INTEGRIGY

WHITE PAPER

Guide to Auditing and Logging Oracle Databases

DECEMBER 2014

GUIDE TO AUDITING AND LOGGING ORACLE DATABASES

Version 1.0 – December 2014

Authors: Michael A. Miller, CISSP-ISSMP and Stephen Kost

If you have any questions, comments, or suggestions regarding this document, please send them via e-mail to info@integrigy.com.

Table of Contents

OVERVIEW

Most Oracle databases do not fully take advantage of the auditing and logging features. These features are sophisticated and are able to satisfy most organization's compliance and security requirements.

The default Oracle database installation only provides a basic set of logging functionality. In Integrigy's experience, the implementation of database and application logging seldom exceeds meeting the needs of basic debugging. Most organizations do not know where to start or how to leverage the built-in auditing and logging features to satisfy their compliance and security requirements.

Even organizations already using centralized logging or Security Incident and Event Management (SIEM) solutions, while being more advanced in the Common Maturity Model (CMM), in Integrigy's experience are commonly challenged by the Oracle database's auditing and logging features and functionality.

This guide presents Integrigy's Framework for auditing and logging for Oracle databases. This Framework is a direct result of Integrigy's consulting experience and will be equally useful to both those wanting to improve their capabilities as well as those just starting to implement logging and auditing. Our goal is to provide a clear explanation of the native auditing and logging features available, present an approach and strategy for using these features and a straight-forward configuration steps to implement the approach.

Integrigy's Framework is also specifically designed to help clients meet compliance and security standards such as Sarbanes-Oxley (SOX), Payment Card Industry (PCI), FISMA, and HIPAA. The foundation of the Framework is PCI DSS requirement 10.2.

To make it easy for clients to implement, the Framework has three maturity levels – which level a client starts at depends on the infrastructure and policies already in place.

The three levels are:

- **Level 1** – Enable baseline auditing and logging for application/database and implement security monitoring and auditing alerts
- **Level 2** – Send audit and log data to a centralized logging solution outside the Oracle Database
- **Level 3** – Extend logging to include functional logging and more complex alerting and monitoring

Oracle 12c and Unified Auditing and Pluggable Databases

Oracle 12c delivers several important new features and functions. Unified Auditing and multi-tenant or pluggable databases are two of the key new Oracle 12c security features. Unified Auditing is free with Enterprise Edition, and Pluggable databases (multi-tenant) requires an additional license.

Integrigy's log and audit Framework can be easily implemented using Oracle 12c. The Framework works with 12c's Unified Auditing, either in Mixed or Purge mode. With regard to pluggable databases, Integrigy's Framework can be implemented either to a single pluggable (tenant) database or to all pluggable databases.

Oracle Audit Vault

Oracle's Audit Vault solution is increasing in popularity having been voted Database Trends and Reader's Choice Best Database Security Solution for 2014[1]. The Oracle Audit Vault provides a comprehensive and flexible monitoring solution by consolidating audit data from Oracle and non-Oracle databases, operating systems, directory, file systems, and application log data.

As a purpose built tool for Oracle security log and audit monitoring, implementing Integrigy's Log and Audit Framework using the Oracle Audit Vault is straight forward and offers compelling advantages, especially if you are using or considering Oracle 12c Unified Auditing. The Oracle Audit Vault is a separately licensed product.

Audience and How to Read This Paper

The intended audience are Oracle DBAs, application administrators, IT security staff, and internal audit staff. A working technical knowledge of the Oracle Databases is recommended.

[1] Reader's Choice Awards 2014
https://blogs.oracle.com/securityinsideout/entry/oracle_audit_vault_and_database1

INTEGRIGY'S FRAMEWORK FOR ORACLE DATABASE AUDITING

The Framework is a result of Integrigy's consulting experience and is based on compliance and security standards such as Payment Card Industry (PCI-DSS), Sarbanes-Oxley (SOX), IT Security (ISO 27001), FISMA (NIST 800-53), and HIPAA.

The foundation of the Framework is the set of security events and actions that should be audited and logged in all Oracle databases. These security events and actions are derived from and mapped back to key compliance and security standards most organizations have to comply with. We view these security events and actions as the core set, and most organizations will need to expand these events and actions to address specific compliance and security requirements, such as functional or change management requirements.

Table 1 presents the core set of audits that, if implemented, will serve as a foundation for more advanced security analytics. Implementing these audits will go a long way toward meeting logging and auditing requirements for most compliance and security standards like PCI requirement 10.2. The numbering scheme used in Table 1 will be referenced throughout the document.

Table 1 – Foundation Events for Logging and Security Framework

Security Events and Actions	PCI DSS 10.2	SOX (COBIT)	HIPAA (NIST 800-66)	IT Security (ISO 27001)	FISMA (NIST 800-53)
E1 - Login	10.2.5	A12.3 DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2
E2 - Logoff	10.2.5	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2
E3 - Unsuccessful login	10.2.4	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1 A.11.5.1	AC-7
E4 - Modify authentication mechanisms	10.2.5	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2
E5 – Create user account	10.2.5	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2
E6 - Modify user account	10.2.5	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2
E7 - Create role	10.2.5	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2
E8 - Modify role	10.2.5	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2

Table 1 – Foundation Events for Logging and Security Framework					
Security Events and Actions	**PCI DSS 10.2**	**SOX (COBIT)**	**HIPAA (NIST 800-66)**	**IT Security (ISO 27001)**	**FISMA (NIST 800-53)**
E9 - Grant/revoke user privileges	10.2.5	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2
E10 - Grant/revoke role privileges	10.2.5	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2
E11 - Privileged commands	10.2.2	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2
E12 - Modify audit and logging	10.2.6	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2 AU-9
E13 - Objects: Create object Modify object Delete object	10.2.7	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2 AU-14
E14 - Modify configuration settings	10.2.2	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2

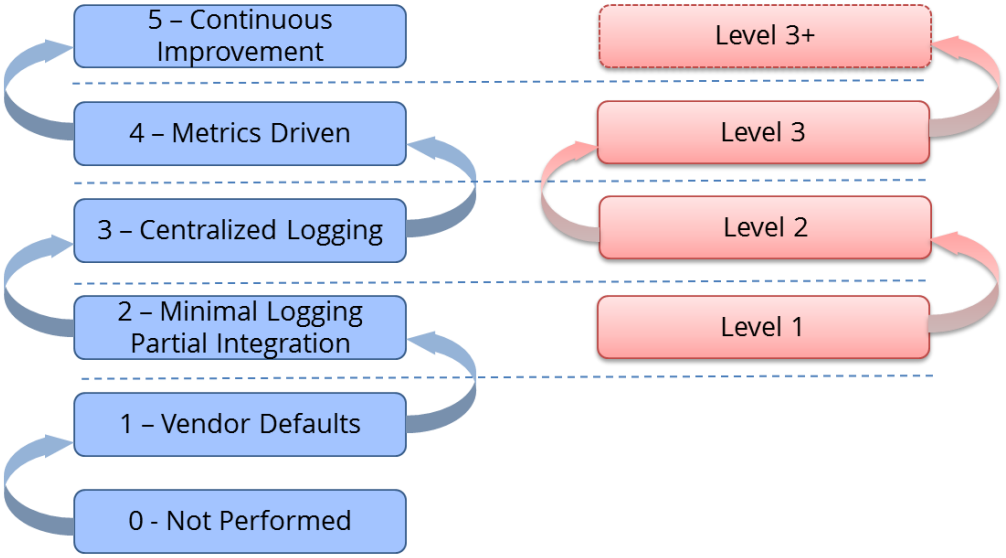
FRAMEWORK APPROACH

Integrigy's Framework has three levels of maturity. Not all organizations will start at the same level. Which level a client starts at depends on the infrastructure and policies an organization already has in place. Integrigy's experience is that using this approach will give both specific guidance as well as a vision.

The levels are:

- **Level 1** – Enable basic logging and implement a best practices checklists for security monitoring and auditing. Implementation focus is on DBAs and application administrators.
- **Level 2** – Send basic log data to a centralized logging solution outside the Oracle Database. Implementation focus is on IT security and internal auditors and their meeting the basic requirements.
- **Level 3** – Expand auditing scope and pass additional database logs to a centralized logging solution. Implementation focus is on IT security and internal auditors to meet advanced requirements for compliance and automation. This is commonly done to meet specific requirements for compliance PCI, SOX, HIPAA and ISO 27001.

Figure 1 - Integrigy Framework Compared to Common Maturity Model



Level 1

The first level focuses on logging and basic monitoring and auditing. Logging, monitoring, and auditing are separate but related disciplines. Logging provides the data for both monitoring and auditing. In the Framework's first level optional logging functionality is enabled. This is functionality not enabled by default and is commonly not used. Once this functionality is in place, the Framework then presents a best practice checklist for security monitoring and auditing for the Oracle RDBMS. For those customers considering a security monitoring and auditing program, this should be an ideal starting point.

Level 2

The second level of maturity focuses on integrating with a centralized logging solution. Given the complexity of the Oracle RDBMS and numerous compliance requirements for protection and non-repudiation of log data, a centralized logging solution is required. Once the solution is in place, Level 2 of the Framework presents where and how to start passing log and audit data from the Oracle Database.

Level 3

The third level of maturity is continuous. Once the basic log data is being passed to a centralized logging solution and/or Security Incident and Event Management (SIEM) system, the Framework identifies additional database and application server logs to be captured. As Level 3 is continuous, as the possibilities of security incident and event correlation rules and filters are only limited by the data within the Oracle RDBMS.

Level 3 is where database log and audit functionality is consolidated with application log and audit functionality. Level 3 is where Oracle Apex and 12c Real Application Security (RAS) auditing would be consolidated as well as logs from Fine Grained Auditing (FGA) and Oracle Label Security (OLS). Moreover, Level 3 is where application logging and auditing should be consolidated and correlated to database logging and auditing. For example,

enterprise applications such as the Oracle E-Business Suite, PeopleSoft, OBIEE and SAP all have extensive log and audit functionality which can and should be utilized as part of Integrigy's Framework.

Figure 2 - Integrigy Framework Auditing and Logging Framework

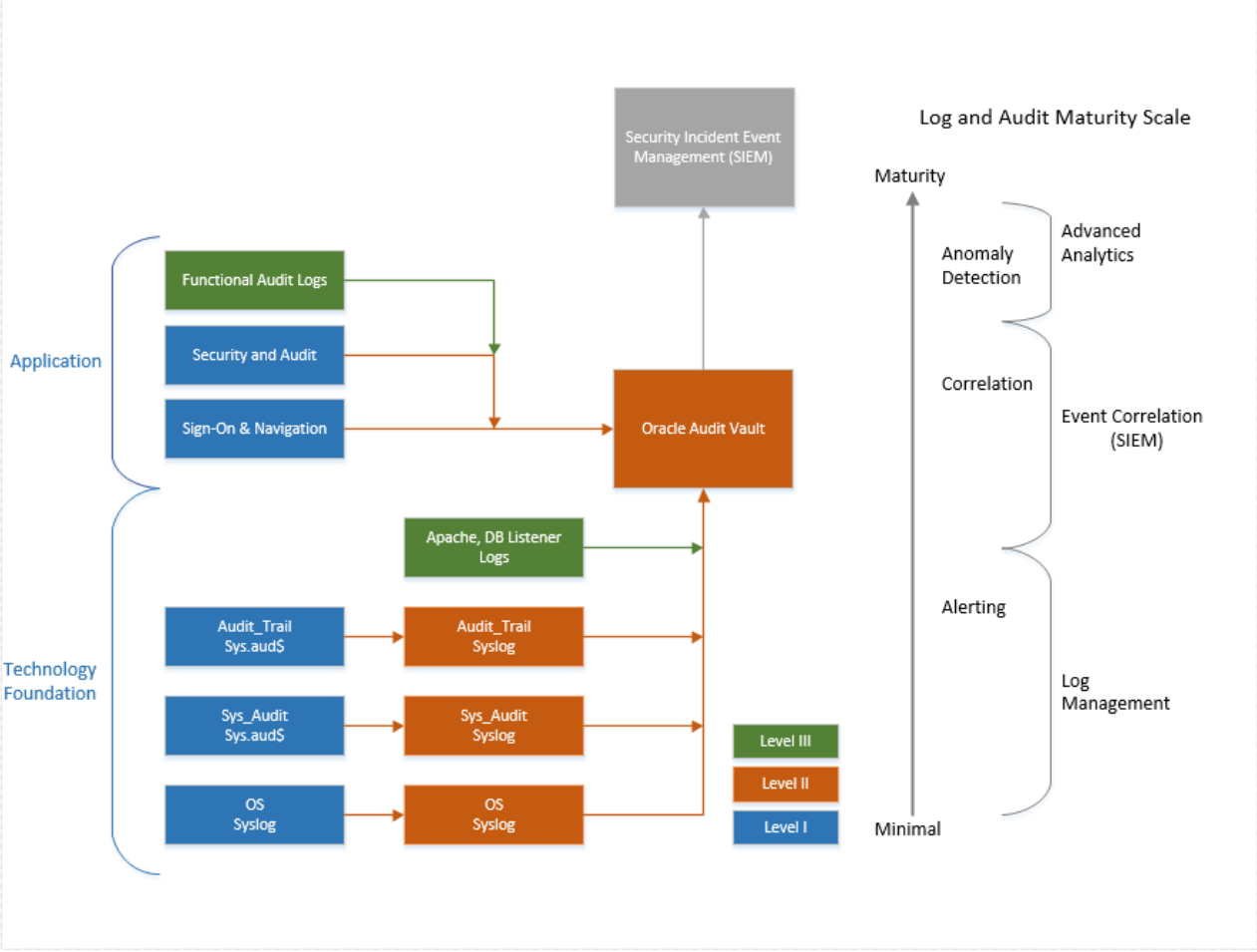
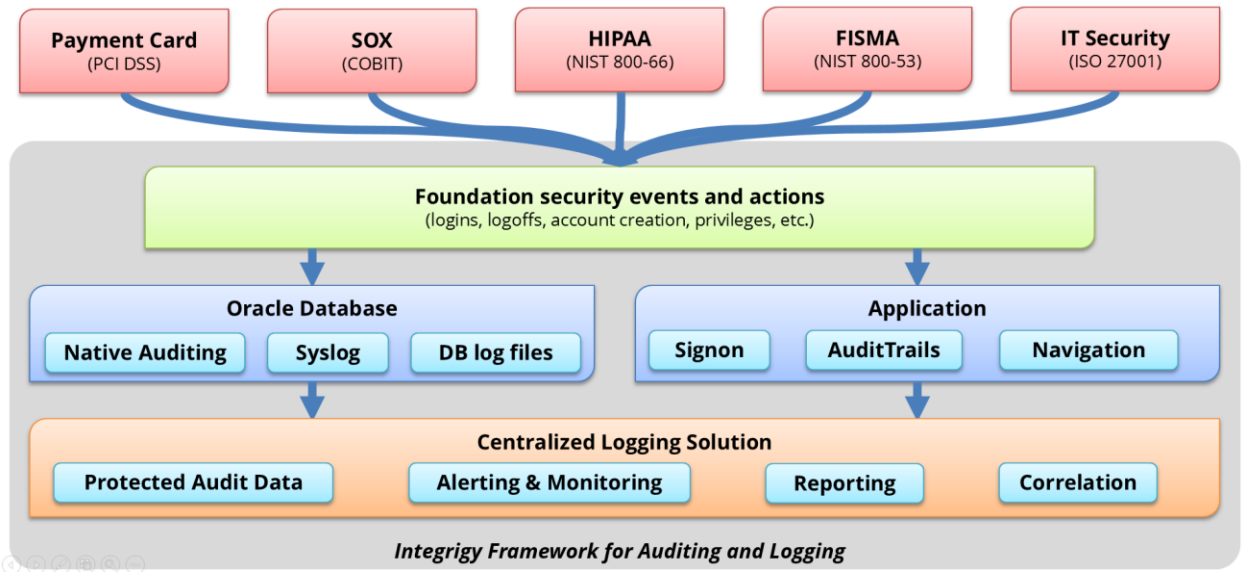


Figure 3 - Integrigy Framework for Auditing and Logging in Oracle RDBMS



Log and Audit Functionality

This section reviews the basic log and audit functionality available in the Oracle Database. Some of this functionality is enabled by default – some of it is optional and needs to be configured. It should also be noted that more audit and monitoring functionality exists than what is discussed here. The scope of this discussion is limited to what is required to implement Integrigy's Framework.

NOTE: This section may be optional if the reader is already familiar with the core auditing and logging functionality in the Oracle. The purpose is to provide an overview of the key auditing and logging features used to implement Integrigy's Framework.

What Is a Log?

A "log" is a collection of messages that "paints a picture" of an event or occurrence. The following are general categories of log messages, all of which are important to Integrigy's Framework:

- **Informational –** benign event occurrence, for example, a system reboot
- **Debug –** information to aid developers and administrators
- **Warning –** events affecting systems and applications
- **Error –** application or system fault
- **Alert –** something interesting has occurred

A log message has three parts:

1. **Timestamp** – when did the event occur
2. **Source** – server, application or person

3. **Data** – system message, SQL statement, debug code, etc.

Operating System Logging

Most, if not all, Oracle RDBMS implementations running on UNIX or Linux will have Syslog enabled by the system administrators and/or hosting provider. Syslog is a standard for UNIX and Linux message logging and supports a wide variety of devices, from printers and network routers to database servers. Syslog messages generated by applications or services are sent to the message store on the system or can be delivered to a centralized server built for the specific purpose of log storage and analysis.

The following basic operating system events are assumed to be collected and available:

- System startup/shutdown
- Logons and attempted logons – IP address, port, time
- Process history and statics

Oracle Database

Oracle Databases offer a rich set of logging and auditing functionality. For Integrigy's Framework, standard Oracle Database auditing and the capability to send database audit logs to Syslog will be leveraged.

Standard Oracle Auditing

Standard auditing is available in all editions of the Oracle RDBMS. It can be used to audit SQL statements, privileges, schemas, objects and network and multitier activity. Standard auditing must be enabled, and once enabled, a regular program for purging data needs to be implemented.

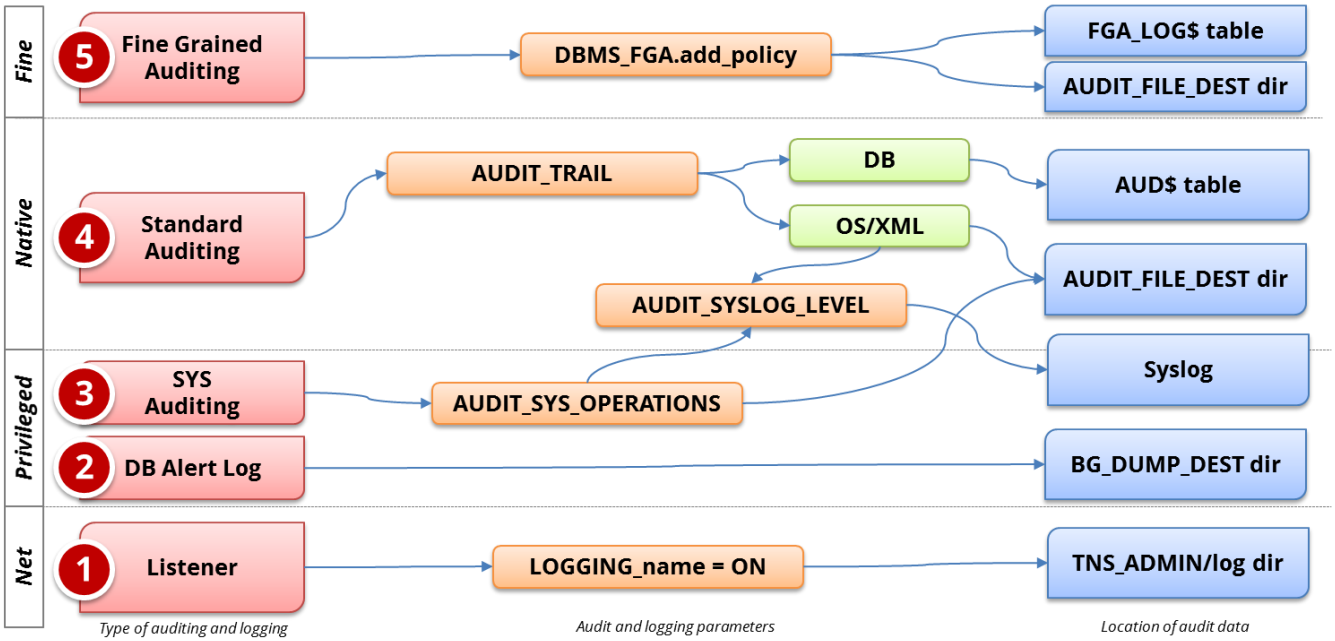
In Oracle 12c standard auditing is referred to as 'Mixed Mode.' Oracle 12 introduces Unified Auditing and has two modes: 'Mixed Mode' (default) that makes use of all pre-12c audit configurations and 'Pure' mode which introduces a whole new set of log and audit configurations. Integrigy's Framework can be implemented using either the default 'Mixed Mode' or 'Pure Mode.' Refer to Integrigy's Whitepaper on Oracle 12c Unified Auditing for more information on Unified Auditing.

The variety and volume of data collected by standard auditing can be large, and the output can be directed either to the database itself or to files in the operating system outside the database. Moving logs outside the reach of DBAs, either into the operating system or sent to a centralized log server, offers many security benefits. For more information on standard auditing refer to the sources in the reference section of this document.

Database Syslog

As noted earlier, Syslog is a standard for UNIX and Linux logging. Oracle Syslog option is a standard database feature that sends Oracle log data to the native operating system Syslog facility, which in turn can be forwarded directly to a centralized Syslog server or collector. The native Oracle Syslog auditing has minimal performance overhead and provides immediate protection of the audit trail. However, it is possible for the DBA to disable auditing and mitigating controls must be established around possible deactivation of the auditing. For more information on Syslog refer to the reference section of this document.

Figure 4 - Database Auditing and Logging



INTEGRIGY FRAMEWORK – LEVEL 1

Level 1 focuses on the basic logging that Integrigy recommends for all Oracle database installations. This logging needs to be in place before proceeding to Levels 2 and 3 of the Framework, but assumes a centralized logging solution is not available yet.

The following summarizes the steps to implement Level 1:

1. Oracle Database logging
    a. Enable standard database auditing to the database (AUD$) per Integrigy's recommendations
    b. Enable AUDIT_SYS_OPERATIONS
2. Set up policies and procedures for security monitoring and auditing

DATABASE AUDITING

Database auditing is vital to application logging and security monitoring as direct database access can be used to circumvent all application controls.

Level 1 assumes there is no centralized logging solution implemented, and the audit data is being written to the database (SYS.AUD$) for monitoring and reporting. Saving audit data to the database is not ideal as the DBA can manipulate the audit data, but provides for much-simplified monitoring and reporting. If a centralized logging solution is implemented, then the database audit data should be written to Syslog per the instructions in Level 2.

Steps for Level 1 database auditing:

1. Enable native database auditing and store audit data to the database. In the init.ora file for the instance set the database initialization parameter **AUDIT_TRAIL** to **DB**. This will write out all logs to the SYS.AUD$ table except for SYS Operations, which are always written to the operating system audit trail.
2. As the SYS user configure database auditing per *Table 2 – Recommended Oracle Database Auditing*.
3. The SYS.AUD$ table needs to be purged on a periodic basis per your organization's policy requirement. All rows should be backed up prior to being purged. Purging is configured through the use **DBMS_AUDIT_MGMT**.
4. In the init.ora file for the database instance, enable auditing of the SYS user by setting the database initialization parameter **AUDIT_SYS_OPERATIONS** to **TRUE**. Logs are written to the operating system's native audit trail.

Table 2 – Recommended Oracle Database Auditing				
Framework Event	**Database Object**	**Oracle Audit Statement (audit {};)**	**Resulting Audited SQL Statements**	**Notes**
E1, E2, E3	Session	session	Database logons and failed logons	▪ All database logons and failed logons ▪ This is highly dependent on database usage and application. With application connection pooling, the number of database session is minimized. However, some frequent interface programs may result in large numbers of sessions.
E5, E6	Users	user	create user alter user drop user	▪ All changes to users ▪ Includes all password changes by users – actual password is not captured
E7, E8	Roles	role	create role alter role drop role	▪ All changes to roles ▪ SET ROLE is excluded which is frequently used and would be included if AUDIT ROLE was used
E13	Database Links Public Database Links	database link public database link	create database link drop database link create public database link drop public database link	▪ Creation and deletion of database links
E11, E14	System	alter system	alter system	▪ Changes to the database configuration ▪ Audits killing of sessions, open/closing wallet, and setting of initialization parameters
	Database	alter database	alter database	▪ Change to database and instance state
E9, E10	Grants (system privileges and roles)	system grant	grant revoke	▪ Captures only grants to system privileges and roles ▪ Grants/revokes on database objects will be captured as part of the object creation
E4	Profiles	profile	create profile alter profile drop profile	▪ All changes to password and resource profiles ▪ Assigning profiles to users will be captured as part of ALTER USER
E9, E10	Directories	grant directory	grant directory revoke directory	▪ Granting of directories

Table 2 – Recommended Oracle Database Auditing				
Framework Event	**Database Object**	**Oracle Audit Statement (audit {};)**	**Resulting Audited SQL Statements**	**Notes**
E9, E10	Procedures Packages Functions Libraries Java Objects	grant procedure	grant <procedural type> revoke <procedural type>	▪ Granting and revoking of procedural objects
E9, E10	Object Grants	grant sequence grant table grant type	grant sequence grant table/view grant type revoke sequence revoke table/view revoke type	▪ Granting on sequence, tables, types, and views ▪ Grant table will also audit grant view
E12	Auditing	system audit	audit noaudit	▪ Changes to database auditing
E11, E14	SYSDBA and SYSOPER	sysdba sysoper	All SQL executed with sysdba and sysoper privileges	▪ Actions taken by DBAs – mostly occurs during weekly maintenance window

As part of the Framework Level 1, we do not recommend enabling extensive auditing of database object (e.g., tables, indexes, procedures, etc.) creation, modification, or deletion since most enterprise application environments will generate a significant amount of audit data. Many enterprise applications create temporary objects, and there are frequent changed due to patching. For example, the Oracle E-Business Suite APPS user account used during these activities and mostly originated from the application or database servers. Thus, the audit trail becomes fairly meaningless.

Refer to Appendix A – Example Level 1 Audit Script for a sample script to enable auditing for Level 1.

INTEGRIGY FRAMEWORK – LEVEL 2

The second level of the Framework focuses on integrating with and/or building a centralized logging solution if such a solution does not exist. Such solutions are commonly built using enterprise logging solutions such as Oracle Audit Vault, Splunk, HP ArcSight, RSA enVision, or Q1 Radar. There are a number of commercial and open-source solutions that can support all the logging and auditing in the Integrigy Framework. For Integrigy's Framework, the specific tool is used is not important. What is important is the solution provides (1) ability to accept logs from Syslog, database connections, and reading files, (2) security and archiving of log data, and (3) unified alerting and reporting capabilities.

Centralized logging solutions protect the log data. Non-repudiation and division of duties is achieved by removing log data from each source and storing it in a secure, central location. Consolidating an organization's log data also offers significantly more options for creating security alerts that cross application, team, and geographic boundaries. Centralized logging is also a key requirement for security standards including PCI and HIPAA.

Once the foundation of centralized logging is created with Level 2, an organization can proceed to Level 3. Contact Integrigy with questions and/or assistance with specific centralized logging tools and/or vendors.

Level 2 Tasks

1. Implement centralized logging solution if it does not exist
2. Redirect database logs to centralized logging
3. Protect application sign-on and navigation audit logs
4. Transition Level 1 alerts and build additional Level 2 alerts

IMPLEMENT CENTRALIZED LOGGING SOLUTION

The installation and configuration of tools such as Oracle Audit Vault, Splunk (Free or Enterprise) or HP ArcSight is beyond the scope of this paper. The first requirement for Level 2 is for such a solution to be in place.

REDIRECT DATABASE LOGS TO CENTRALIZED LOGGING

Writing logs to the operating system is more secure for many reasons, including providing a separation of duties between DBAs and system administrators. There are two steps:

1. To route Oracle database audit logs to the operating system instead of the database set **AUDIT_TRAIL** parameter to **OS** and set **AUDIT_FILE_DEST** to provide a location to write the log files**.**
2. Write logs using the Syslog format. In the init.ora file for the instance set the **AUDIT_TRAIL** parameter to **OS** and **AUDIT_SYSLOG_LEVEL** to 'LOCAL1.WARNING' or another valid Syslog setting. This setting may be used by the logging server to classify the event.

PROTECT APPLICATION SIGN-ON AND NAVIGATION AUDIT LOGS

With the centralized logging solution in place, protect the application sign-on and navigation audit logs using the centralized logging solution. For Oracle Audit Vault, this will start by configuring standard auditing for the tables – see Integrigy's Whitepaper on the Oracle Audit Vault for more information on the Oracle Audit Vault.

For other centralized logging solutions such as Splunk, configure a database connector. Once the connector is in place, pass application sign-on and navigation activity to the centralized logging solution. Custom queries will be required for many of the tables as these tables have referential IDs rather than usable values in many columns (e.g., USER_ID vs. USER_NAME).

Table 6 – Example for Oracle E-Business Suite Sign-on and Navigation Tables (Framework: E1, E2, E3)	
Table	**Description**
APPLSYS.FND_LOGINS	History of each successful end-user login
APPLSYS.FND_LOGIN_RESPONSIBILITIES	History of what menus (responsibilities) were used by each end-user
APPLSYS.FND_LOGIN_RESP_FORMS	History of what individual forms were visited by each end-user
APPLSYS.FND_UNSUCCESSFUL_LOGINS	Unsuccessful login attempts to the Oracle E-Business Suite

TRANSITION LEVEL 1 ALERTS AND BUILD ADDITIONAL LEVEL 2 ALERTS

As much as possible transition all alerting built for Level 1 to the centralized logging solution. Alerting out of the logging solution, (or SIEM) will be more efficient and can provide event correlation capabilities. Moreover, as more alerts will be built, it will consolidate alerting into a single tool.

As with Level 1, the table below is by no means conclusive. Simple things can trigger serious high-risk security events. As such, the table below should be seen as much as a starting point as it is an educational tool. What to monitor for and whom to notify will largely be determined by each client's unique risk profile. The table below suggested several alerts that could be implemented using a purpose built tool such as the Oracle Audit Vault or Splunk to correlate both database and application logon attempts.

Table 7 – Level 2 Monitoring and Auditing Alerts		
Event	**What to Monitor For**	**Description**
E1	Successful or unsuccessful login attempts to the application without network or system login	Logins or attempts to login into the application without first logging onto the network or gaining access to the building should be flagged and investigated.
E1	Successful or unsuccessful logins of named database user without network or system login	Named database accounts, those associated with staff and employees for the purposes of support should be monitored for if the user has first logged on to the network and/or gained access to the building.
E3	Horizontal unsuccessful direct database attempts – more than 5 users more than 5 times within the hour	Attempts to brute force groups of users should be alerted and investigated. This alert may be based per IP address or other system identifiers. The specific alert threshold will be unique to each client.
N/A	Monitor for database attacks	The following standard Oracle error messages may indicate a potential database attack: ORA-29532, ORA-28000, ORA-24247, ORA-29257, ORA-01031

INTEGRIGY FRAMEWORK – LEVEL 3

Level 3 first builds on the connectivity and basic centralized logging established in Level 2 and then identifies additional database and application server logs. Level 3 is also where database log and audit functionality is consolidated with application log and audit functionality. These additions to the centralized logs defined in Level 1 allow clients to meet compliance requirements such as PCI, SOX, and HIPPA and provide automation of the compliance tasks.

For the Framework, the most important concept in Level 3 is the inclusion of the application functional log and audit data into a centralized repository. While applications such as E-Business Suite, PeopleSoft, and SAP are but one application, as the Enterprise Resource Planning (ERP) application, they are the cornerstones of most business processes. This is why Level 3 calls for having the log and audit data for an organization's people, processes and events available for correlation and that this log and audit data is safely stored in a centralized repository.

Level 3 is where Application logging and auditing is consolidated and correlated to database logging and auditing. This is especially important if you are using Oracle Apex because Apex is built within the database. If you are using Oracle12c, Level 3 is where Real Application Security (RAS) auditing would be included. Level 3 is also where Fine Grained Auditing (FGA) and Oracle Label Security (OLS) audit logs are also consolidated.

Level 3 is continuous. Once a baseline is established from which alerts and reports are used to report anomalies, as business processes change, tolerances and alerts need to be adjusted to the new baseline. Level 3 is also continuous not just because of the possibilities of what log and audit data can be created by an individual application but because the possibilities for additional correlations and alerts are only limited by the combining of log and audit data from all applications.

Level 3 Tasks

1. Pass additional database logs and application server logs
2. Begin passing functional log and audit data as well as user navigation logs for applications e.g. E-Business Suite, SAP, PeopleSoft, Apex and OBIEE
3. Integrate network logging and infrastructure systems such as:
    a. Enterprise password safe logon, password create and password retrieval
    b. Door security systems
    c. Network logon
    d. VPN activity
4. Automate compliance tasks
5. Create additional alerts

ORACLE 12C UNIFIED AUDIT

If you are using Oracle 12c, Level 3 is where the new Unified Audit functionality is fully utilized. Oracle 12c by default uses 'Mixed Mode' auditing that utilizes all pre-12c log and audit functionality but exposes them through the new `SYS.UNIFIED_AUDIT_TRAIL` view. This new view consolidates all log and audit activity from standard auditing, Label Security and Real Application Security.

Additional Database and Application Server Logs

Each log management or SIEM vendor will have their own set of log parsers and capabilities. The recommendation for Level 3 is to send additional database, and web server logs to assist with additional logging for who is coming into the applications from where and when.

Apache Logs and Web Server Logs

Apache is commonly utilized by Oracle enterprise applications. Apache server logging is defined in the Apache configuration file (HTTPD.CONF) which for the Oracle E-Business Suite is located at $ORA_CONFIG_HOME/10.1.3/Apache/Apache/conf/httpd.conf. For the Oracle E-Business Suite, the Apache logging level is set by the Autoconfig using the parameter 's_apache_log_level'. Integrigy recommends the default log setting of 'warn'.

Apache Log Levels	Description
emerg	Emergencies, system is not useable
alert	Action must be taken
crit	Critical conditions
error	Error conditions
warn	**Warning conditions - Default**
notice	Normal but significant condition
info	Information
debug	Debug level messages

Apache Logs	Location within E-Business Suite Instance Home
HTTP and PLS access logs	$LOG_HOME/ora/10.1.3/Apache/access_log*
HTTP listener error log for both http & pls	$LOG_HOME/ora/10.1.3/Apache/error_log*
Security Logs	$LOG_HOME/ora/10.1.3/Apache/mod_rewrite.log $LOG_HOME/ora/10.1.3/Apache/sec_audit.log $LOG_HOME/ora/10.1.3/Apache/sec_debug.log

Listener Log

The database listener log provides information regarding database connections, for example, IP addresses of clients, and it should be sent to a centralized logging solution. Within the listener's control file ($TNS_ADMIN/listener.ora), confirm that logging is enabled (LOG_STATUS = On) and the location of the listener log (parameter = LOG_DIRECTORY_listener_name).

For information on how to secure the listener what to monitor for, refer to Integrigy's Whitepaper: Oracle Database Listener Security Guide.

Alert Log

Each database has an alert.log. The alert log of a database is a chronological log of messages and errors, including internal errors (ORA-600/7445), corruption errors , and deadlock errors (ORA-60), administrative operations, and SQL*Plus statements STARTUP, SHUTDOWN, ARCHIVE LOG, and RECOVER.

The alert log should be monitored as part of any centralized logging solution. The table below lists several Errors that should be considered for security monitoring.

Code	Message
ORA-29532	Java call terminated by uncaught Java exception
ORA-24247	Network access denied by access control list (ACL)
ORA-28000	The account is locked
ORA-29257	Host unknown
ORA-01031	Insufficient privileges

The location of the alert.log can be found using the either of following SQL -

```
select name, value from v$parameter where name = 'diagnostic_dest';
select * from v$diag_info;
```

Starting with Oracle 11gR2, the Oracle database alert log is available in the `SYS.X$DBGALERTEXT` table. For example -

```
select * from sys.x$dbgalertext where message_text like '%ORA-%' group by
message_text;
```

To monitor just critical errors, starting with 11gR2 the view `V$DIAG_CRITICAL_ERROR` can be used.

```
select * from v$diag_critical_error;
```

Use V$DIAG_ALERT_EXT to Monitor both Alert and Listener Log

Also starting with 11gR2 it is possible monitor both the Alert and Listener logs using the system table `V$DIAG_ALERT_EXT`.

To query just the alert.log:

```
SELECT *
FROM sys.V$DIAG_ALERT_EXT
WHERE TRIM(COMPONENT_ID)='rdbms';
```

To query just the Listener.log:

```
SELECT *
FROM V$DIAG_ALERT_EXT
WHERE TRIM(COMPONENT_ID)='tnslsnr';
```

Application Functional Setup and Configurations

Level 2 focused on system administration. Level 3 focuses on functional setups and those key controls that can be used to support sophisticated security and audit alerts. Level 3 should be complementary to any Government Risk and Compliance (GRC) implementations. GRC and centralized logging (or SIEM) solutions have similarities but serve separate purposes. Integrigy recommends a GRC solution be implemented to satisfy segregation of duties and functional risk and compliance. However, if no GRC implementation exists, then the centralized logging solution can be expanded to meet many risk mitigation needs.

A first effort could be alert on key roles and menus within the applications. For example, within ERP applications such as the Oracle E-Business Suite, there are several menus that are only infrequently used to configure and setup approval rules, cash controls, or credit card encryption. These menus need to be closely monitored. Only appropriate individuals and/or teams should be using these menus.

For example, an alert or a report could be set to flag a logon to the menu for Payments Setup if it is used outside first shift Eastern Time US.

Application	What	Source
Oracle E-Business Suite	User creation and logon activity should be monitored as well as the consolidation of log and audit data.	APPLSYS.FND_USERS APPLSYS.FND_LOGINS APPLSYS.FND_LOGIN_RESPONSIBILITIES APPLSYS.FND_LOGIN_RESP_FORMS APPLSYS.FND_UNSUCCESSFUL_LOGINS ICX.ICX_FAILURES JTF.JTF_PF_SES_ACTIVITY JTF.JTF_PF_ANON_ACTIVITY JTF.JTF_PF_USER_SUMM APPLSYS.WF_USER_ROLE_ASSIGNMENTS APPLSYS.FND_USER_RESP_GROUPS *_A shadow audit tables
PeopleSoft	User creation and logon activity should be monitored as well as the consolidation of log and audit data.	PSACCESSLOG AUDIT_* PSOPRDEFN PSCLASSDEFN
WebLogic	Refer to log4j properties for logging targets	*DOMAIN_NAME*\servers*SERVER_NAME*\logs*SERVER_NAME*.log Also look in the log4j.properties file
Oracle Business Intelligence Enterprise Edition (OBIEE)	Usage tracking can be optionally enabled to monitor query performance, as well as user activities.	S_NQ_ACCT
Oracle Fine Grained Auditing (FGA)	Audit policies for application tables. Conditional logic can be included. FGA complements	SYS.FAG_LOG$

Application	What	Source
	the new Oracle Database 12c Unified Audit by enabling audit conditions to be associated with specific columns.	
Oracle Apex	Apex user creation and activities are logged in several standard tables.	APEX_WORKSPACE_APEX_USERS APEX_WORKSPACE_ACCESS_LOG APEX_WORKSPACE_ACTIVITY_LOG APEX_WORKSPACE_LOG_SUMMARY APEX_WORKSPACE_LOG_SUMMARY_USR
Oracle Label Security	Label Security allows for strict security requirements to be met such as for "Need to know" Mandatory Access Control. Logs are generated according to the policy name.	SYS.DBA_*policyname*_AUDIT_TRAIL
12c Real Application Security	Creation of user accounts should be monitored along with the consolidation of log data.	SYS.DBA_XS_USERS SYS.UNIFIED_AUDIT_TRAIL
Oracle 12c Unified Audit	For Oracle 12c all log and audit data has been consolidated into a single view	SYS.UNIFIED_AUDIT_TRAIL

Other Features of Note

Level 3 of the Framework is continuous and only limited by the permutations and possibilities allowed by the log and audit data. Several standard features of the Oracle database should be kept in mind when considering what alerts and correlations are possible when combining Oracle and application log and audit data.

Client Identifier

Default Oracle database auditing stores the database username but not the application username. In order to pull the application username into the audit logs, the CLIENT IDENTIFIER attribute needs to be set for the application session which is connecting to the database. The CLIENT_IDENTIFIER is a predefined attribute of the built-in application context namespace, USERENV, and can be used to capture the application user name for use with global application context, or it can be used independently.

CLIENT IDENTIFIER is set using the DBMS_SESSION.SET_IDENTIFIER procedure to store the application username. The CLIENT IDENTIFIER attribute is one the same as V$SESSION.CLIENT_IDENTIFIER. Once set you can query V$SESSION or `SELECT SYS_CONTEXT('USERENV','CLIENT_IDENTIFIER') FROM DUAL.`

The table below offers several examples of how CLIENT_IDENTIFIER is used. For each example, for Level 3 alerts, consider how the value of CLIENT_IDENTIFIER could be used along with network usernames, enterprise application username as well as security and electronic door system activity logs.

Oracle CLIENT_IDENTIFIER	
Application	**Example of how used**
E-Business Suite	As of Release 12, the Oracle E-Business Suite automatically sets and updates CLIENT_IDENTIFIER to the FND_USER.USERNAME of the user logged on. Prior to Release 12, follow Support Note How to add DBMS_SESSION.SET_IDENTIFIER(FND_GLOBAL.USER_NAME) to FND_GLOBAL.APPS_INITIALIZE procedure (Doc ID 1130254.1)
PeopleSoft	Starting with PeopleTools 8.50, the PSOPRID is now additionally set in the Oracle database CLIENT_IDENTIFIER attribute.
SAP	With SAP version 7.10 above, the SAP user name is stored in the CLIENT_IDENTIFIER.
Oracle Business Intelligence Enterprise Edition(OBIEE)	When querying an Oracle database using OBIEE the connection pool username is passed to the database. To also pass the middle-tier username, set the user identifier on the session. To do this in OBIEE, open the RPD, edit the connection pool settings and create a new connection script to run at connect time. Add the following line to the connect script: CALL DBMS_SESSION.SET_IDENTIFIER('VALUEOF(NQ_SESSION.USER)')

Database Link Usage

A database link is a one-way connection between two databases. Starting with Oracle version 11.2.0.3, database session information now can report additional information for those sessions involving database links. As often database links are created between databases of different security profiles; it is important to note that it is now able to log session activity that includes the details of the database link:

DBLINK_INFO returns the source of a database link. Specifically, it returns a string of the form:

SOURCE_GLOBAL_NAME=*dblink_src_global_name*, DBLINK_NAME=*dblink_name*,
SOURCE_AUDIT_SESSIONID=*dblink_src_audit_sessionid*
where:

- *dblink_src_global_name* is the unique global name of the source database
- *dblink_name* is the name of the database link on the source database
- *dblink_src_audit_sessionid* is the audit session ID of the session on the source database that initiated the connection to the remote database using *dblink_name*

You can verify DBLINK_INFO:

- Oracle 12c provides a DBLINK_INFO column in `SYS.UNIFIED_AUDIT_TRAIL`.
- `SELECT SYS_CONTEXT('USERENV','DBLINK_INFO') FROM DUAL`

Last Login

Tracking when database users last logged in is a common compliance requirement. This is required in order to reconcile users and cull stale users. New with Oracle12c, Oracle provides this information for database users. The system table `SYS.DBA_USERS` has a column, last_login.

Example:

```
select username, account_status, common, last_login
from sys.dba_users
order by last_login asc;
```

Username	Account_Status	Common	Last_Login
C##INTEGRIGY	OPEN	YES	05-AUG-14 12.46.52.000000000 PM AMERICA/NEW_YORK
C##INTEGRIGY_TEST_2	OPEN	YES	02-SEP-14 12.29.04.000000000 PM AMERICA/NEW_YORK
XS$NULL	EXPIRED & LOCKED	YES	02-SEP-14 12.35.56.000000000 PM AMERICA/NEW_YORK
SYSTEM	OPEN	YES	04-SEP-14 05.03.53.000000000 PM AMERICA/NEW_YORK

AUTOMATE COMPLIANCE TASKS

Throughout this document, the recommended logging alerts are all able to be mapped back to PCI, HIPAA, NIST 800-53, IS0 27000, and SOX (COBIT). By building automated alerts, staff members do not need to monitor manually and need only to review and confirm. This should largely automate compliance tasks; however, each client will have their own unique compliance requirements.

ADDITIONAL ALERTS

As with Levels 1 and 2, the table below is by no means conclusive. Simple things can trigger serious high-risk security events. As such, the table below should be seen as much as a starting point as it is an educational tool. What to monitor for and whom to notify will largely be determined by each client's unique risk profile.

APPENDIX A – EXAMPLE LEVEL 1 AUDIT SCRIPT

/*
Integrigy Log and Audit Framework Level 1 Script
Oracle Database security check

Version 28 – August 12, 2014

Disclaimer:
As with any script provided by a third-party, this script should be reviewed prior to running in a production environment. With any script, there is always a risk of causing unintended performance or availability issues. This script is provided as-is and is intended for instructional purposes only.

Instructions:
Execute this script using a SQL tool like SQL*Plus, SQL Developer or TOAD.

*/

```
Audit session;
Audit user;
Audit role;
Audit database link;
Audit public database link;
Audit alter system;
Audit alter database;
Audit system grant;
Audit profile;
Audit grant directory;
Audit grant procedure;
Audit grant sequence;
Audit grant table;
Audit grant type;
Audit system audit;
Audit sysdba;
Audit sysoper;
/
```

REFERENCES

GENERAL

- "Building an Audit Trail in an Oracle Applications Environment," Jeff Hare and Stephen Kost, http://www.integrigy.com/files/Building_an_Audit_Trail_in_an_Oracle_Applications_Environment.pdf
- "Real World Database Auditing," Stephen Kost, Collaborate 2009, Session #602, http://www.integrigy.com/files/IOUG%202009%20-%20Real%20World%20Database%20Auditing.pdf
- "Applied Oracle Security," Knox et al., Oracle Press, 2010
- "Protecting Oracle Database 12c," Paul Wright, Apress 2014

ORACLE DOCUMENTATION

- "Oracle Database Security Guide 12c Release 1 (12.2) E48135-09," Oracle Corporation, July 2014 http://docs.oracle.com/database/121/DBSEG/E48135-09.pdf
- "Oracle Database Reference 12c Release 1 (12.1) E41527-12," Oracle Corporation, August 2014 http://docs.oracle.com/database/121/REFRN/refrn23884.htm#REFRN23884
- "Security and Compliance with Oracle Database 12c," Oracle Corporation, April 2014, http://docs.oracle.com/cd/E23574_01/server.103/e16813.pdf
- "Oracle Database Concepts 12*c* Release 1 (12.1) E41396-10" , Oracle Corporation, August 2014 http://docs.oracle.com/database/121/CNCPT/E41396-10.pdf
- "Oracle Database Real Application Security Administrator's and Developer's Guide 12c Release 1 (12.1) E48189-08," Oracle Corporation, July 2014 http://docs.oracle.com/database/121/DBFSG/E48189-08.pdf

ORACLE SUPPORT

- "Troubleshooting (Audit Trail)," Oracle Support Note ID 105624.1, Oracle Corporation, 10 December 2013, https://support.oracle.com/rs?type=doc&id=105624.1
- "Auditing How To, Troubleshooting, and Error Message Document," Oracle Support Note ID 1579731.1, Oracle Corporation, 3 September 2013, https://support.oracle.com/rs?type=doc&id=1579731.1
- "Master Note For Oracle Database Auditing," Oracle Support Note ID 1299033.1, Oracle Corporation, 7 January 2014, https://support.oracle.com/rs?type=doc&id=1299033.1
- "Master Note for Oracle Database Fine-Grained Auditing," Oracle Support Note ID 1533543.1, Oracle Corporation, 25 April 2013, https://support.oracle.com/rs?type=doc&id=1533543.1

ABOUT INTEGRIGY

Integrigy Corporation (www.integrigy.com)

Integrigy Corporation is a leader in application security for enterprise mission-critical applications. AppSentry, our application and database security assessment tool, assists companies in securing their largest and most important applications through detailed security audits and actionable recommendations. AppDefend, our enterprise web application firewall is specifically designed for the Oracle E-Business Suite. Integrigy Consulting offers comprehensive security assessment services for leading databases and ERP applications, enabling companies to leverage our in-depth knowledge of this significant threat to business operations.

Integrigy Corporation
P.O. Box 81545
Chicago, Illinois 60681 USA
888/542-4802
www.integrigy.com