Omega Core Audit ™

For Oracle Database, with built-in SIEM integration

September, 2020

Access control, audit monitoring and protection in real time

Omega Core Audit

For Oracle Database

Evaluation Guide

3.3.0

www.dataplus-al.com

Copyright © 2007-2020 DATAPLUS. All rights reserved. Omega Core Audit is registered at US Copyrights Office and is protected by US and international copyright laws. Omega Core Audit and the DATAPLUS logo are trademarks of DATAPLUS. All other trademarks are the property of their respective owners.

TABLE OF CONTENTS

1 Introduction

1.1 About this Evaluation Guide

This guide is intended to shortly explain the functionalities of the Omega Core Audit for Oracle, once it has been deployed. This is performed using a database evaluation environment, a set of audit/protection policies created for the evaluation and also a benchmarking application.

Instructions for deploying Omega Core Audit are outlined into the Omega Core Audit Deployment Guide. Common instructions are found into the Omega Core Audit User's Guide.

1.2 Functionalities Evaluated

The Omega Core Auditing top functionalities (and evaluated in this guide) are:

Access Control

Evaluated is the database entrance mandatory access control on all connections to the database. This is tested by the Logon test option of the Benchmark software.

Standard Audit

Evaluated is the auditing of the system for user activity, user statements and operations on objects. This is tested by all testing options.

Real-Time Protection DDL

Evaluated is the Structural Change DDL (Data Definition Language - CREATE, ALTER, GRANT, DROP…) audit and protection. This is tested by the DDL test option.

Real-Time Protection DML

Evaluated is the Information Change DML (Data Manipulation Language - SEL, INS, DEL, UPD) audit and protection. This is tested by the DML test option.

1.3 General Prerequisites

Before the benchmark testing for evaluation, the following requirements must have been met:

- Omega Core Audit must have been installed successfully.
- The Omega Core Audit advised After Install actions must have been committed.
- The database evaluation environment must have been installed.
- The evaluation policies must have been installed.

1.4 Omega Core Audit Evaluation Package

Omega Core Audit comes with a full Evaluation Package which is comprised of:

- A database evaluation environment, sample database users and objects.
- Policies created for the evaluation.
- A benchmark application that generates database activity for the evaluation.

Privilege Requirements!

The evaluation environment and the benchmark policies creation can be best performed with DBA privileges!

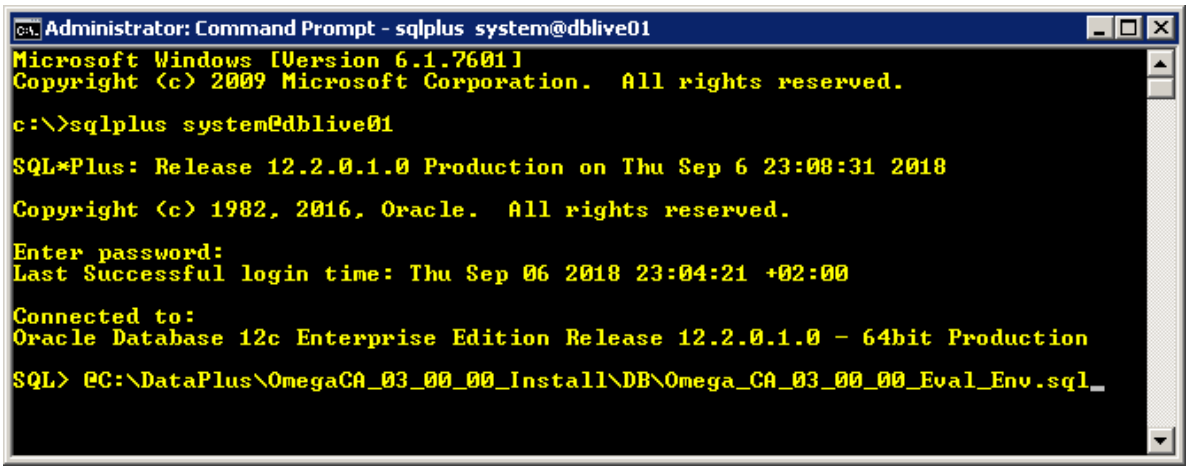
1.4.1 Database Evaluation Environment

The database evaluation environment consists of Oracle database accounts and objects created, simulating in miniature the classical infrastructure: a common Application Schema Owner containing application data, code logic, and working in DB on behalf of all application users; two Developer (privileged) Users and one DBA User.

Install

To install the database evaluation environment, logon in as SQL Command Line prompt execute the script:

Omega_CA_[VS]_[MN]_[PT]_Eval_Env.sql.



To install, pres Enter at the point shown in the figure above!

Check the Omega_CA_[VS]_[MN]_[PT]_Eval_Env.log file, there should be no ORA- errors.

This script will install the following into the database:

OMEGACAEVAPP	Application Schema Owner
OMEGACAEVDEV1	First Developer User
OMEGACAEVDEV2	Second Developer User
OMEGACAEVDBA	DBA User

Objects will be created for OMEGACAEVAPP Application Owner and sample data will be inserted. These objects you will notice on the Simulator Application.

Privileges on the OMEGACAEVAPP's objects are granted to the two Developers, OMEGACAEVDEV1 and OMEGACAEVDEV2. OMEGACAEVDBA is granted the Oracle DBA Role. Other necessary system privileges grants are given to the accounts above to perform the test commands.

Uninstall

Uninstalling the evaluation environment consists in dropping its users and objects. An implicit Uninstall of any currently existing evaluation environment is performed on the Install routine.

To manually drop only the evaluation environment, run the following SQL commands below:

```
drop user OMEGACAEVAPP cascade;
drop user OMEGACAEVDEV1;
drop user OMEGACAEVDEV2;
drop user OMEGACAEVDBA;
```

* The CASCADE option on the drop of the evaluation schema user is used for users that do contain objects; the other evaluation users contain none!

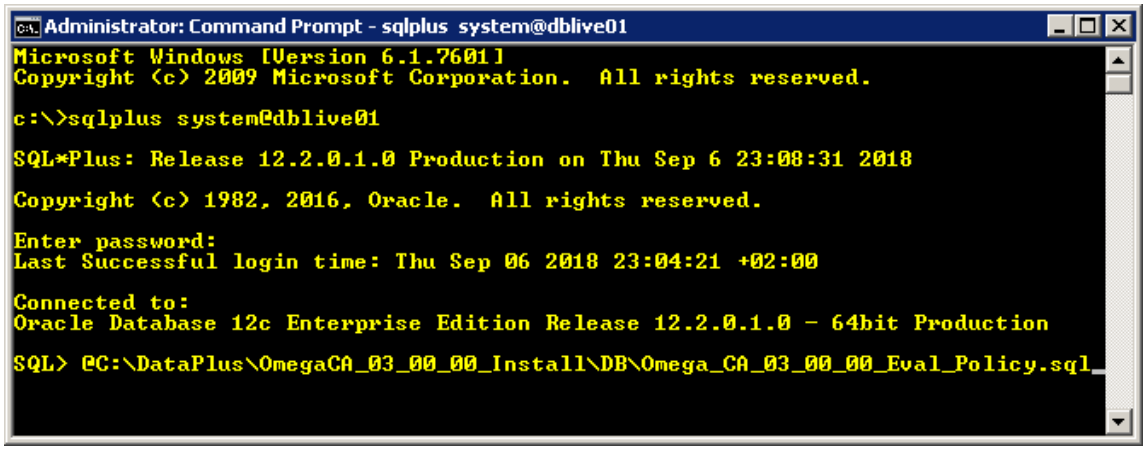
1.4.2 Evaluation Policies

Policies created purposely for the evaluation are made available as an install routine, rather than entering them manually from the Omega Core Audit Application.

Install

To install the evaluation policies into the Omega Core Audit Repository, execute the script:

Omega_CA_[VS]_[MN]_[PT]_Eval_Policy.sql.



To install, pres Enter at the point shown in the figure above!

Check the Omega_CA_[VS]_[MN]_[PT]_Eval_Policy.log file, there should be no ORA- errors.

The evaluation environment install script will create the following evaluation policies into the Omega Core Audit Repository:

Access Control:	"EV Application Access", "EV Developer Access", "EV DBA Access" and "EV Oracle Internals".
Standard Audit:	"EV Developer Audit", "EV App Obj Audit" and "EV DBA Audit".
RTP DDL:	"EV Application DDls", "EV Security Privileges" and "EV Users and Roles"
RTP DML:	"EV Application DMLs".

Important Notes:

1. All evaluation policies are created as Inactive, so are the Rules and Conditions. You will manually activate them during this evaluation guide.
2. All evaluation policies are created with a prefix of "EV" (for evaluation) in their name and description.
3. Where Client Host Factor is used, the provided format-indicating value of <DOMAIN\HOST> should be set as the Benchmark machine name (remove <> brackets). Same for the <DB-SRV>, appearing solely in the policy Access Control "EV Oracle Internals". Real values can be seen in the Security Events trail, field "Userhost"!
4. You can easily convert these evaluation policies to real-life cases.

Uninstall

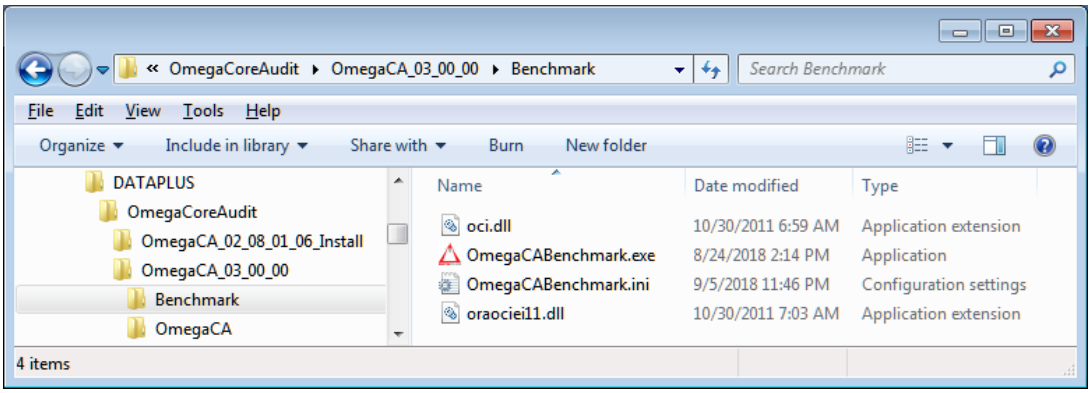
There is no uninstalling routine for the evaluation policies, but you can delete them in the Policy forms of Omega Core Audit's each module.

1.4.3 Omega Core Audit Benchmark

The Omega Core Audit Benchmark has been purposely written for testing and benchmarking the Omega Core Audit solution. It generates database activity related to the evaluation environment and evaluation policies, allowing the testing of the four Audit and Protection modules of the Omega Core Audit.

Deployment

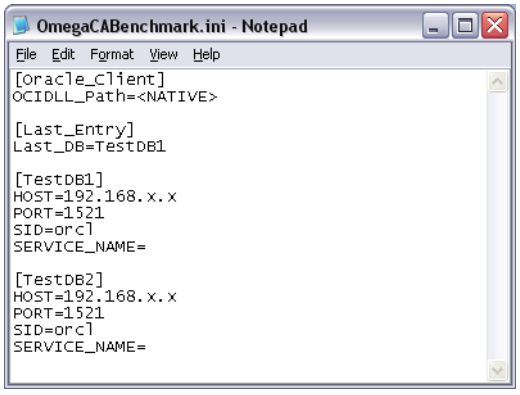
The Omega Core Audit Benchmark software is part of the Omega Core compressed Install package, the software files are found under a single directory named Benchmark. Technology used is the same as that of the Omega Core Audit Application and so is connection to the database server[s]. There is no installation routine in the proper sense, no installer and no registry entries created, just the files you are extracting from the compressed package.



The application files deployed in this version are:

OmegaCABenchmark.exe	The application's executable binary file
OmegaCABenchmark.ini	System parameters initialization file
Oracle DLLs	Oracle 11g R2 Instant Client binaries (oci.dll, oraociei11.dll)

Before you first connect to the target database[s] with the Omega Core Audit Benchmark application, you need to configure the target database[s] section entries in the OmegaCABenchmark.ini initialization parameter file.



OmegaCABenchmark.ini as deployed, comes with two target database connection sections, "TestDB1" and "TestDB2"

For the single parameter under the first section Oracle_Client, the available options are:

- <NATIVE> built-in application connectivity, OCI deployed files, default setting
- <OS> operating system installed and default Oracle client

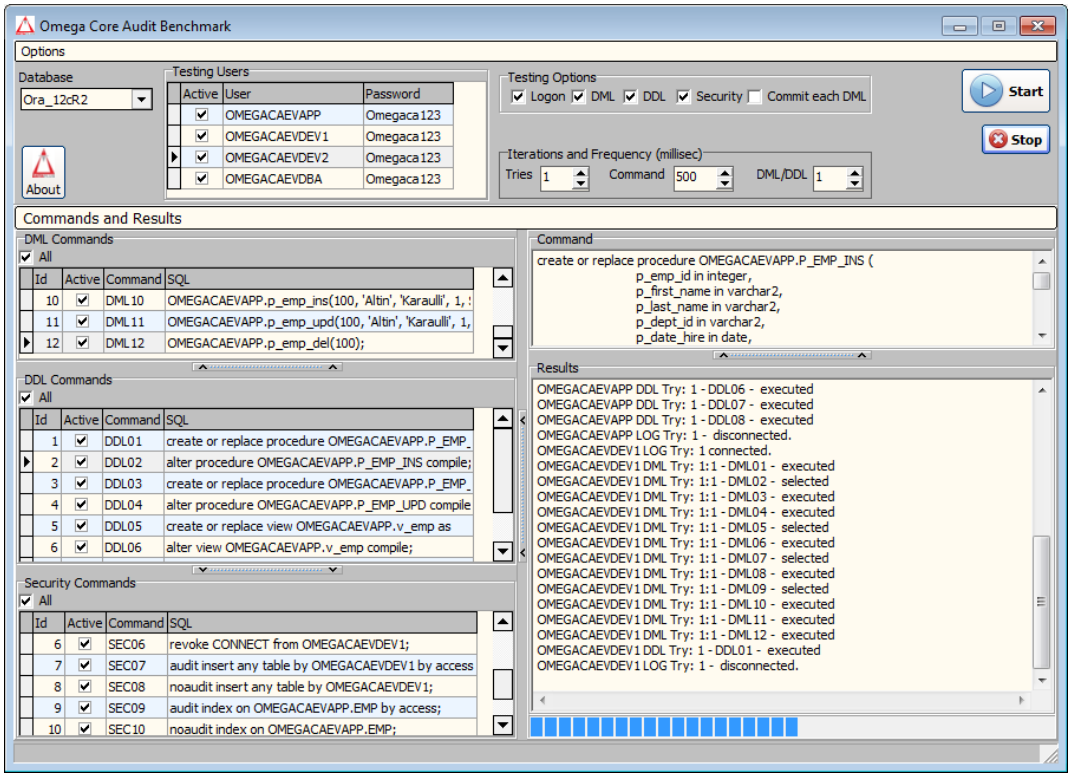
- Manual — manual Oracle client oci.dll path specification, usually for Oracle Instant Client, but others (non-Instant) can be referenced too.

Hint:

In case of a manual choice, set the full path of the Oracle Instant Client, filename included, for example:
C:\oracle\product\11.2.0\client_1\bin\oci.dll

The application

To launch the application, double-click on the OmegaCABenchmark.exe (or its shortcut).



The application's main form contains two panels:

- Options — Benchmark testing options
- Commands and Results — Commands used for testing and result output

The first panel Options features:

Database

The target database on which testing is performed. The Database combo box loads the target database section entries configured in the initialization file OmegaCABenchmark.ini. The last successfully connected is persisted on the next application run.

Testing Users

The four users of the database evaluation environment; least one must be checked (Active) for testing.

Testing Options

This group box contains the following checkboxes optioning for:

Logon	The logon/logoff activity generated for the benchmarking of the Access Control and Standard Audit modules. This is the only mandatory option.
DML	Enables the execution of the selected DML commands for the benchmarking of the Standard Audit and Real-Time Protection DML modules.
DDL	Enables the execution of the selected DDL commands for the benchmarking of the Standard Audit and Real-Time Protection DDL modules.
Security	Enables the execution of the selected security-related commands for the benchmarking of the Standard Audit and Real-Time Protection DDL modules.
Commit Every DML	Enables Commit on every DML command performed.

Iterations and Frequency

This GroupBox contains the following options:

Tries	Number of tries the set of Benchmarking commands (Logon, DML and DDL) are run as a whole for each user.
Command	Pause in milliseconds on each command, default 500, could be higher for real-life conditions, but can be set lower (or 0) for hard-benchmarking!
DML/DDL	The ratio number of times DML Set will be repeated versus the DDL one. Default 1 set (for testing purposes), but to simulate real conditions DML ratio should be much higher *.

* In real-life conditions DDLs do happen much rarely compared to the DMLs!

The second panel Commands and Results features:

DML Commands

The DML Commands grid contains the DML commands used in this Benchmark testing. Commands used are SELECTs, INSERTs, UPDATEs and DELETEs performed on the evaluation environment EMP table, either directly, or through views and stored procedures. Least one must be checked (Active) for testing if DML benchmarking.

DDL Commands

The DDL Commands grid contains the DDL commands used in this Benchmark testing. Commands used are ALTERs and CREATEs performed on the evaluation environment schema procedures, views and functions. Least one must be checked (Active) for testing if DDL benchmarking.

Security Commands

The Security Commands grid contains the security-related commands used in this Benchmark testing. Commands used are GRANTs and REVOKEs performed on the evaluation environment users and objects for System, Object and Role privileges; and also AUDITs and NOAUDITs performed for System and Object privileges. Least one must be checked (Active) for testing if Security benchmarking.

Command

The current SQL benchmark command being executed.

Results

This memo shows the output of all the benchmarked commands.

After the benchmark testing options are set, click on the Start button to initialize the benchmark testing.
Best do not interact with the form while the test is running (especially with any benchmark test option!).
Only exception to the rule above is the stopping the Benchmark test by use of the button Stop.

1.5 Evaluation Reminders

1. The evaluation described later in this guide is cumulative, so while testing the current module, the settings of the previous one might generate Security Events. To simplify testing, when searching Security Events, you can set the Policy Type Search Option field to retrieve only records produced by the module being currently evaluated.

2. The evaluations performed below in this guide are exercised by keeping the default value of Tries = 1 (one try per user)! This parameter logically impacts in proportion the number of Security Events produced - frequently referred to below in this guide.

3. The Benchmark application might "freeze" during testing and long operations, but it will be working instead. Please wait until the operation completes and do not interrupt!

4. For immediate results of Standard Audit and Real-Time Protection DML activity on Security Events trail, you can set a shorter DB Audit Trails Purge interval run, other than the default of 60 seconds; or manually invoke the DB Audit Trails Purge procedure.

2 Evaluating the Access Control

This chapter is a walkthrough on the Access Control module implementation administered through the Omega Core Audit Application and benchmarked via the Omega Core Audit Benchmark software.

2.1 Prerequisites

- The Access Control Module has been activated.
- The Access Control Module remains in Silent mode (default in install).
- The "Privilege encapsulation" is performed on ADMINISTER DATABASE TRIGGER for the testing accounts.

2.2 Benchmark Testing

2.2.1 Initialization

Rationale:
Access Control Module is activated in Silent mode and there is yet no Active Policy.

In the Omega Core Audit Benchmark, Testing Users group, check all the four Users.

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.

In Omega Core Audit, search the Security Events trail. You will see an Access Control trail record (Policy Type equal to Access Control) generated for each testing user. In the Record Details view, check the Trail Evaluation field for each record. It will be empty because there is no policy activated yet. The logons in the Benchmark have been successful only because the whole Access Control Module is in Silent mode (module's specific feature and default on install). Although no policy created yet, it is a specific behavior of the Access Control module that when non-compliant (i.e. no policy evaluated TRUE for whatever reason, even non-existing or all Inactive) a trail record is created.

Note

You should also see Access Control trail records for SYS (and optionally SYSMAN and DBSNMP). These are Oracle own schemas, coming from the local host, usually performed through jobs. They will be treated later in this chapter, topic "Access Control for Oracle Connections"

2.2.2 Access Control for Application Schema Owner

Rationale:
The "EV Application Access" policy establishes a secure logon channel for the Application Schema Owner connections. Policy is created with User Appliance of type "Users Apply" for user OMEGACAEVAPP only.

In Omega Core Audit open the Access Control policy "EV Application Access". Open its only rule "App Owner Access". Activate the two Operand Conditions of Session User and Client Host. For the later set the right "Operand 1" value instead of the provided <DOMAIN\HOST>. Activate the rule "App Owner Access". Activate the policy "EV Application Access".

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.

In Omega Core Audit, search the Security Events trail. You will see an Access Control trail record generated for each testing user. In the Record Details view, check the Trail Evaluation field for each record. It will have a value only for the trail record of the OMEGACAEVAPP user and empty for the other three. The policy name will be "EV Application Access". The policy Result will be TRUE.

2.2.3 Access Control for Developers

Rationale:
The "EV Developer Access" policy establishes a secure logon channel for the two Developer Users. Policy is created with User Appliance of type "Users Apply" for users OMEGACAEVDEV1 and OMEGACAEVDEV2.

In Omega Core Audit open the Access Control policy "EV Developer Access". Open its rule "Developer 01 Access". Activate the two Operand Conditions of Session User and Client Host. For the later set the right "Operand 1" value instead of the provided <DOMAIN\HOST>. Activate the rule "Developer 01 Access". Activate the policy "EV Developer Access".

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.

In Omega Core Audit, search the Security Events trail. You will see an Access Control trail record generated for each testing user. In the Record Details view, check the Trail Evaluation field for each record. It will have a value even for the trail record of the OMEGACAEVDEV1 user and empty for the other two. The policy name will be "EV Developer Access". The policy Result will be TRUE.

Open the rule "Developer 02 Access". Activate the two Operand Conditions of Session User and Client Host. For the later set the right "Operand 1" value instead of the provided <DOMAIN\HOST>. Activate the rule "Developer 02 Access".

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.

In Omega Core Audit, search the Security Events trail. You will see an Access Control trail record generated for each testing user. In the Record Details view, check the Trail Evaluation field for each record. It will have a value even for trail record of the OMEGACAEVDEV2 user and empty for the last OMEGACAEVDBA. The policy name will be "EV Developer Access". The policy Result will be TRUE.

2.2.4 Access Control for DBA

Rationale:
The "EV DBA Access" policy establishes a secure logon channel for the DBA User OMEGACAEVDBA. Policy is created with User Appliance of type "Users Apply" for user OMEGACAEVDBA only.

In Omega Core Audit open the Access Control policy "EV DBA Access". Open its rule "DBA Access". Activate the two Operand Conditions of Session User and Client Host. For the later set the right "Operand 1" value instead of the provided <DOMAIN\HOST>. Activate the rule "DBA Access". Activate the policy "EV DBA Access".

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.

In Omega Core Audit, search the Security Events trail. You will see an Access Control trail record generated for each testing user. In the Record Details view, check the Trail Evaluation field for each record. It will have a value even for the trail record of the OMEGACAEVDBA. The policy name will be "EV DBA Access". The policy Result will be TRUE.

2.2.5 Access Control for Oracle Connections

Rationale:
The "EV Oracle Internals" policy establishes a secure logon channel for the Oracle own schemas SYS, SYSMAN and DBSNMP. Policy is created with User Appliance of type "Users Apply" for users SYS, SYSMAN and DBSNMP.

In Omega Core Audit open the Access Control policy "EV Oracle Internals". Open its only rule "Oracle Connections". Activate the only Operand condition Client Host, set the latter's right "Operand 1" value instead of the provided <DB-SRV>. Activate the rule "Oracle Connections". Activate the policy "EV Oracle Internals".

This is not tested with the Benchmark software!
At this point, to test the policy above you can:

1. Wait for some Oracle SYS/SYSMAN/DBSNMP run jobs to execute.
2. Simulate yourself by manually opening a SYS/SYSMAN session from a SQL terminal, or best a SYS session as SYSDBA (OS authentication) on the DB server.

In Omega Core Audit, search the Security Events trail for Username equal to SYS only. You will see that in the Access Control Trail records generated by SYS, Record Details view, the Trail Evaluation field will have a value and is not empty as before this policy appliance. The policy name will be "EV Oracle Internals". The policy Result will be TRUE.

2.2.6 Tuning the Access Control

Rationale:
Optimize Access Control for auditing knowledgeably and performance.

Having all the Access Control policies above with an Audit Option of "On Success/Failure" might not be a good idea for the Application Access policy. For example, behind the Application Schema Owner Oracle account may be hundreds of application users and new sessions might be (application's developer choice) created at every form open, or refresh, in case of web-design. Having the policy "EV Application Access" on such a configuration could generate hundreds if not thousands of repetitive log entries per minute, so it would be logically to have the trail log only when the policy is not satisfied.

On the Access Control policy "EV Application Access", set the Audit Option to "On Failure".

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.

In Omega Core Audit, search the Security Events trail. You will notice that Access Control trail records will be created for all users, except OMEGACAEVAPP. This we shall avoid the thousands correct logons of the Application Schema Owner, but have all the violations of it!

Note:

Using the "Users Apply" option of the Access Control policies User Appliance setting is a recommended way of reducing even the scope of the policies evaluation; this is further reinforced by the "mandatory" specific of the Access Control. Its counterpart "Users Exclude" would achieve the same goals.

In both cases, and whenever used (also in RTP DML and RTP DDL), always remember to declare the user[s] into the respective Users Apply/Exclude Lists.

2.2.7 Enforcing the Access Control

Rationale:
Enforce Access Control on the database by rejecting non-complying logons.

Until now we have used the Access Control module in its default Silent Mode. This means the logon actions were allowed to continue whatever the compliance evaluation (Policy Result TRUE/FALSE) was.

To test the protective capabilities of the Access Control module, into the Omega Core Audit software, System Components form, Access Control tab, uncheck the Silent Checkbox and press the Set button to set the changed value. Also change the "Developer 02 Access" rule of the "Developer Access" and wrongly set the Client Host Operand Condition's value, so that rule evaluates to FALSE.

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.
You will notice the logon failure "ORA-20010: Access Control Policy non-compliance!" only for the OMEGACAEVDEV2 user. The OMEGACAEVDEV2 logon action has been rejected!

In Omega Core Audit, search the Security Events trail. You will notice the same error code and message in the same Security Events trail fields Return Code and Return Message for the OMEGACAEVDEV2 user. The Trail Evaluation field will be "EV Developer Access". The policy Result will be FALSE.

Note

If you will continue the testing with Silent Mode unset, make sure to have rightly ensured the logon of the four testing users, so that they continue perform other module's commands.

The same is required for the OMEGACAADM (Omega Core Audit pre-configured Administrator) user, needed for operating on the Omega Core Audit Application.

3 Evaluating the Standard Audit

This chapter is a walkthrough on the Standard Audit module implementation administered through the Omega Core Audit Application and benchmarked via the Omega Core Audit Benchmark software.

3.1 Prerequisites

- The database parameter AUDIT_TRAIL has been set to DB_EXTENDED and DB restarted (change effective).
- The Standard Audit module's Map option must have been set. Check in System Components form, tab DB Audit Trails Purge, Group "Std. Aud Map", checkbox Map must be checked. This is not the default setting after the Install!
- The DB Audit Trails Purge job has been activated and the run interval has been set to satisfy the testing needs (can set it less then the default 60 seconds).
- Drop any Oracle Default user-wide audit settings, as advised in Deployment Guide, Appendix Utility Scripts, Topic Oracle Statement and Objects Audits, part ORACLE STATEMENT AUDITS for user-wide Statement Audits. You can re-enable them later through the "Oracle Default Audits" policy.

3.2 Benchmark Testing

3.2.1 Initialization

Rationale:
There is yet no Active Policy. The install provided "Oracle Default Audits" policy is Inactive.

In the Omega Core Audit Benchmark, Testing Options group, check only the DML Checkbox. Check all DML commands in the grid (DML01-DML12). Check all the four Users in the grid.

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.
In the Omega Core Audit manually invoke the DB Audit Trails Purge procedure.

In Omega Core Audit, search the Security Events trail. There will be no Standard Audit records (Policy Type equal to Standard Audit), because no policy's rule has been activated yet, either on user statements or objects.

3.2.2 Standard Audit for Developer Users

Rationale:
The "EV Developer Audit" policy audits statements and system privileges for the Developer Users. Policy is created with a Policy Type of "Statement [Priv.]".

In Omega Core Audit open the Standard Audit policy "EV Developer Audit". Activate the first four rules, namely INSERT/SELECT/DELETE/UPDATE TABLE for the first Developer OMEGACAEVDEV1. Activate the policy "EV Developer Audit".

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.
Wait for the next run of the DB Audit Trails Purge, or invoke it manually.

In Omega Core Audit, search the Security Events trail. Standard Audit trails have been created only for the OMEGACAEVDEV1 user. In the Record Details view, check the Trail Evaluation field. The policy name will be "EV Developer Audit". The policy Result will be TRUE.

Activate the next four rules, namely INSERT/SELECT/DELETE/UPDATE TABLE for the second Developer OMEGACAEVDEV2.

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.
Wait for the next run of the DB Audit Trails Purge, or invoke it manually.

In Omega Core Audit, search the Security Events trail. Standard Audit trails have been created even for the OMEGACAEVDEV2. In the Record Details view, check the Trail Evaluation field. The policy name will be "EV Developer Audit". The policy Result will be TRUE.

3.2.3 Standard Audit for Application Objects

Rationale:
The "EV App Obj Audit" policy audits operations on application schema owner objects. Policy is created with a Policy Type of "Object".

In Omega Core Audit open the Standard Audit policy "EV App Obj Audit". Activate its two rules, namely the DELETE/UPDATE on the EMP table owned by OMEGACAEVAPP. Activate the policy "EV App Obj Audit".

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.
Wait for the next run of the DB Audit Trails Purge, or invoke it manually.

In Omega Core Audit, search the Security Events trail. Standard Audit trails have been created even for the OMEGACAEVAPP and OMEGACAEVDBA users, but only for their DELETEs and UPDATEs on EMP. In the Record Details view, check the Trail Evaluation field. The policy name will be "EV App Obj Audit". The policy Result will be TRUE.

3.2.4 Standard Audit for DBA Users

Rationale:
The "EV DBA Audit" policy audits statements and system privileges for the DBA User. Policy is created with a Policy Type of "Statement [Priv.]".

In Omega Core Audit open the Standard Audit policy "EV DBA Audit". Activate the first three rules, namely the INSERT ANY TABLE / DELETE ANY TABLE / UPDATE ANY TABLE for the OMEGACAEVDBA user. Activate the policy "EV DBA Audit".

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.
Wait for the next run of the DB Audit Trails Purge, or invoke it manually.

In Omega Core Audit, search the Security Events trail. Standard Audit trails have been created even for INSERT on EMP for the OMEGACAEVDBA user. In the Record Details view, check the Trail Evaluation field. The policy name will be "EV DBA Audit". The policy Result will be TRUE.

In the following we will evaluate the Standard Audit of two important security related DDLs, namely the Grant and Revoke commands for managing system privileges, roles and object privileges.

In the Omega Core Audit Benchmark, Testing Options group, uncheck the DML Checkbox and check only the Security Checkbox. In the Security Commands grid check the first six (SEC01-SEC06) Grant/Revoke commands. In the Users grid check only the OMEGACAEVDBA user.

In Omega Core Audit activate the next three rules, namely the SYSTEM GRANT, GRANT ANY OBJECT PRIVILEGE and GRANT ANY ROLE for the OMEGACAEVDBA user.

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.
Wait for the next run of the DB Audit Trails Purge, or invoke it manually.

In Omega Core Audit, search the Security Events trail. Standard Audit trails have been created even for the Grant and Revoke commands performed by the OMEGACAEVDBA user. In the Record Details view, check the Trail Evaluation field. The policy name will be "EV DBA Audit". The policy Result will be TRUE.

Next we will evaluate the Standard Audit of two other important security related DDLs, namely the Audit and NoAudit commands for managing audit settings on user statements and object privileges.

In the Security Commands grid uncheck the first six (SEC01-SEC06) Grant/Revoke commands and check the next four (SEC07-SEC10) Audit/NoAudit commands.

In Omega Core Audit activate the next two rules, namely the SYSTEM AUDIT and AUDIT ANY for the OMEGACAEVDBA user.

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.
Wait for the next run of the DB Audit Trails Purge, or invoke it manually.

In Omega Core Audit, search the Security Events trail. Standard Audit trails have been created even for the Audit and NoAudit commands performed by the OMEGACAEVDBA user. In the Record Details view, check the Trail Evaluation field. The policy name will be "EV DBA Audit". The policy Result will be TRUE.

3.2.5 Tuning the Standard Audit

Rationale:
Optimize Standard Audit for auditing knowledgeably and performance.

As you have noticed, there is no Audit Option configuration into the Standard Audit policy. Meanwhile policies are of two types, according to Oracle user statements or object they can audit with their rules.

Actions performed on the database by Developers and DBA Users are expected to be infrequent and insignificant comparing to the whole system work. But these actions are exercised by privileged users and as such are worthy for audit. User-statement audits are activated on such type of accounts and Standard Audit policies of type Statement [Priv.] do implement them.

For example, auditing the OMEGACAEVDEV1 Developer user for DELETE TABLE will audit him not only for delete of OMEGACAEVAPP.EMP, but even for OMEGACAEVAPP.DEPT or any other schema/table. Users granted privilege of kind ANY (DELETE ANY TABLE) should be audited for ANY statements, like our DBA user OMEGACAEVDBA.

As for auditing actions launched by the Application Owner itself, using exactly the method described so far would not be a good idea, considering the case that behind the Application Schema Owner Oracle account may be hundreds of application users and new sessions might be (application's developer choice) created at every form open, or refresh, in case of web-design. Such a configuration could generate hundreds if not thousands of repetitive log entries per minute.

So it would be logically to implement Standard Audit policies of type "Object" only on the objects of importance, and for specific actions performed, as we have done with the policy ''Application Objects", on the application owner OMEGACAEVAPP table's EMP. Here the audit would be very reasonable (and set) for DELETE and UPDATE commands - usually (in modern transaction based and data-warehouse systems) expected to be few, or even not at all issued from the Application Schema Owner, and as such of much interest. While not set for the SELECT and INSERT - for which a high volume is generally expected from the Application Schema Owner.

Hint:

The usage of SELECT auditing on the DBA has been refrained in the related Standard Audit policy for the same reason!

ALL STATEMENTS auditing

Do not stop your testing here!

Important developments in Oracle security in 11g R2 have introduced the new standard audit shortcut ALL STATEMENTS, which is further supported by other new security developments in Oracle 12C R2 regarding the new audit rail field CURRENT_USER. The first enables Top-Level SQL auditing, the later identification of the same in Oracle dictionary tables/views for Standard Audit and FGA audit trails.

Modify the standard audit policy "EV DBA Audit" to make use of the ALL STATEMENTS. Disable the first three rules INSERT ANY TABLE, DELETE ANY TABLE and UPDATE ANY TABLE for the OMEGACAEVDBA user. Add new rules for ALL STATEMENTS and CREATE SESSION for the OMEGACAEVDBA user.

In the Omega Core Audit Benchmark, Testing Options group, uncheck the Security Checkbox and check only the DML Checkbox. Check only the last three (DML10-DML12) commands *. In the Users grid check the OMEGACAEVDEV2 and OMEGACAEVDBA users.

* These are an INSERT, UPDATE and DELETE performed on the application schema owner's EMP table, not directly, but via the execution of three respective stored procedures, part of the same schema. What is inside an executed procedure is not a Top-Level SQL!

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.

Wait for the next run of the DB Audit Trails Purge, or invoke it manually.

In Omega Core Audit, search the Security Events trail. Standard Audit trails have been created for both users.

Username	Pol. Type	Action Name	Owner	Object Name	Current User
OMEGACAEVDEV2	AUD	INSERT	OMEGACAEVAPP	EMP	OMEGACAEVAPP
OMEGACAEVDEV2	AUD	UPDATE	OMEGACAEVAPP	EMP	OMEGACAEVAPP
OMEGACAEVDEV2	AUD	DELETE	OMEGACAEVAPP	EMP	OMEGACAEVAPP
OMEGACAEVDBA	AUD	LOGON			OMEGACAEVDBA
OMEGACAEVDBA	AUD	PL/SQL EXECUTE			OMEGACAEVDBA
OMEGACAEVDBA	AUD	COMMIT			OMEGACAEVDBA
OMEGACAEVDBA	AUD	UPDATE	OMEGACAEVAPP	EMP	OMEGACAEVAPP
OMEGACAEVDBA	AUD	PL/SQL EXECUTE			OMEGACAEVDBA
OMEGACAEVDBA	AUD	COMMIT			OMEGACAEVDBA
OMEGACAEVDBA	AUD	DELETE	OMEGACAEVAPP	EMP	OMEGACAEVAPP
OMEGACAEVDBA	AUD	PL/SQL EXECUTE			OMEGACAEVDBA
OMEGACAEVDBA	AUD	COMMIT			OMEGACAEVDBA
OMEGACAEVDBA	AUD	LOGOFF			OMEGACAEVDBA

For the Developer user OMEGACAEVDEV2 the result is as expected (as in the prior related topic) - we have three records with actions of INSERT, UPDATE and DELETE, all on object EMP and owner OMEGACAEVAPP. Check the field Trail Evaluation and note the INSERT audited by the "EV Developer Audit" policy, while the two others (UPDATE and DELETE) from the same, plus the "EV App Obj Audit", that monitors for update and delete on EMP.

The result is different for the OMEGACAEVDBA compared to the previous audit settings! The three audit records with action name of PL/SQL EXECUTE belong to the three stored procedures executed and audited by the ALL STATEMENTS rule in the quality of Top-Level SQL statements. In the Record Details view, check the Trail Evaluation field. The policy name will be "EV DBA Audit". The policy Result will be TRUE.

The two records with actions UPDATE and DELETE accounted to user OMEGACAEVDBA are there because of the standard audit policy "EV App Obj Audit", which monitors any same actions on the table EMP; the INSERT meanwhile is missing (as expected) because as the two others above was not performed as a Top-Level SQL (but inside the executed stored procedure)!

Notice the Transaction-Control COMMIT, audited (exclusively) by the ALL STATEMENTS option!

For Oracle database versions 12c R2 and above, the field "Current User" indicates the effective user performing the statement. Notice the difference with the existing field Username, highlighting the Top-Level (vs not) SQL executions!

4 Evaluating the Real-Time Protection DDL

This chapter is a walkthrough on Real-Time Protection DDL implementation administered through the Omega Core Audit Application and benchmarked via the Omega Core Audit Benchmark software.

4.1 Prerequisites

- The Real-Time Protection DDL Module has been activated

4.2 Benchmark Testing

4.2.1 Initialization

Rationale:
Real-Time Protection DDL Module is activated and there is yet no Active Policy.

In the Omega Core Audit Benchmark, Testing Options group, check only the DDL Checkbox. Check all the DDL commands in the grid (DDL01-DDL08). Check all the four Users in the grid.

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.

In Omega Core Audit, search the Security Events trail. There will be no Real-Time Protection DDL records (Policy Type equal to RTP-DDL), because no policy has been activated yet.

4.2.2 Real-Time Protection DDL for Application Objects

Rationale:
The "EV Application DDls" policy establishes a secure channel for DDLs performed by Developers on the Application Owner schema objects. Policy is created with User Appliance of type "All Users".

In Omega Core Audit open the Real-Time Protection DDL policy "EV Application DDls". Open its rule "Developer 01 DDL". Activate the two Operand Conditions of Session User and Client Host. For the later set the right "Operand 1" value instead of the provided <DOMAIN\HOST>. Activate the rule "Developer 01 DDL". Activate the policy "EV Application DDls".

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.

In Omega Core Audit, search the Security Events trail. You will see Real-Time Protection DDL trail records generated for each testing user. In the Record Details view, check the Trail Evaluation field for each record. The policy name will be "EV Application DDls". The policy Result will be TRUE only for the OMEGACAEVDEV1 user and FALSE for all three other users.

Open the rule "Developer 02 DDL". Activate the two Operand Conditions of Session User and Client Host. For the later set the right "Operand 1" value instead of the provided <DOMAIN\HOST>. Activate the rule "Developer 02 DDL".

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.

In Omega Core Audit, search the Security Events trail. You will see Real-Time Protection DDL trail records generated for each testing user. In the Record Details view, check the Trail Evaluation field for each record. The policy name will be "EV Application DDls". The policy Result will be TRUE even for the OMEGACAEVDEV2.

At this stage, only the OMEGACAEVAPP (Schema Owner Itself) and OMEGACAEVDBA's (the DBA) records show evaluate FALSE of the policy, which means are not authorized to perform DDLs on the Application Objects.

4.2.3 Real-Time Protection DDL for Security and Audit

Rationale:
The "EV Security and Audit" policy establishes a secure channel for the Grant and Revoke Security (DDL) commands performed on the database. Policy is created with User Appliance of type "All Users".

In the Omega Core Audit Benchmark, Testing Options group, uncheck the DDL Checkbox and check only the Security Checkbox. Check all the Security Commands - except for the last one (SEC11). In the Users grid check only the OMEGACAEVDBA user.

In Omega Core Audit open the Real-Time Protection DDL policy "EV Security and Audit". Open its rule "DBA Security". Activate the two Operand Conditions of Session User and Client Host. For the later set the right "Operand 1" value instead of the provided <DOMAIN\HOST>. Activate the rule "DBA Security". Activate the policy "EV Security Privileges".

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.

In Omega Core Audit, search the Security Events trail. You will see Real-Time Protection DDL trail records generated for the OMEGACAEVDBA user. In the Record Details view, check the Trail Evaluation field for each record. The policy name will be "EV Security and Audit". The policy Result will be TRUE.

4.2.4 Real-Time Protection DDL for Users and Roles

Rationale:
The "EV Users and Roles" policy establishes a secure channel for any action performed on database Users and Roles. Policy is created with User Appliance of type "All Users".

In Omega Core Audit Benchmark uncheck all the Security Commands and check only the last one (SEC11).

In Omega Core Audit open the Real-Time Protection DDL policy "EV Users and Roles". Open its rule "DBA Security". Activate the two Operand Conditions of Session User and Client Host. For the later set the right "Operand 1" value instead of the provided <DOMAIN\HOST>. Activate the rule "DBA Security". Activate the policy "EV Users and Roles".

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.

In Omega Core Audit, search the Security Events trail. You will see Real-Time Protection DDL trail records generated for the OMEGACAEVDBA user. In the Record Details view, check the Trail Evaluation field for each record. The policy name will be "EV Users and Roles". The policy Result will be TRUE.

4.2.5 Tuning the Real-Time Protection DDL

Rationale:
Optimize Real-Time Protection DDL for auditing knowledgeably and performance.

The Real-Time Protection DDL generates insignificant load to the system, considering that DDL (Data definition language) commands are very rare comparing to the rest of system actions. Due to the importance of the subject, such as structural change is, detailed audit is advised, like existing DDL Body and SQL given.

Although full DDL audit can be performed, it is recommended to narrow the scope of the auditing to the areas of interest. This is approached by featuring of the Secured Areas of the Real-Time Protection DDL policies, using the DDLs action, object owner, name and type as factors in designing an area best narrowed to your needs.

4.2.6 Enforcing the Real-Time Protection DDL

Rationale:
Enforce Real-Time Protection DDL on the database by rejecting non-complying DDLs.

Until now we have used the Real-Time Protection DDL Policies in their default Silent Deny Mode. This means the DDL actions were allowed to continue whatever the compliance evaluation was.

To test the protective capabilities of the Real-Time Protection module, in the Omega Core Audit software, open the Real-Time Protection DDL policy "EV Application DDls". Uncheck the Silent Deny Checkbox and press the Save button to save the policy.

In the Omega Core Audit Benchmark, Testing Options group, uncheck the Security Checkbox and re-check only the DDL Checkbox. Check all the DDL Commands. Check all the four users.

Remember that in the RTP DDL Policy "EV Application DDls", we have authorized only the two Developers to perform DDLs on the Application Schema Owner, through two respective RTP DDL rules.

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.
You will notice DDL failure "ORA-20010: RTP DDL Policy: EV Application DDls violation!" for the OMEGACAEVAPP and OMEGACAEVDBA users. The Application Owner's itself and the DBAs DDL actions have been rejected.

In Omega Core Audit, search the Security Events trail. You will see Real-Time Protection DDL trail records generated for the OMEGACAEVDBA user. In the Record Details view, check the Trail Evaluation field for each record. The policy name will be "EV Application DDls". The policy Result will be TRUE for OMEGACAEVDEV1 and OMEGACAEVDEV2 and FALSE for OMEGACAEVAPP and OMEGACAEVDBA.

You are invited to test the same (disable Silent Deny) with the other policies.

5 Evaluating the Real-Time Protection DML

This chapter is a walkthrough on Real-Time Protection DML implementation administered through the Omega Core Audit Application and benchmarked via the Omega Core Audit Benchmark software.

5.1 Prerequisites

- The Real-Time Protection DML module's Map option must have been set. Check in System Components form, tab DB Audit Trails Purge, Group "RTP DML Map", checkbox Map must be checked. This is not the default setting after the Install! On the same Group "RTP DML Map", check also that "Rtn." checkbox is checked, this is default in install.
- The DB Audit Trails Purge job has been activated and the run interval has been set to satisfy the testing needs (can set it less then the default 60 seconds).

5.2 Benchmark Testing

5.2.1 Initialization

Rationale:
There is yet no Active Policy.

In the Omega Core Audit Benchmark software, Testing Options group, check only the DML Checkbox. Check all the DML commands in the grid. Check all the four Users in the grid.

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.
Wait for the next run of the DB Audit Trails Purge, or invoke it manually.

In Omega Core Audit, search the Security Events trail. There will be no Real-Time Protection DML records (Policy Type equal to RTP-DML), because no policy's rule has been activated yet.

5.2.2 Real-Time Protection DML for Application Objects

Rationale:
The "EV Application DMLs" policy establishes a secure channel for DMLs performed on the Application Owner's table EMP. Policy is created with User Appliance of type "All Users".

In Omega Core Audit open the Real-Time Protection DML policy "EV Application DMLs". Open its rule "EMP Data". When opening the rule, you will get the error: "Oracle Audit policy not found for rule". This is because the policy's rule has not yet been fully created; its Oracle FGA policy part is yet non-existing.

Complete the missing parts of the RTP DML Rule "EMP Data", namely: check all the four SEL, INS, DEL, UPD Statements, set the Audit Trail to DB_EXTENDED and Column Options to ANY_COLUMNS. Do not check any object columns from those listed on the right. In the Rule Authorization group below, radio Group Authorization type, Radio box "No Condition" remains checked (default). Press the SAVE button to save changes made to the "EMP Data" rule! Activate the rule "EMP Data". Activate the policy "EV Application DMLs".

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.
Wait for the next run of the DB Audit Trails Purge, or invoke it manually.

In Omega Core Audit, search the Security Events trail. Real-Time Protection DML trails have been created for all testing users and all their actions. In the Record Details view, check the Trail Evaluation field. The policy name will be "EV Application DMLs". The policy Result will be TRUE. The Audit event has been triggered for all testing users and actions, the DMLs in Benchmark have been successful only because the Rule EMP Data is in Silent Deny mode (default on RTP DML Rule create).

In Omega Core Audit, change the Real-Time Protection DML rule "EMP Data" Rule. Unselect the Statements options INSERT, UPDATE and DELETE, leaving only the SELECT. Save the EMP Data Rule!

In Omega Core Audit Benchmark DML commands list, uncheck all DMLs, except for the SELECT commands (DML02, DML05, DML07 and DML09)!

We will keep it this way the rest of this evaluation only for the sake of simplicity, to have fewer records in the Security Events trail!

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.
Wait for the next run of the DB Audit Trails Purge, or invoke it manually.

In Omega Core Audit, search the Security Events trail. There will be again Real-Time Protection DML trails created for all users, but only for the SELECT action.

Up to this point now, the Real-Time Protection DML Rule looks similar to the Standard Audit Object Rule. However, some extra features that you might have already noted are the ones that make the difference, like: the protective capabilities (Silent Deny), context based authorization (Authorization type of Rule Conditions), audit and protection triggering on columns (conditions) used, row values conditions (SALARY>3000) and the ability to decide completion of SQL Bind and Text fields, latterly discussed.

In the following topics we will explore the advanced capabilities of the Real-Time Protection DML Policy.

5.2.3 Real-Time Protection DML - Rule Conditions Authorization

Rationale:
Using Real-Time Protection DML with Rule Authorization Type of Rule Conditions to achieve user-environment context evaluation, which means evaluating Rule by its Conditions (as for the rules of Access Control and Real-Time Protection DDL policies).

In Omega Core Audit change the "EMP Data" Rule. In the Rule Authorization group below, radio Group Authorization type, check the "Rule Conditions" Radio box. The Rule Conditions tab will open below. Set the Condition Evaluation combo box to "Any True"! Press the SAVE button to save changes made to the "EMP Data" rule!

Activate the two Operand Conditions of Session User and Client Host. For the later set the right "Operand 1" value instead of the provided <DOMAIN\HOST>.

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.
Wait for the next run of the DB Audit Trails Purge, or invoke it manually.

In Omega Core Audit, search the Security Events trail. There will be Real-Time Protection DML trails created for all users, except for the Application Owner OMEGACAEVAPP. This is because with the configuration of the DML Rule above, we have created a secure DML channel for the SELECTs on EMP performed by the Application Owner account (Session User) and coming from the Client Host. We are not requiring an audit event in this case, but in all its violations.

5.2.4 Real-Time Protection DML - row value audit conditioning

Rationale:
The use of Real-Time Protection DML with Rule Authorization Type of Column Condition achieves application context evaluation by setting conditions on application table's row values (row-based audit conditioning).

At this point (for this topic and for the next), you are advised to take a look at the content of the EMP table.
Run the following SQL:

```
select * from
OMEGACAEVAPP.V_EMP
order by EMP_ID
```

This will show the employees data and the Salary - used in this evaluation!

Further narrow the testing events, by testing for the OMEGACAEVDBA user only, to better focus on the features later described. Uncheck all other users in the grid.

In Omega Core Audit, change the "EMP Data" rule. In the Rule Authorization group, radio Group Authorization type, check the "Column Condition" Radio box. The Column Condition tab will open below. Set the Operation Code to > and Operand 1 to 3000. Drag and drop the SALARY column (found in the Object Columns checklist) to the Authorization Condition memo below. The condition SALARY>3000 will be set. Save the "EMP Data" Rule.

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.
Wait for the next run of the DB Audit Trails Purge, or invoke it manually.

In Omega Core Audit, search the Security Events trail. There will be Real-Time Protection DML created only for the DML05, DML07 and DML09 commands, while missing for DML02. In the later because from all employees returned from this query, none has a salary grater then 3000; they all belong to the Operations department, filtered in the query by their Ids (emp_id <= 4)! While the other SQL commands were audited because returned was always least one employee record with a salary grater then 3000, thus the audit event triggered and log generated!

Hint:

It does not matter if the column SALARY itself is part of the dataset returned columns or not, just the row condition satisfied!

5.2.5 Real-Time Protection DML - Column audit conditioning

Rationale:
The use of Real-Time Protection DML with Column Options achieves application context evaluation by setting conditions on application table's column usage (column-based audit conditioning).

In the Omega Core Audit change again the "EMP Data" Rule. In the Rule Authorization group, radio Group Authorization type, check the "No Condition" Radio box. This is done for the sake of simplicity, since the choice of the columns (and their audit-triggering effect) in the RTP DML Rule is independent of the Authorization Type chosen!

Check the Column Options is set to ANY_COLUMNS. In the Object Columns checklist, check the HIRE_DATE and SALARY columns. Save the "EMP Data" Rule.

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.
Wait for the next run of the DB Audit Trails Purge, or invoke it manually.

In Omega Core Audit, search the Security Events trail. There will be Real-Time Protection DML trails created only for the DML07 and DML09 commands, but not for the DML02 and DML05. In the later because none of the

HIRE_DATE and SALARY columns are accessed in any mode in their SQL commands. While the DML07 command is audited because it explicitly accesses the SALARY column; the DML09 command is audited because it access both columns through its asterisk (*) selection.

Hint:

Columns conditioning is triggered not only by columns being present in the dataset returned, but for whatever use of them in the executed SQL clauses, like "where <column_name> ….", or "order by <column_name> ….."!

Change the Column Options to ALL_COLUMNS. In the Object Columns checklist, check the FIRST_NAME, LAST_NAME and SALARY columns. Save the "EMP Data" Rule.

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.
Wait for the next run of the DB Audit Trails Purge, or invoke it manually.

In Omega Core Audit, search the Security Events trail. There will be Real-Time Protection DML trails created only for the DML07 and DML09 commands, but not for the DML02 and DML05. In later because the SALARY column was not accessed in any mode in their SQL commands. While the DML07 command is audited because it explicitly accesses all three columns; the DML09 command is audited because it does the same on all these three columns through its asterisk (*) selection.

5.2.6 Tuning the Real-Time Protection DML

Rationale:
Optimize Real-Time Protection DML for auditing knowledgeably and performance.

As you have noticed, there is no Audit Option configuration into the Real-Time Protection DML policies.

The Real-Time Protection DML mechanism is mostly advised to be used for important objects. Use it mostly to audit (and optionally protect by rejection) privileged accounts actions on such objects and every divergence from normal behavior, implemented in the DML Rule.

Used the way above, the Real-Time Protection DML will generate insignificant load to the system.

In general, tuning the Real-Time Protection DML assembles Tuning Requirements from all other three modules, as the Omega Core Audit internals are built such.

5.2.7 Enforcing the Real-Time Protection DML

Rationale:
Enforce Real-Time Protection DML on the application objects by rejecting non-complying DMLs.

Until now we have used the Real-Time Protection DML Rules in their default Silent Deny Mode. This means the DML actions were allowed to continue even when the audit event is triggered.

To test the protective capabilities of the Real-Time Protection DML rule, into the Omega Core Audit software, change again the DML Rule "EMP Data". In the Rule Authorization group below, radio Group Authorization type, check the "Rule Conditions" Radio box. The Rule Conditions tab will open below. Set the Condition Evaluation combo box to "Any True"! Uncheck all the checked Object Columns and set the Column Options back again to ANY_COLUMNS. Uncheck the Silent Deny Checkbox! Press the Save button to save the "EMP Data" Rule.

For the sake of simplicity, in the rule's Statements keep the single SELECT checked only. And in the Benchmark try the four SELECT DMLs only.

Press the Run button in Omega Core Audit Benchmark and wait until it finishes.

You will notice that only the DMLs of Application Owner OMEGACAEVAPP have been successful, while for all others (the two Developers and the DBA) you will notice the error "ORA-20010: RTP DML Rule violation Id: [Rule_ Id] Rule Name: EMP Data!". The SELECT DML actions on EMP have been rejected for all users other then the Application Schema Owner.

Wait for the next run of the DB Audit Trails Purge, or invoke it manually.

In Omega Core Audit, search the Security Events trail. There will be Real-Time Protection DML trails for the OMEGACAEVDEV1, OMEGACAEVDEV2 and OMEGACAEVDBA users. Notice the error message "RTP DML Policy Rule: EMP Data violation!"

6 Summary and Conclusions

The tests described so far in this document represent only a small view of the Omega Core Audit capabilities. The Omega Core Audit's provided evaluation environment and policies account for a usual production system's schema in miniature, while the Benchmark software, tailored for the environment and polices, generates actions on them in automatic mode.

The above is made to demonstrate the basic features of the Omega Core Audit. You are encouraged to try other combinations of audit/protection settings in Omega Core Audit and test the software's behavior by combining the options into the Omega Core Audit Benchmark.

It would be logical that some knowledge from the Omega User's Guide is required, may be not for exactly the above walkthrough guide, but surely for advanced testing options you may perform, or furthermore in your test-simulated environment and finally your production database.

7 Appendixes

7.1 Appendix A. Evaluation Policies

Below are the evaluation policies, as originally created directly on the repository by the provided SQL script.

7.1.1 Access Control

Policy	Policy Name	Description	Audit Option	User Appliance	Rule Eval.
1.	EV Application Access	EV Application Access	On Success/Failure	Users Apply	Any True
Rules	**Rule Name**	**Rule Description**	**Condition Eval.**		
1.	App. Owner 01 Access	App. Owner 01 Access	All True		
Conditions	**Factor**	**Condition Evaluation**	**Operation Code**	**Operand 1**	**Operand 2**
1.	Session User	Operand	=	OMEGACAEVAPP	
2.	Client Host	Operand	=	<DOMAIN/HOST>	

Policy	Policy Name	Description	Audit Option	User Appliance	Rule Eval.
1.	EV Developer Access	EV Developer Access	On Success/Failure	Users Apply	Any True
Rules	**Rule Name**	**Rule Description**	**Condition Eval.**		
1.	Developer 01 Access	Developer 01 Access	All True		
Conditions	**Factor**	**Condition Evaluation**	**Operation Code**	**Operand 1**	**Operand 2**
1.	Session User	Operand	=	OMEGACAEVDEV1	
2.	Client Host	Operand	=	<DOMAIN/HOST>	
Rules	**Rule Name**	**Rule Description**	**Condition Eval.**		
2.	Developer 02 Access	Developer 02 Access	All True		
Conditions	**Factor**	**Condition Evaluation**	**Operation Code**	**Operand 1**	**Operand 2**
1.	Session User	Operand	=	OMEGACAEVDEV2	
2.	Client Host	Operand	=	<DOMAIN/HOST>	

Policy	Policy Name	Description	Audit Option	User Appliance	Rule Eval.
1.	EV DBA Access	EV DBA Access	On Success/Failure	Users Apply	Any True
Rules	**Rule Name**	**Rule Description**	**Condition Eval.**		
1.	DBA 01 Access	DBA 01 Access	All True		
Conditions	**Factor**	**Condition Evaluation**	**Operation Code**	**Operand 1**	**Operand 2**
1.	Session User	Operand	=	OMEGACAEVDBA	
2.	Client Host	Operand	=	<DOMAIN/HOST>	

Policy	Policy Name	Description	Audit Option	User Appliance	Rule Eval.
1.	EV Oracle Internals	EV Oracle Internals	On Success/Failure	Users Apply	Any True
Rule	**Rule Name**	**Rule Description**	**Condition Eval.**		
1.	Oracle Connections	Oracle Connections	All True		
Condition	**Factor**	**Condition Evaluation**	**Operation Code**	**Operand 1**	**Operand 2**
1.	Client Host	Operand	=	<DB-SRV>	

7.1.2 Standard Audit

Policy	Policy Name	Description	Type
1.	EV Developer Audit	EV Developer Audit	Statement [Priv.]
Rule	**Statement**	**Username**	**Success/Failure**
1.	DELETE TABLE	OMEGACAEVDEV1	Success/Failure
2.	INSERT TABLE	OMEGACAEVDEV1	Success/Failure
3.	SELECT TABLE	OMEGACAEVDEV1	Success/Failure
4.	UPDATE TABLE	OMEGACAEVDEV1	Success/Failure
5.	DELETE TABLE	OMEGACAEVDEV2	Success/Failure
6.	INSERT TABLE	OMEGACAEVDEV2	Success/Failure
7.	SELECT TABLE	OMEGACAEVDEV2	Success/Failure
8.	UPDATE TABLE	OMEGACAEVDEV2	Success/Failure

Policy	Policy Name	Description	Type		

1.	EV App Obj Audit	EV App Obj Audit	Object		
Rule	**Object Owner**	**Object Type**	**Object Name**	**Audit Operation**	**Success/Failure**
1.	OMEGACAEVAPP	TABLE	EMP	DELETE	Success/Failure
2.	OMEGACAEVAPP	TABLE	EMP	UPDATE	Success/Failure

Policy	**Policy Name**	**Description**	**Type**
1.	EV DBA Audit	EV DBA Audit	Statement [Priv.]
Rule	**Statement**	**Username**	**Success/Failure**
1.	DELETE ANY TABLE	OMEGACAEVDBA	Success/Failure
2.	INSERT ANY TABLE	OMEGACAEVDBA	Success/Failure
3.	UPDATE ANY TABLE	OMEGACAEVDBA	Success/Failure
4.	SYSTEM GRANT	OMEGACAEVDBA	Success/Failure
5.	GRANT ANY OBJECT PRIVILEGE	OMEGACAEVDBA	Success/Failure
6.	GRANT ANY ROLE	OMEGACAEVDBA	Success/Failure
7.	SYSTEM AUDIT	OMEGACAEVDBA	Success/Failure
8.	AUDIT ANY	OMEGACAEVDBA	Success/Failure

7.1.3 Real-Time Protection DDL

Policy	**Policy Name**	**Description**	**Audit Option**	**User Appliance**	**Rule Eval.**
1.	EV Application DDls	EV Application DDls	On Success/Failure	All Users	Any True
Rule	**Rule Name**	**Rule Description**	**Condition Eval.**		
1.	Developer 01 DDL	Developer 01 DDL	All True		
Condition	**Factor**	**Condition Evaluation**	**Operation Code**	**Operand 1**	**Operand 2**
1.	Session User	Operand	=	OMEGACAEVDEV1	
2.	Client Host	Operand	=	<DOMAIN/HOST>	
Rule	**Rule Name**	**Rule Description**	**Condition Eval.**		
2.	Developer 02 DDL	Developer 02 DDL	All True		
Condition	**Factor**	**Condition Evaluation**	**Operation Code**	**Operand 1**	**Operand 2**
1.	Session User	Operand	=	OMEGACAEVDEV2	
2.	Client Host	Operand	=	<DOMAIN/HOST>	

Policy	**Policy Name**	**Description**	**Audit Option**	**User Appliance**	**Rule Eval.**
1.	EV Security and Audit	EV Security and Audit	On Success/Failure	All Users	Any True
Rule	**Rule Name**	**Rule Description**	**Condition Eval.**		
1.	DBA Security	DBA Security	All True		
Condition	**Factor**	**Condition Evaluation**	**Operation Code**	**Operand 1**	**Operand 2**
1.	Session User	Operand	=	OMEGACAEVDBA	
2.	Client Host	Operand	=	<DOMAIN/HOST>	

Policy	**Policy Name**	**Description**	**Audit Option**	**User Appliance**	**Rule Eval.**
1.	EV Users and Roles	EV Users and Roles	On Success/Failure	All Users	Any True
Rule	**Rule Name**	**Rule Description**	**Condition Eval.**		
1.	DBA Security	DBA Security	All True		
Condition	**Factor**	**Condition Evaluation**	**Operation Code**	**Operand 1**	**Operand 2**
1.	Session User	Operand	=	OMEGACAEVDBA	
2.	Client Host	Operand	=	<DOMAIN/HOST>	

7.1.4 Real-Time Protection DML

Policy	**Policy Name**	**Description**	**User Appliance**		
1.	EV Application Objects	EV Application Objects	All Users		
Rule	**Rule Name**	**Rule Description**	**Authorization Type**	**Condition Eval.**	
1.	EMP Data	EMP Data	Rule Conditions	Any True	
Condition	**Factor**	**Condition Evaluation**	**Operation Code**	**Operand 1**	**Operand 2**
1.	Session User	Operand	<>	OMEGACAEVAPP	
2.	Client Host	Operand	<>	<DOMAIN/HOST>	

7.2 Appendix B. Support and Licensing

7.2.1 Support

Omega Core Audit users of the free *Standard* edition are entitled to:

- new (Standard) versions
- upgrades and fixes
- new security controls
- online/offline documentations

Commercial annual support is available to the users of the *Professional* edition and also, cumulatively, features:

- discounts for new (Professional) versions
- online support

Professional edition SLA:

Response Time SLA:	2 Business Days
Support Call/Online:	09:00 GMT to 21:00 GMT, Monday to Friday
Emergencies:	we are here to help
Resolution Time:	case-related

For product documentation, forum and knowledge base, visit our site:
www.dataplus-al.com/omega-core-audit

For technical issues, comments, ideas and impressions, e-mail us at:
support@dataplus-al.com

Also follow us on the next social media sites where DATAPLUS is present:

YouTube	https://www.youtube.com/channel/UCa59qQuGg5tvd2vIe1MsMOw
LinkedIn	https://www.linkedin.com/company/dataplus-al
Peerlyst	https://www.peerlyst.com/companies/dataplus/dashboard

7.2.2 Licensing

Omega Core Audit license is perpetual for both Standard and Professional editions. It allows the customer to use the licensed software indefinitely and not tied to the product version. However, in the Professional edition, you are entitled to new versions and upgrades only if you have a valid maintenance and support contract in place.

Omega Core Auditing for Oracle licensing model is based on the Edition of your Oracle database server, the number and type of the processors and the number of cores per processor, where this late applies, of the machine where the database server resides.

Enterprise Edition Per-core licensing:

If the edition of database monitored is the Enterprise Edition, the licensing model is based on the number and type of processors and the number of cores per processor, where this late applies. The licensing formula is:

Number of Licenses = (Processors) * (Cores per Processor) * CPLF

Where:

Processors	Number of physical processors in the machine
Cores per Processor	Number of cores per processor (1 for single processors)
CPLF	Core Processor Licensing Factor

The CPLF (Core Processor Licensing Factor) value varies from 0.5 to 1.0 for different processors vendors and models and can be seen on the table at the end of this topic.

If the "Number of Licenses" has a fractional part, it will be rounded up to the next whole number.
This means a minimal number of 1 (one) license is required!

For example, an Enterprise Edition residing on a multi-core system with 2 x 4-core processors would require 2*4*0.5 = 4 licenses (for a CPLF of 0.5).

CPLF (Core Processor Licensing Factor) Table:

Vendor and Processor	CPLF
Sun and Fujitsu SPARC64 VI, VII	0.75
Sun UltraSPARC IV, IV+, or earlier Multicore chips	0.75
Sun UltraSPARC T2	0.75
HP PA-RISC	0.75
IBM POWER5+ or earlier Multicore chips	0.75
Intel Itanium Series 93XX (For servers purchased on or after Dec 1st, 2010)	1.0
Intel Itanium Series 95XX	1.0
IBM POWER6	1.0
IBM POWER7, IBM POWER7+	1.0
IBM POWER8	1.0
IBM System z (z10 and earlier)	1.0
All Other Multicore chips	0.5
All Single Core Chips	1.0

Standard Edition Per-socket licensing:

If the edition of database monitored is the Standard Edition or Standard Edition One, the licensing model is based on the number of processors. A processor is counted equivalent to an occupied socket; however, in the case of multi-chip modules, each chip in the multi-chip module is counted as one occupied socket. This means that in the formula above (for the Enterprise Edition) the CPLF and Number Cores/Processor are always 1 (one)!
For example, a Standard Edition (One) residing on 2 Processors system you would require 2 licenses.

Note:

Processor technologies of "Soft partitioning" such as VMWare and Microsoft Virtual Server, or others of the same kind, are not recognized in the formula. The pricing is calculated based on the physical processor(s) of the underlying machine hardware.

For pricing and licensing information, contact our Sale specialists at:
sales@dataplus-al.com

Copyright:

Copyright © 2007-2020 DATAPLUS. All rights reserved. Omega Core Audit is registered at US Copyrights Office and is protected by US and international copyright laws. No part of this work may be reproduced, stored in a retrieval system, adopted or transmitted in any form or by any means, electronic or otherwise, translated in any language or computer language, without the prior written permission of DATAPLUS. Omega Core Audit and the DATAPLUS logo are trademarks of DATAPLUS. All other trademarks are the property of their respective owners.