orcl.us.oracle.com\)\)\)](#)

8.3.2. Configuring Integrated Extract in GoldenGate Administration server

- a. Add a new extract for the pluggable database instance **hr**



- b. Choose Integrated Extract as Extract type
- c. Configure Extract Options giving the credential domain, trail subdirectory, and PDB.

Basic Information

* Process Name:
 Description:
 Intent:

► Create new credential

Credential Domain:
 * Credential Alias:
 * Begin:
 * Trail Name:
 Trail Subdirectory:
 Trail Size (MB):
 Trail Sequence:
 Trail Offset:
 Remote: ☐

► Encryption Profile

► Managed Options

Registration Information

CSN:
 Share:
 Optimized: ☐
 * Register to PDBs:

Note: Make sure the subdirectory is created under
<deploymentHome>/var/lib/data/orclpdb

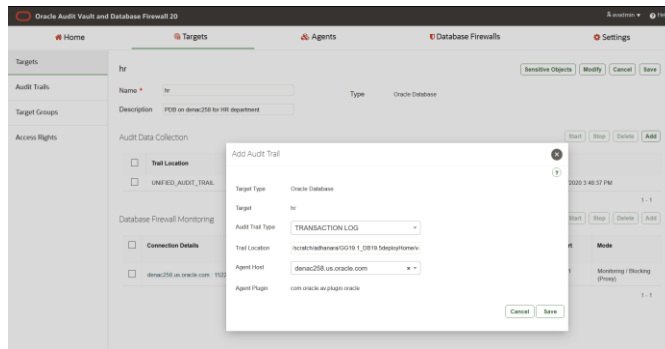
d. Enter the parameter file

Copy the parameter file entry from here:

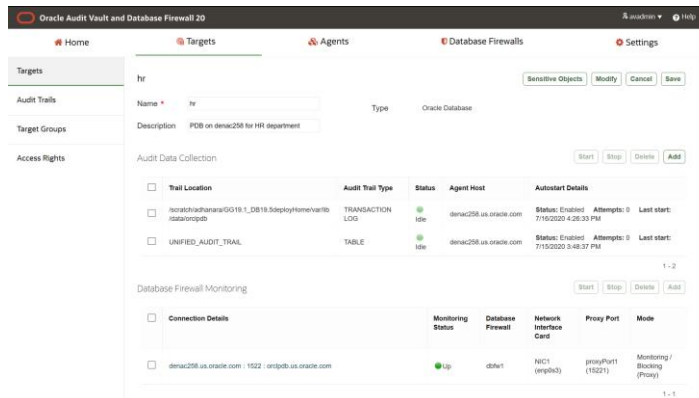
```
extract orclpdb
useridaalias orcl domain orcl
OUTPUTFORMAT XML _AUDIT_VAULT
exttrail orclpdb/or
SOURCECATALOG orclpdb
DDL INCLUDE OBJNAME hcm.*;
TABLE hcm.*;
```

8.4. Configuring transaction log collection in Audit Vault console

- 8.4.1. Configure the collection attribute on the target
 - a. Navigate to target page as AVADMIN
 - b. The collection attribute **av.collector.TimeZoneOffset** needs to be set
 - c. Click **[Modify]** in the home page of the registered target and add the attribute.
 - d. Click **[Save]**.
- 8.4.2. Configure transaction log trail in Audit Vault console
 - a. Navigate to target page as AVADMIN and add transaction log trail
 - b. Give the full path till the subdirectory where trail xml files reside.



- c. The transaction log trail appears as below:



**hcm_workload_script.
sql**

- d. Execute the script against the database target instance
- e. As AVAUDITOR, navigate to **'Reports'** menu and **'Activity Reports'** in the left navigation menu to open the report **Data Modification Before-After Values Report**
- f. Refer to the report values to see the data value changes reported:

Oracle Audit Vault and Database Firewall 20									
Home Targets Policies Alerts Reports Settings									
Activity Reports	Data Modification Before-After Values Report								
Summary Reports									
Compliance Reports									
PDF/XLS Reports									
Saved Reports									
Report Schedules									
Generated Reports									
	Target	User	Event	Object	Data Modification		Event Time	Trail Type	
	hr	hr_lan@example.com	UPDATE	EMP_EXTENDED	Column	Old Value	New Value	7/16/2020 12:57:33 PM	TRANSACTION LOG
					PAYMENTACCOUNTNO	4321123454320000	4321123454320000		
	hr	hr_lan@example.com	UPDATE	SUPPLEMENTAL_DATA	Column	Old Value	New Value	7/16/2020 12:57:33 PM	TRANSACTION LOG
					LAST_HIS_CLAIM	1000	1000		
	hr	hr_arn@example.com	UPDATE	EMPLOYEES	Column	Old Value	New Value	7/16/2020 12:57:16 PM	TRANSACTION LOG
					PHONE_NUMBER	650 670 9800	650 670 9800		
	hr	hr_arn@example.com	UPDATE	REGIONS	Column	Old Value	New Value	7/16/2020 12:57:16 PM	TRANSACTION LOG
					REGION_NAME	Middle East and Africa	EMEA		
	hr	hr_arn@example.com	UPDATE	JOBS	Column	Old Value	New Value	7/16/2020 12:57:16 PM	TRANSACTION LOG
					MN_SALARY	5500	5000		
	hr	hr_arn@example.com	UPDATE	EMP_EXTENDED	Column	Old Value	New Value	7/16/2020 12:57:16 PM	TRANSACTION LOG
					PAYMENTACCOUNTNO	4321123454326181	4321123454320000		
	hr	hr_arn@example.com	UPDATE	EMPLOYEES	Column	Old Value	New Value	7/16/2020 12:57:16 PM	TRANSACTION LOG
					SALARY	3100.00	5000.00		
	hr	HCM	UPDATE	EMPLOYEES	Column	Old Value	New Value	7/16/2020 12:56:13 PM	TRANSACTION LOG
					SALARY	23099.00	14000.00		
	hr	dba_charles@example.com	UPDATE	EMPLOYEES	Column	Old Value	New Value	7/16/2020 12:54:54 PM	TRANSACTION LOG
					SALARY	3000.00	99999.00		

9. <OPTIONAL> NETWORK MONITORING WITH HOST MONITOR

Goal: You will configure customers orders schema in the second pdb instance on the same database server and configure network monitoring of the customer order service with host monitor.

9.1.1. Configure Customer Orders schema in the second PDB instance on the same database server

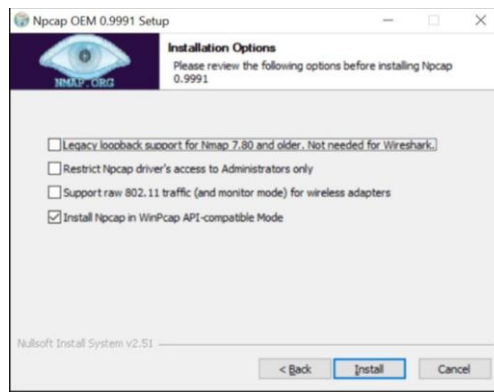
- Go to the GitHub web site: <https://github.com/oracle/db-sample-schemas/releases/tag/v19.2>
- Download the ZIP bundle from GitHub and extract the files. Unzip the file and you will see the folder customer_orders within.
- Follow the instructions in README.txt of customer_orders to create the customer orders schema alone.

9.1.2. Configure host monitor binaries on the database server

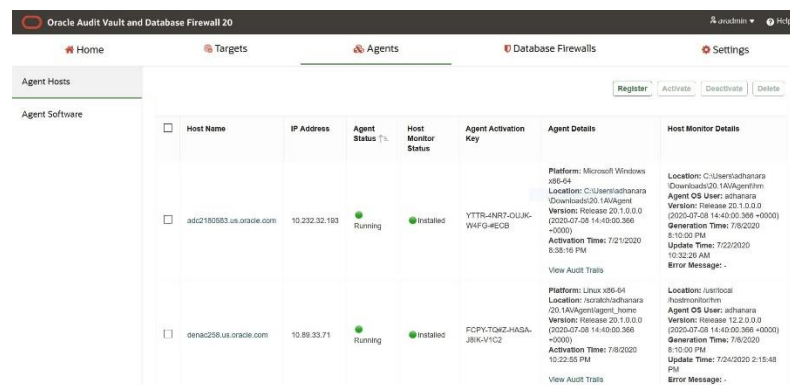
- Navigate to **'Agents'** home page as AVADMIN
- Download the host monitor software corresponding to the OS from **'Agent Software'** menu.

Oracle Audit Vault and Database Firewall 20									
Home Targets Agents Database Firewalls Settings									
Agent Hosts									
Agent Software									
	Platform						Size	Download	
	Agent Release 20.1.0.0.0						34 MB	Download	
	Host Monitor (AIX PPC-64)						35 MB	Download	
	Host Monitor (Linux x86-64)						31 MB	Download	
	Host Monitor (Solaris SPARC 64-bit)						44 MB	Download	
	Host Monitor (Solaris x86-64)						46 MB	Download	

- Ensure the database server has all the pre-requisites outlined in AVDF documentation section [8.2.1 Host Monitor Requirements](#), including the following on Windows OS
 - Npcap installation in WinPcap-API-compatible mode as shown here

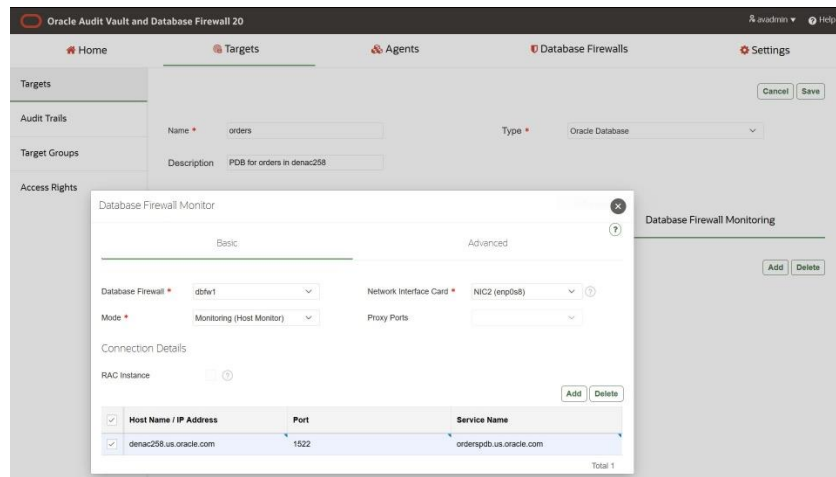


- b. Latest version of OpenSSL (1.1.1g or higher)
- d. Deploy the host monitor software on the server following the instructions in Section [8.2.3 Step 2: Deploy the Audit Vault Agent and Install the Host Monitor](#)
- e. Navigate as AVADMIN to **'Agents'** home page and ensure the host monitor is in **'Installed'** status

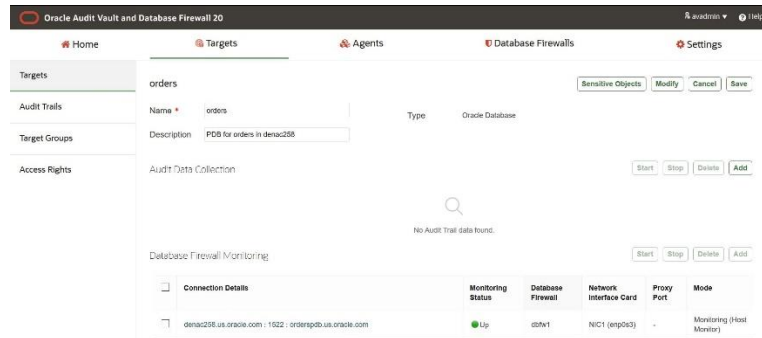


9.1.3. Register the target for network monitoring

- a. Navigate to **'Targets'** home page as AVADMIN,
- b. Click **[Register]** to enter the details of pdb instance with orders schema
- c. Click **[Add]** button in Database Firewall monitoring region.
- d. In the popup,
 - i. Select Database Firewall(**dbfw1**),
 - ii. Traffic Source(**Non-management NIC Interface. Here NIC2**),
Note: Configure the non-management NIC interface if it does not exist already
 - iii. Mode(Monitoring(Host Monitor))
 - iv. Enter the database target identified by the hostname /IP Address , with port and service name in the Target details region
- e. Click **[Save]**

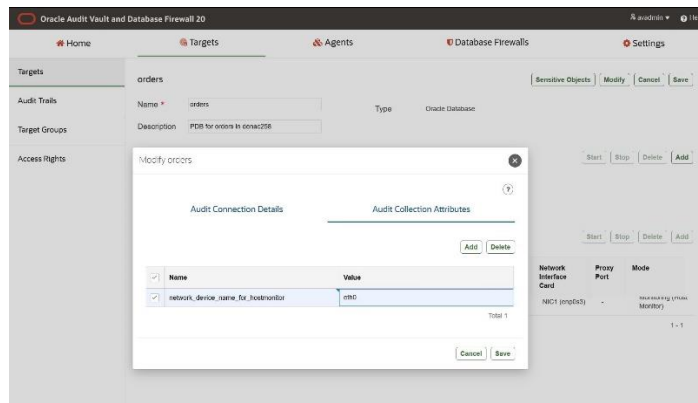


f. Refresh the page and ensure the monitoring point is started as shown here.

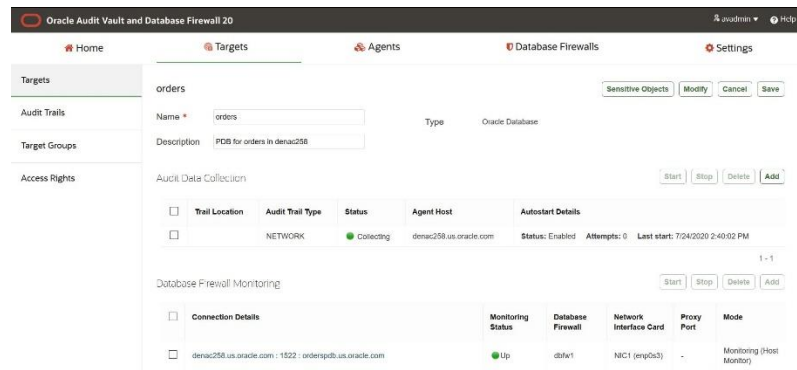


Note: If you are using FQDN name for the target, ensure the DNS system service for Firewall is configured.

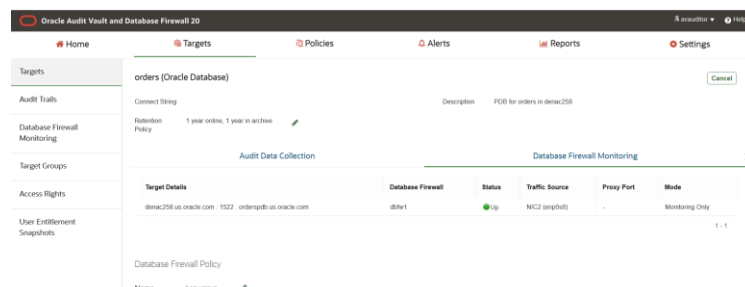
- g. Configure the collection attribute on the target for network monitoring
 1. The collection attribute **network_device_name_for_hostmonitor** ensures reception of the database traffic from the appropriate network interface card of the database server.
 2. Refer to section [8.2.6 Step 5: Create a Network Audit Trail](#) for details on fetching the value from the database server for different operating system.
 3. Click **[Modify]** in the home page of the registered target and add the attribute as shown here.



- 9.1.4. Configure network audit trail
- Click **[Save]**.
 - Click **[Add]** button in Audit Data Collection region.
 - In the popup,
 - Select Audit Trail Type(**NETWORK**),
 - Select Agent Host
 - Click **[Save]**
 - Ensure the status of NETWORK audit trail is in **'Collecting'** status as shown here



- 9.1.5. Simulation of network activity on the Firewall
- Configure pre-defined Database Firewall policy **'Log unique'**
 - Navigate to **'Targets'** page as AVAUDITOR
 - Select the policy **'Log unique'** in Firewall policy dropdown
 - Click the pencil icon to set it against the target as shown here



- Execute the script **sample_queries.sql** in customer_orders folder within unzipped directory against the pdb instance.
- Navigate to **'Reports'** home page, and go to **'Database Firewall Monitored Activity'** to see the SQL traffic as shown here.

Oracle Audit Vault and Database Firewall 20

Home Targets Policies Alerts Reports Settings

Activity Reports

Database Firewall Monitored Activity

Target: 'orders'

Event Time	Target	User	OS User	Client Program	Command Text	Command Class	Policy Name	Action Taken
7/24/2020 5:26:49 PM	orders	SYS	adhamana	SQL Developer	with dates as (select date'#####' + level dt from dual connect by level <= 000), order_totals as (select trunc (o order_datetime) order_date, count (distinct o order_id) number_of_orders, sum (o quantity * oi unit_price) value_of_orders from orders o join order_items oi on o order_id = oi order_id group by trunc (o order_datetime)) select to_char (dt, '#####') sale_date, nvl (number_of_orders, 0) number_of_orders, nvl (value_of_orders, 0) value_of_orders from dates left join order_totals on dt = order_date order by dt	SELECT	Log unique	pass
7/24/2020 5:26:49 PM	orders	SYS	adhamana	SQL Developer	with dates as (select date'#####' + level dt from dual connect by level <= 000), order_totals as (select trunc (o order_datetime) order_date, count (distinct o order_id) number_of_orders, sum (o quantity * oi unit_price) value_of_orders from orders o join order_items oi on o order_id = oi order_id group by trunc (o order_datetime)) select to_char (dt, '#####') sale_date, nvl (number_of_orders, 0) number_of_orders, nvl (value_of_orders, 0) value_of_orders from dates left join order_totals on dt = order_date order by dt	SELECT	Log unique	pass
7/24/2020 5:26:49 PM	orders	SYS	adhamana	SQL Developer	with dates as (select date'#####' + level dt from dual connect by level <= 000), order_totals as (select trunc (o order_datetime) order_date, count (distinct o order_id) number_of_orders, sum (o quantity * oi unit_price) value_of_orders from orders o join order_items oi on o order_id = oi order_id group by trunc (o order_datetime)) select to_char (dt, '#####') sale_date, nvl (number_of_orders, 0) number_of_orders, nvl (value_of_orders, 0) value_of_orders from dates left join order_totals on dt = order_date order by dt	SELECT	Log unique	pass
					with dates as (select date'#####' + level dt from dual connect by level <= 000), order_totals as (select trunc (o order_datetime) order_date, count (distinct o order_id) number_of_orders, sum (o quantity * oi unit_price)			

SUMMARY

In the cookbook, you have learned how to configure targets for database auditing and network monitoring in AVDF20 with ease. You have also learned how to monitor the database activity using the rich reports provided in AVDF20.

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com.
Outside North America, find your local office at oracle.com/contact.



blogs.oracle.com



facebook.com/oracle



twitter.com/oracle

Copyright © 2021, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AVDF20 Cookbook

February, 2021

Angeline Janet Dhanarani, Database Security Product Management

