Auditing for Intrusion Detection on Oracle

David Litchfield
david@davidlitchfield.com
January 2017

Purpose

This document details what actions and events should be audited for non-SYS connections on an Oracle database server to detect when an intrusion may have occurred. These audit options have been chosen so as to provide alerts in the event of a breach whilst minimizing impact on database performance.

Overview

The events and actions that should be audited as listed in this document should occur so infrequently on a production system, if even at all, that there should be no measurable impact on database performance and configuring auditing as described will not inundate the logging system with benign or spurious entries. For many of these audit options, only in the event of an actual breach would you see a large number of audit records being created.

These audit options have been chosen because they are known attack methods and have been used in real world breaches or in exploit tools in the public domain that are used to compromise Oracle database servers. Of course, some of these audit options have been chosen because security relevant changes to the database should be audited even if they've not been observed in real world attacks; there's nothing to say an attacker might take such actions.

Authentication attacks

Brute force attacks, where an attacker attempts to guess a user's password, will result in many failed log in attempts (or up to the lockout threshold if account lock out is enabled), assuming they don't guess the password! If they do guess they password we'd need to know about it – this is recognizable as a login success after many failed attempts. Execute

```
AUDIT CREATE SESSION
```

to capture both failed and successful logins. Note that, if a password rotation occurs, and applications are not updated with the correct password, a stream of failed login attempts can occur so many failed logins may not necessarily be indicative of an attack.

Please note that if an attacker is attempting to guess a database SID this will not be captured at the database level; however, such an attack would be captured in the TNS log file.

Auditing Failures

Whether due to an attacker's typo, missing privileges or the object not actually being present, errors should be captured because on a well configured production system, errors should be rare. Execute

```
AUDIT NOT EXISTS
```

to capture such events. Note – there is a bug in Oracle 12.x and 11.2 where AUDIT NOT EXISTS fails to capture attempts to execute procedures that do not exist (or where a user does not have the privileges to execute it). To capture such events execute

```
AUDIT EXECUTE PROCEDURE WHENEVER NOT SUCCESSFUL
```

This bug can be fixed by installing the April 2016 CPU or later:
http://www.oracle.com/technetwork/security-advisory/cpuapr2016v3-2985753.html

PL/SQL Injection

One of the most common attack vectors used by hackers to attempt to gain elevated privileges is through SQL injection attacks against PL/SQL packages in the database, particularly in the SYS schema. Such attacks follow fairly common patterns and the following audit options attempt to capture these.

Auxiliary Inject Functions

An auxiliary inject function is a function that executes an arbitrary SQL statement and can be injected into a PL/SQL injection vulnerability. The following packages contain functions that can be used to execute arbitrary anonymous PL/SQL blocks or achieve the functional equivalent.

```
DBMS_XMLGEN
DBMS_XMLSTORE
DBMS_XMLSAVE
DBMS_JAVA_TEST
DBMS_AW
KUPP$PROC.CREATE_MASTER_PROCESS
DBMS_REPCAT_RPC
DBMS_SQL
DBMS_SYS_SQL
```

Note, functions on DBMS_SQL and DBMS_SYS_SQL can be used as auxiliary inject functions too, but their use is so prevalent auditing them would likely create thousands of benign entries.

Reference:
http://www.davidlitchfield.com/plsql-injection-create-session.pdf
http://www.davidlitchfield.com/ExploitingPLSQLinOracle11g.pdf
http://www.davidlitchfield.com/cursor-injection.pdf
http://www.davidlitchfield.com/Exploiting_PLSQL_Injection_on_Oracle_12c.pdf
http://www.davidlitchfield.com/DBMS_XMLSTORE_PLSQL_Injection.pdf
http://www.davidlitchfield.com/ExploitingPLSQLInjectionCREATESESSION.pdf
http://www.davidlitchfield.com/OLAPDMLInjection.pdf

If eBusiness Suite has been installed the following packages also contain functions that can be used to execute arbitrary anonymous PL/SQL blocks:

```
APPS.ASG_CUSTOM_PVT
APPS.WIP_MASS_LOAD_UTILITIES
APPS.MSC_GET_NAME
APPS.BSC_UPDATE_UTIL
APPS.PSB_WS_ACCT1
APPS.OKC_WF
APPS.OKC_P_UTIL
```

Reference: http://www.davidlitchfield.com/oracle-apps-to-sys.pdf

Attackers may create their own PL/SQL objects for use as auxiliary inject functions so the following should be audited:

```
AUDIT CREATE PROCEDURE;
AUDIT CREATE TYPE;
```

I**n Oracle 12c, unfortunately, it is not possible to audit the use of the INHERIT PRIVILEGE system privilege.** Earlier versions of Oracle do not have this privilege.

Exfiltration attacks

The following sections detail what to audit to detect attacks attempting to exfiltrate data.

Out-of-band SQL injection attacks

When attempting to steal data, an attacker may use any number of PL/SQL procedures to send data out of the network via an out-of-band channel, for example by sending data over the web using UTL_HTTP, or over DNS using UTL_INADDR or DBMS_LDAP. Execute on the following packages should be audited:

```
UTL_HTTP
UTL_TCP
UTL_INADDR
UTL_MAIL
UTL_SMTP
HTTPURITYPE
DBMS_LDAP
DBMS_LDAP_API_FFI
XDB.DBMS_XSLPROCESSOR
MDSYS.SDO_NET
```

Reference: The Oracle Hacker's Handbook (pages 145-149)

Database links can be used to exfiltrate data, too, so creation of new links should be audited.

```
AUDIT CREATE DATABASE LINK
AUDIT CREATE PUBLIC DATABASE LINK
```

Whilst it is possible to audit accesses via a database link on the target system it is not possible to audit link usage on the source system.

Time-based blind SQL injection attacks

The DBMS_AW package has a function called INTERP that can execute OLAP DML commands – one of which, SLEEP, can be used in time-based blind SQL injection attacks. The RECEIVE_MESSAGE function on DBMS_PIPE can be used too provided the account has the privileges to execute it – by default PUBLIC does not. The hashing and encryption functions on DBMS_CRYPTO and DBMS_OBFUSCATION_TOOLKIT are often used with multiple rounds to cause time delays. Audit execute on the following packages to detect time based SQL injection attacks:

```
DBMS_AW
DBMS_PIPE
DBMS_CRYPTO
DBMS_OBFUSCATION_TOOLKIT
```

Reference: http://www.davidlitchfield.com/sqlinference.pdf

Indirect Privilege Escalation

There are numerous ways for an attacker to indirectly escalate their privileges. An example of this would be exploiting a PL/SQL injection flaw in an MDSYS owned package and then using MDSYS CREATE ANY TRIGGER privileges to create a trigger on a SYSTEM owned table that PUBLIC insert into – for example SYSTEM.OL$. Many indirect privilege escalation exploits use SYSTEM.OL$, probably because it was the first example given. Update, insert, and delete on the following SYSTEM owned tables should be audited.

```
OL$
OL$NODES
OL$HINTS
MVIEW$_ADV_PARTITION
MVIEW$_ADV_OWB
MVIEW$_ADV_INDEX
```

Whilst there are tables that are owned by SYS that PUBLIC can perform DML on, one can't create triggers on SYS owned tables so this privilege escalation technique is not relevant.

Reference: The Oracle Hacker's Handbook

After gaining elevated privileges, for example after executing GRANT DBA TO PIUBLIC an attacker would need to set the role so this should be audited.

```
AUDIT ROLE
```

Note that an attacker need not execute SET ROLE – they could simply log off and log back on again.

Attempts to execute OS commands
An attacker can load a dynamic link library/shared object that can then call, for example, the system() function in libc/msvcrt.dll. To do this they'd need to execute the CREATE LIBRARY DDL command first. This should be audited.

```
AUDIT CREATE LIBRARY
```

From a Java stored procedure an attacker can run OS commands from Runtime().exec() and load a library to do the same through System.load/loadLibrary()

```
AUDIT EXECUTE ON "java/lang/Runtime"
AUDIT EXECUTE ON "java/lang/System"
```

Note – it is not possible to audit these on Oracle 11.2 but you can on 12c.

OS commands can also be executed by DBMS_SCHEDULER, DBMS_JOB, and DBMS_IJOB.

```
AUDIT EXECUTE ON DBMS_SCHEDULER
AUDIT EXECUTE ON DBMS_JOB
AUDIT EXECUTE ON DBMS_IJOB
AUDIT CREATE JOB;
AUDIT CREATE EXTERNAL JOB;
```

Using ALTER SYSTEM to set the PL/SQL make utility can also be used to execute OS commands so ALTER SYSTEM should be audited.

```
AUDIT ALTER SYSTEM
```

Reference: The Oracle Hacker's Handbook (page 131 – 133)

Attempts to access the OS filesystem.
For PL/SQL Oracle provides access to the filesystem through directory objects and UTL_FILE and DBMS_LOB. Creation of new directory objects should be audited along with usage of UTL_FILE and DBMS_LOB.

```
AUDIT DIRECTORY
AUDIT GRANT DIRECTORY
```

```
AUDIT EXECUTE ON SYS.UTL_FILE
AUDIT EXECUTE ON SYS.DBMS_LOB
```

For each directory object defined execute the following

```
AUDIT READ ON DIRECTORY schema.name
AUDIT WRITE ON DIRECTORY schema.name
```

Unfortunately, there is no system wide audit option for read and write on all directories. This is problematic because if an attacker creates a new directory, read and writes to it cannot be audited.

Java can also be used to read and write to files on the file system. Auditing should be enabled for the `FileReader`, `FileWriter`, `FileInputStream` and `FileOutputStream` Java classes:

```
AUDIT EXECUTE ON "java/io/FileWriter";
AUDIT EXECUTE ON "java/io/FileReader";
AUDIT EXECUTE ON "java/io/FileInputStream";
AUDIT EXECUTE ON "java/io/FileOutputStream";
```

Please note, this may cause an ORA-00701 error: "object necessary for warmstarting database cannot be altered". If this happens, you need to stop and then restart the database server then immediately execute the audit statements.

Index attacks

If a user has the index privilege on a table in another user's schema, or the CREATE ANY INDEX system privilege they may create a function based index on that table and that function would execute with the privilege of the table's owner [1]. As such CREATE INDEX should be audited:

```
AUDIT INDEX
```

Reference: http://www.davidlitchfield.com/Privilege_Escalation_via_Oracle_Indexes.pdf

Backdoors/Silent SQL

Attackers may try to backdoor objects so SQL runs automatically in the background when accesses a table or view or logs on. For example, an attacker can use DBMS_RLS, DBMS_FGA or DBMS_REDACT to have a PL/SQL procedure execute when a given table is selected from. The same can be done with triggers, of course.

```
AUDIT CREATE TRIGGER
AUDIT EXECUTE ON DBMS_RLS
AUDIT EXECUTE ON DBMS_FGA
AUDIT EXECUTE ON DBMS_REDACT
AUDIT EXECUTE ON DBMS_REDACT_INT
```

```
AUDIT EXECUTE ON SA_POLICY_ADMIN
AUDIT EXECUTE ON LBACSYS.LBAC_POLICY_ADMIN
AUDIT EXECUTE ON DBMS_RULE_ADM
AUDIT EXECUTE ON DBMS_STREAMS_ADM
AUDIT CREATE CONTEXT
```

It is also possible to have SQL run silently when a user changes their password by setting a password verify function for a profile. As such, both ALTER, DROP, and CREATE PROFILE should be audited.

```
AUDIT PROFILE;
```

ANY Usage
The following ANY system privilege should be audited:

```
AUDIT SELECT ANY DICTIONARY
AUDIT SELECT ANY TABLE
AUDIT INSERT ANY TABLE
AUDIT DELETE ANY TABLE
AUDIT UPDATE ANY TABLE
AUDIT EXECUTE ANY PROCEDURE
AUDIT EXECUTE ANY TYPE
AUDIT EXECUTE ANY LIBRARY
AUDIT ANALYZE ANY
```

CREATE ANY PROCEDURE is already covered by AUDIT CREATE PROCEDURE discussed earlier. This is also true of TYPE, TRIGGER, VIEW, and TABLE.

Database Modifications
To capture any changes to the system execute: `AUDIT ALTER SYSTEM;`

To capture any granting of system privileges or role membership execute: `AUDIT SYSTEM GRANT;`

To capture the granting of any object privilege execute: `AUDIT GRANT ANY OBJECT PRIVILEGE;`

To capture any changes to users on the system execute: `AUDIT USER;` This covers CREATE, ALTER and DROP.

Note, if a user changes their own password this will not be audited.

Note, if a user creates a new user by executing:

```
GRANT DBA TO SOMENEWUSER IDENTIFIED BY PASSWORD1;
```

then neither AUDIT USER nor AUDIT CREATE USER will not capture this, even though a new user is being created. This is a bug and has been reported to Oracle.

Audit Evasion

An attacker may attempt to stop auditing of their actions and try to turn off auditing; or they may attempt to delete their activities. Capturing the following events will catch this:

```
AUDIT AUDIT SYSTEM;
AUDIT SYSTEM AUDIT;
AUDIT DELETE ON AUD$;
AUDIT DELETE ON FGA_LOG$;
AUDIT EXECUTE ON DBMS_AUDIT_MGMT;
```

Some systems may use triggers as a method of auditing. Attackers may attempt to disable such triggers or drop them entirely, or change them. Audit the following to catch such events:

```
AUDIT TRIGGER;
```

Appendix A

```
-- audit.sql

AUDIT CREATE SESSION BY ACCESS;
AUDIT USER BY ACCESS;
AUDIT NOT EXISTS BY ACCESS;
AUDIT EXECUTE PROCEDURE BY ACCESS WHENEVER NOT SUCCESSFUL;
AUDIT DIRECTORY BY ACCESS;
AUDIT GRANT DIRECTORY BY ACCESS;
AUDIT CREATE LIBRARY BY ACCESS;
AUDIT ALTER SYSTEM BY ACCESS;
AUDIT SYSTEM AUDIT BY ACCESS;
AUDIT CONTEXT BY ACCESS;
AUDIT TRIGGER BY ACCESS;
AUDIT CREATE JOB BY ACCESS;
AUDIT CREATE EXTERNAL JOB BY ACCESS;
AUDIT SELECT ANY DICTIONARY BY ACCESS;
AUDIT SELECT ANY TABLE BY ACCESS;
AUDIT INSERT ANY TABLE BY ACCESS;
AUDIT DELETE ANY TABLE BY ACCESS;
AUDIT UPDATE ANY TABLE BY ACCESS;
AUDIT EXECUTE ANY PROCEDURE BY ACCESS;
AUDIT EXECUTE ANY TYPE BY ACCESS;
AUDIT EXECUTE ANY LIBRARY BY ACCESS;
AUDIT ANALYZE ANY BY ACCESS;
AUDIT INDEX BY ACCESS;
AUDIT CREATE PROCEDURE BY ACCESS;
AUDIT CREATE TABLE BY ACCESS;
AUDIT ALTER TABLE BY ACCESS;
```

```
AUDIT CREATE VIEW BY ACCESS;
AUDIT CREATE TYPE BY ACCESS;
AUDIT PROFILE BY ACCESS;
AUDIT SYSTEM GRANT BY ACCESS;
AUDIT GRANT ANY OBJECT PRIVILEGE BY ACCESS;
AUDIT ROLE BY ACCESS;
AUDIT EXEMPT ACCESS POLICY BY ACCESS;
AUDIT CREATE DATABASE LINK BY ACCESS;
AUDIT CREATE PUBLIC DATABASE LINK BY ACCESS;
-- The following 6 may error if the database
-- has not been recently restarted
AUDIT EXECUTE ON "java/lang/Runtime" BY ACCESS;
AUDIT EXECUTE ON "java/lang/System" BY ACCESS;
AUDIT EXECUTE ON "java/io/FileWriter" BY ACCESS;
AUDIT EXECUTE ON "java/io/FileReader" BY ACCESS;
AUDIT EXECUTE ON "java/io/FileInputStream" BY ACCESS;
AUDIT EXECUTE ON "java/io/FileOutputStream" BY ACCESS;
AUDIT EXECUTE ON SYS.DBMS_SCHEDULER BY ACCESS;
AUDIT EXECUTE ON SYS.DBMS_JOB BY ACCESS;
AUDIT EXECUTE ON SYS.DBMS_IJOB BY ACCESS;
AUDIT EXECUTE ON SYS.KUPP$PROC BY ACCESS;
AUDIT EXECUTE ON SYS.UTL_HTTP BY ACCESS;
AUDIT EXECUTE ON SYS.UTL_TCP BY ACCESS;
AUDIT EXECUTE ON SYS.UTL_INADDR BY ACCESS;
-- This may error if the package has not been installed
AUDIT EXECUTE ON SYS.UTL_MAIL BY ACCESS;
AUDIT EXECUTE ON SYS.UTL_SMTP BY ACCESS;
AUDIT EXECUTE ON SYS.HTTPURITYPE BY ACCESS;
AUDIT EXECUTE ON SYS.DBMS_LDAP BY ACCESS;
AUDIT EXECUTE ON SYS.DBMS_LDAP_API_FFI BY ACCESS;
AUDIT EXECUTE ON XDB.DBMS_XSLPROCESSOR BY ACCESS;
-- This may error if the package has not been installed
AUDIT EXECUTE ON MDSYS.SDO_NET BY ACCESS;
AUDIT EXECUTE ON SYS.DBMS_SQL BY ACCESS;
AUDIT EXECUTE ON SYS.DBMS_SYS_SQL BY ACCESS;
AUDIT EXECUTE ON SYS.DBMS_AW BY ACCESS;
AUDIT EXECUTE ON SYS.DBMS_XMLGEN BY ACCESS;
AUDIT EXECUTE ON SYS.DBMS_XMLSTORE BY ACCESS;
AUDIT EXECUTE ON SYS.DBMS_XMLSAVE BY ACCESS;
AUDIT EXECUTE ON SYS.DBMS_SQLHASH BY ACCESS;
AUDIT EXECUTE ON SYS.DBMS_REPCAT_RPC BY ACCESS;
AUDIT EXECUTE ON SYS.UTL_FILE BY ACCESS;
AUDIT EXECUTE ON SYS.DBMS_LOB BY ACCESS;
AUDIT EXECUTE ON SYS.DBMS_PIPE BY ACCESS;
AUDIT EXECUTE ON SYS.DBMS_CRYPTO BY ACCESS;
AUDIT EXECUTE ON SYS.DBMS_OBFUSCATION_TOOLKIT BY ACCESS;
AUDIT EXECUTE ON DBMS_RLS BY ACCESS;
AUDIT EXECUTE ON DBMS_FGA BY ACCESS;
AUDIT EXECUTE ON DBMS_REDACT BY ACCESS;
AUDIT EXECUTE ON DBMS_REDACT_INT BY ACCESS;
-- This may error if the package has not been installed
AUDIT EXECUTE ON LBACSYS.LBAC_POLICY_ADMIN BY ACCESS;
AUDIT EXECUTE ON DBMS_RULE_ADM BY ACCESS;
AUDIT EXECUTE ON DBMS_STREAMS_ADM BY ACCESS;
AUDIT EXECUTE ON DBMS_AUDIT_MGMT BY ACCESS;
AUDIT DELETE ON AUD$ BY ACCESS;
AUDIT DELETE ON FGA_LOG$ BY ACCESS;
```

```
AUDIT INSERT, DELETE, UPDATE ON SYSTEM.OL$ BY ACCESS;
AUDIT INSERT, DELETE, UPDATE ON SYSTEM.OL$NODES BY ACCESS;
AUDIT INSERT, DELETE, UPDATE ON SYSTEM.OL$HINTS BY ACCESS;
-- These will error on 12c
AUDIT INSERT, DELETE, UPDATE ON SYSTEM.MVIEW$_ADV_PARTITION BY ACCESS;
AUDIT INSERT, DELETE, UPDATE ON SYSTEM.MVIEW$_ADV_OWB BY ACCESS;
AUDIT INSERT, DELETE, UPDATE ON SYSTEM.MVIEW$_ADV_INDEX BY ACCESS;
```

```
-- verify_audit_options.sql

SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'CREATE SESSION' AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'USER' AND SUCCESS = 'BY ACCESS' AND FAILURE
= 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'NOT EXISTS' AND SUCCESS = 'BY ACCESS' AND
FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'EXECUTE PROCEDURE' AND SUCCESS = 'NOT SET'
AND FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'DIRECTORY' AND SUCCESS = 'BY ACCESS' AND
FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'GRANT DIRECTORY' AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'CREATE LIBRARY' AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'ALTER SYSTEM' AND SUCCESS = 'BY ACCESS' AND
FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'SYSTEM AUDIT' AND SUCCESS = 'BY ACCESS' AND
FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'CONTEXT' AND SUCCESS = 'BY ACCESS' AND
FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'CREATE JOB' AND SUCCESS = 'BY ACCESS' AND
FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'CREATE EXTERNAL JOB' AND SUCCESS = 'BY
ACCESS' AND FAILURE = 'BY ACCESS';
```

```
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'TRIGGER' AND SUCCESS = 'BY ACCESS' AND
FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'SELECT ANY DICTIONARY' AND SUCCESS = 'BY
ACCESS' AND FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'SELECT ANY TABLE' AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'INSERT ANY TABLE' AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'DELETE ANY TABLE' AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'UPDATE ANY TABLE' AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'EXECUTE ANY PROCEDURE' AND SUCCESS = 'BY
ACCESS' AND FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'EXECUTE ANY TYPE' AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'EXECUTE ANY LIBRARY' AND SUCCESS = 'BY
ACCESS' AND FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'ANALYZE ANY' AND SUCCESS = 'BY ACCESS' AND
FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'INDEX' AND SUCCESS = 'BY ACCESS' AND FAILURE
= 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'CREATE PROCEDURE' AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'CREATE TABLE' AND SUCCESS = 'BY ACCESS' AND
FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'ALTER TABLE' AND SUCCESS = 'BY ACCESS' AND
FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'CREATE VIEW' AND SUCCESS = 'BY ACCESS' AND
FAILURE = 'BY ACCESS';
```

```
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'CREATE TYPE' AND SUCCESS = 'BY ACCESS' AND
FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'PROFILE' AND SUCCESS = 'BY ACCESS' AND
FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'SYSTEM GRANT' AND SUCCESS = 'BY ACCESS' AND
FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'GRANT ANY OBJECT PRIVILEGE' AND SUCCESS =
'BY ACCESS' AND FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'ROLE' AND SUCCESS = 'BY ACCESS' AND FAILURE
= 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'EXEMPT ACCESS POLICY' AND SUCCESS = 'BY
ACCESS' AND FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'CREATE DATABASE LINK' AND SUCCESS = 'BY
ACCESS' AND FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_STMT_AUDIT_OPTS WHERE USER_NAME IS NULL
AND AUDIT_OPTION = 'CREATE PUBLIC DATABASE LINK' AND SUCCESS =
'BY ACCESS' AND FAILURE = 'BY ACCESS';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'java/io/FileWriter' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'java/io/FileReader' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'java/io/FileInputStream' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'java/io/FileOutputStream' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'java/lang/Runtime' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'java/lang/System' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_SCHEDULER' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_JOB' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_IJOB' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'KUPP$PROC' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'UTL_HTTP' AND EXE = 'A/A';
```

```
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'UTL_TCP' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'UTL_INADDR' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'UTL_SMTP' AND EXE = 'A/A';
-- This may return 0 if the package has not been installed
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'UTL_MAIL' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'HTTPURITYPE' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_LDAP' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_LDAP_API_FFI' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'XDB' AND
OBJECT_NAME = 'DBMS_XSLPROCESSOR' AND EXE = 'A/A';
-- This may return 0 if the package has not been installed
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'MDSYS'
AND OBJECT_NAME = 'SDO_NET' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_SQL' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_SYS_SQL' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_AW' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_XMLGEN' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_XMLSAVE' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_XMLSTORE' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_SQLHASH' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_REPCAT_RPC' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'UTL_FILE' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_LOB' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_PIPE' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_CRYPTO' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_OBFUSCATION_TOOLKIT' AND EXE = 'A/A';
```

```
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_RLS' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_FGA' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_REDACT' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_REDACT_INT' AND EXE = 'A/A';
-- This may return 0 if the package has not been installed
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'LBACSYS'
AND OBJECT_NAME = 'LBAC_POLICY_ADMIN' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_RULE_ADM' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_STREAMS_ADM' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'DBMS_AUDIT_MGMT' AND EXE = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'AUD$' AND DEL = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYS' AND
OBJECT_NAME = 'FGA_LOG$' AND DEL = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYSTEM'
AND OBJECT_NAME = 'OL$' AND DEL = 'A/A' AND INS = 'A/A' AND UPD
= 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYSTEM'
AND OBJECT_NAME = 'OL$NODES' AND DEL = 'A/A' AND INS = 'A/A' AND
UPD = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYSTEM'
AND OBJECT_NAME = 'OL$HINTS' AND DEL = 'A/A' AND INS = 'A/A' AND
UPD = 'A/A';
-- On 12c these will return 0
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYSTEM'
AND OBJECT_NAME = 'MVIEW$_ADV_PARTITION' AND DEL = 'A/A' AND INS
= 'A/A' AND UPD = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYSTEM'
AND OBJECT_NAME = 'MVIEW$_ADV_OWB' AND DEL = 'A/A' AND INS =
'A/A' AND UPD = 'A/A';
SELECT COUNT(*) FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER = 'SYSTEM'
AND OBJECT_NAME = 'MVIEW$_ADV_INDEX' AND DEL = 'A/A' AND INS =
'A/A' AND UPD = 'A/A';
```