Center for Internet Security®

CIS Oracle Database 12c Benchmark

v2.0.0 - 12-28-2016

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Public License. The link to the license terms can be found at https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode

To further clarify the Creative Commons license related to CIS Benchmark content, you are authorized to copy and redistribute the content for use by you, within your organization and outside your organization for non-commercial purposes only, provided that (i) appropriate credit is given to CIS, (ii) a link to the license is provided. Additionally, if you remix, transform or build upon the CIS Benchmark(s), you may only distribute the modified materials if they are subject to the same license terms as the original Benchmark license and your derivative will no longer be a CIS Benchmark. Commercial use of CIS Benchmarks is subject to the prior approval of the Center for Internet Security.

Table of Contents

Overview

This document is intended to address the recommended security settings for Oracle Database 12c. This guide was tested against Oracle Database 12c (version 12.1.0.2) installed without pluggable database support running on a Windows Server 2012 R2 instance as a stand-alone system, and running on an Oracle Linux 7 instance also as a stand-alone system. Future Oracle Database 12c critical patch updates (CPUs) may impact the recommendations included in this document.

To obtain the latest version of this guide, please visit http://benchmarks.cisecurity.org. If you have questions, comments, or have identified ways to improve this guide, please write us at feedback@cisecurity.org.

Intended Audience

This benchmark is intended for system and application administrators, security specialists, auditors, help desk, and platform deployment personnel who plan to develop, deploy, assess, or secure solutions that incorporate Oracle Database 12c on Oracle Linux or Microsoft Windows Server.

Consensus Guidance

This benchmark was created using a consensus review process comprised of subject matter experts. Consensus participants provide perspective from a diverse set of backgrounds including consulting, software development, audit and compliance, security research, operations, government, and legal.

Each CIS benchmark undergoes two phases of consensus review. The first phase occurs during initial benchmark development. During this phase, subject matter experts convene to discuss, create, and test working drafts of the benchmark. This discussion occurs until consensus has been reached on benchmark recommendations. The second phase begins after the benchmark has been published. During this phase, all feedback provided by the Internet community is reviewed by the consensus team for incorporation in the benchmark. If you are interested in participating in the consensus process, please visit https://community.cisecurity.org.

Typographical Conventions

The following typographical conventions are used throughout this guide:

Convention	Meaning
`Stylized Monospace font`	Used for blocks of code, command, and script examples. Text should be interpreted exactly as presented.
Monospace font	Used for inline code, commands, or examples. Text should be interpreted exactly as presented.
<italic font in brackets>	Italic texts set in angle brackets denote a variable requiring substitution for a real value.
Italic font	Used to denote the title of a book, article, or other publication.
Note	Additional information or caveats

Scoring Information

A scoring status indicates whether compliance with the given recommendation impacts the assessed target's benchmark score. The following scoring statuses are used in this benchmark:

Scored

Failure to comply with "Scored" recommendations will decrease the final benchmark score. Compliance with "Scored" recommendations will increase the final benchmark score.

Not Scored

Failure to comply with "Not Scored" recommendations will not decrease the final benchmark score. Compliance with "Not Scored" recommendations will not increase the final benchmark score.

Profile Definitions

The following configuration profiles are defined by this Benchmark:

- **Level 1 - RDBMS using Traditional Auditing**

  Items in this profile apply to Oracle Database 12c configured to use Traditional Auditing and intend to:

  - Be practical and prudent;
  - Provide a clear security benefit; and
  - Not inhibit the utility of the technology beyond acceptable means.

- **Level 1 - Linux Host OS using Traditional Auditing**

  Items in this profile apply to Linux Host operating systems with Oracle Database 12c configured to use Traditional Auditing and intend to:

  - Be practical and prudent;
  - Provide a clear security benefit; and
  - Not inhibit the utility of the technology beyond acceptable means.

- **Level 1 - Windows Server Host OS using Traditional Auditing**

  Items in this profile apply to Windows Server operating systems with Oracle Database 12c configured to use Traditional Auditing and intend to:

  - Be practical and prudent;
  - Provide a clear security benefit; and
  - Not inhibit the utility of the technology beyond acceptable means.

- **Level 1 - RDBMS using Unified Auditing**

  Items in this profile apply to Oracle Database 12c configured to use Unified Auditing and intend to:

  - Be practical and prudent;
  - Provide a clear security benefit; and
  - Not inhibit the utility of the technology beyond acceptable means.

- **Level 1 - Linux Host OS using Unified Auditing**

  Items in this profile apply to Linux Host operating systems with Oracle Database 12c configured to use Unified Auditing and intend to:

  - Be practical and prudent;
  - Provide a clear security benefit; and
  - Not inhibit the utility of the technology beyond acceptable means.

- **Level 1 - Windows Server Host OS using Unified Auditing**

  Items in this profile apply to Windows Server operating systems with Oracle Database 12c configured to use Unified Auditing and intend to:

  - Be practical and prudent;
  - Provide a clear security benefit; and
  - Not inhibit the utility of the technology beyond acceptable means.

Acknowledgements

This benchmark exemplifies the great things a community of users, vendors, and subject matter experts can accomplish through consensus collaboration. The CIS community thanks the entire consensus team with special recognition to the following individuals who contributed greatly to the creation of this guide:

Contributor
Kyle Thomason
Justin Brown
Gijs Hasselman
Stephen Dufour
Alexander Kornbrust
S. Brian Suddeth
Pieter Van Puymbroeck
Arman Rawls
Adam Montville
Timothy Harrison
Tung Bui Viet
Jignesh Patel
Than Thi Cham
Vu Dao Quang
Quan Bui
Youfeng Shen
Ole-Andre Jørgensen
Dean Lackey

Editor
Angelo Marcotullio
Jay Mehta

Recommendations

1 Oracle Database Installation and Patching Requirements

One of the best ways to ensure secure Oracle security is to implement Critical Patch Updates (CPUs) as they come out, along with any applicable OS patches that will not interfere with system operations. It is additionally prudent to remove Oracle sample data from production environments.

1.1 Ensure the Appropriate Version/Patches for Oracle Software Is Installed (Not Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle installation version, along with the patch level, should be the most recent that is compatible with the organizations' operational needs.

Rationale:

As using the most recent Oracle database software, along with all applicable patches can help limit the possibilities for vulnerabilities in the software, the installation version and/or patches applied during setup should be established according to the needs of the organization. Ensure you are using a release that is covered by a level of support that includes the generation of Critical Patch Updates.

Audit:

To assess this recommendation, use the following example shell command as appropriate for your environment.

For example, on Unix/Linux systems:

```
opatch lsinventory | grep -e "^.*<latest_patch_version_numer>\s*.*$"
```

For example on Windows systems:

```
opatch lsinventory | find "<latest_patch_version_number>"
```

Remediation:

Download and apply the latest quarterly Critical Patch Update patches.

References:

1. http://www.oracle.com/us/support/assurance/fixing-policies/index.html
2. http://www.oracle.com/technetwork/topics/security/alerts-086861.html
3. http://www.oracle.com/us/support/library/lifetime-support-technology-069183.pdf

1.2 Ensure All Default Passwords Are Changed (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle installation has a view called `DBA_USERS_WITH_DEFPWD`, which keeps a list of all database users making use of default passwords.

Rationale:

Default passwords should be considered "well known" to attackers. Consequently, if default passwords remain in place any attacker with access to the database then has the ability to authenticate as the user with that default password. When default passwords are altered, this circumstance is mitigated.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT USERNAME
FROM DBA_USERS_WITH_DEFPWD
WHERE USERNAME NOT LIKE '%XS$NULL%';
```

The assessment fails if results are returned.

Remediation:

To remediate this recommendation, you may perform either of the following actions.

- Manually issue the following SQL statement for each USERNAME returned in the Audit Procedure:

```
 PASSWORD <username>
```

- Execute the following SQL script to randomly assign passwords:

```
begin
  for r_user in
    (select username from dba_users_with_defpwd where username not like '%XS$NULL%')
    loop
      DBMS_OUTPUT.PUT_LINE('Password for user '||r_user.username||' will be
changed.');
      execute immediate 'alter user "'||r_user.username||'" identified by
"'||DBMS_RANDOM.string('a',16)||'"account lock password expire';
    end loop;
end;
/
```

References:

1. http://docs.oracle.com/database/121/TDPSG/GUID-3EC7A894-D620-4497-AFB1-64EB8C33D854.htm#TDPSG20021

1.3 Ensure All Sample Data And Users Have Been Removed (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

Oracle sample schemas are not needed for the operation of the database. These include, among others, information pertaining to a sample schemas pertaining to Human Resources, Business Intelligence, Order Entry, and the like. These samples create sample users (`BI`,`HR`,`OE`,`PM`,`IX`,`SH`, `SCOTT`), in addition to tables and fictitious data.

Rationale:

The sample data is typically not required for production operations of the database and provides users with well-known default passwords, particular views, and procedures/functions. Such users, views, and/or procedures/functions could be used to launch exploits against production environments.

Audit:

To assess this recommendation, check for the presence of Oracle sample users by executing the following SQL statement.

```
SELECT USERNAME
FROM ALL_USERS
WHERE USERNAME IN ('BI','HR','IX','OE','PM','SCOTT','SH');
```

Remediation:

To remediate this setting, it is recommended that you execute the following SQL script.

```
$ORACLE_HOME/demo/schema/drop_sch.sql
```

Then, execute the following SQL statement.

```
DROP USER SCOTT CASCADE;
```

NOTE: The `recyclebin` is not set to `OFF` within the default drop script, which means that the data will still be present in your environment until the `recyclebin` is emptied.

Impact:

The Oracle sample user names may be in use on a production basis. It is important that you first verify that `BI`, `HR`, `IX`, `OE`, `PM`, `SCOTT`, and/or `SH` are not valid production user names before executing the dropping SQL scripts. This may be particularly true with the `HR` and `BI` users. **If any of these users are present, it is important to be cautious and confirm the schemas present are, in fact, Oracle sample schemas and not production schemas being relied upon by business operations.**

References:

1. http://docs.oracle.com/database/121/COMSC/toc.htm

2 Oracle Parameter Settings

The operation of the Oracle database instance is governed by numerous parameters that are set in specific configuration files and are instance-specific in scope. As alterations of these parameters can cause problems ranging from denial-of-service to theft of proprietary information, these configurations should be carefully considered and maintained.

Note:

For all files that have parameters that can be modified with the OS and/or SQL commands/scripts, these will both be listed where appropriate.

2.1 Listener Settings

Settings for the TNS Listener `listener.ora` file.

2.1.1 Ensure 'SECURE_CONTROL_<listener_name>' Is Set In 'listener.ora' (Scored)

Profile Applicability:

- Level 1 - Linux Host OS using Traditional Auditing
- Level 1 - Windows Server Host OS using Traditional Auditing

Description:

The `SECURE_CONTROL_`<*listener_name*> setting determines the type of control connection the Oracle server requires for remote configuration of the listener.

Rationale:

As listener configuration changes via unencrypted remote connections can result in unauthorized users sniffing the control configuration information from the network, these control values should be set according to the needs of the organization.

Audit:

To audit this recommendation, follow these steps:

- Open the `$ORACLE_HOME/network/admin/listener.ora` file (or `%ORACLE_HOME%\network\admin\listener.ora` on Windows)
- Ensure that each defined listener as an associated `SECURE_CONTROL_`<*listener_name*> directive.

For example:

```
LISTENER1 =
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=TCP) (HOST=sales-server)(PORT=1521))
    (ADDRESS=(PROTOCOL=IPC) (KEY=REGISTER))
    (ADDRESS=(PROTOCOL=TCPS) (HOST=sales-server)(PORT=1522)))
  SECURE_CONTROL_LISTENER1=TCPS
```

Remediation:

Set the `SECURE_CONTROL_`<*listener_name*> for each defined listener in the `listener.ora` file, according to the needs of the organization.

References:

1. http://docs.oracle.com/database/121/NETRF/listener.htm#NETRF327

2.1.2 Ensure 'extproc' Is Not Present in 'listener.ora' (Scored)

Profile Applicability:

- Level 1 - Linux Host OS using Traditional Auditing
- Level 1 - Windows Server Host OS using Traditional Auditing

Description:

Oracle `extproc` allows the database to run procedures from operating system libraries. These library calls can, in turn, run any operating system command.

Rationale:

`extproc` should be removed from the `listener.ora` to mitigate the risk that OS libraries can be invoked by the Oracle instance.

Audit:

To audit this recommendation, execute the following shell commands as appropriate for your Unix/Windows environment.

Unix environment:

```
 grep -i extproc $ORACLE_HOME/network/admin/listener.ora
```

Windows environment:

```
find /I extproc %ORACLE_HOME%\network\admin\listener.ora
```

Ensure `extproc` does not exist.

Remediation:

Remove `extproc` from the `listener.ora` file.

References:

1. http://docs.oracle.com/database/121/DBSEG/app_devs.htm#DBSEG656

2.1.3 Ensure 'ADMIN_RESTRICTIONS_<listener_name>' Is Set to 'ON' (Scored)

Profile Applicability:

- Level 1 - Linux Host OS using Traditional Auditing
- Level 1 - Windows Server Host OS using Traditional Auditing

Description:

The `admin_restrictions_<listener_name>` setting in the `listener.ora` file can require that any attempted real-time alteration of the parameters in the `listener` via the `set` command file be refused unless the `listener.ora` file is manually altered then restarted by a privileged user.

Rationale:

As blocking unprivileged users from making alterations of the `listener.ora` file, where remote data/services are specified, will help protect data confidentiality, this value should be set to the needs of the organization.

Audit:

To audit this recommendation, execute the following shell commands as appropriate for your Unix/Windows environment.

Unix environment:

```
grep -i admin_restrictions $ORACLE_HOME/network/admin/listener.ora
```

Windows environment:

```
find /I admin_restrictions %ORACLE_HOME%|\network\admin\listener.ora
```

Ensure `ADMIN_RESTRICTIONS_<listener_name>` is set to ON for all listeners.

Remediation:

Use a text editor such as `vi` to set the `ADMIN_RESTRICTIONS_<listener_name>` to the value `ON`.

Default Value:

Not set.

References:

1. http://docs.oracle.com/database/121/NETRF/listener.htm#NETRF310

2.1.4 Ensure 'SECURE_REGISTER_<listener_name>' Is Set to 'TCPS' or 'IPC' (Scored)

Profile Applicability:

- Level 1 - Linux Host OS using Traditional Auditing
- Level 1 - Windows Server Host OS using Traditional Auditing

Description:

The `SECURE_REGISTER_<`*listener_name*`>` setting specifies the protocols which are used to connect to the TNS listener.

Rationale:

As listener configuration changes via unencrypted remote connections can result in unauthorized users sniffing the control configuration information from the network, these control values should be set according to the needs of the organization.

Audit:

To audit this recommendation, execute the following shell commands as appropriate for your Unix/Windows environment.

Unix environment:

```
grep -i SECURE_REGISTER $ORACLE_HOME/network/admin/listener.ora
```

Windows environment:

```
find /I SECURE_REGISTER %ORACLE_HOME%\network\admin\listener.ora
```

Ensure `SECURE_REGISTER_<`*listener_name*`>` is set to `TCPS` or `IPC`.

Remediation:

Use a text editor such as `vi` to set the `SECURE_REGISTER_<`*listener_name*`>=TCPS` or `SECURE_REGISTER_<`*listener_name*`>=IPC` for each listener found in `$ORACLE_HOME/network/admin/listener.ora`.

References:

1. http://docs.oracle.com/database/121/NETRF/listener.htm#NETRF328
2. https://support.oracle.com/epmos/faces/ui/km/DocumentDisplay.jspx?id=1453883.1
3. https://support.oracle.com/epmos/faces/ui/km/DocumentDisplay.jspx?id=1340831.1
4. http://www.joxeankoret.com/download/tnspoison.pdf

Notes:

Oracle Real Application Cluster require a different approach to fix the TNS Poisoning problem. See Oracle support note 1453883.1 for details.

2.2 Database settings

This section defines recommendations covering the general security configuration of the database instance. The listed recommendations ensure auditing is enabled, listeners are appropriately confined, and authentication is appropriately configured.

NOTE: The remediation procedures assume the use of a server parameter file, which is often a preferred method of storing server initialization parameters.

```
ALTER SYSTEM SET <configuration_item> = <value> SCOPE = SPFILE;
```

For your environment, leaving off the `SCOPE = SPFILE` directive or substituting that with `SCOPE = BOTH` might be preferred depending on the recommendation.

2.2.1 Ensure 'AUDIT_SYS_OPERATIONS' Is Set to 'TRUE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `AUDIT_SYS_OPERATIONS` setting provides for the auditing of all user activities conducted under the `SYSOPER` and `SYSDBA` accounts.

Rationale:

If the parameter `AUDIT_SYS_OPERATIONS` is `FALSE`, all statements except of Startup/Shutdown and Logon by SYSDBA/SYSOPER users are not audited.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$PARAMETER
WHERE UPPER(NAME) = 'AUDIT_SYS_OPERATIONS';
```

Ensure `VALUE` is set to `TRUE`.

Remediation:

To remediate this setting execute the following SQL statement.

```
ALTER SYSTEM SET AUDIT_SYS_OPERATIONS = TRUE SCOPE=SPFILE;
```

References:

1. http://docs.oracle.com/database/121/REFRN/GUID-58176267-238C-40B5-B1F2-BB8BB9518950.htm#REFRN10005

2.2.2 Ensure 'AUDIT_TRAIL' Is Set to 'OS', 'DB,EXTENDED', or 'XML,EXTENDED' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The `audit_trail` setting determines whether or not Oracle's basic audit features are enabled. These can be set to "`Operating System`"(`OS`), "`DB`", "`DB,EXTENDED`", "`XML`" or "`XML,EXTENDED`".

Rationale:

As enabling the basic auditing features for the Oracle instance permits the collection of data to troubleshoot problems, as well as providing value forensic logs in the case of a system breach, this value should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$PARAMETER
WHERE UPPER(NAME)='AUDIT_TRAIL';
```

Ensure `VALUE` is set to `DB`, `OS`, `XML` or `DB,EXTENDED` or `XML,EXTENDED`.

Remediation:

To remediate this setting execute one of the following SQL statements.

```
ALTER SYSTEM SET AUDIT_TRAIL = DB, EXTENDED SCOPE = SPFILE;
ALTER SYSTEM SET AUDIT_TRAIL = OS SCOPE = SPFILE;
ALTER SYSTEM SET AUDIT_TRAIL = XML, EXTENDED SCOPE = SPFILE;
ALTER SYSTEM SET AUDIT_TRAIL = DB SCOPE = SPFILE;
ALTER SYSTEM SET AUDIT_TRAIL = XML SCOPE = SPFILE;
```

References:

1. http://docs.oracle.com/database/121/REFRN/GUID-BD86F593-B606-4367-9FB6-8DAB2E47E7FA.htm#REFRN10006
2. http://www.oracle.com/technetwork/products/audit-vault/learnmore/twp-security-auditperformance-166655.pdf

2.2.3 Ensure 'GLOBAL_NAMES' Is Set to 'TRUE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `global_names` setting requires that the name of a database link matches that of the remote database it will connect to.

Rationale:

As not requiring database connections to match the domain that is being called remotely could allow unauthorized domain sources to potentially connect via brute-force tactics, this value should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$PARAMETER
WHERE UPPER(NAME)='GLOBAL_NAMES';
```

Ensure `VALUE` is set to `TRUE`.

Remediation:

To remediate this setting execute the following SQL statement.

```
ALTER SYSTEM SET GLOBAL_NAMES = TRUE SCOPE = SPFILE;
```

References:

1. http://docs.oracle.com/database/121/REFRN/GUID-221D0483-D814-4963-84E1-7D39A25048ED.htm#REFRN10065

2.2.4 Ensure 'LOCAL_LISTENER' Is Set Appropriately (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `local_listener` setting specifies a network name that resolves to an address of the Oracle TNS listener.

Rationale:

The TNS poisoning attack allows to redirect TNS network traffic to another system by registering a listener to the TNS listener. This attack can be performed by unauthorized users with network access. By specifying the IPC protocol, it is no longer possible to register listeners via TCP/IP.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$PARAMETER
WHERE UPPER(NAME)='LOCAL_LISTENER';
```

Ensure `VALUE` is set to `(DESCRIPTION=(ADDRESS= (PROTOCOL=IPC)(KEY=REGISTER))).`

Remediation:

To remediate this setting execute the following SQL statement.

```
ALTER SYSTEM SET LOCAL_LISTENER='[description]' SCOPE = BOTH;
```

Replace `[description]` with the appropriate description from your `listener.ora` file, where that description sets the `PROTOCOL` parameter to `IPC`. For example:

```
ALTER SYSTEM SET LOCAL_LISTENER='(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=REGISTER)))'
SCOPE=BOTH;
```

References:

1. http://docs.oracle.com/database/121/REFRN/GUID-70F5D04D-02A3-4E89-8A3F-9410B6861BC4.htm#REFRN10082
2. https://support.oracle.com/epmos/faces/ui/km/DocumentDisplay.jspx?id=1453883.1
3. https://support.oracle.com/epmos/faces/ui/km/DocumentDisplay.jspx?id=1340831.1
4. http://www.joxeankoret.com/download/tnspoison.pdf

Notes:

Oracle Real Application Cluster requires a different approach to fix the TNS Poisoning problem. See Oracle support note 1453883.1 for details.

2.2.5 Ensure 'O7_DICTIONARY_ACCESSIBILITY' Is Set to 'FALSE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `O7_dictionary_accessibility` setting is a database initializations parameter that allows/disallows with the `EXECUTE ANY PROCEDURE` and `SELECT ANY DICTIONARY` access to objects in the `SYS` schema; this functionality was created for the ease of migration from Oracle 7 databases to later versions.

Rationale:

As leaving the `SYS` schema so open to connection could permit unauthorized access to critical data structures, this value should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$PARAMETER
WHERE UPPER(NAME)='O7_DICTIONARY_ACCESSIBILITY';
```

Ensure `VALUE` is set to `FALSE`.

Remediation:

To remediate this setting execute the following SQL statement.

```
ALTER SYSTEM SET O7_DICTIONARY_ACCESSIBILITY=FALSE SCOPE = SPFILE;
```

References:

1. http://docs.oracle.com/database/121/REFRN/GUID-1D1A88F1-B603-48FF-BD30-E6099DB1A1ED.htm#REFRN10133

Notes:

The value for this is "O(oh)7" not "0(Zero)7" for "O7." Also, for "Oracle Applications" up to version 11.5.9, this setting is reversed, the "`O7_dictionary_accessibility=TRUE`" value is required for correct operations.

2.2.6 Ensure 'OS_ROLES' Is Set to 'FALSE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `os_roles` setting permits externally created groups to be applied to database management.

Rationale:

As allowing the OS use external groups for database management could cause privilege overlaps and generally weaken security, this value should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$PARAMETER
WHERE UPPER(NAME)='OS_ROLES';
```

Ensure `VALUE` is set to `FALSE`.

Remediation:

To remediate this setting execute the following SQL statement.

```
ALTER SYSTEM SET OS_ROLES = FALSE SCOPE = SPFILE;
```

References:

1. http://docs.oracle.com/database/121/REFRN/GUID-51CCE2D6-F841-4E02-A89D-EA08FC110CF3.htm#REFRN10153

2.2.7 Ensure 'REMOTE_LISTENER' Is Empty (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `remote_listener` setting determines whether or not a valid listener can be established on a system separate from the database instance.

Rationale:

As permitting a remote listener for connections to the database instance can allow for the potential spoofing of connections and that could compromise data confidentiality and integrity, this value should be disabled/restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$PARAMETER
WHERE UPPER(NAME)='REMOTE_LISTENER';
```

Ensure `VALUE` is empty.

Remediation:

To remediate this setting execute the following SQL statement.

```
ALTER SYSTEM SET REMOTE_LISTENER = '' SCOPE = SPFILE;
```

References:

1. http://docs.oracle.com/database/121/REFRN/GUID-FEE2E8B5-CE02-4158-A6B4-030E59316756.htm#REFRN10183

Notes:

If set at `remote_listener=true`, the address/address list is taken from the TNSNAMES.ORA file

2.2.8 Ensure 'REMOTE_LOGIN_PASSWORDFILE' Is Set to 'NONE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `remote_login_passwordfile` setting specifies whether or not Oracle checks for a password file during login and how many databases can use the password file.

Rationale:

As the use of this sort of password login file could permit unsecured, privileged connections to the database, this value should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$PARAMETER
WHERE UPPER(NAME)='REMOTE_LOGIN_PASSWORDFILE';
```

Ensure `VALUE` is set to `NONE`.

Remediation:

To remediate this setting execute the following SQL statement.

```
ALTER SYSTEM SET REMOTE_LOGIN_PASSWORDFILE = 'NONE' SCOPE = SPFILE;
```

References:

1. http://docs.oracle.com/database/121/REFRN/GUID-6619299E-95E8-4821-B123-3B5899F046C7.htm#REFRN10184

2.2.9 Ensure 'REMOTE_OS_AUTHENT' Is Set to 'FALSE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `remote_os_authent` setting determines whether or not OS 'roles' with the attendant privileges are allowed for remote client connections.

Rationale:

As permitting OS roles for database connections to can allow the spoofing of connections and permit granting the privileges of an OS role to unauthorized users to make connections, this value should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$PARAMETER
WHERE UPPER(NAME)='REMOTE_OS_AUTHENT';
```

Ensure `VALUE` is set to `FALSE`.

Remediation:

To remediate this setting execute the following SQL statement.

```
ALTER SYSTEM SET REMOTE_OS_AUTHENT = FALSE SCOPE = SPFILE;
```

References:

1. http://docs.oracle.com/database/121/REFRN/GUID-AB66C849-FE5A-4E06-A6E1-AEE775D55703.htm#REFRN10185

2.2.10 Ensure 'REMOTE_OS_ROLES' Is Set to 'FALSE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `remote_os_roles` setting permits remote users' OS roles to be applied to database management.

Rationale:

As allowing remote clients OS roles to have permissions for database management could cause privilege overlaps and generally weaken security, this value should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$PARAMETER
WHERE UPPER(NAME)='REMOTE_OS_ROLES';
```

Ensure `VALUE` is set to `FALSE`.

Remediation:

To remediate this setting execute the following SQL statement.

```
ALTER SYSTEM SET REMOTE_OS_ROLES = FALSE SCOPE = SPFILE;
```

References:

1. http://docs.oracle.com/database/121/REFRN/GUID-BAA83447-14C1-4BE7-BB5D-806ED3E00AED.htm#REFRN10186

2.2.11 Ensure 'UTL_FILE_DIR' Is Empty (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `utl_file_dir` setting allows packages like `utl_file` to access (read/write/modify/delete) files specified in `utl_file_dir`. (This is deprecated but usable in 11g.)

Rationale:

As using the `utl_file_dir` to create directories allows the manipulation of files in these directories.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$PARAMETER
WHERE UPPER(NAME)='UTL_FILE_DIR';
```

Ensure `VALUE` is empty.

Remediation:

To remediate this setting execute the following SQL statement.

```
ALTER SYSTEM SET UTL_FILE_DIR = '' SCOPE = SPFILE;
```

References:

1. http://docs.oracle.com/database/121/REFRN/GUID-DCA8A942-ACE1-46D6-876E-3244F390BCAE.htm#REFRN10230

2.2.12 Ensure 'SEC_CASE_SENSITIVE_LOGON' Is Set to 'TRUE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `SEC_CASE_SENSITIVE_LOGON` information determines whether or not case-sensitivity is required for passwords during login.

Due to the security bug CVE-2012-3137 it is recommended to set this parameter to `TRUE` if the October 2012 CPU/PSU or later was applied.

If the patch was not applied it is recommended to set this parameter to `FALSE` to avoid that the vulnerability could be abused.

Rationale:

Oracle 11g databases without CPU October 2012 patch or later are vulnerable to CVE-2012-3137 if case-sensitive SHA-1 password hashes are used. To avoid this kind of attack the old DES-hashes have to be used.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$PARAMETER
WHERE UPPER(NAME)='SEC_CASE_SENSITIVE_LOGON';
```

Ensure `VALUE` is set to `TRUE`.

Remediation:

To remediate this setting execute the following SQL statement.

```
ALTER SYSTEM SET SEC_CASE_SENSITIVE_LOGON = TRUE SCOPE = SPFILE;
```

Impact:

If `SEC_CASE_SENSITIVE_LOGON` is FALSE, all user with SHA-1 hashes only ("`select name,password,spare4 from sys.user$ where password is null and spare4 is not null`") are no longer able to connect to the database. In this case the password for all users without DES hash have to set again.

References:

1. http://docs.oracle.com/database/121/REFRN/GUID-F464653A-0D43-4A70-8F05-0274A12C8578.htm#REFRN10299

2.2.13 Ensure 'SEC_MAX_FAILED_LOGIN_ATTEMPTS' Is '10' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `SEC_MAX_FAILED_LOGIN_ATTEMPTS` parameter determines how many failed login attempts are allowed before Oracle closes the login connection.

Rationale:

As allowing an unlimited number of login attempts for a user connection can facilitate both brute-force login attacks and the occurrence of Denial-of-Service, this value (`10`) should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$PARAMETER
WHERE UPPER(NAME)='SEC_MAX_FAILED_LOGIN_ATTEMPTS';
```

Ensure `VALUE` is set to `10`.

Remediation:

To remediate this setting execute the following SQL statement.

```
ALTER SYSTEM SET SEC_MAX_FAILED_LOGIN_ATTEMPTS = 10 SCOPE = SPFILE;
```

References:

1. http://docs.oracle.com/database/121/REFRN/GUID-DEC2A3B2-F49B-499E-A3CF-D097F3A5BA83.htm#REFRN10274

2.2.14 Ensure 'SEC_PROTOCOL_ERROR_FURTHER_ACTION' Is Set to 'DELAY,3' or 'DROP,3' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `SEC_PROTOCOL_ERROR_FURTHER_ACTION` setting determines the Oracle's server's response to bad/malformed packets received from the client.

Rationale:

As bad packets received from the client can potentially indicate packet-based attacks on the system, such as "TCP SYN Flood" or "Smurf" attacks, which could result in a Denial-of-Service condition, this value should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$PARAMETER
WHERE UPPER(NAME)='SEC_PROTOCOL_ERROR_FURTHER_ACTION';
```

Ensure `VALUE` is set to `DELAY,3` or `DROP,3`.

Remediation:

To remediate this setting execute one of the following SQL statements.

```
ALTER SYSTEM SET SEC_PROTOCOL_ERROR_FURTHER_ACTION = 'DELAY,3' SCOPE = SPFILE;
ALTER SYSTEM SET SEC_PROTOCOL_ERROR_FURTHER_ACTION = 'DROP,3' SCOPE = SPFILE;
```

References:

1. http://docs.oracle.com/database/121/REFRN/GUID-1E8D3C6E-C919-4218-8117-760D31BD0F95.htm#REFRN10282

2.2.15 Ensure 'SEC_PROTOCOL_ERROR_TRACE_ACTION' Is Set to 'LOG' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `SEC_PROTOCOL_ERROR_TRACE_ACTION` setting determines the Oracle's server's logging response level to bad/malformed packets received from the client, by generating `ALERT`, `LOG`, or `TRACE` levels of detail in the log files.

Rationale:

As bad packets received from the client can potentially indicate packet-based attacks on the system, such as "TCP SYN Flood" or "Smurf" attacks, which could result in a Denial-of-Service condition, this diagnostic/logging value for `ALERT`, `LOG`, or `TRACE` conditions should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$PARAMETER
WHERE UPPER(NAME)='SEC_PROTOCOL_ERROR_TRACE_ACTION';
```

Ensure `VALUE` is set to `LOG`.

Remediation:

To remediate this setting execute the following SQL statement.

```
ALTER SYSTEM SET SEC_PROTOCOL_ERROR_TRACE_ACTION=LOG SCOPE = SPFILE;
```

References:

1. http://docs.oracle.com/database/121/REFRN/GUID-AE811BC1-8CED-4B21-B16C-4B712B127535.htm#REFRN10283

Notes:

Setting the value as `SEC_PROTOCOL_ERROR_TRACE_ACTION=TRACE` can generate an enormous amount of log output and should be reserved for debugging only.

2.2.16 Ensure 'SEC_RETURN_SERVER_RELEASE_BANNER' Is Set to 'FALSE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The information about patch/update release number provides information about the exact patch/update release that is currently running on the database.

Rationale:

As allowing the database to return information about the patch/update release number could facilitate unauthorized users' attempts to gain access based upon known patch weaknesses, this value should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$PARAMETER
WHERE UPPER(NAME)='SEC_RETURN_SERVER_RELEASE_BANNER';
```

Ensure `VALUE` is set to `FALSE`.

Remediation:

To remediate this setting execute the following SQL statement.

```
ALTER SYSTEM SET SEC_RETURN_SERVER_RELEASE_BANNER = FALSE SCOPE = SPFILE;
```

References:

1. http://docs.oracle.com/database/121/REFRN/GUID-688102A0-11F5-4F06-8868-934D65C4E878.htm#REFRN10275

2.2.17 Ensure 'SQL92_SECURITY' Is Set to 'TRUE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `SQL92_SECURITY` parameter setting `TRUE` requires that a user must also be granted the `SELECT` object privilege before being able to perform `UPDATE` or `DELETE` operations on tables that have `WHERE` or `SET` clauses.

Rationale:

A user without `SELECT` privilege can still infer the value stored in a column by referring to that column in a `DELETE` or `UPDATE` statement. This setting prevents inadvertent information disclosure by ensuring that only users who already have `SELECT` privilege can execute the statements that would allow them to infer the stored values.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$PARAMETER
WHERE UPPER(NAME)='SQL92_SECURITY';
```

Ensure `VALUE` is set to `TRUE`.

Remediation:

To remediate this setting execute the following SQL statement.

```
ALTER SYSTEM SET SQL92_SECURITY = TRUE SCOPE = SPFILE;
```

Default Value:

`FALSE`

References:

1. http://docs.oracle.com/database/121/REFRN/GUID-E41087C2-250E-4201-908B-79E659B22A4B.htm#REFRN10210

2.2.18 Ensure '_TRACE_FILES_PUBLIC' Is Set to 'FALSE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `_trace_files_public` setting determines whether or not the system's trace file is world readable.

Rationale:

As permitting the read permission to other anyone can read the instance's trace files file which could contain sensitive information about instance operations, this value should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT VALUE
FROM V$PARAMETER
WHERE NAME='_trace_files_public';
```

A `VALUE` equal to `FALSE` or lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
ALTER SYSTEM SET "_trace_files_public" = FALSE SCOPE = SPFILE;
```

References:

1. http://asktom.oracle.com/pls/asktom/f?p=100:11:0::::P11_QUESTION_ID:4295521746131

2.2.19 Ensure 'RESOURCE_LIMIT' Is Set to 'TRUE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

`RESOURCE_LIMIT` determines whether resource limits are enforced in database profiles

Rationale:

If `resource_limit` is set to `FALSE`, none of the system resource limits that are set in any database profiles are enforced. If `resource_limit` is set to TRUE, then the limits set in database profiles are enforced.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$PARAMETER
WHERE UPPER(NAME)='RESOURCE_LIMIT';
```

Ensure `VALUE` is set to `TRUE`.

Remediation:

To remediate this setting execute the following SQL statement.

```
ALTER SYSTEM SET RESOURCE_LIMIT = TRUE SCOPE = SPFILE;
```

Default Value:

FALSE

References:

1. http://docs.oracle.com/database/121/REFRN/GUID-BB0AB177-3867-4D0D-8700-A1AC8BDFEFC3.htm#REFRN10188

3 Oracle Connection and Login Restrictions

The restrictions on Client/User connections to the Oracle database help block unauthorized access to data and services by setting access rules. These security measures help to ensure that successful logins cannot be easily made through brute-force password attacks or intuited by clever social engineering exploits. Settings are generally recommended to be applied to all defined profiles rather than by using only the `DEFAULT` profile. All values assigned below are the recommended minimums or maximums; higher, more restrictive values can be applied at the discretion of the organization by creating a separate profile to assign to a different user group.

3.1 Ensure 'FAILED_LOGIN_ATTEMPTS' Is Less than or Equal to '5' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `failed_login_attempts` setting determines how many failed login attempts are permitted before the system locks the user's account. While different profiles can have different and more restrictive settings, such as USERS and APPS, the minimum(s) recommended here should be set on the DEFAULT profile.

Rationale:

As repeated failed login attempts can indicate the initiation of a brute-force login attack, this value should be set according to the needs of the organization (see **warning** below on a known bug that can make this security measure backfire).

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PROFILE, RESOURCE_NAME, LIMIT
FROM DBA_PROFILES
WHERE RESOURCE_NAME='FAILED_LOGIN_ATTEMPTS'
AND
 (
 LIMIT = 'DEFAULT'
 OR LIMIT = 'UNLIMITED'
 OR LIMIT > 5
 );
```

Lack of results implies compliance.

Remediation:

Remediate this setting by executing the following SQL statement for each `PROFILE` returned by the audit procedure.

```
ALTER PROFILE <profile_name> LIMIT FAILED_LOGIN_ATTEMPTS 5;
```

Warning:

One very great concern with the above is the possibility of this setting being exploited to craft a DDoS attack by using the row-locking delay between failed login attempts (see *Oracle Bug 7715339 – Logon failures causes "row cache lock" waits – Allow disable of logon delay [ID 7715339.8]*, so the configuration of this setting depends on using the bug workaround). Also, while the setting for the `failed_login_attempts` value can also be set in `sqlnet.ora`, this only applies to listed users. The similar setting used to block a DDoS, the `SEC_MAX_FAILED_LOGIN_ATTEMPTS` initialization parameter, can be used to protect unauthorized intruders from attacking the server processes for applications, but this setting does not protect against unauthorized attempts via valid usernames.

3.2 Ensure 'PASSWORD_LOCK_TIME' Is Greater than or Equal to '1' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `PASSWORD_LOCK_TIME` setting determines how many days must pass for the user's account to be unlocked after the set number of failed login attempts has occurred.

Rationale:

As locking the user account after repeated failed login attempts can block further brute-force login attacks, but can create administrative headaches as this account unlocking process always requires DBA intervention, this value should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PROFILE, RESOURCE_NAME, LIMIT
FROM DBA_PROFILES
WHERE RESOURCE_NAME='PASSWORD_LOCK_TIME'
AND
 (
 LIMIT = 'DEFAULT'
 OR LIMIT = 'UNLIMITED'
 OR LIMIT < 1
 );
```

Lack of results implies compliance.

Remediation:

Remediate this setting by executing the following SQL statement for each PROFILE returned by the audit procedure.

```
ALTER PROFILE <profile_name> LIMIT PASSWORD_LOCK_TIME 1;
```

3.3 Ensure 'PASSWORD_LIFE_TIME' Is Less than or Equal to '90' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `password_life_time` setting determines how long a password may be used before the user is required to be change it.

Rationale:

As allowing passwords to remain unchanged for long periods makes the success of brute-force login attacks more likely, this value should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PROFILE, RESOURCE_NAME, LIMIT
FROM DBA_PROFILES
WHERE RESOURCE_NAME='PASSWORD_LIFE_TIME'
AND
 (
 LIMIT = 'DEFAULT'
 OR LIMIT = 'UNLIMITED'
 OR LIMIT > 90
 );
```

Lack of results implies compliance.

Remediation:

Remediate this setting by executing the following SQL statement for each PROFILE returned by the audit procedure.

```
ALTER PROFILE <profile_name> LIMIT PASSWORD_LIFE_TIME 90;
```

3.4 Ensure 'PASSWORD_REUSE_MAX' Is Greater than or Equal to '20' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `password_reuse_max` setting determines how many different passwords must be used before the user is allowed to reuse a prior password.

Rationale:

As allowing reuse of a password within a short period of time after the password's initial use can make the success of both social-engineering and brute-force password-based attacks more likely, this value should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PROFILE, RESOURCE_NAME, LIMIT
FROM DBA_PROFILES
WHERE RESOURCE_NAME='PASSWORD_REUSE_MAX'
AND
 (
 LIMIT = 'DEFAULT'
 OR LIMIT = 'UNLIMITED'
 OR LIMIT < 20
 );
```

Lack of results implies compliance.

Remediation:

Remediate this setting by executing the following SQL statement for each `PROFILE` returned by the audit procedure.

```
ALTER PROFILE <profile_name> LIMIT PASSWORD_REUSE_MAX 20;
```

Notes:

The above restriction should be applied along with the `password_reuse_time` setting.

3.5 Ensure 'PASSWORD_REUSE_TIME' Is Greater than or Equal to '365' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `password_reuse_time` setting determines the amount of time in days that must pass before the same password may be reused.

Rationale:

As reusing the same password after only a short period of time has passed makes the success of brute-force login attacks more likely, this value should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PROFILE, RESOURCE_NAME, LIMIT
FROM DBA_PROFILES
WHERE RESOURCE_NAME='PASSWORD_REUSE_TIME'
AND
 (
 LIMIT = 'DEFAULT'
 OR LIMIT = 'UNLIMITED'
 OR LIMIT < 365
 );
```

Lack of results implies compliance.

Remediation:

Remediate this setting by executing the following SQL statement for each `PROFILE` returned by the audit procedure.

```
ALTER PROFILE <profile_name> LIMIT PASSWORD_REUSE_TIME 365;
```

Notes:

The above restriction should be applied along with the `password_reuse_max` setting.

3.6 Ensure 'PASSWORD_GRACE_TIME' Is Less than or Equal to '5' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `password_grace_time` setting determines how many days can pass after the user's password expires before the user's login capability is automatically locked out.

Rationale:

As locking the user account after the expiration of the password change requirement's grace period can help prevent password-based attack against a forgotten or disused accounts, while still allowing the account and its information to be accessible by DBA intervention, this value should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PROFILE, RESOURCE_NAME, LIMIT
FROM DBA_PROFILES
WHERE RESOURCE_NAME='PASSWORD_GRACE_TIME'
AND
 (
 LIMIT = 'DEFAULT'
 OR LIMIT = 'UNLIMITED'
 OR LIMIT > 5
 );
```

Lack of results implies compliance.

Remediation:

Remediate this setting by executing the following SQL statement for each `PROFILE` returned by the audit procedure.

```
ALTER PROFILE <profile_name> LIMIT PASSWORD_GRACE_TIME 5;
```

3.7 Ensure 'DBA_USERS.PASSWORD' Is Not Set to 'EXTERNAL' for Any User (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `password='EXTERNAL'` setting determines whether or not a user can be authenticated by a remote OS to allow access to the database with full authorization.

Rationale:

As allowing remote OS authentication of a user to the database can potentially allow supposed "privileged users" to connect as "authenticated," even when the remote system is compromised, these logins should be disabled/restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT USERNAME
FROM DBA_USERS
WHERE PASSWORD='EXTERNAL';
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
ALTER USER <username> IDENTIFIED BY <password>;
```

Notes:

The `PASSWORD` keyword (column) used in the SQL for prior Oracle versions has been deprecated from version 11.2 onward in favor of the new `AUTHENTICATION_TYPE` keyword (column) for the `DBA_USERS` table. However, the PASSWORD column has still been retained for backward-compatability.

3.8 Ensure 'PASSWORD_VERIFY_FUNCTION' Is Set for All Profiles (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `password_verify_function` determines password settings requirements when a user password is changed at the SQL command prompt. This setting does not apply for users managed by the Oracle password file.

Rationale:

As requiring users to apply the 11gr2 security features in password creation, such as forcing mixed-case complexity, the blocking of simple combinations, and change/history settings can potentially thwart logins by unauthorized users, this function should be applied/enabled according to the needs of the organization.

Audit:

To assess this recommendation execute the following SQL statement.

```
SELECT PROFILE, RESOURCE_NAME
FROM DBA_PROFILES
WHERE RESOURCE_NAME='PASSWORD_VERIFY_FUNCTION'
AND (LIMIT = 'DEFAULT' OR LIMIT = 'NULL');
```

Lack of results implies compliance.

Remediation:

Create a custom password verification function which fulfills the password requirements of the organization.

3.9 Ensure 'SESSIONS_PER_USER' Is Less than or Equal to '10' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `SESSIONS_PER_USER` (Number of sessions allowed) determines the maximum number of user sessions that are allowed to be open concurrently.

Rationale:

As limiting the number of the `SESSIONS_PER_USER` can help prevent memory resource exhaustion by poorly formed requests or intentional Denial-of-Service attacks, this value should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PROFILE, RESOURCE_NAME, LIMIT
FROM DBA_PROFILES
WHERE RESOURCE_NAME='SESSIONS_PER_USER'
AND
  (
    LIMIT = 'DEFAULT'
    OR LIMIT = 'UNLIMITED'
    OR LIMIT > 10
  );
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement for each `PROFILE` returned by the audit procedure.

```
ALTER PROFILE <profile_name> LIMIT SESSIONS_PER_USER 10;
```

Notes:

The `SESSIONS_PER_USER` profile management capability was created to prevent resource(s) exhaustion at a time when these were very expensive. As current database design may require much higher limits on this parameter if one "user" handles all processing for specific types of batch/customer connections, this must be handled via a new user profile.

3.10 Ensure No Users Are Assigned the 'DEFAULT' Profile (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

Upon creation database users are assigned to the DEFAULT profile unless otherwise specified.

Rationale:

It is recommended that users be created with function-appropriate profiles. The `DEFAULT` profile, being defined by Oracle, is subject to change at any time (e.g. by patch or version update). The `DEFAULT` profile has unlimited settings that are often required by the `SYS` user when patching; such unlimited settings should be tightly reserved and not applied to unnecessary users.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT USERNAME
FROM DBA_USERS
WHERE PROFILE='DEFAULT'
AND ACCOUNT_STATUS='OPEN'
AND USERNAME NOT IN
  ('ANONYMOUS', 'CTXSYS',   'DBSNMP', 'EXFSYS', 'LBACSYS',
   'MDSYS',     'MGMT_VIEW','OLAPSYS','OWBSYS', 'ORDPLUGINS',
   'ORDSYS',    'OUTLN',    'SI_INFORMTN_SCHEMA','SYS',
   'SYSMAN',    'SYSTEM',   'TSMSYS', 'WK_TEST', 'WKSYS',
   'WKPROXY',   'WMSYS',    'XDB', 'CISSCAN');
```

Lack of results implies compliance.

Remediation:

To remediate this recommendation, execute the following SQL statement for each user returned by the audit query using a functional-appropriate profile.

```
ALTER USER <username> PROFILE <appropriate_profile>
```

4 Oracle User Access and Authorization Restrictions

The capability to use database resources at a given level, or user authorization rules, allows for user manipulation of the various parts of the Oracle database. These authorizations must be structured to block unauthorized use and/or corruption of vital data and services by setting restrictions on user capabilities, particularly those of the user `PUBLIC`. Such security measures help to ensure that successful logins cannot be easily redirected.
IMPORTANT: Use caution when revoking privileges from PUBLIC. Oracle and third-party products explicitly require default grants to PUBLIC for commonly used functions, objects, and in view definitions. After revoking any privilege from PUBLIC, verify that applications keep running properly. After revoking privileges from PUBLIC, recompile invalid database objects. Specific grants to users and roles may be needed to make all objects valid.
Please see the following Oracle support document which provides further information and SQL statements that can be used to determine dependencies that require explicit grants.
Be Cautious When Revoking Privileges Granted to PUBLIC (Doc ID 247093.1)
Always test database changes in Development and Test environments before making changes to Production databases.

4.1 Default Public Privileges for Packages and Object Types

Revoke default public execute privileges from powerful packages and object types.

4.1.1 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_ADVISOR' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database DBMS_ADVISOR package can be used to write files located on the server where the Oracle instance is installed.

Rationale:

As use of the `DBMS_ADVISOR` package could allow an unauthorized user to corrupt operating system files on the instance's host, use of this package should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_ADVISOR';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_ADVISOR FROM PUBLIC;
```

References:

1. http://docs.oracle.com/database/121/ARPLS/d_advis.htm#ARPLS350

4.1.2 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_CRYPTO' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `DBMS_CRYPTO` settings provide a toolset that determines the strength of the encryption algorithm used to encrypt application data and is part of the `SYS` schema. The DES (56-bit key), 3DES (168-bit key), 3DES-2KEY (112-bit key), AES (128/192/256-bit keys), and RC4 are available.

Rationale:

As execution of these cryptography procedures by the user PUBLIC can potentially endanger portions of or all of the data storage, this value should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND TABLE_NAME='DBMS_CRYPTO';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_CRYPTO FROM PUBLIC;
```

References:

1. http://docs.oracle.com/database/121/ARPLS/d_crypto.htm#ARPLS664

4.1.3 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_JAVA' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DBMS_JAVA` package can run Java classes (e.g. OS commands) or grant Java privileges.

Rationale:

`The DBMS_JAVA` package could allow an attacker to run operating system commands from the database.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_JAVA';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_JAVA FROM PUBLIC;
```

References:

1. http://docs.oracle.com/database/121/JJDEV/appendixa.htm#JJDEV13000

4.1.4 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_JAVA_TEST' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DBMS_JAVA_TEST` package can run Java classes (e.g. OS commands) or grant Java privileges.

Rationale:

The `DBMS_JAVA_TEST` package could allow an attacker to run operating system commands from the database.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_JAVA_TEST';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_JAVA_TEST FROM PUBLIC;
```

Notes:

Undocumented

4.1.5 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_JOB' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DBMS_JOB` package schedules and manages the jobs sent to the job queue and has been superseded by the `DBMS_SCHEDULER` package, even though `DBMS_JOB` has been retained for backwards compatibility.

Rationale:

As use of the `DBMS_JOB` package could allow an unauthorized user to disable or overload the job queue and has been superseded by the `DBMS_SCHEDULER` package, this package should be disabled or restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_JOB';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_JOB FROM PUBLIC;
```

References:

1. http://docs.oracle.com/database/121/ARPLS/d_job.htm#ARPLS019

4.1.6 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_LDAP' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DBMS_LDAP` package contains functions and procedures that enable programmers to access data from LDAP servers.

Rationale:

As use of the `DBMS_LDAP` package can be used to create specially crafted error messages or send information via DNS to the outside.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_LDAP';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_LDAP FROM PUBLIC;
```

References:

1. http://docs.oracle.com/database/121/ARPLS/d_ldap.htm#ARPLS360

4.1.7 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_LOB' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DBMS_LOB` package provides subprograms that can manipulate and read/write on BLOBs, CLOBs, NCLOBs, BFILEs, and temporary LOBs.

Rationale:

As use of the `DBMS_LOB` package could allow an unauthorized user to manipulate BLOBs, CLOBs, NCLOBs, BFILEs, and temporary LOBs on the instance, either destroying data or causing a Denial-of-Service condition due to corruption of disk space, use of this package should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_LOB';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_LOB FROM PUBLIC;
```

References:

1. http://docs.oracle.com/database/121/ARPLS/d_lob.htm#ARPLS600

4.1.8 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_OBFUSCATION_TOOLKIT' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `DBMS_OBFUSCATION_TOOLKIT` settings provide one of the tools that determine the strength of the encryption algorithm used to encrypt application data and is part of the `SYS` schema. The DES (56-bit key) and 3DES (168-bit key) are the only two types available.

Rationale:

As allowing the PUBLIC user privileges to access this capability can be potentially harm the data storage, this access should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_OBFUSCATION_TOOLKIT';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_OBFUSCATION_TOOLKIT FROM PUBLIC;
```

4.1.9 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_RANDOM' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DBMS_RANDOM` package is used for generating random numbers but should not be used for cryptographic purposes.

Rationale:

As assignment of use of the `DBMS_RANDOM` package can allow the unauthorized application of the random number-generating function, this capability should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_RANDOM';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_RANDOM FROM PUBLIC;
```

References:

1. http://docs.oracle.com/cd/E11882_01/appdev.112/e25788/d_random.htm

Notes:

The OEM cautions that removing this from PUBLIC may break certain applications.

4.1.10 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_SCHEDULER' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DBMS_SCHEDULER` package schedules and manages the database and operating system jobs.

Rationale:

As use of the `DBMS_SCHEDULER` package could allow an unauthorized user to run database or operating system jobs.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_SCHEDULER';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_SCHEDULER FROM PUBLIC;
```

References:

1. http://docs.oracle.com/database/121/ARPLS/d_sched.htm#ARPLS72235

4.1.11 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_SQL' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DBMS_SQL` package is used for running dynamic SQL statements.

Rationale:

The `DBMS_SQL` package could allow privilege escalation if the input validation is not done properly.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_SQL';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_SQL FROM PUBLIC;
```

References:

1. http://docs.oracle.com/database/121/ARPLS/d_sql.htm#ARPLS058

4.1.12 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_XMLGEN' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The `DBMS_XMLGEN` package takes an arbitrary SQL query as input, converts it to XML format, and returns the result as a CLOB.

Rationale:

The package `DBMS_XMLGEN` can be used to search the entire database for critical information like credit card numbers, and other sensitive information.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_XMLGEN';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_XMLGEN FROM PUBLIC;
```

References:

1. http://docs.oracle.com/database/121/ARPLS/d_xmlgen.htm#ARPLS374
2. http://www.red-database-security.com/wp/confidence2009.pdf

4.1.13 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_XMLQUERY' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle package `DBMS_XMLQUERY` takes an arbitrary SQL query, converts it to XML format, and returns the result. This package is similar to DBMS_XMLGEN.

Rationale:

The package `DBMS_XMLQUERY` can be used to search the entire database for critical information like credit card numbers and other sensitive information.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_XMLQUERY';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_XMLQUERY FROM PUBLIC;
```

References:

1. http://docs.oracle.com/database/121/ARPLS/d_xmlque.htm#ARPLS376

4.1.14 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'UTL_FILE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `UTL_FILE` package can be used to read/write files located on the server where the Oracle instance is installed.

Rationale:

As use of the `UTL_FILE` package could allow an user to read files at the operating system. These files could contain sensitive information (e.g. passwords in `.bash_history`).

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='UTL_FILE';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON UTL_FILE FROM PUBLIC;
```

References:

1. http://docs.oracle.com/database/121/ARPLS/u_file.htm#ARPLS069

4.1.15 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'UTL_INADDR' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `UTL_INADDR` package can be used to create specially crafted error messages or send information via DNS to the outside.

Rationale:

As use of the `UTL_INADDR` package is often used in SQL Injection attacks from the web it should be revoked from public.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='UTL_INADDR';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON UTL_INADDR FROM PUBLIC;
```

References:

1. http://docs.oracle.com/database/121/ARPLS/u_inaddr.htm#ARPLS071

4.1.16 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'UTL_TCP' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `UTL_TCP` package can be used to read/write file to TCP sockets on the server where the Oracle instance is installed.

Rationale:

As use of the `UTL_TCP` package could allow an unauthorized user to corrupt the TCP stream used for carry the protocols that communicate with the instance's external communications, use of this package should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='UTL_TCP';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON UTL_TCP FROM PUBLIC;
```

References:

1. http://docs.oracle.com/database/121/ARPLS/u_tcp.htm#ARPLS075

4.1.17 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'UTL_MAIL' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `UTL_MAIL` package can be used to send email from the server where the Oracle instance is installed.

Rationale:

As use of the `UTL_MAIL` package could allow an unauthorized user to corrupt the SMTP function to accept or generate junk mail that can result in a Denial-of-Service condition due to network saturation, use of this package should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='UTL_MAIL';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON UTL_MAIL FROM PUBLIC;
```

References:

1. http://docs.oracle.com/database/121/ARPLS/u_mail.htm#ARPLS384

4.1.18 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'UTL_SMTP' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `UTL_SMTP` package can be used to send email from the server where the Oracle instance is installed.

Rationale:

As use of the `UTL_SMTP` package could allow an unauthorized user to corrupt the SMTP function to accept or generate junk mail that can result in a Denial-of-Service condition due to network saturation, use of this package should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='UTL_SMTP';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON UTL_SMTP FROM PUBLIC;
```

References:

1. http://docs.oracle.com/database/121/ARPLS/u_smtp.htm#ARPLS074

4.1.19 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'UTL_DBWS' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `UTL_DBWS` package can be used to read/write file to web-based applications on the server where the Oracle instance is installed. This package is not automatically installed for security reasons.

Rationale:

As use of the `UTL_DBWS` package could allow an unauthorized user to corrupt the HTTP stream used for carry the protocols that communicate with the instance's web-based external communications, use of this package should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='UTL_DBWS';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON UTL_DBWS FROM 'PUBLIC';
```

References:

1. https://docs.oracle.com/database/121/JJPUB/intro.htm#BHCIBFGJ

4.1.20 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'UTL_ORAMTS' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database UTL_ORAMTS package can be used to perform HTTP-requests. This could be used to send information to the outside.

Rationale:

As use of the `UTL_ORAMTS` package could be used to send (sensitive) information to external websites. The use of this package should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='UTL_ORAMTS';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON UTL_ORAMTS FROM PUBLIC;
```

References:

1. http://docs.oracle.com/database/121/NTMTS/recovery.htm#sthref73

4.1.21 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'UTL_HTTP' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `UTL_HTTP` package can be used to perform HTTP-requests. This could be used to send information to the outside.

Rationale:

As use of the `UTL_HTTP` package could be used to send (sensitive) information to external websites. The use of this package should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='UTL_HTTP';
The assessment fails if results are returned.
```

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON UTL_HTTP FROM PUBLIC;
```

References:

1. http://docs.oracle.com/database/121/ARPLS/u_http.htm#ARPLS070

4.1.22 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'HTTPURITYPE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `HTTPURITYPE` object type can be used to perform HTTP-requests.

Rationale:

The ability to perform HTTP requests could be used to leak information from the database to an external destination.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='HTTPURITYPE';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON HTTPURITYPE FROM PUBLIC;
```

References:

1. http://docs.oracle.com/database/121/ARPLS/t_dburi.htm#ARPLS71705

4.2 Revoke Non-Default Privileges for Packages and Object Types

The recommendations within this section revoke excessive privileges for packages and object types.

4.2.1 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_SYS_SQL' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DBMS_SYS_SQL` package is shipped as undocumented.

Rationale:

As use of the `DBMS_SYS_SQL` package could allow an user to run code as a different user without entering user credentials.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_SYS_SQL';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_SYS_SQL FROM PUBLIC;
```

References:

1. http://asktom.oracle.com/pls/asktom/f?p=100:11:0::::P11_QUESTION_ID:1325202421535

4.2.2 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_BACKUP_RESTORE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DBMS_BACKUP_RESTORE` package is used for applying PL/SQL commands to the native RMAN sequences.

Rationale:

As assignment of use of the `DBMS_BACKUP_RESTORE` package can allow to access file permissions on operating system level.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_BACKUP_RESTORE';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_BACKUP_RESTORE FROM PUBLIC;
```

References:

1. http://psoug.org/reference/dbms_backup_restore.html
2. http://davidalejomarcos.wordpress.com/2011/09/13/how-to-list-files-on-a-directory-from-oracle-database/

4.2.3 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_AQADM_SYSCALLS' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DBMS_AQADM_SYSCALLS` package is shipped as undocumented and allows to run SQL commands as user SYS.

Rationale:

As use of the `DBMS_AQADM_SYSCALLS` package could allow an unauthorized user to run SQL commands as user SYS.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_AQADM_SYSCALLS';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_AQADM_SYSCALLS FROM PUBLIC;
```

References:

1. http://securityvulns.ru/files/ohh-indirect-privilege-escalation.pdf

4.2.4 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_REPCAT_SQL_UTL' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DBMS_REPCAT_SQL_UTL` package is shipped as undocumented and allows to run SQL commands as user SYS.

Rationale:

As use of the `DBMS_REPCAT_SQL_UTL` package could allow an unauthorized user to run SQL commands as user SYS.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_REPCAT_SQL_UTL';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
revoke execute on DBMS_REPCAT_SQL_UTL FROM PUBLIC;
```

References:

1. http://securityvulns.ru/files/ohh-indirect-privilege-escalation.pdf

4.2.5 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'INITJVMAUX' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `INITJVMAUX` package is shipped as undocumented and allows to run SQL commands as user SYS.

Rationale:

As use of the `INITJVMAUX` package could allow an unauthorized user to run SQL commands as user SYS.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='INITJVMAUX';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON INITJVMAUX FROM PUBLIC;
```

References:

1. http://securityvulns.ru/files/ohh-indirect-privilege-escalation.pdf

4.2.6 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_STREAMS_ADM_UTL' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DBMS_STREAMS_ADM_UTL` package is shipped as undocumented and allows to run SQL commands as user SYS.

Rationale:

As use of the `DBMS_STREAMS_ADM_UTL` package could allow an unauthorized user to run SQL commands as user SYS.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_STREAMS_ADM_UTL';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_STREAMS_ADM_UTL FROM PUBLIC;
```

References:

1. http://securityvulns.ru/files/ohh-indirect-privilege-escalation.pdf

4.2.7 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_AQADM_SYS' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DBMS_AQADM_SYS` package is shipped as undocumented and allows to run SQL commands as user SYS.

Rationale:

As use of the `DBMS_AQADM_SYS` package could allow an unauthorized user to run SQL commands as user SYS.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_AQADM_SYS';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_AQADM_SYS FROM PUBLIC;
```

4.2.8 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_STREAMS_RPC' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DBMS_STREAMS_RPC` package is shipped as undocumented and allows to run SQL commands as user SYS.

Rationale:

As use of the `DBMS_STREAMS_RPC` package could allow an unauthorized user to run SQL commands as user SYS.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_STREAMS_RPC';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_STREAMS_RPC FROM PUBLIC;
```

References:

1. http://securityvulns.ru/files/ohh-indirect-privilege-escalation.pdf

4.2.9 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_PRVTAQIM' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DBMS_PRVTAQIM` package is shipped as undocumented and allows to run SQL commands as user SYS.

Rationale:

As use of the `DBMS_PRVTAQIM` package could allow an unauthorized user to escalate privileges because any SQL statements could be executed as user SYS.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_PRVTAQIM';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_PRVTAQIM FROM PUBLIC;
```

References:

1. http://securityvulns.ru/files/ohh-indirect-privilege-escalation.pdf

4.2.10 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'LTADM' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `LTADM` package is shipped as undocumented and allows privilege escalation if granted to unprivileged users.

Rationale:

As use of the `LTADM` package could allow an unauthorized user to run any SQL command as user SYS.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='LTADM';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON LTADM FROM PUBLIC;
```

References:

1. http://securityvulns.ru/files/ohh-indirect-privilege-escalation.pdf

4.2.11 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'WWV_DBMS_SQL' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `WWV_DBMS_SQL` package is shipped as undocumented and allows Oracle Application Express to run dynamic SQL statements.

Rationale:

As use of the `WWV_DBMS_SQL` package could allow an unauthorized user to run SQL statements as Application Express (APEX) user.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='WWV_DBMS_SQL';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON WWV_DBMS_SQL FROM PUBLIC;
```

4.2.12 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'WWV_EXECUTE_IMMEDIATE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `WWV_EXECUTE_IMMEDIATE` package is shipped as undocumented and allows Oracle Application Express to run dynamic SQL statements.

Rationale:

As use of the `WWV_EXECUTE_IMMEDIATE` package could allow an unauthorized user to run SQL statements as Application Express (APEX) user.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='WWV_EXECUTE_IMMEDIATE';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON WWV_EXECUTE_IMMEDIATE FROM PUBLIC;
```

References:

1. http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2008-1811

4.2.13 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_IJOB' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DBMS_IJOB` package is shipped as undocumented and allows to run database jobs in the context of another user.

Rationale:

As use of the `DBMS_IJOB` package could allow an attacker to change identities by using a different username to execute a database job.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_IJOB';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_IJOB FROM PUBLIC;
```

4.2.14 Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_FILE_TRANSFER' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DBMS_FILE_TRANSFER` package allows to transfer files from one database server to another.

Rationale:

As use of the `DBMS_FILE_TRANSFER` package could allow to transfer files from one database server to another.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME='DBMS_FILE_TRANSFER';
```

The assessment fails if results are returned.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ON DBMS_FILE_TRANSFER FROM PUBLIC;
```

References:

1. http://docs.oracle.com/database/121/ARPLS/d_ftran.htm#ARPLS095

4.3 Revoke Excessive System Privileges

The recommendations within this section revoke excessive system privileges.

4.3.1 Ensure 'SELECT_ANY_DICTIONARY' Is Revoked from Unauthorized 'GRANTEE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `SELECT ANY DICTIONARY` privilege allows the designated user to access SYS schema objects.

Rationale:

The Oracle database `SELECT ANY DICTIONARY` privilege allows the designated user to access SYS schema objects. The Oracle password hashes are part of the SYS schema and can be selected using SELECT ANY DICTIONARY privileges.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='SELECT ANY DICTIONARY'
AND GRANTEE NOT IN ('DBA','DBSNMP','OEM_MONITOR',
                    'OLAPSYS','ORACLE_OCM','SYSMAN','WMSYS','SYSBACKUP','SYSDG');
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE SELECT_ANY_DICTIONARY FROM <grantee>;
```

References:

1. http://docs.oracle.com/database/121/DBSEG/authorization.htm#DBSEG99870
2. http://docs.oracle.com/database/121/REFRN/GUID-10024282-6729-4C66-8679-FD653C9C7DE7.htm#REFRN-GUID-10024282-6729-4C66-8679-FD653C9C7DE7
3. http://arup.blogspot.de/2011/07/difference-between-select-any.html

4.3.2 Ensure 'SELECT ANY TABLE' Is Revoked from Unauthorized 'GRANTEE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `SELECT ANY TABLE` privilege allows the designated user to open any table, except of SYS, to view it.

Rationale:

As assignment of the `SELECT ANY TABLE` privilege can allow the unauthorized viewing of sensitive data, this capability should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='SELECT ANY TABLE'
AND GRANTEE NOT IN ('DBA', 'MDSYS', 'SYS', 'IMP_FULL_DATABASE', 'EXP_FULL_DATABASE',
                    'DATAPUMP_IMP_FULL_DATABASE', 'WMSYS', 'SYSTEM','OLAP_DBA',
                    'DV_REALM_OWNER');
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE SELECT ANY TABLE FROM <grantee>;
```

References:

1. http://docs.oracle.com/database/121/SQLRF/statements_10002.htm#SQLRF01702

Notes:

If the '`O7_DICTIONARY_ACCESSIBILITY`' has been set to `TRUE` (non-default setting) then the `SELECT ANY TABLE` privilege provides access to `SYS` objects.

4.3.3 Ensure 'AUDIT SYSTEM' Is Revoked from Unauthorized 'GRANTEE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `AUDIT SYSTEM` privilege allows the change auditing activities on the system.

Rationale:

As assignment of the `AUDIT SYSTEM` privilege can allow the unauthorized alteration of system audit activities, disabling the creation of audit trails, this capability should be restricted according to the needs of the organization.

Audit:

To asses this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='AUDIT SYSTEM'
AND GRANTEE NOT IN ('DBA','DATAPUMP_IMP_FULL_DATABASE','IMP_FULL_DATABASE',
                    'SYS','AUDIT_ADMIN');
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE AUDIT SYSTEM FROM <grantee>;
```

References:

1. http://docs.oracle.com/database/121/SQLRF/statements_4007.htm#SQLRF01107
2. http://docs.oracle.com/database/121/SQLRF/statements_4008.htm#SQLRF56110

4.3.4 Ensure 'EXEMPT ACCESS POLICY' Is Revoked from Unauthorized 'GRANTEE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `EXEMPT ACCESS POLICY` keyword provides the user the capability to access all the table rows regardless of row-level security lockouts.

Rationale:

As assignment of the `EXEMPT ACCESS POLICY` privilege can allow an unauthorized user to potentially access/change confidential data, this capability should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='EXEMPT ACCESS POLICY';
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXEMPT ACCESS POLICY FROM <grantee>;
```

References:

1. http://docs.oracle.com/database/121/DBSEG/audit_config.htm#DBSEG703
2. http://docs.oracle.com/database/121/DBSEG/vpd.htm#CIHEEAFJ

4.3.5 Ensure 'BECOME USER' Is Revoked from Unauthorized 'GRANTEE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `BECOME USER` privilege allows the designated user to inherit the rights of another user.

Rationale:

As assignment of the `BECOME USER` privilege can allow the unauthorized use of another user's privileges, this capability should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='BECOME USER'
AND GRANTEE NOT IN ('DBA','SYS','IMP_FULL_DATABASE');
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE BECOME USER FROM <grantee>;
```

References:

1. http://docs.oracle.com/database/121/DBSEG/guidelines.htm#DBSEG499

4.3.6 Ensure 'CREATE_PROCEDURE' Is Revoked from Unauthorized 'GRANTEE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `CREATE PROCEDURE` privilege allows the designated user to create a stored procedure that will fire when given the correct command sequence.

Rationale:

As assignment of the `CREATE PROCEDURE` privilege can lead to severe problems in unauthorized hands, such as rogue procedures facilitating data theft or Denial-of-Service by corrupting data tables, this capability should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='CREATE PROCEDURE'
AND GRANTEE NOT IN ( 'DBA','DBSNMP','MDSYS','OLAPSYS','OWB$CLIENT',
        'OWBSYS','RECOVERY_CATALOG_OWNER','SPATIAL_CSW_ADMIN_USR',
        'SPATIAL_WFS_ADMIN_USR','SYS','APEX_030200','APEX_040000',
        'APEX_040100','APEX_040200','DVF','RESOURCE','DV_REALM_RESOURCE',
'APEX_GRANTS_FOR_NEW_USERS_ROLE','APEX_050000','MGMT_VIEW',
'SYSMAN_MDS','SYSMAN_OPSS','SYSMAN_RO','SYSMAN_STB');
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE CREATE_PROCEDURE FROM <grantee>;
```

References:

1. http://docs.oracle.com/database/121/DBSEG/guidelines.htm#DBSEG499

4.3.7 Ensure 'ALTER SYSTEM' Is Revoked from Unauthorized 'GRANTEE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `ALTER SYSTEM` privilege allows the designated user to dynamically alter the instance's running operations.

Rationale:

As assignment of the `ALTER SYSTEM` privilege can lead to severe problems, such as the instance's session being killed or the stopping of redo log recording, which would make transactions unrecoverable, this capability should be severely restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='ALTER SYSTEM'
AND GRANTEE NOT IN ('SYS','SYSTEM','APEX_030200','APEX_040000',
 'APEX_040100','APEX_040200','DBA','EM_EXPRESS_ALL','SYSBACKUP','GSMADMIN_ROLE',
 'GSM_INTERNAL','SYSDG','GSMADMIN_INTERNAL');
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE ALTER SYSTEM FROM <grantee>;
```

References:

1. http://docs.oracle.com/database/121/DBSEG/guidelines.htm#DBSEG499

4.3.8 Ensure 'CREATE ANY LIBRARY' Is Revoked from Unauthorized 'GRANTEE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `CREATE ANY LIBRARY` privilege allows the designated user to create objects that are associated to the shared libraries.

Rationale:

As assignment of the `CREATE ANY LIBRARY` privilege can allow the creation of numerous library-associated objects and potentially corrupt the libraries' integrity, this capability should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='CREATE ANY LIBRARY'
AND GRANTEE NOT IN ('SYS','SYSTEM','DBA','IMP_FULL_DATABASE');
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE CREATE ANY LIBRARY FROM <grantee>;
```

References:

1. http://docs.oracle.com/database/121/DBSEG/guidelines.htm#DBSEG499
2. http://docs.oracle.com/database/121/ADMIN/manproc.htm#ADMIN00501

Notes:

Oracle has 2 identical privileges: CREATE LIBRARY and CREATE ANY LIBRARY.

4.3.9 Ensure 'CREATE LIBRARY' Is Revoked from Unauthorized 'GRANTEE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `CREATE LIBRARY` privilege allows the designated user to create objects that are associated to the shared libraries.

Rationale:

As assignment of the `CREATE LIBRARY` privilege can allow the creation of numerous library-associated objects and potentially corrupt the libraries' integrity, this capability should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='CREATE LIBRARY'
AND GRANTEE NOT IN ('SYS','SYSTEM','DBA','MDSYS','SPATIAL_WFS_ADMIN_USR',
                    'SPATIAL_CSW_ADMIN_USR','DVSYS','GSMADMIN_INTERNAL','XDB');
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE CREATE LIBRARY FROM <grantee>;
```

References:

1. http://docs.oracle.com/database/121/DBSEG/guidelines.htm#DBSEG499
2. http://docs.oracle.com/database/121/ADMIN/manproc.htm#ADMIN00501

Notes:

Oracle has 2 identical privileges: CREATE LIBRARY and CREATE ANY LIBRARY.

4.3.10 Ensure 'GRANT ANY OBJECT PRIVILEGE' Is Revoked from Unauthorized 'GRANTEE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `GRANT ANY OBJECT PRIVILEGE` keyword provides the grantee the capability to grant access to any single or multiple combinations of objects to any grantee in the catalog of the database.

Rationale:

As authorization to use the `GRANT ANY OBJECT PRIVILEGE` capability can allow an unauthorized user to potentially access/change confidential data or damage the data catalog due to potential complete instance access, this capability should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='GRANT ANY OBJECT PRIVILEGE'
AND GRANTEE NOT IN ('DBA','SYS','IMP_FULL_DATABASE','DATAPUMP_IMP_FULL_DATABASE',
                    'EM_EXPRESS_ALL', 'DV_REALM_OWNER');
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE GRANT ANY OBJECT PRIVILEGE FROM <grantee>;
```

References:

1. http://docs.oracle.com/database/121/DBSEG/authorization.htm#DBSEG99914

4.3.11 Ensure 'GRANT ANY ROLE' Is Revoked from Unauthorized 'GRANTEE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `GRANT ANY ROLE` keyword provides the grantee the capability to grant any single role to any grantee in the catalog of the database.

Rationale:

As authorization to use the `GRANT ANY ROLE` capability can allow an unauthorized user to potentially access/change confidential data or damage the data catalog due to potential complete instance access, this capability should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='GRANT ANY ROLE'
AND GRANTEE NOT IN ('DBA','SYS','DATAPUMP_IMP_FULL_DATABASE','IMP_FULL_DATABASE',
                    'SPATIAL_WFS_ADMIN_USR','SPATIAL_CSW_ADMIN_USR',
                    'GSMADMIN_INTERNAL','DV_REALM_OWNER', 'EM_EXPRESS_ALL',
'DV_OWNER');
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE GRANT ANY ROLE FROM <grantee>;
```

References:

1. http://docs.oracle.com/database/121/DBSEG/authorization.htm#DBSEG99945

4.3.12 Ensure 'GRANT ANY PRIVILEGE' Is Revoked from Unauthorized 'GRANTEE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `GRANT ANY PRIVILEGE` keyword provides the grantee the capability to grant any single privilege to any item in the catalog of the database.

Rationale:

As authorization to use the `GRANT ANY PRIVILEGE` capability can allow an unauthorized user to potentially access/change confidential data or damage the data catalog due to potential complete instance access, this capability should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='GRANT ANY PRIVILEGE'
AND GRANTEE NOT IN ('DBA','SYS','IMP_FULL_DATABASE','DATAPUMP_IMP_FULL_DATABASE'
                    'DV_REALM_OWNER', 'EM_EXPRESS_ALL');
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE GRANT ANY PRIVILEGE FROM <grantee>;
```

References:

1. http://docs.oracle.com/database/121/DBSEG/authorization.htm#DBSEG99945

4.4 Revoke Role Privileges

The recommendations within this section intend to revoke powerful roles where they are likely not needed.

4.4.1 Ensure 'DELETE_CATALOG_ROLE' Is Revoked from Unauthorized 'GRANTEE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DELETE_CATALOG_ROLE` provides `DELETE` privileges for the records in the system's audit table (`AUD$`).

Rationale:

As permitting unauthorized access to the `DELETE_CATALOG_ROLE` can allow the destruction of audit records vital to the forensic investigation of unauthorized activities, this capability should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, GRANTED_ROLE
FROM DBA_ROLE_PRIVS
WHERE granted_role='DELETE_CATALOG_ROLE'
AND GRANTEE NOT IN ('DBA','SYS');
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE DELETE_CATALOG_ROLE FROM <grantee>;
```

References:

1. http://docs.oracle.com/database/121/DBSEG/authorization.htm#BABFCAFH

4.4.2 Ensure 'SELECT_CATALOG_ROLE' Is Revoked from Unauthorized 'GRANTEE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `SELECT_CATALOG_ROLE` provides `SELECT` privileges on all data dictionary views held in the `SYS` schema.

Rationale:

As permitting unauthorized access to the `SELECT_CATALOG_ROLE` can allow the disclosure of all dictionary data, this capability should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, GRANTED_ROLE
FROM DBA_ROLE_PRIVS
WHERE granted_role='SELECT_CATALOG_ROLE'
AND grantee not in ('DBA','SYS','IMP_FULL_DATABASE','EXP_FULL_DATABASE',
'OEM_MONITOR', 'SYSBACKUP','EM_EXPRESS_BASIC','SYSMAN');
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE SELECT_CATALOG_ROLE FROM <grantee>;
```

References:

1. http://docs.oracle.com/database/121/DBSEG/authorization.htm#BABFCAFH

4.4.3 Ensure 'EXECUTE_CATALOG_ROLE' Is Revoked from Unauthorized 'GRANTEE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `EXECUTE_CATALOG_ROLE` provides `EXECUTE` privileges for a number of packages and procedures in the data dictionary in the `SYS` schema.

Rationale:

As permitting unauthorized access to the `EXECUTE_CATALOG_ROLE` can allow the disruption of operations by initialization of rogue procedures, this capability should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, GRANTED_ROLE
FROM DBA_ROLE_PRIVS
WHERE granted_role='EXECUTE_CATALOG_ROLE'
AND grantee not in ('DBA','SYS','IMP_FULL_DATABASE','EXP_FULL_DATABASE');
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE_CATALOG_ROLE FROM <grantee>;
```

References:

1. http://docs.oracle.com/database/121/DBSEG/authorization.htm#BABFCAFH

4.4.4 Ensure 'DBA' Is Revoked from Unauthorized 'GRANTEE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database DBA role is the default database administrator role provided for the allocation of administrative privileges.

Rationale:

As assignment of the `DBA` role to an ordinary user can provide a great number of unnecessary privileges to that user and opens the door to data breaches, integrity violations, and Denial-of-Service conditions, application of this role should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, GRANTED_ROLE
FROM DBA_ROLE_PRIVS
WHERE GRANTED_ROLE='DBA'
AND GRANTEE NOT IN ('SYS','SYSTEM');
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE DBA FROM <grantee>;
```

References:

1. http://docs.oracle.com/database/121/DBSEG/authorization.htm#DBSEG4414

4.5 Revoke Excessive Table and View Privileges

The recommendations within this section intend to revoke excessive table and view privileges.

4.5.1 Ensure 'ALL' Is Revoked from Unauthorized 'GRANTEE' on 'AUD$' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `SYS.AUD$` table contains all the audit records for the database of the non-Data Manipulation Language (DML) events, such as `ALTER`, `DROP`, `CREATE`, and so forth. (DML changes need trigger-based audit events to record data alterations.)

Rationale:

As permitting non-privileged users the authorization to manipulate the `SYS_AUD$` table can allow distortion of the audit records, hiding unauthorized activities, this capability should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE TABLE_NAME='AUD$'
AND GRANTEE NOT IN ('DELETE_CATALOG_ROLE');
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE ALL ON AUD$ FROM <grantee>;
```

References:

1. http://docs.oracle.com/database/121/DBSEG/audit_admin.htm#DBSEG629

4.5.2 Ensure 'ALL' Is Revoked from Unauthorized 'GRANTEE' on 'USER_HISTORY$' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `SYS.USER_HISTORY$` table contains all the audit records for the user's password change history. (This table gets updated by password changes if the user has an assigned profile that has password reuse limit set, e.g., `PASSWORD_REUSE_TIME` set to other than `UNLIMITED`.)

Rationale:

As permitting non-privileged users the authorization to manipulate the records in the `SYS.USER_HISTORY$` table can allow distortion of the audit trail, potentially hiding unauthorized data confidentiality attacks or integrity changes, this capability should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE TABLE_NAME='USER_HISTORY$' AND OWNER = 'SYS';
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE ALL ON USER_HISTORY$ FROM <grantee>;
```

References:

1. http://marcel.vandewaters.nl/oracle/database-oracle/password-history-reusing-a-password

Notes:

USER_HISTORY$ contains only the old, case-insensitive passwords.

4.5.3 Ensure 'ALL' Is Revoked from Unauthorized 'GRANTEE' on 'LINK$' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `SYS.LINK$` table contains all the user's password information and data table link information.

Rationale:

As permitting non-privileged users to manipulate or view the `SYS.LINK$` table can allow capture of password information and/or corrupt the primary database linkages, this capability should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE TABLE_NAME='LINK$'
AND GRANTEE NOT IN ('DV_SECANALYST')
AND OWNER='SYS';
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE ALL ON LINK$ FROM <grantee>;
```

4.5.4 Ensure 'ALL' Is Revoked from Unauthorized 'GRANTEE' on 'SYS.USER$' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `SYS.USER$` table contains the users' hashed password information.

Rationale:

As permitting non-privileged users the authorization to open the `SYS.USER$` table can allow the capture of password hashes for the later application of password cracking algorithms to breach confidentiality, this capability should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE TABLE_NAME='USER$' AND OWNER='SYS'
AND GRANTEE NOT IN ('CTXSYS','XDB','APEX_030200','SYSMAN',
     'APEX_040000','APEX_040100','APEX_040200','DV_SECANALYST','DVSYS','ORACLE_OCM');
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE ALL ON SYS.USER$ FROM <username>;
```

References:

1. http://dba.stackexchange.com/questions/17513/what-do-the-columns-in-sys-user-represent

4.5.5 Ensure 'ALL' Is Revoked from Unauthorized 'GRANTEE' on 'DBA_%' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `DBA_` views show all information which is relevant to administrative accounts.

Rationale:

As permitting users the authorization to manipulate the `DBA_` views can expose sensitive data.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT grantee||'.'||table_name FROM DBA_TAB_PRIVS
WHERE TABLE_NAME LIKE 'DBA_%'
AND GRANTEE NOT IN ('DBA','AUDIT_ADMIN','AUDIT_VIEWER','CAPTURE_ADMIN','DVSYS',

'SYSDG','DV_SECANALYST','SYSKM','DV_MONITOR','ORACLE_OCM','DV_ACCTMGR',

'GSMADMIN_INTERNAL','XDB','SYS','APPQOSSYS','AQ_ADMINISTRATOR_ROLE','CTXSYS',

'EXFSYS','MDSYS','OLAP_XS_ADMIN','OLAPSYS','ORDSYS','OWB$CLIENT','OWBSYS',
                    'SELECT_CATALOG_ROLE','WM_ADMIN_ROLE','WMSYS','XDBADMIN',
                    'LBACSYS', 'ADM_PARALLEL_EXECUTE_TASK','CISSCANROLE')
AND NOT REGEXP_LIKE(grantee,'^APEX_0[3-9][0-9][0-9][0-9][0-9]$');
```

Lack of results implies compliance.

Note: An organization should perform proper impact analysis before revoking grants on DBA_ objects.

Remediation:

Replace <*Non-DBA/SYS grantee*>, in the query below, with the Oracle login(s) or role(s) returned from the associated audit procedure and execute:

```
REVOKE ALL ON DBA_ FROM <Non-DBA/SYS grantee>;
```

References:

1. http://docs.oracle.com/database/121/REFRN/GUID-10024282-6729-4C66-8679-FD653C9C7DE7.htm#REFRN-GUID-10024282-6729-4C66-8679-FD653C9C7DE7

4.5.6 Ensure 'ALL' Is Revoked from Unauthorized 'GRANTEE' on 'SYS.SCHEDULER$_CREDENTIAL' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `SCHEDULER$_CREDENTIAL` table contains the database scheduler credential information.

Rationale:

As permitting non-privileged users the authorization to open the `SYS.SCHEDULER$_CREDENTIAL` table.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_TAB_PRIVS
WHERE TABLE_NAME='SCHEDULER$_CREDENTIAL' AND OWNER='SYS';
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE ALL ON SYS.SCHEDULER$_CREDENTIAL FROM <username>;
```

References:

1. http://docs.oracle.com/database/121/ADMIN/schedadmin.htm#ADMIN12073
2. http://berxblog.blogspot.de/2012/02/restore-dbmsschedulercreatecredential.html

Notes:

** *_SCHEDULER_CREDENTIALS is deprecated in Oracle Database 12*c*, but remains available, for reasons of backward compatibility.

4.5.7 Ensure 'SYS.USER$MIG' Has Been Dropped (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The table `sys.user$mig` is created during migration and contains the Oracle password hashes before the migration starts.

Rationale:

The table `sys.user$mig` is not deleted after the migration. An attacker could access the table containing the Oracle password hashes.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT OWNER, TABLE_NAME
FROM ALL_TABLES
WHERE OWNER='SYS'
AND TABLE_NAME='USER$MIG';
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
DROP TABLE SYS.USER$MIG;
```

4.6 Ensure '%ANY%' Is Revoked from Unauthorized 'GRANTEE' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `ANY` keyword provides the user the capability to alter any item in the catalog of the database.

Rationale:

As authorization to use the `ANY` expansion of a privilege can allow an unauthorized user to potentially change confidential data or damage the data catalog, this capability should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE LIKE '%ANY%'
AND GRANTEE NOT IN ('AQ_ADMINISTRATOR_ROLE','DBA','DBSNMP','EXFSYS',
          'EXP_FULL_DATABASE','IMP_FULL_DATABASE','DATAPUMP_IMP_FULL_DATABASE',
          'JAVADEBUGPRIV','MDSYS','OEM_MONITOR','OLAPSYS','OLAP_DBA','ORACLE_OCM',
          'OWB$CLIENT','OWBSYS','SCHEDULER_ADMIN','SPATIAL_CSW_ADMIN_USR',
          'SPATIAL_WFS_ADMIN_USR','SYS','SYSMAN','SYSTEM','WMSYS','APEX_030200',
          'APEX_040000','APEX_040100','APEX_040200','LBACSYS','SYSBACKUP',
          'CTXSYS','OUTLN','DVSYS','ORDPLUGINS','ORDSYS','RECOVERY_CATALOG_OWNER_VPD',
          'GSMADMIN_INTERNAL','XDB','SYSDG','AUDIT_ADMIN','DV_OWNER','DV_REALM_OWNER',
          'EM_EXPRESS_ALL','RECOVERY_CATALOG_OWNER','APEX_050000','SYSMAN_STB',
          'SYSMAN_TYPES');
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE '<ANY Privilege>' FROM <grantee>;
```

References:

1. http://docs.oracle.com/database/121/DBSEG/authorization.htm#DBSEG99877

4.7 Ensure 'DBA_SYS_PRIVS.%' Is Revoked from Unauthorized 'GRANTEE' with 'ADMIN_OPTION' Set to 'YES' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `WITH_ADMIN` privilege allows the designated user to grant another user the same privileges.

Rationale:

As assignment of the `WITH_ADMIN` privilege can allow the granting of a restricted privilege to an unauthorized user, this capability should be restricted according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE ADMIN_OPTION='YES'
AND GRANTEE not in ('AQ_ADMINISTRATOR_ROLE','DBA','OWBSYS',
                    'SCHEDULER_ADMIN','SYS','SYSTEM','WMSYS',
                    'DVSYS','SYSKM','DV_ACCTMGR')
AND NOT REGEXP_LIKE(grantee,'^APEX_0[3-9][0-9][0-9][0-9][0-9]$');
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE <privilege> FROM <grantee>;
```

4.8 Ensure Proxy Users Have Only 'CONNECT' Privilege (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

Do not grant privileges directly to proxy users.

Rationale:

A proxy user should only have the ability to connect to the database.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE,GRANTED_ROLE FROM DBA_ROLE_PRIVS WHERE GRANTEE IN (SELECT PROXY FROM
DBA_PROXIES) AND GRANTED_ROLE NOT IN ('CONNECT')
UNION
SELECT GRANTEE,PRIVILEGE FROM DBA_SYS_PRIVS WHERE GRANTEE IN (SELECT PROXY FROM
DBA_PROXIES) AND PRIVILEGE NOT IN ('CREATE SESSION')
UNION
SELECT GRANTEE,PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTEE IN (SELECT PROXY FROM
DBA_PROXIES);
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement for each `[PRIVILEGE]` returned by running the audit procedure.

```
REVOKE [PRIVILEGE] FROM <proxy_user>;
```

4.9 Ensure 'EXECUTE ANY PROCEDURE' Is Revoked from 'OUTLN' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

Remove unneeded privileges from OUTLN.

Rationale:

Migrated OUTLN users have more privileges than required.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='EXECUTE ANY PROCEDURE'
AND GRANTEE='OUTLN';
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ANY PROCEDURE FROM OUTLN;
```

4.10 Ensure 'EXECUTE ANY PROCEDURE' Is Revoked from 'DBSNMP' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing
- Level 1 - RDBMS using Unified Auditing

Description:

Remove unneeded privileges from DBSNMP.

Rationale:

Migrated DBSNMP users have more privileges than required.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='EXECUTE ANY PROCEDURE'
AND GRANTEE='DBSNMP';
```

Lack of results implies compliance.

Remediation:

To remediate this setting execute the following SQL statement.

```
REVOKE EXECUTE ANY PROCEDURE FROM DBSNMP;
```

5 Audit/Logging Policies and Procedures

The ability to audit database activities is among the most important of all database security features. Decisions must be made regarding the scope of auditing since auditing has costs - in storage for the audit trail and in performance impact on audited operations - and perhaps even the database or system in general. There is also the additional cost to manage (store, backup, secure) and review the data in audit trail.

Measures must be taken to protect the audit trail itself, for it may be targeted for alteration or destruction to hide unauthorized activity. For an audit destination outside the database, the recommendations are elsewhere in this document. Auditing recommendations for potential database audit destinations is below.

Auditing "by session" typically creates fewer (until 11g) and slightly smaller audit records, but is discouraged in most situations since there is some loss of fidelity (e.g. object privilege `GRANTEE`). More detailed auditing creates larger audit records. The `AUDIT_TRAIL` initialization parameter (for DB|XML, extended - or not) is the main determining factor for the size of a given audit record - and a notable factor in the performance cost, although the largest of the latter is DB versus OS or XML.

This section deals with standard Oracle auditing since auditing of privileged connections (as `sysdba` or `sysoper`) is configured via the `AUDIT_SYS_OPERATIONS` initialization parameter and is otherwise not configurable. The basic types of standard auditing are object auditing, statement auditing and privilege auditing and each behaves differently.

Object auditing applies to specific objects for which it is invoked and always applies to all users. This type of auditing is usually employed to audit application-specific sensitive objects, but can be used to protect the audit trail in the database.

Privilege auditing audits the use of specific system privileges, but typically only if the user actually possesses the audited privilege. Attempts that fail for lack of the audited privilege are typically not audited. This is the main weakness of privilege auditing and why statement auditing is usually preferred, if the option exists.

Statement auditing audits the issuance of certain types of statements, usually without regard to privilege or lack thereof. Both privilege and statement audits may be specified for specific users or all users (the default).

5.1 Traditional Auditing

This section is to be followed if traditional auditing is implemented.

5.1.1 Enable 'USER' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The USER object in the Oracle database an account through which a connection may be made to interact with the database according to the roles and privileges allotted to account. It is also a schema which may own database objects. This audits all activities and requests to create, drop or alter a user, including a user changing their own password. (The latter is not audited by 'audit ALTER USER'.)

Rationale:

Any unauthorized attempts to create, drop or alter a user should cause concern, whether successful or not. It can also be useful in forensics if an account is compromised and is mandated by many common security initiatives. An abnormally high number of these activities in a given period might be worth investigation. Any failed attempt to drop a user or create a user may be worth further review.

Audit:

To assess this recommendation, execute the following SQL Statement.

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='USER'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT USER;
```

Impact:

This would the current 5.2 (audit CREATE USER), 5.3 (audit ALTER USER), and 5.4 (audit DROP USER) privilege audits with the single statement auditing option "audit USER". Any action audited by those three privilege audits would also be audited by this. In addition, this would audit:

1. Attempts to create user by anyone without the CREATE USER system privilege.
2. Attempts to drop user by anyone without the DROP USER system privilege
3. Attempts to alter user by anyone without the ALTER USER system privilege
4. Users changing or attempting to change their own passwords (which is not done by auditing ALTER USER).

5.1.2 Enable 'ALTER USER' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The `USER` object for the Oracle database is a specification of an object which is an account through which either a human or an application can connect to, via a JDBC or log into, via a CLI, and interact with the database instance according to the roles and privileges allotted to account.

Rationale:

As the logging of user activities involving the creation, alteration, or dropping of a USER can provide forensic evidence about a pattern of suspect/unauthorized activities, the audit capability should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='ALTER USER'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT ALTER USER;
```

5.1.3 Enable 'DROP USER' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The `USER` object for the Oracle database is a specification of an object which is an account through which either a human or an application can connect to, via a JDBC or log into, via a CLI, and interact with the database instance according to the roles and privileges allotted to account.

Rationale:

As the logging of user activities involving the creation, alteration, or dropping of a USER can provide forensic evidence about a pattern of suspect/unauthorized activities, the audit capability should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='DROP USER'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT DROP USER;
```

5.1.4 Enable 'ROLE' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The ROLE object allows for the creation of a set of privileges that can be granted to users or other roles. This audits all attempts, successful or not, to create, drop, alter or set roles.

Rationale:

Roles are a key database security infrastructure component. Any attempt to create, drop or alter a role should be audited. This statement auditing option also audits attempts, successful or not, to set a role in a session. Any unauthorized attempts to create, drop or alter a role may be worthy of investigation. Attempts to set a role by users without the role privilege may warrant investigation.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='ROLE'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting:

```
AUDIT ROLE;
```

Impact:

The change to the audit/check is to ensure that the audit is in effect for all users, regardless of proxy or success.

The change to the title, description and rationale are to better clarify what it actually does. (e.g. It does NOT audit "all ROLE activities/requests". For example, it does not audit role grants and revokes.)

5.1.5 Enable 'SYSTEM GRANT' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

This will audit any attempt, successful or not, to grant or revoke any system privilege or role - regardless of privilege held by the user attempting the operation.

Rationale:

Logging of all grant and revokes (roles and system privileges) can provide forensic evidence about a pattern of suspect/unauthorized activities. Any unauthorized attempt may be cause for further investigation.

Audit:

To assess this recommendation execute the following SQL statement.

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='SYSTEM GRANT'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT SYSTEM GRANT;
```

5.1.6 Enable 'PROFILE' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The PROFILE object allows for the creation of a set of database resource limits that can be assigned to a user, so that that user cannot exceed those resource limitations. This will audit all attempts, successful or not, to create, drop or alter any profile.

Rationale:

As profiles are part of the database security infrastructure, auditing the modification of profiles is recommended.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='PROFILE'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT PROFILE;
```

Impact:

The statement auditing option 'audit PROFILE' audits everything that the three privilege audits 'audit CREATE PROFILE', 'audit DROP PROFILE' and 'audit ALTER PROFILE' do, but also audits:

1. Attempts to create a profile by a user without the CREATE PROFILE system privilege.
2. Attempts to drop a profile by a user without the DROP PROFILE system privilege
3. Attempts to alter a profile by a user without the ALTER PROFILE system privilege.

5.1.7 Enable 'ALTER PROFILE' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The `PROFILE` object allows for the creation of a set of database resource limits that can be assigned to a user, so that that user cannot exceed those resource limitations.

Rationale:

As the logging of user activities involving the creation, alteration, or dropping of a `PROFILE` can provide forensic evidence about a pattern of unauthorized activities, the audit capability should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='ALTER PROFILE'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT ALTER PROFILE;
```

5.1.8 Enable 'DROP PROFILE' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The `PROFILE` object allows for the creation of a set of database resource limits that can be assigned to a user, so that that user cannot exceed those resource limitations.

Rationale:

As the logging of user activities involving the creation, alteration, or dropping of a `PROFILE` can provide forensic evidence about a pattern of unauthorized activities, the audit capability should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='DROP PROFILE'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT DROP PROFILE;
```

5.1.9 Enable 'DATABASE LINK' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

All activities on database links should be audited.

Rationale:

As the logging of user activities involving the creation or dropping of a `DATABASE LINK` can provide forensic evidence about a pattern of unauthorized activities, the audit capability should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='DATABASE LINK'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT DATABASE LINK;
```

References:

1. http://docs.oracle.com/database/121/DBSEG/audit_config.htm#DBSEG1115

5.1.10 Enable 'PUBLIC DATABASE LINK' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The `PUBLIC DATABASE LINK` object allows for the creation of a public link for an application-based "user" to access the database for connections/session creation.

Rationale:

As the logging of user activities involving the creation, alteration, or dropping of a `PUBLIC DATABASE LINK` can provide forensic evidence about a pattern of unauthorized activities, the audit capability should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='PUBLIC DATABASE LINK'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT PUBLIC DATABASE LINK;
```

5.1.11 Enable 'PUBLIC SYNONYM' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The `PUBLIC SYNONYM` object allows for the creation of an alternate description of an object and public synonyms are accessible by all users that have the appropriate privileges to the underlying object.

Rationale:

As the logging of user activities involving the creation or dropping of a `PUBLIC SYNONYM` can provide forensic evidence about a pattern of unauthorized activities, the audit capability should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='PUBLIC SYNONYM'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT PUBLIC SYNONYM;
```

5.1.12 Enable 'SYNONYM' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The `SYNONYM` operation allows for the creation of an alternative name for a database object such as a Java class schema object, materialized view, operator, package, procedure, sequence, stored function, table, view, user-defined object type, even another synonym; this synonym puts a dependency on its target and is rendered invalid if the target object is changed/dropped.

Rationale:

As the logging of user activities involving the creation or dropping of a `SYNONYM` can provide forensic evidence about a pattern of suspect/unauthorized activities, the audit capability should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='SYNONYM'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT SYNONYM;
```

References:

1. http://docs.oracle.com/database/121/DBSEG/audit_config.htm#DBSEG1115

5.1.13 Enable 'GRANT DIRECTORY' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The `DIRECTORY` object allows for the creation of a directory object that specifies an alias for a directory on the server file system, where the external binary file LOBs (BFILEs)/ table data are located.

Rationale:

As the logging of user activities involving the creation or dropping of a DIRECTORY can provide forensic evidence about a pattern of unauthorized activities, the audit capability should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='GRANT DIRECTORY'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT GRANT DIRECTORY;
```

References:

1. http://docs.oracle.com/database/121/SQLRF/statements_4007.htm#SQLRF01107

Notes:

Grant directory is a shortcut
for GRANT privilege ON directory, REVOKE privilege ON directory.

5.1.14 Enable 'SELECT ANY DICTIONARY' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The `SELECT ANY DICTIONARY` capability allows the user to view the definitions of all schema objects in the database.

Rationale:

As the logging of user activities involving the capability to access the description of all schema objects in the database can provide forensic evidence about a pattern of unauthorized activities, the audit capability should be set according to the needs of the organization.

Audit:

To assess this recommendation execute the following SQL statement.

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='SELECT ANY DICTIONARY'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT SELECT ANY DICTIONARY;
```

References:

1. http://docs.oracle.com/database/121/DBSEG/guidelines.htm#DBSEG500

5.1.15 Enable 'GRANT ANY OBJECT PRIVILEGE' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

`GRANT ANY OBJECT PRIVILEGE` allows the user to grant or revoke any object privilege, which includes privileges on tables, directories, mining models, etc. This audits all uses of that privilege.

Rationale:

Logging of privilege grants that can lead to the creation, alteration, or deletion of critical data, the modification of objects, object privilege propagation and other such activities can be critical to forensic investigations.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE, SUCCESS, FAILURE
FROM DBA_PRIV_AUDIT_OPTS
WHERE PRIVILEGE='GRANT ANY OBJECT PRIVILEGE'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT GRANT ANY OBJECT PRIVILEGE;
```

5.1.16 Enable 'GRANT ANY PRIVILEGE' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

This audits all uses of the system privilege named `GRANT ANY PRIVILEGE`. Actions by users *not* holding this privilege are *not* audited.

Rationale:

`GRANT ANY PRIVILEGE` allows a user to grant any system privilege, including the most powerful privileges typically available only to administrators - to change the security infrastructure, to drop/add/modify users and more. Auditing the use of this privilege is part of a comprehensive auditing policy that can help in detecting issues and can be useful in forensics.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT PRIVILEGE, SUCCESS, FAILURE
FROM DBA_PRIV_AUDIT_OPTS
WHERE PRIVILEGE='GRANT ANY PRIVILEGE'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT GRANT ANY PRIVILEGE;
```

5.1.17 Enable 'DROP ANY PROCEDURE' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The `AUDIT DROP ANY PROCEDURE` command is auditing the creation of procedures in other schema.

Rationale:

Dropping procedures of another user could be part of an privilege escalation exploit and should be audited.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='DROP ANY PROCEDURE'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT DROP ANY PROCEDURE;
```

5.1.18 Enable 'ALL' Audit Option on 'SYS.AUD$' (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The logging of attempts to alter the audit trail in the `SYS.AUD$` table (open for read/update/delete/view) will provide a record of any activities that may indicate unauthorized attempts to access the audit trail.

Rationale:

As the logging of attempts to alter the `SYS.AUD$` table can provide forensic evidence of the initiation of a pattern of unauthorized activities, this logging capability should be set according to the needs of the organization.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT *
FROM DBA_OBJ_AUDIT_OPTS
WHERE OBJECT_NAME='AUD$'
AND ALT='A/A'
AND AUD='A/A'
AND COM='A/A'
AND DEL='A/A'
AND GRA='A/A'
AND IND='A/A'
AND INS='A/A'
AND LOC='A/A'
AND REN='A/A'
AND SEL='A/A'
AND UPD='A/A'
AND FBK='A/A';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT ALL ON SYS.AUD$ BY ACCESS;
```

5.1.19 Enable 'PROCEDURE' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

In this statement audit, "PROCEDURE" means any procedure, function, package or library. Any attempt, successful or not, to create or drop any of these types of objects is audited, regardless of privilege or lack thereof. Java schema objects (sources, classes, and resources) are considered the same as procedures for purposes of auditing SQL statements.

Rationale:

Any unauthorized attempts to create or drop a procedure in another's schema should cause concern, whether successful or not. Changes to critical store code can dramatically change the behavior of the application and produce serious security consequences, including privilege escalation and introducing SQL injection vulnerabilities. Audit records of such changes can be helpful in forensics.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='PROCEDURE'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT PROCEDURE;
```

Notes:

Be aware that not all auditing options work alike. In particular, the statement auditing option "audit PROCEDURE" does indeed audit create and drop library as well as all types of procedures and java schema objects. However, privilege audits do not work this way. So, for example, none of "audit CREATE ANY PROCEDURE", "audit DROP ANY PROCEDURE". '"audit CREATE PROCEDURE" will audit create or drop library activities. In statement auditing, "PROCEDURE" has a larger scope than in privilege auditing, where it is specific to functions, packages and procedures, but excludes libraries and perhaps other object types.

"Audit PROCEDURE" does not audit altering procedures, either in your own schema or in another via the ALTER ANY PROCEDURE system privilege. There seems to be no statement audit that is a better replacement for "Audit ALTER ANY PROCEDURE", but beware that will not create any audit records for users that do not have the privilege. Thus, attempts to alter procedures in one's own schema are never audited and attempts to alter procedures in another's schema that fail for lack of the ALTER ANY PROCEDURE privilege are not audited. This is simply a weakness in the current state of Oracle auditing. Fortunately, though, all that the ALTER command can be used for regarding procedures, functions, packages and libraries is compile options, so the inability to comprehensively audit alter procedure activities and requests is not as bad as it would be for other object types (USER, PROFIULE, etc.).

5.1.20 Enable 'ALTER SYSTEM' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

This will audit all attempts to ALTER SYSTEM, whether successful or not and regardless of whether or not the ALTER SYSTEM privilege is held by the user attempting the action.

Rationale:

Alter system allows one to change instance settings, including security settings and auditing options. Additionally, alter system can be used to run operating system commands using undocumented Oracle functionality. Any unauthorized attempt to alter the system should be cause for concern. Alterations outside of some specified maintenance window may be of concern. In forensics, these audit records could be quite useful.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='ALTER SYSTEM'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT ALTER SYSTEM;
```

5.1.21 Enable 'TRIGGER' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

A TRIGGER may be used to modify DML actions or invoke other (recursive) actions when some types of user-initiated actions occur. This will audit any attempt, successful or not, to create, drop, enable or disable any schema trigger in any schema regardless of privilege or lack thereof. For enabling and disabling a trigger, it covers both alter trigger and alter table.

Rationale:

Triggers are often part of schema security, data validation and other critical constraints upon actions and data. A trigger in another schema may be used to escalate privileges, redirect operations, transform data and perform other sorts of perhaps undesired actions. Any unauthorized attempt to create, drop or alter a trigger in another schema may be cause for investigation.

Audit:

To assess this recommendation execute the following SQL statement.

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='TRIGGER'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT TRIGGER;
```

Impact:

The statement auditing option 'audit TRIGGER' audits almost everything that the three privilege audits "audit CREATE ANY TRIGGER", "audit ALTER ANY TRIGGER" and "audit DROP ANY TRIGGER" audit, but also audits:

1. Statements to create, drop, enable or disable a trigger in the user's own schema.
2. Attempts to create a trigger by a user without the CREATE TRIGGER system privilege.
3. Attempts to create a trigger in another schema by users without the CREATE ANY TRIGGER privilege.
4. Attempts to drop a trigger in another schema by users without the DROP ANY TRIGGER privilege.
5. Attempts to disable or enable a trigger in another schema by users without the ALTER ANY TRIGGER privilege.

The one thing is audited by any of the three privilege audits that is not audited by this is "alter trigger ...compile" if the trigger is in another's schema, which is audited by "audit ALTER ANY TRIGGER"', but only if the user attempting the alteration actually holds the ALTER ANY TRIGGER system privilege. "Audit TRIGGER" only audits "alter table" or "alter trigger" statements used to enable or disable triggers. It does not audit alter trigger or alter table statements used only with compile options.

5.1.22 Enable 'CREATE SESSION' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

Audit all attempts to connect to the database, whether successful or not. Also, audits session disconnects/logoffs. The commands to audit SESSION, CONNECT or CREATE SESSION all accomplish *exactly* the same thing - they initiate statement auditing of the connect statement used to create a database session.

Rationale:

Auditing attempts to connect to the database is basic and mandated by most security initiatives. Any attempt to logon to a locked account, failed attempts to logon to default accounts or an unusually high number of failed logon attempts of any sort, for any user, in a particular time period may indicate an intrusion attempt. In forensics, the logon record may be first in a chain of evidence and contains information found in no other type of audit record for the session. Logon and logoff in the audit trail define the period and duration of the session.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION, SUCCESS, FAILURE
FROM DBA_STMT_AUDIT_OPTS
WHERE AUDIT_OPTION='CREATE SESSION'
AND USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
AUDIT SESSION;
```

Notes:

Although listing in the documentation as a privilege audit, 'audit CREATE SESSION' actually audits the CONNECT statement - as evidenced by the undocumented 'audit CONNECT' - which does exactly the same thing as 'audit SESSION' or 'audit CREATE SESSION'. There is no system privilege named either 'SESSION' or 'CONNECT' (CONNECT is a role, not a system privilege). Also, it behaves as statement auditing rather than privilege auditing in that it audits all attempts to create a session, even if the user does not hold the 'CREATE SESSION' system privilege.

5.2 Unified Auditing

This section is to be followed if unified auditing is implemented.

5.2.1 Enable 'CREATE USER' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

An Oracle database user is an account through which a connection to the database is established. A schema is associated with the user account which stores data. A user account may belong to an individual person or can be used by a device, process, and job or connection pool. `CREATE USER` statement is used to create Oracle database accounts and assign database properties to it. This unified audit action enables logging of all `CREATE USER` statements, whether successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to create user, whether successful or unsuccessful, may provide clues and forensic evidences about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all activities involving `CREATE USER`.

Audit:

To assess this recommendation, execute the following SQL Statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'CREATE USER'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies a finding.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
CREATE USER;
```

If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.2 Enable 'ALTER USER' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

An Oracle database user is an account through which a connection to the database is established. A schema is associated with the user account which stores data. A user account may belong to an individual person or can be used by a device, process, and job or connection pool. `ALTER USER` statement is used to change database users' passwords or to lock an account or expire passwords. In addition, this statement is used to change database properties of user accounts such as database profiles, default and temporary tablespaces or assign tablespace quotas. This unified audit action enables logging of all `ALTER USER` statements, whether successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to alter user, whether successful or unsuccessful, may provide clues and forensic evidences about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all activities involving `ALTER USER`.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'ALTER USER'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
ALTER USER;
```

If you do not have CIS_UNIFIED_AUDIT_POLICY, please create one using CREATE AUDIT POLICY statement.

5.2.3 Enable 'DROP USER' Audit Option (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

An Oracle database user is an account through which a connection to the database is established. A schema is associated with the user account which stores data. A user account may belong to an individual person or can be used by a device, process, and job or connection pool. `DROP USER` statement is used to drop Oracle database accounts and schemas associated with them. This unified audit action enables logging of all `DROP USER` statements, whether successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to drop user, whether successful or unsuccessful, may provide clues and forensic evidences about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all activities involving `DROP USER`.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'DROP USER'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
DROP USER;
```

If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.4 Enable 'CREATE ROLE' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

An Oracle database role is a collection or set of privileges that can be granted to users or other roles. Roles may include system privileges, object privileges or other roles. This unified audit action enables logging of all `CREATE ROLE` statements, whether successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to create roles, whether successful or unsuccessful, may provide clues and forensic evidences about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving `CREATE ROLE`.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'CREATE ROLE'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
CREATE ROLE;
```

If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.5 Enable 'ALTER ROLE' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

An Oracle database role is a collection or set of privileges that can be granted to users or other roles. Roles may include system privileges, object privileges or other roles. `ALTER ROLE` statement is used to change the authorization needed to enable a role. This unified audit action enables logging of all `ALTER ROLE` statements, whether successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to alter roles, whether successful or unsuccessful, may provide clues and forensic evidences about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving alteration of roles.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'ALTER ROLE'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
ALTER ROLE;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.6 Enable 'DROP ROLE' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

An Oracle database role is a collection or set of privileges that can be granted to users or other roles. Roles may include system privileges, object privileges or other roles. This unified audit action enables logging of all `DROP ROLE` statements, successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to drop roles, whether successful or unsuccessful, may provide clues and forensic evidences about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving `DROP ROLE`.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES  AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'DROP ROLE'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
DROP ROLE;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.7 Enable 'GRANT' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

`GRANT` SQL statements are used to grant privileges to Oracle database users and roles, including the most powerful privileges and roles typically available to the database administrators. This unified audit action enables logging of all `GRANT` statements, whether successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

With unauthorized grants and permissions, a malicious user may be able to change the security of the database, access/update confidential data or compromise integrity of the database. Logging and monitoring of all attempts to grant system privileges, object privileges or roles, whether successful or unsuccessful, may provide forensic evidence about potential suspicious/unauthorized activities as well as privilege escalation activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving `GRANT`.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'GRANT'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
GRANT;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.8 Enable 'REVOKE' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

`REVOKE` SQL statements are used to revoke privileges from Oracle database users and roles. This unified audit action enables logging of all `REVOKE` statements, successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to revoke system privileges, object privileges or roles, whether successful or unsuccessful, may provide clues and forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving `REVOKE`.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'REVOKE'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
REVOKE;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.9 Enable 'CREATE PROFILE' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

Oracle Database Profiles are used to enforce resource usage limits and implement password policies such as password complexity rules and reuse restrictions. This unified audit action enables logging of all `CREATE PROFILE` statements, whether successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to create profiles, whether successful or unsuccessful, may provide clues and forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving creation of database profiles.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'CREATE PROFILE'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
CREATE PROFILE;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.10 Enable 'ALTER PROFILE' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

Oracle Database Profiles are used to enforce resource usage limits and implement password policies such as password complexity rules and reuse restrictions. This unified audit action enables logging of all `ALTER PROFILE` statements, whether successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to alter profiles, whether successful or unsuccessful, may provide forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving alteration of database profiles.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'ALTER PROFILE'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
ALTER PROFILE;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.11 Enable 'DROP PROFILE' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

Oracle Database Profiles are used to enforce resource usage limits and implement password policies such as password complexity rules and reuse restrictions. This unified audit action enables logging of all `DROP PROFILE` statements, whether successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to drop profiles, whether successful or unsuccessful, may provide clues and forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving dropping of database profiles.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'DROP PROFILE'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
DROP PROFILE;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.12 Enable 'CREATE DATABASE LINK' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

Oracle Database Links are used to establish database-to-database connections to other databases. These connections are available without further authentication once the link is established. This unified audit action enables logging of all `CREATE DATABASE` or `CREATE PUBLIC DATABASE` statements, whether successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to create database links, whether successful or unsuccessful, may provide forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving creation of database links.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'CREATE DATABASE LINK'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
CREATE DATABASE LINK;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.13 Enable 'ALTER DATABASE LINK' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

Oracle Database Links are used to establish database-to-database connections to other databases. These connections are always available without further authentication once the link is established. This unified audit action enables logging of all `ALTER DATABASE` or `ALTER PUBLIC DATABASE` statements, whether successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to alter database links, whether successful or unsuccessful, may provide forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving alteration of database links.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES  AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'ALTER DATABASE LINK'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
ALTER DATABASE LINK;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.14 Enable 'DROP DATABASE LINK' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

Oracle Database Links are used to establish database-to-database connections to other databases. These connections are always available without further authentication once the link is established. This unified audit action enables logging of all `DROP DATABASE` or `DROP PUBLIC DATABASE`, whether successful or unsuccessful, statements issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to drop database links, whether successful or unsuccessful, may provide forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving dropping of database links.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'DROP DATABASE LINK'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
DROP DATABASE LINK;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.15 Enable 'CREATE SYNONYM' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

An Oracle database synonym is used to create an alternative name for a database object such as table, view, procedure, java object or even another synonym, etc. This unified audit action enables logging of all `CREATE SYNONYM` or `CREATE PUBLIC SYNONYM` statements, whether successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to create synonyms, whether successful or unsuccessful, may provide clues and forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving creation of synonyms or public synonyms.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'CREATE SYNONYM'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
CREATE SYNONYM;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.16 Enable 'ALTER SYNONYM' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

An Oracle database synonym is used to create an alternative name for a database object such as table, view, procedure, java object or even another synonym, etc. This unified audit action enables logging of all `ALTER SYNONYM` or `ALTER PUBLIC SYNONYM` statements, whether successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to alter synonyms, whether successful or unsuccessful, may provide clues and forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving alteration of synonyms or public synonyms.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'ALTER SYNONYM'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
ALTER SYNONYM;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.17 Enable 'DROP SYNONYM' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

An Oracle database synonym is used to create an alternative name for a database object such as table, view, procedure, java object or even another synonym, etc. This unified audit action enables logging of all `DROP SYNONYM` or `DROP PUBLIC SYNONYM` statements, whether successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to drop synonyms, whether successful or unsuccessful, may provide forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving dropping of synonyms or public synonyms.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'DROP SYNONYM'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
DROP SYNONYM;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.18 Enable 'SELECT ANY DICTIONARY' Privilege Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

`SELECT ANY DICTIONARY` system privilege allows the user to view the definition of all schema objects in the database. It grants `SELECT` privileges on the data dictionary objects to the grantees, including `SELECT` on `DBA_` views, `V$` views, `X$` views and underlying `SYS` tables such as `TAB$`, `OBJ$`, etc. This privilege also allows grantees to create stored objects such as procedures, packages or views on the underlying data dictionary objects. Please note that this privilege doesn't grant `SELECT` on tables with password hashes such as `USER$`, `DEFAULT_PWD$`, `LINK$`, and `USER_HISTORY$`. This audit enables logging of activities that exercise this privilege.

Rationale:

Logging and monitoring of all attempts to access data dictionary, whether successful or unsuccessful, may provide clues and forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving access to the database.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'SELECT ANY DICTIONARY'
AND AUD.AUDIT_OPTION_TYPE = 'SYSTEM PRIVILEGE'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
PRIVILEGES
SELECT ANY DICTIONARY;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.19 Enable 'UNIFIED_AUDIT_TRAIL' Access Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

`UNIFIED_AUDIT_TRAIL` view holds audit trail records generated by the database. This audit action enables logging of all access attempts to the `UNIFIED_AUDIT_TRAIL` view, whether successful or unsuccessful, regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to access the `UNIFIED_AUDIT_TRAIL` view, whether successful or unsuccessful, may provide clues and forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving access to this view.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'ALL'
AND AUD.AUDIT_OPTION_TYPE = 'OBJECT ACTION'
AND AUD.OBJECT_SCHEMA = 'SYS'
AND AUD.OBJECT_NAME = 'UNIFIED_AUDIT_TRAIL'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
ALL on SYS.UNIFIED_AUDIT_TRAIL;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.20 Enable 'CREATE PROCEDURE/FUNCTION/PACKAGE/PACKAGE BODY' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

Oracle database procedures, functions and packages, which are stored within the database, are created to perform business functions and access database as defined by PL/SQL code and SQL statements contained within these objects. This unified audit action enables logging of all `CREATE PROCEDURE`, `CREATE FUNCTION`, `CREATE PACKAGE` or `CREATE PACKAGE BODY`, successful or unsuccessful, statements issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to create procedures, functions, packages or package bodies, whether successful or unsuccessful, may provide clues and forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving creation of procedures, functions, packages or package bodies.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT *
FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS'
AND EXISTS ( SELECT 'x'
             FROM AUDIT_UNIFIED_POLICIES  AUD
             WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
             AND AUD.AUDIT_OPTION = 'CREATE PROCEDURE'
             AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION')
AND EXISTS ( SELECT 'x'
             FROM AUDIT_UNIFIED_POLICIES  AUD
             WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
             AND AUD.AUDIT_OPTION = 'CREATE FUNCTION'
             AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION')
AND EXISTS ( SELECT 'x'
             FROM AUDIT_UNIFIED_POLICIES  AUD
             WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
             AND AUD.AUDIT_OPTION = 'CREATE PACKAGE'
             AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION')
AND EXISTS ( SELECT 'x'
             FROM AUDIT_UNIFIED_POLICIES  AUD
             WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
             AND AUD.AUDIT_OPTION = 'CREATE PACKAGE BODY'
             AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION');
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
CREATE PROCEDURE,
CREATE FUNCTION,
CREATE PACKAGE,
CREATE PACKAGE BODY;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.21 Enable 'ALTER PROCEDURE/FUNCTION/PACKAGE/PACKAGE BODY' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

Oracle database procedures, functions and packages, which are stored within the database, are created to carry out business functions and access database as defined by PL/SQL code and SQL statements contained within these objects. This unified audit action enables logging of all, successful or unsuccessful, `ALTER PROCEDURE`, `ALTER FUNCTION`, `ALTER PACKAGE` or `ALTER PACKAGE BODY` statements issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Unauthorized alteration of procedures, functions, packages or package bodies may impact critical business functions or compromise integrity of the database. Logging and monitoring of all attempts, whether successful or unsuccessful, to alter procedures, functions, packages or package bodies may provide clues and forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving alteration of procedures, functions, packages or package bodies.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT *
FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS'
AND EXISTS ( SELECT 'x'
             FROM AUDIT_UNIFIED_POLICIES  AUD
             WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
             AND AUD.AUDIT_OPTION = 'ALTER PROCEDURE'
             AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION')
AND EXISTS ( SELECT 'x'
             FROM AUDIT_UNIFIED_POLICIES  AUD
             WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
             AND AUD.AUDIT_OPTION = 'ALTER FUNCTION'
             AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION')
AND EXISTS ( SELECT 'x'
             FROM AUDIT_UNIFIED_POLICIES  AUD
             WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
             AND AUD.AUDIT_OPTION = 'ALTER PACKAGE'
             AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION')
AND EXISTS ( SELECT 'x'
             FROM AUDIT_UNIFIED_POLICIES  AUD
             WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
             AND AUD.AUDIT_OPTION = 'ALTER PACKAGE BODY'
             AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION') ;
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
ALTER PROCEDURE,
ALTER FUNCTION,
ALTER PACKAGE,
ALTER PACKAGE BODY;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.22 Enable 'DROP PROCEDURE/FUNCTION/PACKAGE/PACKAGE BODY' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

Oracle database procedures, functions and packages, which are stored within the database, are created to carry out business functions and access database as defined by PL/SQL code and SQL statements contained within these objects. This unified audit action enables logging of all, successful or unsuccessful, `DROP PROCEDURE`, `DROP FUNCTION`, `DROP PACKAGE` or `DROP PACKAGE BODY` statements issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts, whether successful or unsuccessful, to drop procedures, functions, packages or package bodies may provide forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving dropping of procedures, functions, packages or package bodies.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT *
FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS'
AND EXISTS ( SELECT 'x'
             FROM AUDIT_UNIFIED_POLICIES  AUD
             WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
             AND AUD.AUDIT_OPTION = 'DROP PROCEDURE'
             AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION')
AND EXISTS ( SELECT 'x'
             FROM AUDIT_UNIFIED_POLICIES  AUD
             WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
             AND AUD.AUDIT_OPTION = 'DROP FUNCTION'
             AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION')
AND EXISTS ( SELECT 'x'
             FROM AUDIT_UNIFIED_POLICIES  AUD
             WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
             AND AUD.AUDIT_OPTION = 'DROP PACKAGE'
             AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION')
AND EXISTS ( SELECT 'x'
             FROM AUDIT_UNIFIED_POLICIES  AUD
             WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
             AND AUD.AUDIT_OPTION = 'DROP PACKAGE BODY'
             AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION');
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
DROP PROCEDURE,
DROP FUNCTION,
DROP PACKAGE,
DROP PACKAGE BODY;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.23 Enable 'ALTER SYSTEM' Privilege Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

`ALTER SYSTEM` privilege allows the user to change instance settings which could impact security posture, performance or normal operation of the database. Additionally, `ALTER SYSTEM` privilege may be used to run operating system commands using undocumented Oracle functionality. This unified audit enables logging of activities that involve exercise of this privilege, whether successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to execute `ALTER SYSTEM` statements, whether successful or unsuccessful, may provide forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities that involve `ALTER SYSTEM` statements.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'ALTER SYSTEM'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
ALTER SYSTEM;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.24 Enable 'CREATE TRIGGER' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

Oracle database triggers are executed automatically when specified conditions on the underlying objects occur. Trigger bodies contain the code, quite often to perform data validation, ensure data integrity/security or enforce critical constraints on allowable actions on data. This unified audit action enables logging of all `CREATE TRIGGER` statements, whether successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to create triggers, whether successful or unsuccessful, may provide clues and forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving creation of triggers.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'CREATE TRIGGER'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
CREATE TRIGGER;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.25 Enable 'ALTER TRIGGER' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

Oracle database triggers are executed automatically when specified conditions on the underlying objects occur. Trigger bodies contain the code, quite often to perform data validation, ensure data integrity/security or enforce critical constraints on allowable actions on data. This unified audit action enables logging of all `ALTER TRIGGER` statements, whether successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Unauthorized alteration of triggers may impact critical business functions or compromise integrity/security of the database. Logging and monitoring of all attempts to alter triggers, whether successful or unsuccessful, may provide clues and forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving alteration of triggers.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'ALTER TRIGGER'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
ALTER TRIGGER;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.26 Enable 'DROP TRIGGER' Action Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

Oracle database triggers are executed automatically when specified conditions on the underlying objects occur. Trigger bodies contain the code, quite often to perform data validation, ensure data integrity/security or enforce critical constraints on allowable actions on data. This unified audit action enables logging of all `DROP TRIGGER` statements, whether successful or unsuccessful, issued by the users regardless of the privileges held by the users to issue such statements.

Rationale:

Logging and monitoring of all attempts to drop triggers, whether successful or unsuccessful, may provide forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving dropping of triggers.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT AUD.POLICY_NAME, AUD.AUDIT_OPTION, AUD.AUDIT_OPTION_TYPE
FROM AUDIT_UNIFIED_POLICIES AUD, AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
AND AUD.AUDIT_OPTION = 'DROP TRIGGER'
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS';
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
DROP TRIGGER;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

5.2.27 Enable 'LOGON' AND 'LOGOFF' Actions Audit (Scored)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

Oracle database users log on to the database to perform their work. This unified audit action enables logging of all `LOGON` actions, whether successful or unsuccessful, issued by the users regardless of the privileges held by the users to log into the database. In addition, `LOGOFF` action audit captures logoff activities. This audit action also captures logon/logoff to the open database by `SYSDBA` and `SYSOPER`.

Rationale:

Logging and monitoring of all attempts to logon to the database, whether successful or unsuccessful, may provide forensic evidence about potential suspicious/unauthorized activities. Any such activities may be a cause for further investigation. In addition, organization security policies and industry/government regulations may require logging of all user activities involving `LOGON` and `LOGOFF`.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT *
FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
WHERE ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPT = 'BY'
AND ENABLED.USER_NAME = 'ALL USERS'
AND EXISTS ( SELECT 'x'
             FROM AUDIT_UNIFIED_POLICIES  AUD
             WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
             AND AUD.AUDIT_OPTION = 'LOGON'
             AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION')
AND EXISTS ( SELECT 'x'
             FROM AUDIT_UNIFIED_POLICIES  AUD
             WHERE AUD.POLICY_NAME = ENABLED.POLICY_NAME
             AND AUD.AUDIT_OPTION = 'LOGOFF'
             AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION');
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
LOGON,
LOGOFF;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using `CREATE AUDIT POLICY` statement.

6 Appendix: Establishing an Audit/Scan User

This document has been authored with the expectation that a user with appropriate permissions will be used to execute the queries and perform other assessment actions. While this could be accomplished by granting DBA privileges to a given user, the preferred approach is to create a dedicated user and granting only the specific permissions required to perform the assessments expressed herein. Doing this avoids the necessity for any user assessing the system needs to be granted DBA privileges.

The recommendations expressed in this document assume the presence of a role named CISSCANROLE and a user named CISSCAN. This role and user should be created by executing the following SQL statements, being careful to substitute an appropriate password for <password>.

```
-- Create the role
CREATE ROLE CISSCANROLE;
-- Grant necessary privileges to the role
GRANT CREATE SESSION TO CISSCANROLE;
GRANT SELECT ON V_$PARAMETER TO CISSCANROLE;
GRANT SELECT ON DBA_TAB_PRIVS TO CISSCANROLE;
GRANT SELECT ON DBA_PROFILES TO CISSCANROLE;
GRANT SELECT ON DBA_SYS_PRIVS TO CISSCANROLE;
GRANT SELECT ON DBA_STMT_AUDIT_OPTS TO CISSCANROLE;
GRANT SELECT ON DBA_ROLE_PRIVS TO CISSCANROLE;
GRANT SELECT ON DBA_OBJ_AUDIT_OPTS TO CISSCANROLE;
GRANT SELECT ON DBA_PRIV_AUDIT_OPTS TO CISSCANROLE;
GRANT SELECT ON DBA_PROXIES TO CISSCANROLE;
GRANT SELECT ON DBA_USERS TO CISSCANROLE;
GRANT SELECT ON DBA_USERS_WITH_DEFPWD TO CISSCANROLE;
GRANT AUDIT_VIEWER TO CISSCANROLE;
-- Create the user and assign the user to the role
CREATE USER CISSCAN IDENTIFIED BY <password>;
GRANT CISSCANROLE TO CISSCAN;
```

If you rely on similar roles and/or users, but which are not named as CISSCANROLE or CISSCAN, or if you have roles or users named CISSCANROLE or CISSCAN intended to be used for different purposes, be aware that some recommendations herein explicitly name CISSCANROLE and CISSCAN.

These are:

- 3.10 Ensure No Users Are Assigned the 'DEFAULT' Profile
- 4.5.5 Ensure 'ALL' Is Revoked from Unauthorized 'GRANTEE' on 'DBA_%'

Note that different organizations may wish to follow the instructions in this appendix in different ways. For more permanent or regular assessment scans, it may be acceptable to retain the `CISSCANROLE` and `CISSCAN` user indefinitely. However, in a consultative context where an assessment is perhaps run at the outset of the consulting engagement and again closer to the end, after any remediation has been performed, the `CISSCANROLE` role and `CISSCAN` user may be dropped. Such a decision is ultimately left up to the implementing organization.

Appendix: Summary Table

Control		Set Correctly	
		Yes	No
1	**Oracle Database Installation and Patching Requirements**		
1.1	Ensure the Appropriate Version/Patches for Oracle Software Is Installed (Not Scored)	☐	☐
1.2	Ensure All Default Passwords Are Changed (Scored)	☐	☐
1.3	Ensure All Sample Data And Users Have Been Removed (Scored)	☐	☐
2	**Oracle Parameter Settings**		
2.1	**Listener Settings**		
2.1.1	Ensure 'SECURE_CONTROL_<listener_name>' Is Set In 'listener.ora' (Scored)	☐	☐
2.1.2	Ensure 'extproc' Is Not Present in 'listener.ora' (Scored)	☐	☐
2.1.3	Ensure 'ADMIN_RESTRICTIONS_<listener_name>' Is Set to 'ON' (Scored)	☐	☐
2.1.4	Ensure 'SECURE_REGISTER_<listener_name>' Is Set to 'TCPS' or 'IPC' (Scored)	☐	☐
2.2	**Database settings**		
2.2.1	Ensure 'AUDIT_SYS_OPERATIONS' Is Set to 'TRUE' (Scored)	☐	☐
2.2.2	Ensure 'AUDIT_TRAIL' Is Set to 'OS', 'DB,EXTENDED', or 'XML,EXTENDED' (Scored)	☐	☐
2.2.3	Ensure 'GLOBAL_NAMES' Is Set to 'TRUE' (Scored)	☐	☐
2.2.4	Ensure 'LOCAL_LISTENER' Is Set Appropriately (Scored)	☐	☐
2.2.5	Ensure 'O7_DICTIONARY_ACCESSIBILITY' Is Set to 'FALSE' (Scored)	☐	☐
2.2.6	Ensure 'OS_ROLES' Is Set to 'FALSE' (Scored)	☐	☐
2.2.7	Ensure 'REMOTE_LISTENER' Is Empty (Scored)	☐	☐
2.2.8	Ensure 'REMOTE_LOGIN_PASSWORDFILE' Is Set to 'NONE' (Scored)	☐	☐
2.2.9	Ensure 'REMOTE_OS_AUTHENT' Is Set to 'FALSE' (Scored)	☐	☐
2.2.10	Ensure 'REMOTE_OS_ROLES' Is Set to 'FALSE' (Scored)	☐	☐
2.2.11	Ensure 'UTL_FILE_DIR' Is Empty (Scored)	☐	☐
2.2.12	Ensure 'SEC_CASE_SENSITIVE_LOGON' Is Set to 'TRUE' (Scored)	☐	☐
2.2.13	Ensure 'SEC_MAX_FAILED_LOGIN_ATTEMPTS' Is '10' (Scored)	☐	☐
2.2.14	Ensure 'SEC_PROTOCOL_ERROR_FURTHER_ACTION' Is Set to 'DELAY,3' or 'DROP,3' (Scored)	☐	☐
2.2.15	Ensure 'SEC_PROTOCOL_ERROR_TRACE_ACTION' Is Set to 'LOG' (Scored)	☐	☐
2.2.16	Ensure 'SEC_RETURN_SERVER_RELEASE_BANNER' Is Set to 'FALSE' (Scored)	☐	☐

2.2.17	Ensure 'SQL92_SECURITY' Is Set to 'TRUE' (Scored)	☐	☐
2.2.18	Ensure '_TRACE_FILES_PUBLIC' Is Set to 'FALSE' (Scored)	☐	☐
2.2.19	Ensure 'RESOURCE_LIMIT' Is Set to 'TRUE' (Scored)	☐	☐
3	**Oracle Connection and Login Restrictions**		
3.1	Ensure 'FAILED_LOGIN_ATTEMPTS' Is Less than or Equal to '5' (Scored)	☐	☐
3.2	Ensure 'PASSWORD_LOCK_TIME' Is Greater than or Equal to '1' (Scored)	☐	☐
3.3	Ensure 'PASSWORD_LIFE_TIME' Is Less than or Equal to '90' (Scored)	☐	☐
3.4	Ensure 'PASSWORD_REUSE_MAX' Is Greater than or Equal to '20' (Scored)	☐	☐
3.5	Ensure 'PASSWORD_REUSE_TIME' Is Greater than or Equal to '365' (Scored)	☐	☐
3.6	Ensure 'PASSWORD_GRACE_TIME' Is Less than or Equal to '5' (Scored)	☐	☐
3.7	Ensure 'DBA_USERS.PASSWORD' Is Not Set to 'EXTERNAL' for Any User (Scored)	☐	☐
3.8	Ensure 'PASSWORD_VERIFY_FUNCTION' Is Set for All Profiles (Scored)	☐	☐
3.9	Ensure 'SESSIONS_PER_USER' Is Less than or Equal to '10' (Scored)	☐	☐
3.10	Ensure No Users Are Assigned the 'DEFAULT' Profile (Scored)	☐	☐
4	**Oracle User Access and Authorization Restrictions**		
4.1	**Default Public Privileges for Packages and Object Types**		
4.1.1	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_ADVISOR' (Scored)	☐	☐
4.1.2	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_CRYPTO' (Scored)	☐	☐
4.1.3	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_JAVA' (Scored)	☐	☐
4.1.4	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_JAVA_TEST' (Scored)	☐	☐
4.1.5	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_JOB' (Scored)	☐	☐
4.1.6	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_LDAP' (Scored)	☐	☐
4.1.7	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_LOB' (Scored)	☐	☐
4.1.8	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_OBFUSCATION_TOOLKIT' (Scored)	☐	☐
4.1.9	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_RANDOM' (Scored)	☐	☐
4.1.10	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on	☐	☐

	'DBMS_SCHEDULER' (Scored)		
4.1.11	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_SQL' (Scored)	☐	☐
4.1.12	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_XMLGEN' (Scored)	☐	☐
4.1.13	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_XMLQUERY' (Scored)	☐	☐
4.1.14	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'UTL_FILE' (Scored)	☐	☐
4.1.15	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'UTL_INADDR' (Scored)	☐	☐
4.1.16	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'UTL_TCP' (Scored)	☐	☐
4.1.17	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'UTL_MAIL' (Scored)	☐	☐
4.1.18	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'UTL_SMTP' (Scored)	☐	☐
4.1.19	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'UTL_DBWS' (Scored)	☐	☐
4.1.20	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'UTL_ORAMTS' (Scored)	☐	☐
4.1.21	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'UTL_HTTP' (Scored)	☐	☐
4.1.22	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'HTTPURITYPE' (Scored)	☐	☐
4.2	**Revoke Non-Default Privileges for Packages and Object Types**		
4.2.1	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_SYS_SQL' (Scored)	☐	☐
4.2.2	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_BACKUP_RESTORE' (Scored)	☐	☐
4.2.3	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_AQADM_SYSCALLS' (Scored)	☐	☐
4.2.4	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_REPCAT_SQL_UTL' (Scored)	☐	☐
4.2.5	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'INITJVMAUX' (Scored)	☐	☐
4.2.6	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_STREAMS_ADM_UTL' (Scored)	☐	☐
4.2.7	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_AQADM_SYS' (Scored)	☐	☐
4.2.8	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_STREAMS_RPC' (Scored)	☐	☐
4.2.9	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_PRVTAQIM' (Scored)	☐	☐

4.2.10	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'LTADM' (Scored)	☐	☐
4.2.11	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'WWV_DBMS_SQL' (Scored)	☐	☐
4.2.12	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'WWV_EXECUTE_IMMEDIATE' (Scored)	☐	☐
4.2.13	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_IJOB' (Scored)	☐	☐
4.2.14	Ensure 'EXECUTE' Is Revoked from 'PUBLIC' on 'DBMS_FILE_TRANSFER' (Scored)	☐	☐
4.3	**Revoke Excessive System Privileges**		
4.3.1	Ensure 'SELECT_ANY_DICTIONARY' Is Revoked from Unauthorized 'GRANTEE' (Scored)	☐	☐
4.3.2	Ensure 'SELECT ANY TABLE' Is Revoked from Unauthorized 'GRANTEE' (Scored)	☐	☐
4.3.3	Ensure 'AUDIT SYSTEM' Is Revoked from Unauthorized 'GRANTEE' (Scored)	☐	☐
4.3.4	Ensure 'EXEMPT ACCESS POLICY' Is Revoked from Unauthorized 'GRANTEE' (Scored)	☐	☐
4.3.5	Ensure 'BECOME USER' Is Revoked from Unauthorized 'GRANTEE' (Scored)	☐	☐
4.3.6	Ensure 'CREATE_PROCEDURE' Is Revoked from Unauthorized 'GRANTEE' (Scored)	☐	☐
4.3.7	Ensure 'ALTER SYSTEM' Is Revoked from Unauthorized 'GRANTEE' (Scored)	☐	☐
4.3.8	Ensure 'CREATE ANY LIBRARY' Is Revoked from Unauthorized 'GRANTEE' (Scored)	☐	☐
4.3.9	Ensure 'CREATE LIBRARY' Is Revoked from Unauthorized 'GRANTEE' (Scored)	☐	☐
4.3.10	Ensure 'GRANT ANY OBJECT PRIVILEGE' Is Revoked from Unauthorized 'GRANTEE' (Scored)	☐	☐
4.3.11	Ensure 'GRANT ANY ROLE' Is Revoked from Unauthorized 'GRANTEE' (Scored)	☐	☐
4.3.12	Ensure 'GRANT ANY PRIVILEGE' Is Revoked from Unauthorized 'GRANTEE' (Scored)	☐	☐
4.4	**Revoke Role Privileges**		
4.4.1	Ensure 'DELETE_CATALOG_ROLE' Is Revoked from Unauthorized 'GRANTEE' (Scored)	☐	☐
4.4.2	Ensure 'SELECT_CATALOG_ROLE' Is Revoked from Unauthorized 'GRANTEE' (Scored)	☐	☐
4.4.3	Ensure 'EXECUTE_CATALOG_ROLE' Is Revoked from Unauthorized 'GRANTEE' (Scored)	☐	☐
4.4.4	Ensure 'DBA' Is Revoked from Unauthorized 'GRANTEE' (Scored)	☐	☐

4.5	**Revoke Excessive Table and View Privileges**		
4.5.1	Ensure 'ALL' Is Revoked from Unauthorized 'GRANTEE' on 'AUD$' (Scored)	☐	☐
4.5.2	Ensure 'ALL' Is Revoked from Unauthorized 'GRANTEE' on 'USER_HISTORY$' (Scored)	☐	☐
4.5.3	Ensure 'ALL' Is Revoked from Unauthorized 'GRANTEE' on 'LINK$' (Scored)	☐	☐
4.5.4	Ensure 'ALL' Is Revoked from Unauthorized 'GRANTEE' on 'SYS.USER$' (Scored)	☐	☐
4.5.5	Ensure 'ALL' Is Revoked from Unauthorized 'GRANTEE' on 'DBA_%' (Scored)	☐	☐
4.5.6	Ensure 'ALL' Is Revoked from Unauthorized 'GRANTEE' on 'SYS.SCHEDULER$_CREDENTIAL' (Scored)	☐	☐
4.5.7	Ensure 'SYS.USER$MIG' Has Been Dropped (Scored)	☐	☐
4.6	Ensure '%ANY%' Is Revoked from Unauthorized 'GRANTEE' (Scored)	☐	☐
4.7	Ensure 'DBA_SYS_PRIVS.%' Is Revoked from Unauthorized 'GRANTEE' with 'ADMIN_OPTION' Set to 'YES' (Scored)	☐	☐
4.8	Ensure Proxy Users Have Only 'CONNECT' Privilege (Scored)	☐	☐
4.9	Ensure 'EXECUTE ANY PROCEDURE' Is Revoked from 'OUTLN' (Scored)	☐	☐
4.10	Ensure 'EXECUTE ANY PROCEDURE' Is Revoked from 'DBSNMP' (Scored)	☐	☐
5	**Audit/Logging Policies and Procedures**		
5.1	**Traditional Auditing**		
5.1.1	Enable 'USER' Audit Option (Scored)	☐	☐
5.1.2	Enable 'ALTER USER' Audit Option (Scored)	☐	☐
5.1.3	Enable 'DROP USER' Audit Option (Scored)	☐	☐
5.1.4	Enable 'ROLE' Audit Option (Scored)	☐	☐
5.1.5	Enable 'SYSTEM GRANT' Audit Option (Scored)	☐	☐
5.1.6	Enable 'PROFILE' Audit Option (Scored)	☐	☐
5.1.7	Enable 'ALTER PROFILE' Audit Option (Scored)	☐	☐
5.1.8	Enable 'DROP PROFILE' Audit Option (Scored)	☐	☐
5.1.9	Enable 'DATABASE LINK' Audit Option (Scored)	☐	☐
5.1.10	Enable 'PUBLIC DATABASE LINK' Audit Option (Scored)	☐	☐
5.1.11	Enable 'PUBLIC SYNONYM' Audit Option (Scored)	☐	☐
5.1.12	Enable 'SYNONYM' Audit Option (Scored)	☐	☐
5.1.13	Enable 'GRANT DIRECTORY' Audit Option (Scored)	☐	☐
5.1.14	Enable 'SELECT ANY DICTIONARY' Audit Option (Scored)	☐	☐
5.1.15	Enable 'GRANT ANY OBJECT PRIVILEGE' Audit Option (Scored)	☐	☐
5.1.16	Enable 'GRANT ANY PRIVILEGE' Audit Option (Scored)	☐	☐
5.1.17	Enable 'DROP ANY PROCEDURE' Audit Option (Scored)	☐	☐

5.1.18	Enable 'ALL' Audit Option on 'SYS.AUD$' (Scored)	☐	☐
5.1.19	Enable 'PROCEDURE' Audit Option (Scored)	☐	☐
5.1.20	Enable 'ALTER SYSTEM' Audit Option (Scored)	☐	☐
5.1.21	Enable 'TRIGGER' Audit Option (Scored)	☐	☐
5.1.22	Enable 'CREATE SESSION' Audit Option (Scored)	☐	☐
5.2	**Unified Auditing**		
5.2.1	Enable 'CREATE USER' Action Audit (Scored)	☐	☐
5.2.2	Enable 'ALTER USER' Action Audit (Scored)	☐	☐
5.2.3	Enable 'DROP USER' Audit Option (Scored) (Scored)	☐	☐
5.2.4	Enable 'CREATE ROLE' Action Audit (Scored) (Scored)	☐	☐
5.2.5	Enable 'ALTER ROLE' Action Audit (Scored) (Scored)	☐	☐
5.2.6	Enable 'DROP ROLE' Action Audit (Scored) (Scored)	☐	☐
5.2.7	Enable 'GRANT' Action Audit (Scored) (Scored)	☐	☐
5.2.8	Enable 'REVOKE' Action Audit (Scored) (Scored)	☐	☐
5.2.9	Enable 'CREATE PROFILE' Action Audit (Scored) (Scored)	☐	☐
5.2.10	Enable 'ALTER PROFILE' Action Audit (Scored) (Scored)	☐	☐
5.2.11	Enable 'DROP PROFILE' Action Audit (Scored) (Scored)	☐	☐
5.2.12	Enable 'CREATE DATABASE LINK' Action Audit (Scored) (Scored)	☐	☐
5.2.13	Enable 'ALTER DATABASE LINK' Action Audit (Scored) (Scored)	☐	☐
5.2.14	Enable 'DROP DATABASE LINK' Action Audit (Scored) (Scored)	☐	☐
5.2.15	Enable 'CREATE SYNONYM' Action Audit (Scored) (Scored)	☐	☐
5.2.16	Enable 'ALTER SYNONYM' Action Audit (Scored) (Scored)	☐	☐
5.2.17	Enable 'DROP SYNONYM' Action Audit (Scored) (Scored)	☐	☐
5.2.18	Enable 'SELECT ANY DICTIONARY' Privilege Audit (Scored) (Scored)	☐	☐
5.2.19	Enable 'UNIFIED_AUDIT_TRAIL' Access Audit (Scored) (Scored)	☐	☐
5.2.20	Enable 'CREATE PROCEDURE/FUNCTION/PACKAGE/PACKAGE BODY' Action Audit (Scored) (Scored)	☐	☐
5.2.21	Enable 'ALTER PROCEDURE/FUNCTION/PACKAGE/PACKAGE BODY' Action Audit (Scored) (Scored)	☐	☐
5.2.22	Enable 'DROP PROCEDURE/FUNCTION/PACKAGE/PACKAGE BODY' Action Audit (Scored) (Scored)	☐	☐
5.2.23	Enable 'ALTER SYSTEM' Privilege Audit (Scored) (Scored)	☐	☐
5.2.24	Enable 'CREATE TRIGGER' Action Audit (Scored) (Not Scored)	☐	☐
5.2.25	Enable 'ALTER TRIGGER' Action Audit (Scored) (Scored)	☐	☐
5.2.26	Enable 'DROP TRIGGER' Action Audit (Scored) (Scored)	☐	☐
5.2.27	Enable 'LOGON' AND 'LOGOFF' Actions Audit (Scored)	☐	☐

	(Scored)		
6	**Appendix: Establishing an Audit/Scan User**		

Appendix: Change History

Date	Version	Changes for this version
04-29-2015	1.0.0	Initial Release
04-30-2015	1.1.0	Ticket #204: Clarification in overview for benchmark non-pluggable applicability
06-29-2015	1.1.0	Ticket #209: Add workflow advice to appendix about scan user
06-29-2015	1.1.0	Ticket #217: Corrected type of "repact" with "repcat"
06-29-2015	1.1.0	Ticket #216: Updated remediation to reference [PRIVILEGE] list
06-29-2015	1.1.0	Ticket #213: Updated audit query for regex on APEX users
06-29-2015	1.1.0	Ticket #212: Corrected confusion between DBMS_RANDOM and DBMS_BACKUP_RESTORE
06-29-2015	1.1.0	Ticket #211: Corrected incorrect recommendation from 'FALSE' to 'TRUE'
06-29-2015	1.1.0	Ticket #203: Updated references from 11g R2 to 12c where possible
03-31-2016	1.2.0	Ticket #259: Added SYSMAN to list of authorized grantees for 4.4.2
03-31-2016	1.2.0	Ticket #258: Added APEX_050000;MGMT_VIEW;SYSMAN_MDS;SYSMAN_OPSS;SYSMAN_RO;SYSMAN_STB to list of authorized grantees in 4.3.6
03-31-2016	1.2.0	Ticket #256: Added SYSBACKUP and SYSDG to grantee list for 4.3.1

03-31-2016	1.2.0	Ticket #254: Updated recommendation text to say 'Less than or Equal to 10' on 2.13
03-31-2016	1.2.0	Ticket #241: Added missing semicolon in audit query on 5.1
03-31-2016	1.2.0	Ticket #253: Removed quotes from remediation command on 2.2.2
03-31-2016	1.2.0	Ticket #261: Added SYS to table owners and SYSMAN to list of authorized grantees for 4.5.4
03-31-2016	1.2.0	Ticket #263: Added SYS to list of table owners
03-31-2016	1.2.0	Ticket #264: Added APEX_050000;SYSMAN_STB;SYSMAN_TYPES to list of authorized grantees
03-31-2016	1.2.0	Ticket #225: Updated description and rationale for 2.2.17
03-31-2016	1.2.0	Ticket #251: Added AUDIT_ADMIN, AUDIT_VIEWER, CAPTURE_ADMIN, DBA, GSMADMIN_INTERNAL, ORACLE_OCM, SYSDG, SYSKM, XDB to list of authorized grantees
03-31-2016	1.2.0	Ticket #215: Revised LISTENER sections and included LISTENER_HOME references
03-31-2016	1.2.0	Ticket #242: Added missing semicolon to 4.1.4
03-31-2016	1.2.0	Ticket #266: Updated audit query to check for all privileges, not only roles
03-31-2016	1.2.0	Ticket #265: Added APEX_050000 to list of authorized grantees on 4.7
03-31-2016	1.2.0	Ticket #252: Update profile text (minor)
02-29-2016	2.0.0	Ticket #267: Added a caution statement about revoking privileges from PUBLIC.

10-18-2016	2.0.0	Ticket #207: Moved existing auditing recommendations to a subsection named Traditional Auditing (5.1) and added unified auditing recommendations under a sibling subsection called Unified Auditing (5.2).
10-18-2016	2.0.0	Ticket #275: Corrected reference included for 2.2.2
10-18-2016	2.0.0	Ticket #276: Added 'DB' and 'XML' as valid parameter values for 2.2.2
12-01-2016	2.0.0	Ticket #262: Updated Grantee list and added a not regarding PUBLIC grants for 4.5.5
12-01-2016	2.0.0	Ticket #282: Corrected typo in 2.2.11 where it specified UTIL_FILE_DIR instead of UTL_FILE_DIR
12-01-2016	2.0.0	Ticket #283: Updated title to read "Ensure 'SEC_MAX_FAILED_LOGIN_ATTEMPTS' is '10'" for 2.2.13
12-01-2016	2.0.0	Ticket #284: Added "and OWNER='SYS'" to the query for 4.5.2
12-01-2016	2.0.0	Ticket #285: Added "and OWNER='SYS'" to the query for 4.5.3
12-01-2016	2.0.0	Ticket #286: Added "and OWNER='SYS'" to the query for 4.5.4
12-01-2016	2.0.0	Ticket #287: Added "and OWNER='SYS'" to the query for 4.5.6
12-28-2016	2.0.0	Planned Update