CIS Benchmarks

CIS Oracle Database 18c Benchmark

v1.0.0 - 07-16-2020

Terms of Use

Please see the below link for our current terms of use:

https://www.cisecurity.org/cis-securesuite/cis-securesuite-membership-terms-of-use/

Table of Contents

Overview

This document is intended to address the recommended security settings for Oracle Database 18c. This guide was tested against Oracle Database 18c installed with and without pluggable database support running on a Windows Server instance as a stand-alone system and running on an Oracle Linux instance also as a stand-alone system. Future Oracle Database 18c 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 18c 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://workbench.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

Assessment Status

An assessment status is included for every recommendation. The assessment status indicates whether the given recommendation can be automated or requires manual steps to implement. Both statuses are equally important and are determined and supported as defined below:

Automated

Represents recommendations for which assessment of a technical control can be fully automated and validated to a pass/fail state. Recommendations will include the necessary information to implement automation.

Manual

Represents recommendations for which assessment of a technical control cannot be fully automated and requires all or some manual steps to validate that the configured state is set as expected. The expected state can vary depending on the environment.

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 18c 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**

  This profile extends the "RDBMS using Traditional Auditing" profile. Items in this profile apply to RDBMS running on a Linux Host operating system with Oracle Database 18c 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**

  This profile extends the "RDBMS using Traditional Auditing" profile. Items in this profile apply to RDBMS running on a Windows Server operating system with Oracle Database 18c 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 18c 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**

  This profile extends the "RDBMS using Unified Auditing" profile. Items in this profile apply to RDBMS running on a Linux Host operating system with Oracle Database 18c configured to use Unified and intend to:

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

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

  This profile extends the "RDBMS using Unified Auditing" profile. Items in this profile apply to RDBMS running on a Windows Server operating system with Oracle Database 18c configured to use Unified and intend to:

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

Acknowledgements

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 (Unscored)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle installation version and patches should be the most recent that are compatible with the organization's operational needs.

Rationale:

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 Linux systems:

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

For example, on Windows systems:

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

Remediation:

Perform the following step for 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

CIS Controls:

Version 6

2 Inventory of Authorized and Unauthorized Software
Inventory of Authorized and Unauthorized Software

Version 7

2.2 Ensure Software is Supported by Vendor
Ensure that only software applications or operating systems currently supported by the software's vendor are added to the organization's authorized software inventory. Unsupported software should be tagged as unsupported in the inventory system.

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

This section defines recommendations for the settings for the TNS Listener `listener.ora` file.

2.1.1 Ensure 'SECURE_CONTROL_' Is Set In 'listener.ora' (Automated)

Profile Applicability:

- Level 1 - Linux Host OS using Traditional Auditing

- Level 1 - Windows Server Host OS using Traditional Auditing

- Level 1 - Linux Host OS using Unified Auditing

- Level 1 - Windows Server Host OS using Unified 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:

Listener configuration changes via unencrypted remote connections can result in unauthorized users sniffing control configuration information from the network.

Audit:

To audit this recommendation, follow these steps:

1. Open the `$ORACLE_HOME/network/admin/listener.ora` file (or `%ORACLE_HOME%\network\admin\listener.ora` on Windows)

2. 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:

To remediate this recommendation:
Set the `SECURE_CONTROL_<listener_name>` for each defined listener in the `listener.ora` file.

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/netrf/Oracle-Net-Listener-parameters-in-listener-ora-file.html

CIS Controls:

Version 6

3.4 Use Only Secure Channels For Remote System Administration
   Perform all remote administration of servers, workstation, network devices, and similar equipment over secure channels. Protocols such as telnet, VNC, RDP, or others that do not actively support strong encryption should only be used if they are performed over a secondary encryption channel, such as SSL, TLS or IPSEC.

Version 7

4.5 Use Multifactor Authentication For All Administrative Access
   Use multi-factor authentication and encrypted channels for all administrative account access.

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

Profile Applicability:

- Level 1 - Linux Host OS using Traditional Auditing

- Level 1 - Windows Server Host OS using Traditional Auditing

- Level 1 - Linux Host OS using Unified Auditing

- Level 1 - Windows Server Host OS using Unified Auditing

Description:

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

Rationale:

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

Audit:

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

Linux 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:

To remediate this recommendation:
Remove `extproc` from the `listener.ora` file.

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/netrf/Oracle-Net-Listener-parameters-in-listener-ora-file.html

CIS Controls:

Version 6

18.9 <u>Sanitize Deployed Software Of Development Artifacts</u>
For in-house developed applications, ensure that development artifacts (sample data and scripts; unused libraries, components, debug code; or tools) are not included in the deployed software, or accessible in the production environment.

Version 7

9.2 <u>Ensure Only Approved Ports, Protocols and Services Are Running</u>
Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.

2.1.3 Ensure 'ADMIN_RESTRICTIONS_' Is Set to 'ON' (Automated)

Profile Applicability:

- Level 1 - Linux Host OS using Traditional Auditing

- Level 1 - Windows Server Host OS using Traditional Auditing

- Level 1 - Linux Host OS using Unified Auditing

- Level 1 - Windows Server Host OS using Unified 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:

Blocking unprivileged users from making alterations of the `listener.ora` file, where remote data/service settings are specified, will help protect data confidentiality.

Audit:

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

Linux 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:

To remediate this recommendation:
Use a text editor such as `vi` to set the `admin_restrictions_<listener_name>` to the value `ON`.

Default Value:

Not set.

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/netrf/Oracle-Net-Listener-parameters-in-listener-ora-file.html

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

4.3 Ensure the Use of Dedicated Administrative Accounts
Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities.

2.1.4 Ensure 'SECURE_REGISTER_' Is Set to 'TCPS' or 'IPC' (Automated)

Profile Applicability:

- Level 1 - Linux Host OS using Traditional Auditing

- Level 1 - Windows Server Host OS using Traditional Auditing

- Level 1 - Linux Host OS using Unified Auditing

- Level 1 - Windows Server Host OS using Unified Auditing

Description:

The `SECURE_REGISTER_<listener_name>` setting specifies the protocols used to connect to the TNS listener. Each setting should have a value of either TCPS or IPC based on the needs for its protocol.

Rationale:

Listener configuration changes via unencrypted remote connections can result in unauthorized users sniffing control configuration information from the network.

Audit:

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

Linux 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:

To remediate this recommendation:
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. https://docs.oracle.com/en/database/oracle/oracle-database/18/netrf/Oracle-Net-Listener-parameters-in-listener-ora-file.html
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.

CIS Controls:

Version 6

14.2 Encrypt All Sensitive Information Over Less-trusted Networks
All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.

Version 7

14.4 Encrypt All Sensitive Information in Transit
Encrypt all sensitive information in transit.

2.2 Database Settings

This section defines recommendations covering the general security configuration of the database instance. The 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 it with SCOPE = BOTH might be preferred depending on the recommendation.

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

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The AUDIT_SYS_OPERATIONS setting provides for the auditing of all user activities conducted under the SYSOPER and SYSDBA accounts. The setting should be set to TRUE to enable this auditing.

Rationale:

If the parameter AUDIT_SYS_OPERATIONS is FALSE, all statements except for 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$SYSTEM_PARAMETER
WHERE UPPER(NAME) = 'AUDIT_SYS_OPERATIONS';
```

Ensure VALUE is set to TRUE.

Remediation:

To remediate this setting, execute the following SQL statement and restart the instance.

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

Default Value:

```
TRUE
```

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/AUDIT_SYS_OPERATIONS.html#GUID-58176267-238C-40B5-B1F2-BB8BB9518950

CIS Controls:

Version 6

5.4 Log Administrative User Addition And Removal
Configure systems to issue a log entry and alert when an account is added to or removed from a domain administrators' group, or when a new local administrator account is added on a system.

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

4.8 Log and Alert on Changes to Administrative Group Membership
Configure systems to issue a log entry and alert when an account is added to or removed from any group assigned administrative privileges.

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

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

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The `audit_trail` setting determines whether or not Oracle's basic audit features are enabled. It can be set to "Operating System"(`OS`); `DB`; `DB,EXTENDED`; `XML`; or `XML,EXTENDED`. The value should be set according to the needs of the organization.

Rationale:

Enabling the basic auditing features for the Oracle instance permits the collection of data to troubleshoot problems, as well as provides valuable 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$SYSTEM_PARAMETER
WHERE UPPER(NAME)='AUDIT_TRAIL';
```

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

Remediation:

To remediate this setting, execute one of the following SQL statements and restart the instance.

```
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. https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/AUDIT_TRAIL.html#GUID-BD86F593-B606-4367-9FB6-8DAB2E47E7FA

2. http://www.oracle.com/technetwork/products/audit-vault/learnmore/twp-security-auditperformance-166655.pdf

CIS Controls:

Version 6

6 Maintenance, Monitoring, and Analysis of Audit Logs
Maintenance, Monitoring, and Analysis of Audit Logs

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

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

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. This setting should have a value of `TRUE`.

Rationale:

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.

Audit:

Non multi-tenant or pluggable database only:

To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:

This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT DISTINCT UPPER(V.VALUE),
DECODE (V.CON_ID,0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B
                      WHERE V.CON_ID = B.CON_ID))
FROM V$SYSTEM_PARAMETER V
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. https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/GLOBAL_NAMES.html#GUID-221D0483-D814-4963-84E1-7D39A25048ED

CIS Controls:

Version 6

9 Limitation and Control of Network Ports, Protocols, and Services
Limitation and Control of Network Ports, Protocols, and Services

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

2.2.4 Ensure 'O7_DICTIONARY_ACCESSIBILITY' Is Set to 'FALSE' (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

- Level 1 - RDBMS using Unified Auditing

Description:

The `O7_dictionary_accessibility` setting is a database initialization parameter that allows/disallows access to objects with the **ANY** privileges (`SELECT ANY TABLE`, `DELETE ANY TABLE`, `EXECUTE ANY PROCEDURE`, etc.). This functionality was created for the ease of migration from Oracle 7 databases to later versions. The setting should have a value of `FALSE`.

Note: The `O7_dictionary_accessibility` parameter has been deprecated in 12.2 and higher versions.

Rationale:

Leaving the `SYS` schema so open to connection could permit unauthorized access to critical data structures.

Audit:

Non multi-tenant or pluggable database only:

To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:

This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT DISTINCT UPPER(V.VALUE),
DECODE (V.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B
                      WHERE V.CON_ID = B.CON_ID))
```

```
FROM V$SYSTEM_PARAMETER V
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. https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/O7_DICTIONARY_ACCESSIBILITY.html#GUID-1D1A88F1-B603-48FF-BD30-E6099DB1A1ED

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.

CIS Controls:

Version 6

9.1 Limit Open Ports, Protocols, and Services
Ensure that only ports, protocols, and services with validated business needs are running on each system.

Version 7

9.2 Ensure Only Approved Ports, Protocols and Services Are Running
Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.

2.2.5 Ensure 'OS_ROLES' Is Set to 'FALSE' (Automated)

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:

Allowing the OS to use external groups for database management could cause privilege overlaps and generally weaken security.

Audit:

Non multi-tenant or pluggable database only:

To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:

This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT DISTINCT UPPER(V.VALUE),
DECODE (V.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B
                        WHERE V.CON_ID = B.CON_ID))
FROM V$SYSTEM_PARAMETER V
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. https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/OS_ROLES.html#GUID-51CCE2D6-F841-4E02-A89D-EA08FC110CF3

CIS Controls:

Version 6

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

2.2.6 Ensure 'REMOTE_LISTENER' Is Empty (Automated)

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. This setting should be empty unless the organization specifically needs a valid listener on a separate system or on nodes running Oracle RAC instances.

Rationale:

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.

Audit:

Non multi-tenant or pluggable database only:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$SYSTEM_PARAMETER
WHERE UPPER(NAME)='REMOTE_LISTENER' AND VALUE IS NOT NULL;
```

Multi-tenant in the container database:

This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT DISTINCT UPPER(V.VALUE),
DECODE (V.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B
                     WHERE V.CON_ID = B.CON_ID))
FROM V$SYSTEM_PARAMETER V
WHERE UPPER(NAME) = 'REMOTE_LISTENER' AND VALUE IS NOT NULL;
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/REMOTE_LISTENER.html#GUID-FEE2E8B5-CE02-4158-A6B4-030E59316756

Notes:

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

CIS Controls:

Version 6

9 Limitation and Control of Network Ports, Protocols, and Services
Limitation and Control of Network Ports, Protocols, and Services

Version 7

9.2 Ensure Only Approved Ports, Protocols and Services Are Running
Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.

2.2.7 Ensure 'REMOTE_LOGIN_PASSWORDFILE' Is Set to 'NONE' (Automated)

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. The setting should have a value of `NONE` or in the event you are running DR/Data Guard, `EXCLUSIVE` is an allowable value.

Rationale:

The use of this sort of password login file could permit unsecured, privileged connections to the database.

Audit:

To assess this recommendation, execute the following SQL statement.

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

Ensure `VALUE` is set to `NONE` or in the event you are running DR/Data Guard, `EXCLUSIVE` is an allowable VALUE.

Remediation:

To remediate this setting, execute the following SQL statement.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/REMOTE_LOGIN_PASSWORDFILE.html#GUID-6619299E-95E8-4821-B123-3B5899F046C7

CIS Controls:

Version 6

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

16.2 Configure Centralized Point of Authentication
Configure access for all accounts through as few centralized points of authentication as possible, including network, security, and cloud systems.

2.2.8 Ensure 'REMOTE_OS_AUTHENT' Is Set to 'FALSE' (Automated)

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. This setting should have a value of `FALSE`.

Note: This parameter has been deprecated in 12.1 and higher versions.

Rationale:

Permitting OS roles for database connections 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$SYSTEM_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. https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/REMOTE_OS_AUTHENT.html#GUID-AB66C849-FE5A-4E06-A6E1-AEE775D55703

CIS Controls:

Version 6

   16 <u>Account Monitoring and Control</u>
   Account Monitoring and Control

Version 7

   16.2 <u>Configure Centralized Point of Authentication</u>
   Configure access for all accounts through as few centralized points of authentication as possible, including network, security, and cloud systems.

2.2.9 Ensure 'REMOTE_OS_ROLES' Is Set to 'FALSE' (Automated)

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. This setting should have a value of `FALSE`.

Rationale:

Allowing remote clients OS roles to have permissions for database management could cause privilege overlaps and generally weaken security.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$SYSTEM_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. https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/REMOTE_OS_ROLES.html#GUID-BAA83447-14C1-4BE7-BB5D-806ED3E00AED

CIS Controls:

Version 6

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

16.2 Configure Centralized Point of Authentication

Configure access for all accounts through as few centralized points of authentication as possible, including network, security, and cloud systems.

2.2.10 Ensure 'SEC_CASE_SENSITIVE_LOGON' Is Set to 'TRUE' (Automated)

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.

Note: This parameter has been deprecated in 12.1 and higher versions.

Rationale:

Oracle database password case-sensitivity increases the pool of characters that can be chosen for the passwords, making brute-force password attacks quite difficult.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$SYSTEM_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;
```

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/SEC_CASE_SENSITIVE_LOGON.html#GUID-F464653A-0D43-4A70-8F05-0274A12C8578

CIS Controls:

Version 6

16 <u>Account Monitoring and Control</u>
Account Monitoring and Control

Version 7

4.4 <u>Use Unique Passwords</u>
Where multi-factor authentication is not supported (such as local administrator, root, or service accounts), accounts will use passwords that are unique to that system.

2.2.11 Ensure 'SEC_MAX_FAILED_LOGIN_ATTEMPTS' Is '3' or Less (Automated)

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:

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.

Audit:

To assess this recommendation, execute the following SQL statement.

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

Ensure `VALUE` is set to `3`.

Remediation:

To remediate this setting, execute the following SQL statement.

```
ALTER SYSTEM SET SEC_MAX_FAILED_LOGIN_ATTEMPTS = 3 SCOPE = SPFILE;
```

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/SEC_MAX_FAILED_LOGIN_ATTEMPTS.html#GUID-DEC2A3B2-F49B-499E-A3CF-D097F3A5BA83

CIS Controls:

Version 6

16.7 <u>Configure Account Lockouts</u>

Use and configure account lockouts such that after a set number of failed login attempts the account is locked for a standard period of time.

Version 7

16.7 <u>Establish Process for Revoking Access</u>

Establish and follow an automated process for revoking system access by disabling accounts immediately upon termination or change of responsibilities of an employee or contractor . Disabling these accounts, instead of deleting accounts, allows preservation of audit trails.

2.2.12 Ensure 'SEC_PROTOCOL_ERROR_FURTHER_ACTION' Is Set to '(DROP,3)' (Automated)

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 server's response to bad/malformed packets received from the client. This setting should have a value of `(DROP,3)`, which will cause a connection to be dropped after three bad/malformed packets.

Rationale:

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$SYSTEM_PARAMETER
WHERE UPPER(NAME)='SEC_PROTOCOL_ERROR_FURTHER_ACTION';
```

Ensure `VALUE` is set to `(DROP,3)`.

Remediation:

To remediate this setting, execute the following SQL statement.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/SEC_PROTOCOL_ERROR_FURTHER_ACTION.html#GUID-1E8D3C6E-C919-4218-8117-760D31BD0F95

CIS Controls:

Version 6

18 Application Software Security
Application Software Security

2.2.13 Ensure 'SEC_PROTOCOL_ERROR_TRACE_ACTION' Is Set to 'LOG' (Automated)

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. This setting should have a value of `LOG` unless the organization has a compelling reason to use a different value because `LOG` should cause the necessary information to be logged. Setting the value as `TRACE` can generate an enormous amount of log output and should be reserved for debugging only.

Rationale:

Bad packets received from the client can potentially indicate packet-based attacks on the system, which could result in a denial-of-service condition.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$SYSTEM_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. https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/SEC_PROTOCOL_ERROR_TRACE_ACTION.html#GUID-AE811BC1-8CED-4B21-B16C-4B712B127535

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting

Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging

Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging

Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

2.2.14 Ensure 'SEC_RETURN_SERVER_RELEASE_BANNER' Is Set to 'FALSE' (Automated)

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. This is sensitive information that should not be revealed to anyone who requests it.

Rationale:

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.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT UPPER(VALUE)
FROM V$SYSTEM_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. https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/SEC_RETURN_SERVER_RELEASE_BANNER.html#GUID-688102A0-11F5-4F06-8868-934D65C4E878

CIS Controls:

Version 6

   9 <u>Limitation and Control of Network Ports, Protocols, and Services</u>
   Limitation and Control of Network Ports, Protocols, and Services

Version 7

   14.6 <u>Protect Information through Access Control Lists</u>
   Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

2.2.15 Ensure 'SQL92_SECURITY' Is Set to 'TRUE' (Automated)

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. The setting should have a value of TRUE.

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:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT DISTINCT UPPER(V.VALUE),
DECODE (V.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B
                      WHERE V.CON_ID = B.CON_ID))
FROM V$SYSTEM_PARAMETER V
WHERE UPPER(NAME) = 'SQL92_SECURITY';
```

Remediation:

To remediate this setting, execute the following SQL statement.

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

Default Value:

```
FALSE
```

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/SQL92_SECURITY.html#GUID-E41087C2-250E-4201-908B-79E659B22A4B

CIS Controls:

Version 6

18 Application Software Security
Application Software Security

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

2.2.16 Ensure '_trace_files_public' Is Set to 'FALSE' (Automated)

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. This setting should have a value of FALSE to restrict trace file access.

Rationale:

Making the file world readable means anyone can read the instance's trace file, which could contain sensitive information about instance operations.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT A.KSPPINM, B.KSPPSTVL
FROM SYS.X_$KSPPI a, SYS.X_$KSPPCV b
WHERE A.INDX=B.INDX
AND A.KSPPINM LIKE '\_%trace_files_public' escape '\';
```

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

Please note that the assessment SQL relies on `X_$` views which should be created per Appendix 7.

BELOW SQL NO LONGER WORKS FOR Oracle12c FOR UNDOCUMENTED PARAMETERS.

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

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:4295521
746131

CIS Controls:

Version 6

14.4 Protect Information With Access Control Lists
All information stored on systems shall be protected with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

2.2.17 Ensure 'RESOURCE_LIMIT' Is Set to 'TRUE' (Automated)

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. This setting should have a value of TRUE.

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, the limits set in database profiles are enforced.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT DISTINCT UPPER(V.VALUE),
DECODE (V.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B
                        WHERE V.CON_ID = B.CON_ID))
FROM V$SYSTEM_PARAMETER V
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. https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/RESOURCE_LIMIT.html#GUID-BB0AB177-3867-4D0D-8700-A1AC8BDFEFC3

CIS Controls:

Version 6

14.4 Protect Information With Access Control Lists
All information stored on systems shall be protected with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

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' (Automated)

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:

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 the **Notes** for a warning on a known bug that can make this security measure backfire.)

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT P.PROFILE, P.RESOURCE_NAME, P.LIMIT
FROM DBA_PROFILES P
WHERE TO_NUMBER(DECODE(P.LIMIT,
                    'DEFAULT',(SELECT DISTINCT
DECODE(LIMIT,'UNLIMITED',9999,LIMIT)
                                FROM DBA_PROFILES
```

```
                                WHERE PROFILE='DEFAULT'
                                AND RESOURCE_NAME='FAILED_LOGIN_ATTEMPTS'),
                  'UNLIMITED','9999',
                   P.LIMIT)) > 5
AND P.RESOURCE_NAME = 'FAILED_LOGIN_ATTEMPTS'
AND EXISTS ( SELECT 'X' FROM DBA_USERS U WHERE U.PROFILE = P.PROFILE ) ;
```

Multi-tenant in the container database:

This query will also give you the name of the CDB/PDB that has the issue. To assess this
recommendation, execute the following SQL statement.

```
SELECT P.PROFILE, P.RESOURCE_NAME, P.LIMIT,
DECODE (P.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                 1,(SELECT NAME FROM V$DATABASE),
                   (SELECT NAME FROM V$PDBS B
                        WHERE P.CON_ID = B.CON_ID)) DATABASE
FROM CDB_PROFILES P
WHERE TO_NUMBER(DECODE(P.LIMIT,
                       'DEFAULT',(SELECT DECODE(LIMIT,'UNLIMITED',9999,LIMIT)
                                  FROM CDB_PROFILES
                                  WHERE PROFILE='DEFAULT'
                                  AND RESOURCE_NAME='FAILED_LOGIN_ATTEMPTS'
                                AND CON_ID = P.CON_ID),
                       'UNLIMITED','9999',P.LIMIT)) > 5
AND P.RESOURCE_NAME = 'FAILED_LOGIN_ATTEMPTS'
AND EXISTS ( SELECT 'X' FROM CDB_USERS U WHERE U.PROFILE = P.PROFILE )
ORDER BY CON_ID, PROFILE, RESOURCE_NAME;
```

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;
```

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/CREATE-PROFILE.html#GUID-ABC7AE4D-64A8-4EA9-857D-BEF7300B64C3

Notes:

Warning: One 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.

CIS Controls:

Version 6

16.7 Configure Account Lockouts
Use and configure account lockouts such that after a set number of failed login attempts the account is locked for a standard period of time.

Version 7

16.7 Establish Process for Revoking Access
Establish and follow an automated process for revoking system access by disabling accounts immediately upon termination or change of responsibilities of an employee or contractor . Disabling these accounts, instead of deleting accounts, allows preservation of audit trails.

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

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. The suggested value for this is one day or greater.

Rationale:

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.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT P.PROFILE, P.RESOURCE_NAME, P.LIMIT
FROM DBA_PROFILES P
WHERE TO_NUMBER(DECODE(P.LIMIT,
                       'DEFAULT',(SELECT DISTINCT
DECODE(LIMIT,'UNLIMITED',9999,LIMIT)
                                   FROM DBA_PROFILES
                                   WHERE PROFILE='DEFAULT'
                                   AND RESOURCE_NAME='PASSWORD_LOCK_TIME'),
                       'UNLIMITED','9999',
                        P.LIMIT)) < 1
AND P.RESOURCE_NAME = 'PASSWORD_LOCK_TIME'
AND EXISTS ( SELECT 'X' FROM DBA_USERS U WHERE U.PROFILE = P.PROFILE );
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT P.PROFILE, P.RESOURCE_NAME, P.LIMIT,
DECODE (P.CON_ID,0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B
```

```
                     WHERE P.CON_ID = B.CON_ID)) DATABASE
FROM CDB_PROFILES P
WHERE TO_NUMBER(DECODE(P.LIMIT,
                      'DEFAULT',(SELECT DECODE(LIMIT,'UNLIMITED',9999,LIMIT)
                                 FROM CDB_PROFILES
                                 WHERE PROFILE='DEFAULT'
                                 AND RESOURCE_NAME='PASSWORD_LOCK_TIME'
                                 AND CON_ID = P.CON_ID),
                      'UNLIMITED','9999',P.LIMIT)) < 1
AND P.RESOURCE_NAME = 'PASSWORD_LOCK_TIME'
AND EXISTS ( SELECT 'X' FROM CDB_USERS U WHERE U.PROFILE = P.PROFILE )
ORDER BY CON_ID, PROFILE, RESOURCE_NAME;
```

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;
```

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/CREATE-PROFILE.html#GUID-ABC7AE4D-64A8-4EA9-857D-BEF7300B64C3

CIS Controls:

Version 6

16.7 Configure Account Lockouts
  Use and configure account lockouts such that after a set number of failed login attempts the account is locked for a standard period of time.

Version 7

16.7 Establish Process for Revoking Access
  Establish and follow an automated process for revoking system access by disabling accounts immediately upon termination or change of responsibilities of an employee or contractor . Disabling these accounts, instead of deleting accounts, allows preservation of audit trails.

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

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. The suggested value for this is 90 days or less.

Rationale:

Allowing passwords to remain unchanged for long periods makes the success of brute-force login attacks more likely.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT P.PROFILE, P.RESOURCE_NAME, P.LIMIT
FROM DBA_PROFILES P
WHERE TO_NUMBER(DECODE(P.LIMIT,
                    'DEFAULT',(SELECT DISTINCT
DECODE(LIMIT,'UNLIMITED',9999,LIMIT)
                                FROM DBA_PROFILES
                                WHERE PROFILE='DEFAULT'
                                AND RESOURCE_NAME='PASSWORD_LIFE_TIME'),
                    'UNLIMITED','9999',P.LIMIT)) > 90 AND
     P.RESOURCE_NAME = 'PASSWORD_LIFE_TIME' AND
 EXISTS ( SELECT 'X' FROM DBA_USERS U WHERE U.PROFILE = P.PROFILE );
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT P.PROFILE, P.RESOURCE_NAME, P.LIMIT,
DECODE (P.CON_ID,0,(SELECT NAME FROM V$DATABASE),
            1,(SELECT NAME FROM V$DATABASE),
                (SELECT NAME FROM V$PDBS B
                    WHERE P.CON_ID = B.CON_ID)) DATABASE
FROM CDB_PROFILES P
WHERE TO_NUMBER(DECODE(P.LIMIT,
                    'DEFAULT',(SELECT DECODE(LIMIT,'UNLIMITED',9999,LIMIT)
```

```
                              FROM CDB_PROFILES
                              WHERE PROFILE='DEFAULT'
                              AND RESOURCE_NAME='PASSWORD_LIFE_TIME'
                            AND CON_ID = P.CON_ID),
                    'UNLIMITED','9999',
                    P.LIMIT)) > 90
AND P.RESOURCE_NAME = 'PASSWORD_LIFE_TIME'
AND EXISTS ( SELECT 'X' FROM CDB_USERS U WHERE U.PROFILE = P.PROFILE )
ORDER BY CON_ID, PROFILE, RESOURCE_NAME;
```

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;
```

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/CREATE-PROFILE.html#GUID-ABC7AE4D-64A8-4EA9-857D-BEF7300B64C3

CIS Controls:

Version 6

16 Account Monitoring and Control
Account Monitoring and Control

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

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. The suggested value for this is 20 passwords or greater.

Rationale:

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.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT P.PROFILE, P.RESOURCE_NAME, P.LIMIT
FROM DBA_PROFILES P
WHERE TO_NUMBER(DECODE(P.LIMIT,
                       'DEFAULT',(SELECT DISTINCT
DECODE(LIMIT,'UNLIMITED',9999,LIMIT)
                                 FROM DBA_PROFILES
                                 WHERE PROFILE='DEFAULT'
                                 AND RESOURCE_NAME='PASSWORD_REUSE_MAX'),
                       'UNLIMITED','9999',P.LIMIT)) < 20 AND
      P.RESOURCE_NAME = 'PASSWORD_REUSE_MAX' AND
 EXISTS ( SELECT 'X' FROM DBA_USERS U WHERE U.PROFILE = P.PROFILE );
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT P.PROFILE, P.RESOURCE_NAME, P.LIMIT,
DECODE (P.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B
                    WHERE P.CON_ID = B.CON_ID)) DATABASE
```

```
FROM CDB_PROFILES P
WHERE TO_NUMBER(DECODE(P.LIMIT,
                       'DEFAULT',(SELECT DECODE(LIMIT,'UNLIMITED',9999,LIMIT)
                                   FROM CDB_PROFILES
                                   WHERE PROFILE='DEFAULT'
                                   AND RESOURCE_NAME='PASSWORD_REUSE_MAX'
                                 AND CON_ID = P.CON_ID),
                       'UNLIMITED','9999',P.LIMIT)) < 20
AND P.RESOURCE_NAME = 'PASSWORD_REUSE_MAX'
AND EXISTS ( SELECT 'X' FROM CDB_USERS U WHERE U.PROFILE = P.PROFILE )
ORDER BY CON_ID, PROFILE, RESOURCE_NAME;
```

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;
```

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/CREATE-PROFILE.html#GUID-ABC7AE4D-64A8-4EA9-857D-BEF7300B64C3

Notes:

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

CIS Controls:

Version 6

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

4.4 Use Unique Passwords
Where multi-factor authentication is not supported (such as local administrator, root, or service accounts), accounts will use passwords that are unique to that system.

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

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. The suggested value for this is 365 days or greater.

Rationale:

Reusing the same password after only a short period of time has passed makes the success of brute-force login attacks more likely.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT P.PROFILE, P.RESOURCE_NAME, P.LIMIT
FROM DBA_PROFILES P
WHERE TO_NUMBER(DECODE(P.LIMIT,
                       'DEFAULT',(SELECT DISTINCT
DECODE(LIMIT,'UNLIMITED',9999,LIMIT)
                                 FROM DBA_PROFILES
                                 WHERE PROFILE='DEFAULT'
                                 AND RESOURCE_NAME='PASSWORD_REUSE_TIME'),
                       'UNLIMITED','9999',P.LIMIT)) < 365 AND
      P.RESOURCE_NAME = 'PASSWORD_REUSE_TIME' AND
 EXISTS ( SELECT 'X' FROM DBA_USERS U WHERE U.PROFILE = P.PROFILE );
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT P.PROFILE, P.RESOURCE_NAME, P.LIMIT,
DECODE (P.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B
                        WHERE P.CON_ID = B.CON_ID)) DATABASE
FROM CDB_PROFILES P
```

```
WHERE TO_NUMBER(DECODE(P.LIMIT,
                       'DEFAULT',(SELECT DECODE(LIMIT,'UNLIMITED',9999,LIMIT)
                                  FROM CDB_PROFILES
                                  WHERE PROFILE='DEFAULT'
                                  AND RESOURCE_NAME='PASSWORD_REUSE_TIME'
                                AND CON_ID = P.CON_ID),
                       'UNLIMITED','9999',P.LIMIT)) < 365
AND P.RESOURCE_NAME = 'PASSWORD_REUSE_TIME'
AND EXISTS ( SELECT 'X' FROM CDB_USERS U WHERE U.PROFILE = P.PROFILE )
ORDER BY CON_ID, PROFILE, RESOURCE_NAME;
```

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;
```

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/CREATE-PROFILE.html#GUID-ABC7AE4D-64A8-4EA9-857D-BEF7300B64C3

Notes:

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

CIS Controls:

Version 6

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

4.4 Use Unique Passwords
Where multi-factor authentication is not supported (such as local administrator, root, or service accounts), accounts will use passwords that are unique to that system.

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

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. The suggested value for this is five days or less.

Rationale:

Locking the user account after the expiration of the password change requirement's grace period can help prevent password-based attacks against any forgotten or disused accounts, while still allowing the account and its information to be accessible by DBA intervention.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT P.PROFILE, P.RESOURCE_NAME, P.LIMIT
FROM DBA_PROFILES P
WHERE TO_NUMBER(DECODE(P.LIMIT,
                       'DEFAULT',(SELECT DISTINCT
DECODE(LIMIT,'UNLIMITED',9999,LIMIT)
                                  FROM DBA_PROFILES
                                  WHERE PROFILE='DEFAULT'
                                  AND RESOURCE_NAME='PASSWORD_GRACE_TIME'),
                       'UNLIMITED','9999',P.LIMIT)) > 5 AND
      P.RESOURCE_NAME = 'PASSWORD_GRACE_TIME' AND
 EXISTS ( SELECT 'X' FROM DBA_USERS U WHERE U.PROFILE = P.PROFILE );
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT P.PROFILE, P.RESOURCE_NAME, P.LIMIT,
DECODE (P.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B
                      WHERE P.CON_ID = B.CON_ID)) DATABASE
```

```
FROM CDB_PROFILES P
WHERE TO_NUMBER(DECODE(P.LIMIT,
                       'DEFAULT',(SELECT DECODE(LIMIT,'UNLIMITED',9999,LIMIT)
                                  FROM CDB_PROFILES
                                  WHERE PROFILE='DEFAULT'
                                  AND RESOURCE_NAME='PASSWORD_GRACE_TIME'
                                AND CON_ID = P.CON_ID),
                       'UNLIMITED','9999',P.LIMIT)) > 5
AND P.RESOURCE_NAME = 'PASSWORD_GRACE_TIME'
AND EXISTS ( SELECT 'X' FROM CDB_USERS U WHERE U.PROFILE = P.PROFILE )
ORDER BY CON_ID, PROFILE, RESOURCE_NAME;
```

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;
```

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/CREATE-PROFILE.html#GUID-ABC7AE4D-64A8-4EA9-857D-BEF7300B64C3

CIS Controls:

Version 6

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

16.10 Ensure All Accounts Have An Expiration Date
Ensure that all accounts have an expiration date that is monitored and enforced.

16.9 Disable Dormant Accounts
Automatically disable dormant accounts after a set period of inactivity.

3.7 Ensure 'PASSWORD_VERIFY_FUNCTION' Is Set for All Profiles (Automated)

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. It should be set for all profiles. Note that this setting does not apply for users managed by the Oracle password file.

Rationale:

Through Oracle database profiles, password complexity rules (mixed cases with digits and special characters), blocking of simple combinations, and enforcing change/history settings can potentially thwart unauthorized logins by an unauthorized user.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT P.PROFILE, P.RESOURCE_NAME, P.LIMIT
FROM DBA_PROFILES P
WHERE DECODE(P.LIMIT,
             'DEFAULT',(SELECT LIMIT
                        FROM DBA_PROFILES
                        WHERE PROFILE='DEFAULT'
                        AND RESOURCE_NAME = P.RESOURCE_NAME),
             LIMIT) = 'NULL'
AND P.RESOURCE_NAME = 'PASSWORD_VERIFY_FUNCTION'
AND EXISTS ( SELECT 'X'
             FROM DBA_USERS U
             WHERE U.PROFILE = P.PROFILE );
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT P.PROFILE, P.RESOURCE_NAME, P.LIMIT,
DECODE (P.CON_ID,0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B
```

```
                       WHERE P.CON_ID = B.CON_ID)) DATABASE
FROM CDB_PROFILES P
WHERE DECODE(P.LIMIT,
             'DEFAULT',(SELECT LIMIT
                         FROM CDB_PROFILES
                         WHERE PROFILE='DEFAULT'
                         AND RESOURCE_NAME = P.RESOURCE_NAME
                        AND CON_ID = P.CON_ID),
             LIMIT) = 'NULL'
AND P.RESOURCE_NAME = 'PASSWORD_VERIFY_FUNCTION'
AND EXISTS ( SELECT 'X'
             FROM CDB_USERS U
             WHERE U.PROFILE = P.PROFILE )
ORDER BY CON_ID, PROFILE, RESOURCE_NAME;
```

Lack of results implies compliance.

Remediation:

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/CREATE-PROFILE.html#GUID-ABC7AE4D-64A8-4EA9-857D-BEF7300B64C3

CIS Controls:

Version 6

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

4.4 Use Unique Passwords
Where multi-factor authentication is not supported (such as local administrator, root, or service accounts), accounts will use passwords that are unique to that system.

3.8 Ensure 'SESSIONS_PER_USER' Is Less than or Equal to '10' (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

- Level 1 - RDBMS using Unified Auditing

Description:

The `SESSIONS_PER_USER` setting determines the maximum number of user sessions that are allowed to be open concurrently. The suggested value for this is 10 or less.

Rationale:

Limiting the number of the `SESSIONS_PER_USER` can help prevent memory resource exhaustion by poorly formed requests or intentional denial-of-service attacks.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT P.PROFILE, P.RESOURCE_NAME, P.LIMIT
FROM DBA_PROFILES P
WHERE TO_NUMBER(DECODE(P.LIMIT,
                       'DEFAULT',(SELECT DISTINCT
DECODE(LIMIT,'UNLIMITED',9999,LIMIT)
                                 FROM DBA_PROFILES
                                 WHERE PROFILE='DEFAULT'
                                 AND RESOURCE_NAME='SESSIONS_PER_USER'),
                       'UNLIMITED','9999',P.LIMIT)) > 10
AND P.RESOURCE_NAME = 'SESSIONS_PER_USER'
AND EXISTS ( SELECT 'X' FROM DBA_USERS U WHERE U.PROFILE = P.PROFILE );
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT P.PROFILE, P.RESOURCE_NAME, P.LIMIT,
DECODE (P.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                 1,(SELECT NAME FROM V$DATABASE),
                   (SELECT NAME FROM V$PDBS B
                        WHERE P.CON_ID = B.CON_ID)) DATABASE
FROM CDB_PROFILES P
WHERE TO_NUMBER(DECODE(P.LIMIT,
                       'DEFAULT',(SELECT DECODE(LIMIT,'UNLIMITED',9999,LIMIT)
```

```
                             FROM CDB_PROFILES
                             WHERE PROFILE='DEFAULT'
                             AND RESOURCE_NAME='SESSIONS_PER_USER'
                           AND CON_ID = P.CON_ID),
                  'UNLIMITED','9999',P.LIMIT)) > 10
AND P.RESOURCE_NAME = 'SESSIONS_PER_USER'
AND EXISTS ( SELECT 'X' FROM CDB_USERS U WHERE U.PROFILE = P.PROFILE )
ORDER BY CON_ID, PROFILE, RESOURCE_NAME;
```

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;
```

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/CREATE-PROFILE.html#GUID-ABC7AE4D-64A8-4EA9-857D-BEF7300B64C3

Notes:

The `SESSIONS_PER_USER` profile management capability was created to prevent resource(s) exhaustion at a time when resource usage was 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.

CIS Controls:

Version 6

18 Application Software Security
Application Software Security

Version 7

16.7 Establish Process for Revoking Access
Establish and follow an automated process for revoking system access by disabling accounts immediately upon termination or change of responsibilities of an employee or contractor . Disabling these accounts, instead of deleting accounts, allows preservation of audit trails.

3.9 Ensure 'INACTIVE_ACCOUNT_TIME' Is Less than or Equal to '120' (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

- Level 1 - RDBMS using Unified Auditing

Description:

The 'INACTIVE_ACCOUNT_TIME' setting determines the maximum number of days of inactivity (no logins at all) after which the account will be locked. The suggested value for this is 120 or less.

Rationale:

Setting 'INACTIVE_ACCOUNT_TIME' can help with deactivation of "inactive" or "unused" accounts.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT P.PROFILE, P.RESOURCE_NAME, P.LIMIT
FROM DBA_PROFILES P
WHERE TO_NUMBER(DECODE(P.LIMIT,'DEFAULT',(SELECT DISTINCT
DECODE(LIMIT,'UNLIMITED',9999,LIMIT)
                                FROM DBA_PROFILES
                                WHERE PROFILE='DEFAULT'
                                AND
RESOURCE_NAME='INACTIVE_ACCOUNT_TIME'),
                        'UNLIMITED','9999',
                        P.LIMIT)) > 120
AND P.RESOURCE_NAME = 'INACTIVE_ACCOUNT_TIME'
AND EXISTS ( SELECT 'X' FROM DBA_USERS U WHERE U.PROFILE = P.PROFILE );
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT DISTINCT P.PROFILE, P.RESOURCE_NAME, P.LIMIT,
DECODE (P.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                    (SELECT NAME FROM V$PDBS B
                        WHERE P.CON_ID = B.CON_ID)) DATABASE
```

```
FROM CDB_PROFILES P
WHERE TO_NUMBER(DECODE(P.LIMIT,
                       'DEFAULT',(SELECT DISTINCT
DECODE(LIMIT,'UNLIMITED',9999,LIMIT)
                              FROM CDB_PROFILES
                              WHERE PROFILE='DEFAULT'
                              AND RESOURCE_NAME='INACTIVE_ACCOUNT_TIME'
                           AND CON_ID = P.CON_ID),
                    'UNLIMITED','9999',
                    P.LIMIT)) > 120
AND P.RESOURCE_NAME = 'INACTIVE_ACCOUNT_TIME'
AND EXISTS ( SELECT 'X' FROM CDB_USERS U WHERE U.PROFILE = P.PROFILE );
```

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 INACTIVE_ACCOUNT_TIME 120;
```

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/CREATE-PROFILE.html#GUID-ABC7AE4D-64A8-4EA9-857D-BEF7300B64C3

Notes:

The SESSIONS_PER_USER profile management capability was created to prevent resource(s) exhaustion at a time when resource usage was 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.

CIS Controls:

Version 6

18 Application Software Security
Application Software Security

Version 7

16.9 Disable Dormant Accounts
Automatically disable dormant accounts after a set period of inactivity.

4 Users

4.1 Ensure All Default Passwords Are Changed (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

- Level 1 - RDBMS using Unified Auditing

Description:

Default passwords should not be used by Oracle database users.

Rationale:

Default passwords should be considered "well known" to attackers. Consequently, if default passwords remain in place, any attacker with access to the database can authenticate as the user with that default password.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT DISTINCT A.USERNAME
FROM DBA_USERS_WITH_DEFPWD A, DBA_USERS B
WHERE A.USERNAME = B.USERNAME
AND B.ACCOUNT_STATUS = 'OPEN';
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT DISTINCT A.USERNAME,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_USERS_WITH_DEFPWD A, CDB_USERS C
WHERE A.USERNAME = C.USERNAME
AND C.ACCOUNT_STATUS = 'OPEN';
```

Lack of results implies compliance.

The view called `CDB_USERS_WITH_DEFPWD` and `DBA_USERS_WITH_DEFPWD` shows a list of all database users making use of default passwords. Lack of results implies compliance.

Note: Per Oracle Support Document 2173962.1, "after creation of a new…database, the `SYS` and `SYSTEM` accounts [may be] listed in `DBA_USERS_WITH_DEFPWD` even though the accounts were created with non-default passwords. Setting the same passwords again with `ALTER USER` correctly recognizes that the accounts do not have default passwords."

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

- Manually issue the following SQL statement for each USERNAME returned in the Audit Procedure:
- Execute the following SQL script to assign a randomly generated password to each account using a default password:

```
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. https://docs.oracle.com/en/database/oracle/oracle-database/18/dbseg/keeping-your-oracle-database-secure.html#GUID-5D5EDAA6-AB24-424D-ABA1-F3869E445544
2. https://support.oracle.com/epmos/faces/DocumentDisplay?id=2173962.1

CIS Controls:

Version 6

5.3 Change Default Passwords On All New Devices
Before deploying any new devices in a networked environment, change all default passwords for applications, operating systems, routers, firewalls, wireless access points, and other systems to have values consistent with administration-level accounts.

Version 7

4.2 <u>Change Default Passwords</u>

Before deploying any new asset, change all default passwords to have values consistent with administrative level accounts.

4.2 Ensure All Sample Data And Users Have Been Removed (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

- Level 1 - RDBMS using Unified Auditing

Description:

Oracle sample schemas can be used to create sample users (`BI,HR,IX,OE,PM,SCOTT,SH`), with well-known default passwords, particular views, and procedures/functions, in addition to tables and fictitious data. The sample schemas should be removed.

Rationale:

The sample schemas are typically not required for production operations of the database. The default users, views, and/or procedures/functions created by sample schemas could be used to launch exploits against production environments.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT DISTINCT A.USERNAME,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_USERS A
WHERE A.USERNAME IN ('BI','HR','IX','OE','PM','SCOTT','SH');
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to run the drop 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 usernames 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 usernames 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. https://docs.oracle.com/en/database/oracle/oracle-database/18/comsc/toc.htm

CIS Controls:

Version 6

18.9 Sanitize Deployed Software Of Development Artifacts
For in-house developed applications, ensure that development artifacts (sample data and scripts; unused libraries, components, debug code; or tools) are not included in the deployed software, or accessible in the production environment.

Version 7

4.7 Limit Access to Script Tools
Limit access to scripting tools (such as Microsoft PowerShell and Python) to only administrative or development users with the need to access those capabilities.

4.3 Ensure 'DBA_USERS.AUTHENTICATION_TYPE' Is Not Set to 'EXTERNAL' for Any User (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

- Level 1 - RDBMS using Unified Auditing

Description:

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

Rationale:

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.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT A.USERNAME,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B
                     WHERE A.CON_ID = B.CON_ID))
FROM CDB_USERS A
WHERE AUTHENTICATION_TYPE = 'EXTERNAL';
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

```
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 compatibility.

CIS Controls:

Version 6

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

16.2 Configure Centralized Point of Authentication
Configure access for all accounts through as few centralized points of authentication as possible, including network, security, and cloud systems.

4.4 Ensure No Users Are Assigned the 'DEFAULT' Profile (Automated)

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. No users should be assigned to that profile.

Rationale:

Users should 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:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT USERNAME
FROM DBA_USERS
WHERE PROFILE='DEFAULT'
AND ACCOUNT_STATUS='OPEN'
AND ORACLE_MAINTAINED = 'N';
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT A.USERNAME,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_USERS A
WHERE A.PROFILE='DEFAULT'
AND A.ACCOUNT_STATUS='OPEN'
AND A.ORACLE_MAINTAINED = 'N';
```

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, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

CIS Controls:

Version 6

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

4.3 Ensure the Use of Dedicated Administrative Accounts
Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities.

4.5 Ensure 'SYS.USER$MIG' Has Been Dropped (Automated)

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. This table should be dropped.

Rationale:

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

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT OWNER, TABLE_NAME,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_TABLES A
WHERE TABLE_NAME='USER$MIG' AND OWNER='SYS';
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

CIS Controls:

Version 6

16.14 Encrypt/Hash All Authentication Files And Monitor Their Access
Verify that all authentication files are encrypted or hashed and that these files cannot be accessed without root or administrator privileges. Audit all access to password files in the system.

Version 7

16.4 Encrypt or Hash all Authentication Credentials
Encrypt or hash with a salt all authentication credentials when stored.

4.6 Ensure No Public Database Links Exist (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

Public Database links are used to allow connections between databases.

Rationale:

Using public database links in the database can allow anyone with a connection to the database to query, update, insert, delete data on a remote database depending on the userid that is part of the link.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT DB_LINK, HOST FROM DBA_DB_LINKS WHERE OWNER = 'PUBLIC';
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT DB_LINK, HOST,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_DB_LINKS A
WHERE OWNER = 'PUBLIC';
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

```
DROP PUBLIC DATABASE LINK <DB_LINK>;
```

CIS Controls:

Version 7

14.6 <u>Protect Information through Access Control Lists</u>

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5 Privileges & Grants & ACLs

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 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 and 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.

5.1 Excessive Table, View and Package Privileges

This section contains recommendations that revoke default public execute privileges from powerful packages and object types.

5.1.1 Public Privileges

This section contains recommendations that revoke default public execute privileges from powerful packages and object types.

5.1.1.1 Ensure 'EXECUTE' is revoked from 'PUBLIC' on "Network" Packages (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

- Level 1 - RDBMS using Unified Auditing

Description:

As described below, Oracle Database PL/SQL "Network" packages - `DBMS_LDAP`, `UTL_INADDR`, `UTL_TCP`, `UTL_MAIL`, `UTL_SMTP`, `UTL_DBWS`, `UTL_ORAMTS`, `UTL_HTTP` and type `HTTPURITYPE` – provide PL/SQL APIs to interact or access remote servers. The PUBLIC should not be able to execute these packages.

- The Oracle database `DBMS_LDAP` package contains functions and procedures that enable programmers to access data from LDAP servers.
- The Oracle database `UTL_INADDR` package provides an API to retrieve host names and IP addresses of local and remote hosts.
- 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.
- The Oracle database `UTL_MAIL` package can be used to send email from the server where the Oracle instance is installed.
- The Oracle database `UTL_SMTP` package can be used to send email from the server where the Oracle instance is installed. The user `PUBLIC` should not be able to execute `UTL_SMTP`.
- 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.
- The Oracle database `UTL_ORAMTS` package can be used to perform HTTP requests. This could be used to send information to the outside.

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

Rationale:

As described below, Oracle Database PL/SQL packages - `DBMS_LDAP`, `UTL_INADDR`, `UTL_TCP`, `UTL_MAIL`, `UTL_SMTP`, `UTL_DBWS`, `UTL_ORAMTS`, `UTL_HTTP` and type `HTTPURITYPE` can be used by unauthorized users to create specially crafted error messages or send information to external servers. The `PUBLIC` should not be able to execute these packages.

- The use of the `DBMS_LDAP` package can be used to create specially crafted error messages or send information via DNS to the outside.
- The `UTL_INADDR` package can be used to create specially crafted error messages or send information via DNS to the outside.
- The `UTL_TCP` package could allow an unauthorized user to corrupt the TCP stream used to carry the protocols that communicate with the instance's external communications.
- 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.
- 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.
- The `UTL_DBWS` package could allow an unauthorized user to corrupt the HTTP stream used to carry the protocols that communicate for the instance's web-based external communications.
- 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.
- 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.
- The ability to perform HTTP requests could be used to leak information from the database to an external destination.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT TABLE_NAME, PRIVILEGE, GRANTEE
FROM DBA_TAB_PRIVS
```

```
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME IN
('DBMS_LDAP','UTL_INADDR','UTL_TCP','UTL_MAIL','UTL_SMTP','UTL_DBWS','UTL_ORA
MTS','UTL_HTTP','HTTPURITYPE');
```

Lack of results implies compliance.

Multi-tenant in the container database:

This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT TABLE_NAME, PRIVILEGE, GRANTEE,DECODE (A.CON_ID,0,(SELECT NAME FROM
V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                   (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_TAB_PRIVS A
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME IN
('DBMS_LDAP','UTL_INADDR','UTL_TCP','UTL_MAIL','UTL_SMTP','UTL_DBWS','UTL_ORA
MTS','UTL_HTTP','HTTPURITYPE')
ORDER BY CON_ID, TABLE_NAME;
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

```
REVOKE EXECUTE ON DBMS_LDAP FROM PUBLIC;
REVOKE EXECUTE ON UTL_INADDR FROM PUBLIC;
REVOKE EXECUTE ON UTL_TCP FROM PUBLIC;
REVOKE EXECUTE ON UTL_MAIL FROM PUBLIC;
REVOKE EXECUTE ON UTL_SMTP FROM PUBLIC;
REVOKE EXECUTE ON UTL_DBWS FROM PUBLIC;
REVOKE EXECUTE ON UTL_ORAMTS FROM PUBLIC;
REVOKE EXECUTE ON UTL_HTTP FROM PUBLIC;
REVOKE EXECUTE ON HTTPURITYPE FROM PUBLIC;
```

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.1.1.2 Ensure 'EXECUTE' is revoked from 'PUBLIC' on "File System" Packages (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

- Level 1 - RDBMS using Unified Auditing

Description:

As described below, Oracle Database PL/SQL "File System" packages - `DBMS_ADVISOR`, `DBMS_LOB` and `UTL_FILE` – provide PL/SQL APIs to access files on the servers. The user `PUBLIC` should not be able to execute these packages.

- The Oracle database `DBMS_ADVISOR` package can be used to write files located on the server where the Oracle instance is installed. The user PUBLIC should not be able to execute `DBMS_ADVISOR`.
- The Oracle database `DBMS_LOB` package provides subprograms that can manipulate and read/write on `BLOB`'s, `CLOB`'s, `NCLOB`'s, `BFILE`'s, and temporary `LOB`'s. The user `PUBLIC` should not be able to execute `DBMS_LOB`.
- The Oracle database `UTL_FILE` package can be used to read/write files located on the server where the Oracle instance is installed. The user `PUBLIC` should not be able to execute `UTL_FILE`.

Rationale:

As described below, Oracle Database PL/SQL "File System" packages - `DBMS_ADVISOR`, `DBMS_LOB` and `UTL_FILE` – should not be granted to `PUBLIC`.

- Use of the `DBMS_ADVISOR` package could allow an unauthorized user to corrupt operating system files on the instance's host.
- Use of the `DBMS_LOB` package could allow an unauthorized user to manipulate `BLOB`'s, `CLOB`'s, `NCLOB`'s, `BFILE`'s, and temporary LOBs on the instance, either destroying data or causing a denial-of-service condition due to corruption of disk space.
- Use of the `UTL_FILE` package could allow a user to read OS files. These files could contain sensitive information (e.g. passwords in `.bash_history`)

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT TABLE_NAME, PRIVILEGE, GRANTEE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME IN ('DBMS_ADVISOR','DBMS_LOB','UTL_FILE');
```

Lack of results implies compliance.

Multi-tenant in the container database:

This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT TABLE_NAME, PRIVILEGE, GRANTEE,DECODE (A.CON_ID,0,(SELECT NAME FROM
V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_TAB_PRIVS A
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME IN ('DBMS_ADVISOR','DBMS_LOB','UTL_FILE')
ORDER BY CON_ID, TABLE_NAME;
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the

principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.1.1.3 Ensure 'EXECUTE' is revoked from 'PUBLIC' on "Encryption" Packages (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

- Level 1 - RDBMS using Unified Auditing

Description:

As described below, Oracle Database PL/SQL "Encryption" packages - `DBMS_CRYPTO`, `DBMS_OBFUSCATION_TOOLKIT` and `DBMS_RANDOM` – provide PL/SQL APIs to perform functions related to cryptography. The `PUBLIC` should not be able to execute these packages.

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

Rationale:

As described below, Oracle Database PL/SQL Encryption packages - `DBMS_CRYPTO`, `DBMS_OBFUSCATION_TOOLKIT` and `DBMS_RANDOM` – should not be granted to `PUBLIC`.

- Execution of the `DBMS_CRYPTO` procedures by the `PUBLIC` can potentially endanger portions of or all of the data storage.
- Allowing the `PUBLIC` privileges to access this capability can be potentially harm data storage.
- Use of the `DBMS_RANDOM` package can allow the unauthorized application of the random number-generating function.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT TABLE_NAME, PRIVILEGE, GRANTEE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME IN ('DBMS_CRYPTO','DBMS_OBFUSCATION_TOOLKIT', 'DBMS_RANDOM');
```

Lack of results implies compliance.

Multi-tenant in the container database:

This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT TABLE_NAME, PRIVILEGE, GRANTEE,DECODE (A.CON_ID,0,(SELECT NAME FROM
V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_TAB_PRIVS A
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME IN ('DBMS_CRYPTO','DBMS_OBFUSCATION_TOOLKIT', 'DBMS_RANDOM')
ORDER BY CON_ID, TABLE_NAME;
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the

principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.1.1.4 Ensure 'EXECUTE' is revoked from 'PUBLIC' on "Java" Packages (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

- Level 1 - RDBMS using Unified Auditing

Description:

As described below, Oracle Database PL/SQL "Java" packages - `DBMS_JAVA` and `DBMS_JAVA_TEST` – provide APIs to run Java classes or grant Java packages. The user `PUBLIC` should not be able to execute these packages.

- The Oracle database `DBMS_JAVA` package can run Java classes (e.g. OS commands) or grant Java privileges. The user `PUBLIC` should not be able to execute `DBMS_JAVA`.
- The Oracle database `DBMS_JAVA_TEST` package can run Java classes (e.g. OS commands) or grant Java privileges. The user `PUBLIC` should not be able to execute `DBMS_JAVA_TEST`.

Rationale:

As described below, Oracle Database PL/SQL "Java" packages - `DBMS_JAVA` and `DBMS_JAVA_TEST` – should not be granted to `PUBLIC`.

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

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT TABLE_NAME, PRIVILEGE, GRANTEE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME IN ('DBMS_JAVA','DBMS_JAVA_TEST');
```

Lack of results implies compliance.

Multi-tenant in the container database:

This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT TABLE_NAME, PRIVILEGE, GRANTEE,DECODE (A.CON_ID,0,(SELECT NAME FROM
V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_TAB_PRIVS A
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME IN ('DBMS_JAVA','DBMS_JAVA_TEST')
ORDER BY CON_ID, TABLE_NAME;
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.1.1.5 Ensure 'EXECUTE' is revoked from 'PUBLIC' on "Job Scheduler" Packages (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

- Level 1 - RDBMS using Unified Auditing

Description:

As described below, Oracle Database PL/SQL "Job Scheduler" packages - `DBMS_SCHEDULER` and `DBMS_JOB` – provide APIs to schedule jobs. The user `PUBLIC` should not be able to execute these packages.

- The Oracle database `DBMS_SCHEDULER` package schedules and manages the database and operating system jobs. The user `PUBLIC` should not be able to execute `DBMS_SCHEDULER`.
- 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. The user `PUBLIC` should not be able to execute `DBMS_JOB`.

Rationale:

As described below, Oracle Database PL/SQL "Job Scheduler" packages - `DBMS_SCHEDULER` and `DBMS_JOB` – should not be granted to the user `PUBLIC`.

- Use of the `DBMS_SCHEDULER` package could allow an unauthorized user to run database or operating system jobs.
- Use of the `DBMS_JOB` package could allow an unauthorized user to disable or overload the job queue. It has been superseded by the `DBMS_SCHEDULER` package.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT TABLE_NAME, PRIVILEGE, GRANTEE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME IN ('DBMS_SCHEDULER','DBMS_JOB');
```

Lack of results implies compliance.

Multi-tenant in the container database:

This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT TABLE_NAME, PRIVILEGE, GRANTEE,DECODE (A.CON_ID,0,(SELECT NAME FROM
V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_TAB_PRIVS A
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME IN ('DBMS_SCHEDULER','DBMS_JOB')
ORDER BY CON_ID, TABLE_NAME;
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

CIS Controls:

Version 6

   5.1 Minimize And Sparingly Use Administrative Privileges
   Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

   14.6 Protect Information through Access Control Lists
   Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.1.1.6 Ensure 'EXECUTE' is revoked from 'PUBLIC' on "SQL Injection Helper" Packages (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

- Level 1 - RDBMS using Unified Auditing

Description:

As described below, Oracle Database PL/SQL "SQL Injection Helper Packages" packages - `DBMS_SQL`, `DBMS_XMLGEN`, `DBMS_XMLQUERY`, `DBMS_XLMSTORE`, `DBMS_XLMSAVE` and `DBMS_REDACT` – provide APIs to schedule jobs. The user `PUBLIC` should not be able to execute these packages.

- The Oracle database `DBMS_SQL` package is used for running dynamic SQL statements.
- The `DBMS_XMLGEN` package takes an arbitrary SQL query as input, converts it to XML format, and returns the result as a `CLOB`.
- 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`.
- The `DBMS_XLMSTORE` package provides XML functionality. It accepts a table name and XML as input to perform `DML` operations against the table.
- The `DBMS_XLMSAVE` package provides XML functionality. It accepts a table name and XML as input and then inserts into or updates that table.
- The DBMS_REDACT package provides an interface to Oracle Data Redaction, which enables you to mask (redact) data that is returned from queries issued by low-privileged users or an application.

Rationale:

As described below, Oracle Database PL/SQL "SQL Injection Helper Packages" packages - `DBMS_SQL`, `DBMS_XMLGEN`, `DBMS_XMLQUERY`, `DBMS_XLMSTORE`, `DBMS_XLMSAVE` and 'DBMS_REDACT' – should not be granted to `PUBLIC`.

- The `DBMS_SQL` package could allow privilege escalation if input validation is not done properly.
- The package `DBMS_XMLGEN` can be used to search the entire database for sensitive information like credit card numbers
- The package `DBMS_XMLQUERY` can be used to search the entire database for sensitive information like credit card numbers. Malicious users may be able to exploit this package as an auxiliary inject function in a SQL injection attack.
- Malicious users may be able to exploit the `DBMS_XLMSTORE` package as an auxiliary inject function in a SQL injection attack.

- Malicious users may be able to exploit the `DBMS_XLMSAVE` package as an auxiliary inject function in a SQL injection attack.
- Malicious users may be able to exploit DBMS_REDACT as an auxiliary inject function in a SQL injection attack.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT TABLE_NAME, PRIVILEGE, GRANTEE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME IN ('DBMS_SQL', 'DBMS_XMLGEN',
'DBMS_XMLQUERY','DBMS_XMLSTORE','DBMS_XMLSAVE','DBMS_AW','OWA_UTIL','DBMS_RED
ACT');
```

Lack of results implies compliance.

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT TABLE_NAME, PRIVILEGE, GRANTEE,DECODE (A.CON_ID,0,(SELECT NAME FROM
V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                   (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_TAB_PRIVS A
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME IN ('DBMS_SQL', 'DBMS_XMLGEN',
'DBMS_XMLQUERY','DBMS_XMLSTORE','DBMS_XMLSAVE','DBMS_AW','OWA_UTIL','DBMS_RED
ACT')
ORDER BY CON_ID, TABLE_NAME;
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

```
REVOKE EXECUTE ON DBMS_SQL FROM PUBLIC;
REVOKE EXECUTE ON DBMS_XMLGEN FROM PUBLIC;
REVOKE EXECUTE ON DBMS_XMLQUERY FROM PUBLIC;
REVOKE EXECUTE ON DBMS_XMLSAVE FROM PUBLIC;
REVOKE EXECUTE ON DBMS_XMLSTORE FROM PUBLIC;
REVOKE EXECUTE ON DBMS_AW FROM PUBLIC;
```

```
REVOKE EXECUTE ON OWA_UTIL FROM PUBLIC;
REVOKE EXECUTE ON DBMS_REDACT FROM PUBLIC;
```

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.1.2 Non-Default Privileges

5.1.2.1 Ensure 'EXECUTE' is not granted to 'PUBLIC' on "Non-default" Packages (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

- Level 1 - RDBMS using Unified Auditing

Description:

The packages described in this control are not granted to `PUBLIC` by default ("Non-default" packages). These packages should not be granted to `PUBLIC`.

- The Oracle database `DBMS_BACKUP_RESTORE` package is used for applying PL/SQL commands to the native `RMAN` sequences.
- The Oracle database `DBMS_FILE_TRANSFER` package allows a user to transfer files from one database server to another.
- The Oracle database `DBMS_SYS_SQL`, `DBMS_REPCAT_SQL_UTL`, `INITJVMAUX`, `DBMS_AQADM_SYS`, `DBMS_STREAMS_RPC`, `DBMS_PRVTAQIM`, `LTADM` and `DBMS_IJOB` packages are shipped as undocumented.

Rationale:

As described below, these "non-default" group of PL/SQL packages, which are not granted to `PUBLIC` by default, packages should not be granted to `PUBLIC`.

- The `DBMS_BACKUP_RESTORE` package can allow access to OS files.
- The `DBMS_FILE_TRANSFER` package could allow to transfer files from one database server to another without authorization to do so.
- The `DBMS_SYS_SQL` package could allow a user to run code as a different user without entering valid credentials.
- The `DBMS_REPCAT_SQL_UTL` package could allow an unauthorized user to run SQL commands as user `SYS`.
- The `INITJVMAUX` package could allow an unauthorized user to run SQL commands as user `SYS`.
- The `DBMS_AQADM_SYS` package could allow an unauthorized user to run SQL commands as user `SYS`.
- The `DBMS_STREAMS_RPC` package could allow an unauthorized user to run SQL commands as user `SYS`.

- The `DBMS_PRVTAQIM` package could allow an unauthorized user to escalate privileges because any SQL statements could be executed as user `SYS`.
- The `LTADM` package could allow an unauthorized user to run any SQL command as user `SYS`. It allows privilege escalation if granted to unprivileged users.
- The `DBMS_IJOB` package could allow an attacker to change identities by using a different username to execute a database job. It allows a user to run database jobs in the context of another user.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT TABLE_NAME, PRIVILEGE, GRANTEE
FROM DBA_TAB_PRIVS
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME IN
('DBMS_BACKUP_RESTORE','DBMS_FILE_TRANSFER','DBMS_SYS_SQL','DBMS_REPCAT_SQL_U
TL','INITJVMAUX',
'DBMS_AQADM_SYS','DBMS_STREAMS_RPC','DBMS_PRVTAQIM','LTADM',
'DBMS_IJOB','DBMS_PDB_EXEC_SQL');
```

Lack of results implies compliance.

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT TABLE_NAME, PRIVILEGE, GRANTEE,DECODE (A.CON_ID,0,(SELECT NAME FROM
V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_TAB_PRIVS A
WHERE GRANTEE='PUBLIC'
AND PRIVILEGE='EXECUTE'
AND TABLE_NAME IN
('DBMS_BACKUP_RESTORE','DBMS_FILE_TRANSFER','DBMS_SYS_SQL','DBMS_REPCAT_SQL_U
TL','INITJVMAUX',
'DBMS_AQADM_SYS','DBMS_STREAMS_RPC','DBMS_PRVTAQIM','LTADM',
'DBMS_IJOB','DBMS_PDB_EXEC_SQL')
ORDER BY CON_ID, TABLE_NAME;
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

```
REVOKE EXECUTE ON DBMS_BACKUP_RESTORE FROM PUBLIC;
REVOKE EXECUTE ON DBMS_FILE_TRANSFER FROM PUBLIC;
REVOKE EXECUTE ON DBMS_SYS_SQL FROM PUBLIC;
REVOKE EXECUTE ON DBMS_REPCAT_SQL_UTL FROM PUBLIC;
REVOKE EXECUTE ON INITJVMAUX FROM PUBLIC;
REVOKE EXECUTE ON DBMS_AQADM_SYS FROM PUBLIC;
REVOKE EXECUTE ON DBMS_STREAMS_RPC FROM PUBLIC;
REVOKE EXECUTE ON DBMS_PRVTAQIM FROM PUBLIC;
REVOKE EXECUTE ON LTADM FROM PUBLIC;
REVOKE EXECUTE ON DBMS_IJOB FROM PUBLIC;
REVOKE EXECUTE ON DBMS_PDB_EXEC_SQL FROM PUBLIC;
```

CIS Controls:

Version 6

18 Application Software Security
Application Software Security

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.1.3 Other Privileges

5.1.3.1 Ensure 'ALL' Is Revoked from Unauthorized 'GRANTEE' on 'AUD$' (Automated)

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`, and `CREATE`, and so forth. (DML changes need trigger-based audit events to record data alterations.) Unauthorized grantees should not have full access to that table.

Rationale:

Permitting non-privileged users the authorization to manipulate the `SYS.AUD$` table can allow distortion of the audit records, hiding unauthorized activities.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Lack of results implies compliance.

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_TAB_PRIVS A
```

```
WHERE TABLE_NAME='AUD$'
AND OWNER = 'SYS';
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/dbseg/administering-the-audit-trail.html#GUID-9F298B8A-6196-4206-A889-A7CEB0924CF1

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.1.3.2 Ensure 'ALL' Is Revoked from Unauthorized 'GRANTEE' on 'DBA_%' (Automated)

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. Unauthorized grantees should not have full access to those views.

Rationale:

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

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE,TABLE_NAME
FROM DBA_TAB_PRIVS
WHERE TABLE_NAME LIKE 'DBA_%'
AND OWNER = 'SYS'
AND GRANTEE NOT IN (SELECT USERNAME FROM DBA_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE,TABLE_NAME,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_TAB_PRIVS A
WHERE TABLE_NAME LIKE 'DBA_%'
AND OWNER = 'SYS'
AND GRANTEE NOT IN (SELECT USERNAME FROM CDB_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM CDB_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Lack of results implies compliance.

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, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke:

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/static-data-dictionary-views.html#GUID-8865F65B-EF6D-44A5-B0A1-3179EFF0C36A

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.1.3.3 Ensure 'ALL' Is Revoked on 'Sensitive' Tables (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database tables listed below may contain sensitive information, and should not be accessible to unauthorized users.

- `USER$`, `USER_HISTORY$`, `XS$VERIFIERS` and `DEFAULT_PWD$` may contain password hashes.
- `CDB_LOCAL_ADMINAUTH$` and `PDB_SYNC$` may contain DDLs.
- `LINK$` and `SCHEDULER$_CREDENTIAL` may contain encrypted passwords.
- `ENC$` may contains encryption keys.
- `HISTGRM$` and `HIST_HEAD$` may contain sensitive data.

Rationale:

Access to sensitive information such as hashed passwords may allow unauthorized users to decrypt the passwords hashes which could potentially result in complete compromise of the database.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE, TABLE_NAME
FROM DBA_TAB_PRIVS
WHERE TABLE_NAME in
('CDB_LOCAL_ADMINAUTH$','DEFAULT_PWD$','ENC$','HISTGRM$','HIST_HEAD$','LINK$'
,'PDB_SYNC$','SCHEDULER$_CREDENTIAL','USER$','USER_HISTORY$','XS$VERIFIERS')
AND OWNER = 'SYS'
AND GRANTEE NOT IN (SELECT USERNAME FROM DBA_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Lack of results implies compliance.

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT TABLE_NAME, PRIVILEGE, GRANTEE,DECODE (A.CON_ID,0,(SELECT NAME FROM
V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
DATABASE
FROM CDB_TAB_PRIVS A
WHERE TABLE_NAME in
('CDB_LOCAL_ADMINAUTH$','DEFAULT_PWD$','ENC$','HISTGRM$','HIST_HEAD$','LINK$'
,'PDB_SYNC$','SCHEDULER$_CREDENTIAL','USER$','USER_HISTORY$','XS$VERIFIERS')
AND OWNER = 'SYS'
AND GRANTEE NOT IN (SELECT USERNAME FROM DBA_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y')
ORDER BY CON_ID, TABLE_NAME;
```

Lack of results implies compliance.

Remediation:

Execute applicable SQLs listed below to remediate:

```
REVOKE ALL ON SYS.CDB_LOCAL_ADMINAUTH$ FROM <grantee>;
REVOKE ALL ON SYS.DEFAULT_PWD$ FROM <grantee>;
REVOKE ALL ON SYS.ENC$ FROM <grantee>;
REVOKE ALL ON SYS.HISTGRM$ FROM <grantee>;
REVOKE ALL ON SYS.HIST_HEAD$ FROM <grantee>;
REVOKE ALL ON SYS.LINK$ FROM <grantee>;
REVOKE ALL ON SYS.PDB_SYNC$ FROM <grantee>;
REVOKE ALL ON SYS.SCHEDULER$_CREDENTIAL FROM <grantee>;
REVOKE ALL ON SYS.USER$ FROM <grantee>;
REVOKE ALL ON SYS.USER_HISTORY$ FROM <grantee>;
REVOKE ALL ON SYS.XS$VERIFIERS FROM <grantee>;
```

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.2 Excessive System Privileges

The recommendations within this section revoke excessive system privileges.

5.2.1 Ensure '%ANY%' Is Revoked from Unauthorized 'GRANTEE' (Automated)

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. Unauthorized grantees should not have that keyword assigned to them.

Rationale:

Authorization to use the `ANY` expansion of a privilege can allow an unauthorized user to potentially change confidential data or damage the data catalog.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE LIKE '%ANY%'
AND GRANTEE NOT IN (SELECT USERNAME FROM DBA_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_SYS_PRIVS A
WHERE PRIVILEGE LIKE '%ANY%'
```

```
AND GRANTEE NOT IN (SELECT USERNAME FROM CDB_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM CDB_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is
granted in both container and pluggable database, you must connect to both places to
revoke.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-
   database/18/dbseg/configuring-privilege-and-role-authorization.html#GUID-
   6429A893-1D8F-49E8-9022-61B5D30B885E

CIS Controls:

Version 6

14.4 Protect Information With Access Control Lists
   All information stored on systems shall be protected with file system, network share,
claims, application, or database specific access control lists. These controls will enforce the
principle that only authorized individuals should have access to the information based on
their need to access the information as a part of their responsibilities.

Version 7

14.6 Protect Information through Access Control Lists
   Protect all information stored on systems with file system, network share, claims,
application, or database specific access control lists. These controls will enforce the
principle that only authorized individuals should have access to the information based on
their need to access the information as a part of their responsibilities.

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

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. Unauthorized grantees should not have that privilege.

Rationale:

Assignment of the `WITH_ADMIN` privilege can allow the granting of a restricted privilege to an unauthorized user.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE ADMIN_OPTION='YES'
AND GRANTEE NOT IN (SELECT USERNAME FROM DBA_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_SYS_PRIVS A
WHERE ADMIN_OPTION='YES'
AND GRANTEE NOT IN (SELECT USERNAME FROM CDB_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM CDB_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.2.3 Ensure 'EXECUTE ANY PROCEDURE' Is Revoked from 'OUTLN' (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

- Level 1 - RDBMS using Unified Auditing

Description:

Remove unneeded `EXECUTE ANY PROCEDURE` privileges from `OUTLN`.

Rationale:

Migrated `OUTLN` users have more privileges than required.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_SYS_PRIVS A
WHERE PRIVILEGE='EXECUTE ANY PROCEDURE'
AND GRANTEE='OUTLN';
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.2.4 Ensure 'EXECUTE ANY PROCEDURE' Is Revoked from 'DBSNMP' (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

- Level 1 - RDBMS using Unified Auditing

Description:

Remove unneeded `EXECUTE ANY PROCEDURE` privileges from `DBSNMP`.

Rationale:

Migrated `DBSNMP` users have more privileges than required.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_SYS_PRIVS A
WHERE PRIVILEGE='EXECUTE ANY PROCEDURE'
AND GRANTEE='DBSNMP';
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.2.5 Ensure 'SELECT ANY DICTIONARY' Is Revoked from Unauthorized 'GRANTEE' (Automated)

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. Unauthorized grantees should not have that privilege.

Rationale:

`SELECT ANY DICTIONARY` is a powerful system privilege which would allow an unauthorized user to gather information about the database through data dictionary objects. Information collected could potentially be used to exploit the database.

Audit:

Non multi-tenant or pluggable database only:
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 (SELECT USERNAME FROM DBA_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
             1,(SELECT NAME FROM V$DATABASE),
                (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_SYS_PRIVS A
WHERE PRIVILEGE='SELECT ANY DICTIONARY'
AND GRANTEE NOT IN (SELECT USERNAME FROM CDB_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM CDB_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/dbseg/configuring-privilege-and-role-authorization.html#GUID-89CE989D-C97F-4CFD-941F-18203090A1AC
2. http://arup.blogspot.de/2011/07/difference-between-select-any.html

CIS Controls:

Version 6

14.4 Protect Information With Access Control Lists
   All information stored on systems shall be protected with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

Version 7

14.6 Protect Information through Access Control Lists
   Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.2.6 Ensure 'SELECT ANY TABLE' Is Revoked from Unauthorized 'GRANTEE' (Automated)

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 `SYS`, to view it. Unauthorized grantees should not have that privilege.

Rationale:

Assignment of the `SELECT ANY TABLE` privilege can allow the unauthorized viewing of sensitive data.

Audit:

Non multi-tenant or pluggable database only:
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 (SELECT USERNAME FROM DBA_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_SYS_PRIVS A
WHERE PRIVILEGE='SELECT ANY TABLE'
AND GRANTEE NOT IN (SELECT USERNAME FROM CDB_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM CDB_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/dbseg/configuring-privilege-and-role-authorization.html#GUID-89CE989D-C97F-4CFD-941F-18203090A1AC

Notes:

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

CIS Controls:

Version 6

14.4 Protect Information With Access Control Lists
   All information stored on systems shall be protected with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

Version 7

14.6 Protect Information through Access Control Lists
   Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.2.7 Ensure 'AUDIT SYSTEM' Is Revoked from Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

- Level 1 - RDBMS using Unified Auditing

Description:

The Oracle database `AUDIT SYSTEM` privilege allows changes to auditing activities on the system. Unauthorized grantees should not have that privilege.

Rationale:

The `AUDIT SYSTEM` privilege can allow the unauthorized alteration of system audit activities, such as disabling the creation of audit trails.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='AUDIT SYSTEM'
AND GRANTEE NOT IN (SELECT USERNAME FROM DBA_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_SYS_PRIVS A
WHERE PRIVILEGE='AUDIT SYSTEM'
AND GRANTEE NOT IN (SELECT USERNAME FROM CDB_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM CDB_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/AUDIT-Traditional-Auditing.html#GUID-ADF45B07-547A-4096-8144-50241FA2D8DD

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.2.8 Ensure 'EXEMPT ACCESS POLICY' Is Revoked from Unauthorized 'GRANTEE' (Automated)

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. Unauthorized grantees should not have that keyword assigned to them.

Rationale:

The `EXEMPT ACCESS POLICY` privilege can allow an unauthorized user to potentially access and change data.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='EXEMPT ACCESS POLICY'
AND GRANTEE NOT IN (SELECT USERNAME FROM DBA_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
              1,(SELECT NAME FROM V$DATABASE),
                (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_SYS_PRIVS A
WHERE PRIVILEGE='EXEMPT ACCESS POLICY'
AND GRANTEE NOT IN (SELECT USERNAME FROM CDB_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM CDB_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/dbseg/configuring-audit-policies.html#GUID-C43651C6-A35C-4EEF-BEA7-EADA408BFF67

CIS Controls:

Version 6

14.4 Protect Information With Access Control Lists
All information stored on systems shall be protected with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.2.9 Ensure 'BECOME USER' Is Revoked from Unauthorized 'GRANTEE' (Automated)

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. Unauthorized grantees should not have that privilege.

Rationale:

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:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='BECOME USER'
AND GRANTEE NOT IN (SELECT USERNAME FROM DBA_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_SYS_PRIVS A
WHERE PRIVILEGE='BECOME USER'
AND GRANTEE NOT IN (SELECT USERNAME FROM CDB_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM CDB_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/dbseg/configuring-audit-policies.html#GUID-89871AE1-F8E3-4C48-B88F-3F2CD37BBF96

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.2.10 Ensure 'CREATE PROCEDURE' Is Revoked from Unauthorized 'GRANTEE' (Automated)

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. Unauthorized grantees should not have that privilege.

Rationale:

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.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='CREATE PROCEDURE'
AND GRANTEE NOT IN (SELECT USERNAME FROM DBA_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_SYS_PRIVS A
WHERE PRIVILEGE='CREATE PROCEDURE'
AND GRANTEE NOT IN (SELECT USERNAME FROM CDB_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM CDB_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/dbseg/configuring-audit-policies.html#GUID-89871AE1-F8E3-4C48-B88F-3F2CD37BBF96

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.2.11 Ensure 'ALTER SYSTEM' Is Revoked from Unauthorized 'GRANTEE' (Automated)

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. Unauthorized grantees should not have that privilege.

Rationale:

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.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='ALTER SYSTEM'
AND GRANTEE NOT IN (SELECT USERNAME FROM DBA_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
              1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_SYS_PRIVS A
WHERE PRIVILEGE='ALTER SYSTEM'
AND GRANTEE NOT IN (SELECT USERNAME FROM CDB_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM CDB_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/dbseg/configuring-audit-policies.html#GUID-89871AE1-F8E3-4C48-B88F-3F2CD37BBF96

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.2.12 Ensure 'CREATE ANY LIBRARY' Is Revoked from Unauthorized 'GRANTEE' (Automated)

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. Unauthorized grantees should not have that privilege.

Rationale:

The `CREATE ANY LIBRARY` privilege can allow the creation of numerous library-associated objects and potentially corrupt the libraries' integrity.

Audit:

Non multi-tenant or pluggable database only:
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 (SELECT USERNAME FROM DBA_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_SYS_PRIVS A
WHERE PRIVILEGE='CREATE ANY LIBRARY'
AND GRANTEE NOT IN (SELECT USERNAME FROM CDB_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM CDB_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/dbseg/configuring-audit-policies.html#GUID-89871AE1-F8E3-4C48-B88F-3F2CD37BBF96

Notes:

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

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.2.13 Ensure 'CREATE LIBRARY' Is Revoked from Unauthorized 'GRANTEE' (Automated)

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. Unauthorized grantees should not have that privilege.

Rationale:

The `CREATE LIBRARY` privilege can allow the creation of numerous library-associated objects and potentially corrupt the libraries' integrity.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE
FROM DBA_SYS_PRIVS
WHERE PRIVILEGE='CREATE LIBRARY'
AND GRANTEE NOT IN (SELECT USERNAME FROM DBA_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_SYS_PRIVS A
WHERE PRIVILEGE='CREATE LIBRARY'
AND GRANTEE NOT IN (SELECT USERNAME FROM CDB_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM CDB_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/dbseg/configuring-audit-policies.html#GUID-89871AE1-F8E3-4C48-B88F-3F2CD37BBF96

Notes:

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

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.2.14 Ensure 'GRANT ANY OBJECT PRIVILEGE' Is Revoked from Unauthorized 'GRANTEE' (Automated)

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. Unauthorized grantees should not have that keyword assigned to them.

Rationale:

The `GRANT ANY OBJECT PRIVILEGE` capability can allow an unauthorized user to potentially access or change confidential data, or damage the data catalog due to potential complete instance access.

Audit:

Non multi-tenant or pluggable database only:
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 (SELECT USERNAME FROM DBA_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_SYS_PRIVS A
WHERE PRIVILEGE='GRANT ANY OBJECT PRIVILEGE'
AND GRANTEE NOT IN (SELECT USERNAME FROM CDB_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM CDB_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/dbseg/configuring-privilege-and-role-authorization.html#GUID-0A8EAA3F-CD48-4F60-97ED-556CEF7DA021

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.2.15 Ensure 'GRANT ANY ROLE' Is Revoked from Unauthorized 'GRANTEE' (Automated)

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. Unauthorized grantees should not have that keyword assigned to them.

Rationale:

The `GRANT ANY ROLE` capability can allow an unauthorized user to potentially access or change confidential data or damage the data catalog due to potential complete instance access.

Audit:

Non multi-tenant or pluggable database only:
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 (SELECT USERNAME FROM DBA_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_SYS_PRIVS A
WHERE PRIVILEGE='GRANT ANY ROLE'
AND GRANTEE NOT IN (SELECT USERNAME FROM CDB_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM CDB_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/dbseg/configuring-privilege-and-role-authorization.html#GUID-FEC9C8E4-B935-405B-8994-9AC5A71C6DD7

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.2.16 Ensure 'GRANT ANY PRIVILEGE' Is Revoked from Unauthorized 'GRANTEE' (Automated)

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. Unauthorized grantees should not have that privilege.

Rationale:

The `GRANT ANY PRIVILEGE` capability can allow an unauthorized user to potentially access or change confidential data or damage the data catalog due to potential complete instance access.

Audit:

Non multi-tenant or pluggable database only:
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 (SELECT USERNAME FROM DBA_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, PRIVILEGE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_SYS_PRIVS A
WHERE PRIVILEGE='GRANT ANY PRIVILEGE'
AND GRANTEE NOT IN (SELECT USERNAME FROM CDB_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM CDB_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/dbseg/configuring-privilege-and-role-authorization.html#GUID-6429A893-1D8F-49E8-9022-61B5D30B885E

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
   Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
   Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.3 Excessive Role Privileges

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

5.3.1 Ensure 'SELECT_CATALOG_ROLE' Is Revoked from Unauthorized 'GRANTEE' (Automated)

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. Unauthorized grantees should not have that role.

Rationale:

Permitting unauthorized access to the `SELECT_CATALOG_ROLE` can allow the disclosure of all dictionary data.

Audit:

Non multi-tenant or pluggable database only:
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 (SELECT USERNAME FROM DBA_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, GRANTED_ROLE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_ROLE_PRIVS A
WHERE GRANTED_ROLE='SELECT_CATALOG_ROLE'
```

```
AND GRANTEE NOT IN (SELECT USERNAME FROM CDB_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM CDB_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/dbseg/configuring-privilege-and-role-authorization.html#GUID-F4BE7D94-B60E-42CE-9AFA-2D0C0443C31F

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
    Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
    Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.3.2 Ensure 'EXECUTE_CATALOG_ROLE' Is Revoked from Unauthorized 'GRANTEE' (Automated)

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. Unauthorized grantees should not have that role.

Rationale:

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:

Non multi-tenant or pluggable database only: 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 (SELECT USERNAME FROM DBA_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM DBA_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Multi-tenant in the container database: This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT GRANTEE, GRANTED_ROLE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_ROLE_PRIVS A
WHERE GRANTED_ROLE='EXECUTE_CATALOG_ROLE'
AND GRANTEE NOT IN (SELECT USERNAME FROM CDB_USERS WHERE
ORACLE_MAINTAINED='Y')
AND GRANTEE NOT IN (SELECT ROLE FROM CDB_ROLES WHERE ORACLE_MAINTAINED='Y');
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is granted in both container and pluggable database, you must connect to both places to revoke.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/dbseg/configuring-privilege-and-role-authorization.html#GUID-F4BE7D94-B60E-42CE-9AFA-2D0C0443C31F

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

5.3.3 Ensure 'DBA' Is Revoked from Unauthorized 'GRANTEE' (Automated)

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. Unauthorized grantees should not have that role.

Rationale:

Assignment of the DBA role to an ordinary user can provide a great number of unnecessary privileges to that user and open the door to data breaches, integrity violations, and denial-of-service conditions.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

```
SELECT 'GRANT' AS PATH, GRANTEE, GRANTED_ROLE
FROM DBA_ROLE_PRIVS
WHERE GRANTED_ROLE = 'DBA' AND GRANTEE NOT IN ('SYS', 'SYSTEM')
UNION
SELECT 'PROXY', PROXY || '-' || CLIENT, 'DBA'
FROM DBA_PROXIES
WHERE CLIENT IN (SELECT GRANTEE
                 FROM DBA_ROLE_PRIVS
                 WHERE GRANTED_ROLE = 'DBA');
```

Multi-tenant in the container database: This query will also give you the name of the CDB/PDB that has the issue. To assess this recommendation, execute the following SQL statement.

```
SELECT 'GRANT' AS PATH, GRANTEE, GRANTED_ROLE,
DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID)) CON
FROM CDB_ROLE_PRIVS A
WHERE GRANTED_ROLE='DBA'
AND GRANTEE NOT IN ('SYS', 'SYSTEM')
UNION
```

```
SELECT 'PROXY', PROXY || '-' || CLIENT, 'DBA',
       DECODE (A.CON_ID,0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID)) CON
FROM CDB_PROXIES A
WHERE CLIENT IN (SELECT GRANTEE
                 FROM CDB_ROLE_PRIVS B
                 WHERE GRANTED_ROLE = 'DBA'
                 AND A.CON_ID = B.CON_ID);
```

Lack of results implies compliance.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this is
granted in both container and pluggable database, you must connect to both places to
revoke.

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

References:

1. https://docs.oracle.com/en/database/oracle/oracle-
   database/18/dbseg/configuring-privilege-and-role-authorization.html#GUID-
   A5B26A03-32CF-4F5D-A6BE-F2452AD8CB8A

CIS Controls:

Version 6

5.1 Minimize And Sparingly Use Administrative Privileges
   Minimize administrative privileges and only use administrative accounts when they are
required. Implement focused auditing on the use of administrative privileged functions and
monitor for anomalous behavior.

Version 7

14.6 Protect Information through Access Control Lists
   Protect all information stored on systems with file system, network share, claims,
application, or database specific access control lists. These controls will enforce the
principle that only authorized individuals should have access to the information based on
their need to access the information as a part of their responsibilities.

6 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 the 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 are 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, statement 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 also 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).

6.1 Traditional Auditing

The recommendations in this section should be followed if traditional auditing is implemented.

6.1.1 Ensure the 'USER' Audit Option Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The `USER` object allows for creating accounts that can interact with the database according to the roles and privileges allotted to the account. It may also own database objects. Enabling the audit option causes auditing of 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. Auditing can also be useful in forensics if an account is compromised, and auditing 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:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has auditing turned on. To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION,SUCCESS,FAILURE,
       DECODE (A.CON_ID,
               0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_STMT_AUDIT_OPTS A
WHERE USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS'
AND AUDIT_OPTION='USER';
```

Lack of results implies a finding.

Remediation:

To remediate this setting, execute the following SQL statement in either the non multi-tenant or container database, it does NOT need run in the pluggable.

```
AUDIT USER;
```

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.1.2 Ensure the 'ROLE' Audit Option Is Enabled (Automated)

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. Enabling the audit option causes auditing of 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:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has auditing turned on. To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION,SUCCESS,FAILURE,
       DECODE (A.CON_ID,
                0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_STMT_AUDIT_OPTS A
WHERE USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
```

```
AND FAILURE = 'BY ACCESS'
AND AUDIT_OPTION='ROLE';
```

Lack of results implies a finding.

Remediation:

To remediate this setting, execute the following SQL statement in either the non multi-tenant or container database, it does NOT need run in the pluggable.

```
AUDIT ROLE;
```

Notes:

This option does not audit role grants and revokes.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.1.3 Ensure the 'SYSTEM GRANT' Audit Option Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

Enabling the audit option for the SYSTEM GRANT object causes auditing of 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:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has auditing turned on. To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION,SUCCESS,FAILURE,
       DECODE (A.CON_ID,
               0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_STMT_AUDIT_OPTS A
WHERE USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS'
AND AUDIT_OPTION='SYSTEM GRANT';
```

Lack of results implies a finding.

Remediation:

To remediate this setting, execute the following SQL statement in either the non multi-tenant or container database, it does NOT need run in the pluggable.

```
AUDIT SYSTEM GRANT;
```

CIS Controls:

Version 6

5.4 Log Administrative User Addition And Removal
Configure systems to issue a log entry and alert when an account is added to or removed from a domain administrators' group, or when a new local administrator account is added on a system.

Version 7

4.8 Log and Alert on Changes to Administrative Group Membership
Configure systems to issue a log entry and alert when an account is added to or removed from any group assigned administrative privileges.

6.1.4 Ensure the 'PROFILE' Audit Option Is Enabled (Automated)

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. Enabling the audit option causes auditing of all attempts, successful or not, to create, drop or alter any profile.

Rationale:

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

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has auditing turned on. To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION,SUCCESS,FAILURE,
       DECODE (A.CON_ID,
               0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_STMT_AUDIT_OPTS A
WHERE USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS'
AND AUDIT_OPTION='PROFILE';
```

Lack of results implies a finding.

Remediation:

To remediate this setting, execute the following SQL statement in either the non multi-tenant or container database, it does NOT need run in the pluggable.

```
AUDIT PROFILE;
```

Notes:

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.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.1.5 Ensure the 'DATABASE LINK' Audit Option Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

Enabling the audit option for the DATABASE LINK object causes all activities on database links to 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 enabled.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has auditing turned on. To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION,SUCCESS,FAILURE,
       DECODE (A.CON_ID,
               0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_STMT_AUDIT_OPTS A
WHERE USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS'
AND AUDIT_OPTION='DATABASE LINK';
```

Lack of results implies a finding.

Remediation:

To remediate this setting, execute the following SQL statement in either the non multitenant or container database, it does NOT need run in the pluggable.

```
AUDIT DATABASE LINK;
```

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/dbseg/configuring-audit-policies.html#GUID-89871AE1-F8E3-4C48-B88F-3F2CD37BBF96

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.1.6 Ensure the 'PUBLIC DATABASE LINK' Audit Option Is Enabled (Automated)

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. Enabling the audit option causes all user activities involving the creation, alteration, or dropping of public links to be audited.

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 enabled.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has auditing turned on. To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION,SUCCESS,FAILURE,
       DECODE (A.CON_ID,
               0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_STMT_AUDIT_OPTS A
WHERE USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
```

```
AND FAILURE = 'BY ACCESS'
AND AUDIT_OPTION='PUBLIC DATABASE LINK';
```

Lack of results implies a finding.

Remediation:

To remediate this setting, execute the following SQL statement in either the non multi-tenant or container database, it does NOT need run in the pluggable.

```
AUDIT PUBLIC DATABASE LINK;
```

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.1.7 Ensure the 'PUBLIC SYNONYM' Audit Option Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The `PUBLIC SYNONYM` object allows for the creation of an alternate description of an object. Public synonyms are accessible by all users that have the appropriate privileges to the underlying object. Enabling the audit option causes all user activities involving the creation or dropping of public synonyms to be audited.

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 enabled.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has auditing turned on. To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION,SUCCESS,FAILURE,
       DECODE (A.CON_ID,
               0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_STMT_AUDIT_OPTS A
WHERE USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
```

```
AND FAILURE = 'BY ACCESS'
AND AUDIT_OPTION='PUBLIC SYNONYM';
```

Lack of results implies a finding.

Remediation:

To remediate this setting, execute the following SQL statement in either the non multi-tenant or container database, it does NOT need run in the pluggable.

```
AUDIT PUBLIC SYNONYM;
```

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.1.8 Ensure the 'SYNONYM' Audit Option Is Enabled (Automated)

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, or even another synonym. This synonym puts a dependency on its target and is rendered invalid if the target object is changed/dropped. Enabling the audit option causes all user activities involving the creation or dropping of synonyms to be audited.

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 enabled.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has auditing turned on. To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION,SUCCESS,FAILURE,
       DECODE (A.CON_ID,
               0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_STMT_AUDIT_OPTS A
WHERE USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
```

```
AND FAILURE = 'BY ACCESS'
AND AUDIT_OPTION='SYNONYM';
```

Lack of results implies a finding.

Remediation:

To remediate this setting, execute the following SQL statement in either the non multi-tenant or container database, it does NOT need run in the pluggable.

```
AUDIT SYNONYM;
```

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/dbseg/configuring-audit-policies.html#GUID-89871AE1-F8E3-4C48-B88F-3F2CD37BBF96

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.1.9 Ensure the 'DIRECTORY' Audit Option Is Enabled (Automated)

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. Enabling this audit option causes all user activities involving the creation or dropping of a directory alias to be audited.

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 enabled.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has auditing turned on. To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION,SUCCESS,FAILURE,
       DECODE (A.CON_ID,
               0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_STMT_AUDIT_OPTS A
WHERE USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS'
AND AUDIT_OPTION='DIRECTORY';
```

Lack of results implies a finding.

Remediation:

To remediate this setting, execute the following SQL statement in either the non multi-tenant or container database, it does NOT need run in the pluggable.

```
AUDIT DIRECTORY;
```

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/dbseg/configuring-audit-policies.html#GUID-C0070008-D2BB-425A-9DC3-153FB1575445

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.1.10 Ensure the 'SELECT ANY DICTIONARY' Audit Option Is Enabled (Automated)

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. Enabling the audit option causes all user activities involving this capability to be audited.

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 enabled.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has auditing turned on. To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION,SUCCESS,FAILURE,
       DECODE (A.CON_ID,
               0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_STMT_AUDIT_OPTS A
WHERE USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS'
AND AUDIT_OPTION='SELECT ANY DICTIONARY';
```

Lack of results implies a finding.

Remediation:

To remediate this setting, execute the following SQL statement in either the non multi-tenant or container database, it does NOT need run in the pluggable.

```
AUDIT SELECT ANY DICTIONARY;
```

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/18/dbseg/configuring-audit-policies.html#GUID-C0070008-D2BB-425A-9DC3-153FB1575445

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
   Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
   Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.1.11 Ensure the 'GRANT ANY OBJECT PRIVILEGE' Audit Option Is Enabled (Automated)

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. Enabling this audit option causes auditing of 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:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has auditing turned on. To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION,SUCCESS,FAILURE,
       DECODE (A.CON_ID,
               0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_STMT_AUDIT_OPTS A
WHERE USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS'
AND AUDIT_OPTION='GRANT ANY OBJECT PRIVILEGE';
```

Lack of results implies a finding.

Remediation:

To remediate this setting, execute the following SQL statement, keeping in mind if this needs to be done in both container and pluggable database, you must connect to both places to do the audit statement.

```
AUDIT GRANT ANY OBJECT PRIVILEGE;
```

Notes:

This does NOT audit all attempts to grant or revoke object privileges since this can also be done by anyone who was granted an object privilege with the grant option. Also, this never creates an audit record for anyone who does not hold the `GRANT ANY OBJECT PRIVILEGE` system privilege. Therefore, many attempts, successful or not, to grant and revoke object privileges are not audited by this.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.1.12 Ensure the 'GRANT ANY PRIVILEGE' Audit Option Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

`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.

Rationale:

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:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has auditing turned on. To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION,SUCCESS,FAILURE,
       DECODE (A.CON_ID,
               0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_STMT_AUDIT_OPTS A
WHERE USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS'
AND AUDIT_OPTION='GRANT ANY PRIVILEGE';
```

Lack of results implies a finding.

Remediation:

To remediate this setting, execute the following SQL statement in either the non multi-tenant or container database, it does NOT need run in the pluggable.

```
AUDIT GRANT ANY PRIVILEGE;
```

Notes:

This does NOT audit all attempts to grant or revoke system privileges since this can also be done by anyone who was granted a system privilege with the admin option. Also, this never creates an audit record for anyone who does not hold the `GRANT ANY PRIVILEGE` system privilege. Thus, many attempts, successful or not, to grant and revoke system privileges are not audited by this.

CIS Controls:

Version 6

5.4 Log Administrative User Addition And Removal
Configure systems to issue a log entry and alert when an account is added to or removed from a domain administrators' group, or when a new local administrator account is added on a system.

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

4.8 Log and Alert on Changes to Administrative Group Membership
Configure systems to issue a log entry and alert when an account is added to or removed from any group assigned administrative privileges.

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 <u>Enable Detailed Logging</u>

Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.1.13 Ensure the 'DROP ANY PROCEDURE' Audit Option Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

The `AUDIT DROP ANY PROCEDURE` command is auditing the dropping of procedures. Enabling the option causes auditing of all such activities.

Rationale:

Dropping procedures of another user could be part of a privilege escalation exploit and should be audited.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has auditing turned on. To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION,SUCCESS,FAILURE,
        DECODE (A.CON_ID,
                0,(SELECT NAME FROM V$DATABASE),
                1,(SELECT NAME FROM V$DATABASE),
                  (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_STMT_AUDIT_OPTS A
WHERE USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS'
AND AUDIT_OPTION='DROP ANY PROCEDURE';
```

Lack of results implies a finding.

Remediation:

To remediate this setting, execute the following SQL statement in either the non multi-tenant or container database, it does NOT need run in the pluggable.

```
AUDIT DROP ANY PROCEDURE;
```

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.1.14 Ensure the 'ALL' Audit Option on 'SYS.AUD$' Is Enabled (Automated)

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. Enabling the audit option will cause these activities to be audited.

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 enabled.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT *
FROM CDB_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:

To remediate this setting, execute the following SQL statement in either the non multi-tenant or container database, it does NOT need run in the pluggable.

```
AUDIT ALL ON SYS.AUD$ BY ACCESS;
```

CIS Controls:

Version 6

6.2 <u>Ensure Audit Log Settings Support Appropriate Log Entry Formatting</u>
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 <u>Activate audit logging</u>
Ensure that local logging has been enabled on all systems and networking devices.

6.3 <u>Enable Detailed Logging</u>
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.1.15 Ensure the 'PROCEDURE' Audit Option Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

In this statement audit, `PROCEDURE` means any procedure, function, package or library. Enabling this audit option causes any attempt, successful or not, to create or drop any of these types of objects to be audited, regardless of privilege or lack thereof. Java schema objects (sources, classes, and resources) are considered the same as procedures for the 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 stored code can dramatically change the behavior of the application and produce serious security consequences, including enabling privilege escalation and introducing SQL injection vulnerabilities. Audit records of such changes can be helpful in forensics.

Audit:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has auditing turned on. To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION,SUCCESS,FAILURE,
       DECODE (A.CON_ID,
               0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_STMT_AUDIT_OPTS A
WHERE USER_NAME IS NULL
AND PROXY_NAME IS NULL
```

```
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS'
AND AUDIT_OPTION='PROCEDURE';
```

Lack of results implies a finding.

Remediation:

To remediate this setting, execute the following SQL statement in either the non multi-tenant or container database, it does NOT need run in the pluggable.

```
AUDIT PROCEDURE;
```

Notes:

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`, or `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`, `PROFILE`, etc.)

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event

Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.1.16 Ensure the 'ALTER SYSTEM' Audit Option Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

`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. Enabling the audit option will audit all attempts to perform `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:

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:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has auditing turned on. To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION,SUCCESS,FAILURE,
       DECODE (A.CON_ID,
               0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_STMT_AUDIT_OPTS A
WHERE USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
```

```
AND FAILURE = 'BY ACCESS'
AND AUDIT_OPTION='ALTER SYSTEM';
```

Lack of results implies a finding.

Remediation:

To remediate this setting, execute the following SQL statement in either the non multi-tenant or container database, it does NOT need run in the pluggable.

```
AUDIT ALTER SYSTEM;
```

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.1.17 Ensure the 'TRIGGER' Audit Option Is Enabled (Automated)

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. Enabling this audit option will cause auditing of 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:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has auditing turned on. To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION,SUCCESS,FAILURE,
       DECODE (A.CON_ID,
               0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
                 (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_STMT_AUDIT_OPTS A
WHERE USER_NAME IS NULL
AND PROXY_NAME IS NULL
```

```
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS'
AND AUDIT_OPTION='TRIGGER';
```

Lack of results implies a finding.

Remediation:

To remediate this setting, execute the following SQL statement in either the non multi-tenant or container database, it does NOT need run in the pluggable.

```
AUDIT TRIGGER;
```

Notes:

There is no current CIS recommendation to audit the use of the system privilege `CREATE TRIGGER`, as there is for `CREATE SYNONYM`, `CREATE PROCEDURE` and some other types of objects, so this is actually a scope escalation also - to audit such actions in one's own schema. However, this is the only way to comprehensively audit things like attempts to create, drop or alter triggers in another's schema if the user attempting to operation does not hold the required `ANY` privilege - and these are exactly the sorts of things that should raise a large red flag.

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` do, 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.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.1.18 Ensure the 'CREATE SESSION' Audit Option Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Traditional Auditing

Description:

Enabling this audit option will cause auditing of all attempts to connect to the database, whether successful or not, as well as audit session disconnects/logoffs. The commands to audit `SESSION`, `CONNECT` or `CREATE SESSION` all accomplish 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 contain 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:

Non multi-tenant or pluggable database only:
To assess this recommendation, execute the following SQL statement.

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

Multi-tenant in the container database:
This query will also give you the name of the CDB/PDB that has auditing turned on. To assess this recommendation, execute the following SQL statement.

```
SELECT AUDIT_OPTION,SUCCESS,FAILURE,
       DECODE (A.CON_ID,
               0,(SELECT NAME FROM V$DATABASE),
               1,(SELECT NAME FROM V$DATABASE),
```

```
                        (SELECT NAME FROM V$PDBS B WHERE A.CON_ID = B.CON_ID))
FROM CDB_STMT_AUDIT_OPTS A
WHERE USER_NAME IS NULL
AND PROXY_NAME IS NULL
AND SUCCESS = 'BY ACCESS'
AND FAILURE = 'BY ACCESS'
AND AUDIT_OPTION='CREATE SESSION';
```

Lack of results implies a finding.

Remediation:

To remediate this setting, execute the following SQL statement in either the non multi-tenant or container database, it does NOT need run in the pluggable.

```
AUDIT SESSION;
```

Notes:

Although listed in the documentation as a privilege audit, audit `CREATE SESSION` actually audits the `CONNECT` statement. This is evidenced by the undocumented audit `CONNECT` which has the same result 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.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2 Unified Auditing

The recommendations in this section should be followed if unified auditing is implemented.

6.2.1 Ensure the 'CREATE USER' Action Audit Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

The `CREATE USER` statement is used to create Oracle database accounts and assign database properties to them. Enabling this unified action audit causes 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 accounts, 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.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY('CREATE USER' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('CREATE USER' )
        AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
        AND EXISTS (SELECT *
                    FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                    WHERE ENABLED.SUCCESS = 'YES'
                        AND ENABLED.FAILURE = 'YES'
                        AND ENABLED.ENABLED_OPTION = 'BY USER'
                        AND ENABLED.ENTITY_NAME = 'ALL USERS'
                        AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
```

```
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL ;
```

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 USER;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.2 Ensure the 'ALTER USER' Action Audit Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

The `ALTER USER` statement is used to change database users' password, lock accounts, and expire passwords. In addition, this statement is used to change database properties of user accounts such as database profiles, default and temporary tablespaces, and 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 accounts, 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.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY('ALTER USER' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('ALTER USER' )
       AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
       AND EXISTS (SELECT *
                   FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                   WHERE ENABLED.SUCCESS = 'YES'
                       AND ENABLED.FAILURE = 'YES'
                       AND ENABLED.ENABLED_OPTION = 'BY USER'
                       AND ENABLED.ENTITY_NAME = 'ALL USERS'
                       AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
```

```
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.3 Ensure the 'DROP USER' Audit Option Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

The DROP USER statement is used to drop Oracle database accounts and schemas associated with them. Enabling this unified action audit 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 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 activities involving DROP USER.

Audit:

To assess this recommendation, execute the following SQL statement.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY('DROP USER' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('DROP USER' )
      AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
      AND EXISTS (SELECT *
                  FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                  WHERE ENABLED.SUCCESS = 'YES'
                      AND ENABLED.FAILURE = 'YES'
                      AND ENABLED.ENABLED_OPTION = 'BY USER'
                      AND ENABLED.ENTITY_NAME = 'ALL USERS'
                      AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.4 Ensure the 'CREATE ROLE' Action Audit Is Enabled (Automated)

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. Enabling 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 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 CREATE ROLE.

Audit:

To assess this recommendation, execute the following SQL statement.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY('CREATE ROLE' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('CREATE ROLE' )
       AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
       AND EXISTS (SELECT *
                    FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                    WHERE ENABLED.SUCCESS = 'YES'
                          AND ENABLED.FAILURE = 'YES'
                          AND ENABLED.ENABLED_OPTION = 'BY USER'
                          AND ENABLED.ENTITY_NAME = 'ALL USERS'
                          AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.5 Ensure the 'ALTER ROLE' Action Audit Is Enabled (Automated)

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. The `ALTER ROLE` statement is used to change the authorization needed to enable a role. Enabling this unified action audit causes 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 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 roles.

Audit:

To assess this recommendation, execute the following SQL statement.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY('ALTER ROLE' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('ALTER ROLE' )
      AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
      AND EXISTS (SELECT *
                    FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                    WHERE ENABLED.SUCCESS = 'YES'
                        AND ENABLED.FAILURE = 'YES'
                        AND ENABLED.ENABLED_OPTION = 'BY USER'
                        AND ENABLED.ENTITY_NAME = 'ALL USERS'
                        AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
```

```
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.6 Ensure the 'DROP ROLE' Action Audit Is Enabled (Automated)

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. Enabling 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 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 `DROP ROLE`.

Audit:

To assess this recommendation, execute the following SQL statement.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY('DROP ROLE' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('DROP ROLE' )
        AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
        AND EXISTS (SELECT *
                    FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                    WHERE ENABLED.SUCCESS = 'YES'
                        AND ENABLED.FAILURE = 'YES'
                        AND ENABLED.ENABLED_OPTION = 'BY USER'
                        AND ENABLED.ENTITY_NAME = 'ALL USERS'
                        AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.7 Ensure the 'GRANT' Action Audit Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

GRANT 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. Enabling this unified action audit 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 the 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.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY('GRANT' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('GRANT' )
      AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
      AND EXISTS (SELECT *
                  FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                  WHERE ENABLED.SUCCESS = 'YES'
                      AND ENABLED.FAILURE = 'YES'
                      AND ENABLED.ENABLED_OPTION = 'BY USER'
                      AND ENABLED.ENTITY_NAME = 'ALL USERS'
                      AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
```

```
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.8 Ensure the 'REVOKE' Action Audit Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

REVOKE statements are used to revoke privileges from Oracle database users and roles. Enabling this unified action audit 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.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY('REVOKE' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('REVOKE' )
       AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
       AND EXISTS (SELECT *
                   FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                   WHERE ENABLED.SUCCESS = 'YES'
                       AND ENABLED.FAILURE = 'YES'
                       AND ENABLED.ENABLED_OPTION = 'BY USER'
                       AND ENABLED.ENTITY_NAME = 'ALL USERS'
                       AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.9 Ensure the 'CREATE PROFILE' Action Audit Is Enabled (Automated)

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. Enabling this unified action audit 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.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY('CREATE PROFILE' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('CREATE PROFILE')
       AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
       AND EXISTS (SELECT *
                    FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                    WHERE ENABLED.SUCCESS = 'YES'
                         AND ENABLED.FAILURE = 'YES'
                         AND ENABLED.ENABLED_OPTION = 'BY USER'
                         AND ENABLED.ENTITY_NAME = 'ALL USERS'
                         AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 PROFILE;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.10 Ensure the 'ALTER PROFILE' Action Audit Is Enabled (Automated)

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. Enabling this unified action audit 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.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY('ALTER PROFILE' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('ALTER PROFILE' )
       AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
       AND EXISTS (SELECT *
                      FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                      WHERE ENABLED.SUCCESS = 'YES'
                            AND ENABLED.FAILURE = 'YES'
                            AND ENABLED.ENABLED_OPTION = 'BY USER'
                            AND ENABLED.ENTITY_NAME = 'ALL USERS'
                            AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.11 Ensure the 'DROP PROFILE' Action Audit Is Enabled (Automated)

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. Enabling this unified action audit 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 database profiles.

Audit:

To assess this recommendation, execute the following SQL statement.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY('DROP PROFILE' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('DROP PROFILE' )
       AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
       AND EXISTS (SELECT *
                     FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                     WHERE ENABLED.SUCCESS = 'YES'
                          AND ENABLED.FAILURE = 'YES'
                          AND ENABLED.ENABLED_OPTION = 'BY USER'
                          AND ENABLED.ENTITY_NAME = 'ALL USERS'
                          AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.12 Ensure the 'CREATE DATABASE LINK' Action Audit Is Enabled (Automated)

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. Enabling this unified action audit causes logging of all `CREATE DATABASE` and `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.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY('CREATE DATABASE LINK' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('CREATE DATABASE LINK' )
        AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
        AND EXISTS (SELECT *
                    FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                    WHERE ENABLED.SUCCESS = 'YES'
                          AND ENABLED.FAILURE = 'YES'
                          AND ENABLED.ENABLED_OPTION = 'BY USER'
                          AND ENABLED.ENTITY_NAME = 'ALL USERS'
                          AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
```

```
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.13 Ensure the 'ALTER DATABASE LINK' Action Audit Is Enabled (Automated)

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. Enabling this unified action audit causes logging of all `ALTER DATABASE` and `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.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY('ALTER DATABASE LINK' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('ALTER DATABASE LINK' )
       AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
       AND EXISTS (SELECT *
                   FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                   WHERE ENABLED.SUCCESS = 'YES'
                         AND ENABLED.FAILURE = 'YES'
                         AND ENABLED.ENABLED_OPTION = 'BY USER'
                         AND ENABLED.ENTITY_NAME = 'ALL USERS'
                         AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
```

```
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.14 Ensure the 'DROP DATABASE LINK' Action Audit Is Enabled (Automated)

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. Enabling this unified action audit causes logging of all `DROP DATABASE` and `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 database links.

Audit:

To assess this recommendation, execute the following SQL statement.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY('DROP DATABASE LINK' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('DROP DATABASE LINK' )
       AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
       AND EXISTS (SELECT *
                    FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                    WHERE ENABLED.SUCCESS = 'YES'
                         AND ENABLED.FAILURE = 'YES'
                         AND ENABLED.ENABLED_OPTION = 'BY USER'
                         AND ENABLED.ENTITY_NAME = 'ALL USERS'
                         AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
```

```
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.15 Ensure the 'CREATE SYNONYM' Action Audit Is Enabled (Automated)

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. Enabling this unified action audit causes logging of all `CREATE SYNONYM` and `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.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY( 'CREATE SYNONYM' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('CREATE SYNONYM' )
       AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
       AND EXISTS (SELECT *
                    FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                   WHERE ENABLED.SUCCESS = 'YES'
                        AND ENABLED.FAILURE = 'YES'
                        AND ENABLED.ENABLED_OPTION = 'BY USER'
                        AND ENABLED.ENTITY_NAME = 'ALL USERS'
                        AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
```

```
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.16 Ensure the 'ALTER SYNONYM' Action Audit Is Enabled (Automated)

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, or java object, or even another synonym. Enabling this unified action audit causes logging of all `ALTER SYNONYM` and `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.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY('ALTER SYNONYM' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('ALTER SYNONYM' )
        AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
        AND EXISTS (SELECT *
                    FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                    WHERE ENABLED.SUCCESS = 'YES'
                        AND ENABLED.FAILURE = 'YES'
                        AND ENABLED.ENABLED_OPTION = 'BY USER'
                        AND ENABLED.ENTITY_NAME = 'ALL USERS'
                        AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
```

```
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.17 Ensure the 'DROP SYNONYM' Action Audit Is Enabled (Automated)

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, or java object, or even another synonym. Enabling his unified action audit causes logging of all `DROP SYNONYM` and `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.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY( 'DROP SYNONYM' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('DROP SYNONYM' )
       AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
       AND EXISTS (SELECT *
                    FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                    WHERE ENABLED.SUCCESS = 'YES'
                          AND ENABLED.FAILURE = 'YES'
                          AND ENABLED.ENABLED_OPTION = 'BY USER'
                          AND ENABLED.ENTITY_NAME = 'ALL USERS'
                          AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.18 Ensure the 'SELECT ANY DICTIONARY' Privilege Audit Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

The `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$` and `OBJ$`. This privilege also allows grantees to create stored objects such as procedures, packages and views on the underlying data dictionary objects. Please note that this privilege does not grant `SELECT` on tables with password hashes such as `USER$`, `DEFAULT_PWD$`, `LINK$`, and `USER_HISTORY$`. Enabling this audit causes logging of activities that exercise this privilege.

Rationale:

Logging and monitoring of all attempts to access a 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.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY( 'SELECT ANY DICTIONARY' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('SELECT ANY DICTIONARY' )
        AND AUD.AUDIT_OPTION_TYPE = 'SYSTEM PRIVILEGE'
        AND EXISTS (SELECT *
                    FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                    WHERE ENABLED.SUCCESS = 'YES'
                          AND ENABLED.FAILURE = 'YES'
                          AND ENABLED.ENABLED_OPTION = 'BY USER'
                          AND ENABLED.ENTITY_NAME = 'ALL USERS'
```

```
                        AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL ;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
   Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
   Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.19 Ensure the 'AUDSYS.AUD$UNIFIED' Access Audit Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

The `AUDSYS.AUD$UNIFIED` holds audit trail records generated by the database. Enabling this audit action causes logging of all access attempts to the `AUDSYS.AUD$UNIFIED`, 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 `AUDSYS.AUD$UNIFIED`, 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 table.

Audit:

To assess this recommendation, execute the following SQL statement.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY( 'AUD$UNIFIED' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT OBJECT_NAME
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('ALL' )
      AND AUD.AUDIT_OPTION_TYPE = 'OBJECT ACTION'
      AND AUD.OBJECT_SCHEMA = 'AUDSYS'
      AND AUD.OBJECT_NAME = 'AUD$UNIFIED'
      AND EXISTS (SELECT *
                    FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                    WHERE ENABLED.SUCCESS = 'YES'
                        AND ENABLED.FAILURE = 'YES'
                        AND ENABLED.ENABLED_OPTION = 'BY USER'
                        AND ENABLED.ENTITY_NAME = 'ALL USERS'
                        AND ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
```

```
LEFT JOIN AUDIT_ENABLED E
ON C.AUDIT_OPTION = E.OBJECT_NAME
WHERE E.OBJECT_NAME   IS NULL;
```

Lack of results implies compliance.

Remediation:

For Oracle 12.2 and above, execute the following SQL statement to remediate this setting.

```
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY
ADD
ACTIONS
ALL on AUDSYS.AUD$UNIFIED;
```

Note: If you do not have `CIS_UNIFIED_AUDIT_POLICY`, please create one using the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.20 Ensure the 'CREATE PROCEDURE/FUNCTION/PACKAGE/PACKAGE BODY' Action Audit Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

Oracle database procedures, function, packages, and package bodies, 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. Enabling this unified action audit causes logging of all `CREATE PROCEDURE`, `CREATE FUNCTION`, `CREATE PACKAGE` and `CREATE PACKAGE BODY` statements, 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.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY(
  'CREATE PROCEDURE','CREATE FUNCTION','CREATE PACKAGE','CREATE PACKAGE BODY'
) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('CREATE PROCEDURE','CREATE FUNCTION','CREATE
PACKAGE','CREATE PACKAGE BODY' )
        AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
        AND EXISTS (SELECT *
                    FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                    WHERE ENABLED.SUCCESS = 'YES'
                        AND ENABLED.FAILURE = 'YES'
                        AND ENABLED.ENABLED_OPTION = 'BY USER'
```

```
                            AND ENABLED.ENTITY_NAME = 'ALL USERS'
                            AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.21 Ensure the 'ALTER PROCEDURE/FUNCTION/PACKAGE/PACKAGE BODY' Action Audit Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

Oracle database procedures, functions, packages, and package bodies, 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. Enabling this unified action audit causes logging of all `ALTER PROCEDURE`, `ALTER FUNCTION`, `ALTER PACKAGE` and `ALTER PACKAGE BODY` statements, successful or unsuccessful, 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.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY(
  'ALTER PROCEDURE','ALTER FUNCTION','ALTER PACKAGE','ALTER PACKAGE BODY' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('ALTER PROCEDURE','ALTER FUNCTION','ALTER
PACKAGE','ALTER PACKAGE BODY' )
        AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
        AND EXISTS (SELECT *
                    FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                    WHERE ENABLED.SUCCESS = 'YES'
```

```
                    AND ENABLED.FAILURE = 'YES'
                    AND ENABLED.ENABLED_OPTION = 'BY USER'
                    AND ENABLED.ENTITY_NAME = 'ALL USERS'
                    AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
    Validate audit log settings for each hardware device and the software installed on it,
ensuring that logs include a date, timestamp, source addresses, destination addresses, and
various other useful elements of each packet and/or transaction. Systems should record
logs in a standardized format such as syslog entries or those outlined by the Common Event
Expression initiative. If systems cannot generate logs in a standardized format, log
normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
    Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
    Enable system logging to include detailed information such as an event source, date,
user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.22 Ensure the 'DROP PROCEDURE/FUNCTION/PACKAGE/PACKAGE BODY' Action Audit Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

Oracle database procedures, functions, packages, and package bodies, 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. Enabling this unified action audit causes logging of all `DROP PROCEDURE`, `DROP FUNCTION`, `DROP PACKAGE` or `DROP PACKAGE BODY` 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, 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 procedures, functions, packages or package bodies.

Audit:

To assess this recommendation, execute the following SQL statement.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY(
  'DROP PROCEDURE','DROP FUNCTION','DROP PACKAGE','DROP PACKAGE BODY' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('DROP PROCEDURE','DROP FUNCTION','DROP
PACKAGE','DROP PACKAGE BODY' )
      AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
      AND EXISTS (SELECT *
                  FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                  WHERE ENABLED.SUCCESS = 'YES'
                      AND ENABLED.FAILURE = 'YES'
                      AND ENABLED.ENABLED_OPTION = 'BY USER'
                      AND ENABLED.ENTITY_NAME = 'ALL USERS'
```

```
                          AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.23 Ensure the 'ALTER SYSTEM' Privilege Audit Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

The `ALTER SYSTEM` privilege allows the user to change instance settings which could impact security posture, performance or normal operation of the database. Additionally, the `ALTER SYSTEM` privilege may be used to run operating system commands using undocumented Oracle functionality. Enabling this unified audit causes 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.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY( 'ALTER SYSTEM' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('ALTER SYSTEM' )
      AND AUD.AUDIT_OPTION_TYPE = 'SYSTEM PRIVILEGE'
      AND EXISTS (SELECT *
                   FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                   WHERE ENABLED.SUCCESS = 'YES'
                       AND ENABLED.FAILURE = 'YES'
                       AND ENABLED.ENABLED_OPTION = 'BY USER'
                       AND ENABLED.ENTITY_NAME = 'ALL USERS'
                       AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
```

```
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.24 Ensure the 'CREATE TRIGGER' Action Audit Is Enabled (Automated)

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. Enabling this unified audit causes 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.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY( 'CREATE TRIGGER' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('CREATE TRIGGER' )
      AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
      AND EXISTS (SELECT *
                   FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                  WHERE ENABLED.SUCCESS = 'YES'
                       AND ENABLED.FAILURE = 'YES'
                       AND ENABLED.ENABLED_OPTION = 'BY USER'
                       AND ENABLED.ENTITY_NAME = 'ALL USERS'
                       AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
```

```
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.25 Ensure the 'ALTER TRIGGER' Action Audit IS Enabled (Automated)

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. Enabling this unified audit causes 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.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY( 'ALTER TRIGGER' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('ALTER TRIGGER' )
        AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
        AND EXISTS (SELECT *
                    FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                    WHERE ENABLED.SUCCESS = 'YES'
                          AND ENABLED.FAILURE = 'YES'
                          AND ENABLED.ENABLED_OPTION = 'BY USER'
                          AND ENABLED.ENTITY_NAME = 'ALL USERS'
                          AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
```

```
LEFT JOIN AUDIT_ENABLED E
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.26 Ensure the 'DROP TRIGGER' Action Audit Is Enabled (Automated)

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. Enabling this unified audit causes 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 triggers.

Audit:

To assess this recommendation, execute the following SQL statement.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY( 'DROP TRIGGER' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('DROP TRIGGER' )
       AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
       AND EXISTS (SELECT *
                    FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                    WHERE ENABLED.SUCCESS = 'YES'
                        AND ENABLED.FAILURE = 'YES'
                        AND ENABLED.ENABLED_OPTION = 'BY USER'
                        AND ENABLED.ENTITY_NAME = 'ALL USERS'
                        AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
```

```
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

6.2.27 Ensure the 'LOGON' AND 'LOGOFF' Actions Audit Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS using Unified Auditing

Description:

Oracle database users log on to the database to perform their work. Enabling this unified audit causes 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.

```
WITH
CIS_AUDIT(AUDIT_OPTION) AS
(
SELECT * FROM TABLE( DBMSOUTPUT_LINESARRAY( 'LOGON','LOGOFF' ) )
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUDIT_OPTION
  FROM AUDIT_UNIFIED_POLICIES AUD
  WHERE AUD.AUDIT_OPTION IN ('LOGON','LOGOFF' )
       AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
       AND EXISTS (SELECT *
                   FROM AUDIT_UNIFIED_ENABLED_POLICIES ENABLED
                   WHERE ENABLED.SUCCESS = 'YES'
                       AND ENABLED.FAILURE = 'YES'
                       AND ENABLED.ENABLED_OPTION = 'BY USER'
                       AND ENABLED.ENTITY_NAME = 'ALL USERS'
                       AND  ENABLED.POLICY_NAME = AUD.POLICY_NAME)
)
SELECT C.AUDIT_OPTION
FROM CIS_AUDIT C
LEFT JOIN AUDIT_ENABLED E
```

```
ON C.AUDIT_OPTION = E.AUDIT_OPTION
WHERE E.AUDIT_OPTION IS NULL;
```

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 the `CREATE AUDIT POLICY` statement.

CIS Controls:

Version 6

6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting
Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

7 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 grant only the specific permissions required to perform the assessments expressed herein. Doing this avoids the necessity for any user assessing the system to be granted `DBA` privileges.

NON-MULTI-TENANT Create this in non-container or Pluggable Database (if you do not have access to the container database).

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_$PARAMETER TO CISSCANROLE ;
GRANT SELECT ON DBA_TAB_PRIVS TO CISSCANROLE ;
GRANT SELECT ON DBA_TABLES TO CISSCANROLE;
GRANT SELECT ON DBA_PROFILES TO CISSCANROLE ;
GRANT SELECT ON DBA_SYS_PRIVS TO CISSCANROLE ;
GRANT SELECT ON DBA_STMT_AUDIT_OPTS TO CISSCANROLE ;
GRANT SELECT ON DBA_ROLE_PRIVS TO CISSCANROLE ;
GRANT SELECT ON DBA_OBJ_AUDIT_OPTS TO CISSCANROLE ;
GRANT SELECT ON DBA_PRIV_AUDIT_OPTS TO CISSCANROLE ;
GRANT SELECT ON DBA_PROXIES TO CISSCANROLE ;
GRANT SELECT ON DBA_USERS TO CISSCANROLE ;
GRANT SELECT ON DBA_USERS_WITH_DEFPWD TO CISSCANROLE ;
GRANT SELECT ON DBA_DB_LINKS TO CISSCANROLE ;
GRANT SELECT ON DBA_ROLES TO CISSCANROLE;
GRANT SELECT ON V_$INSTANCE to CISSCANROLE ;
GRANT SELECT ON V_$DATABASE to CISSCANROLE ;
GRANT SELECT ON V_$PDBS to CISSCANROLE ;
GRANT SELECT ON V_$SYSTEM_PARAMETER 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;
CREATE VIEW SYS.X_$KSPPI AS SELECT * FROM SYS.X$KSPPI;
CREATE VIEW SYS.X_$KSPPCV AS SELECT * FROM SYS.X$KSPPCV;
GRANT SELECT ON SYS.X_$KSPPI to CISSCANROLE;
GRANT SELECT ON SYS.X_$KSPPCV to CISSCANROLE;
```

MULTI-TENANT - Create this in the Container Database.

The recommendations expressed in this document assume the presence of a common role named `C##CISSCANROLE` and a common user named `C##CISSCAN`. This common role and common user should be created by executing the following SQL statements, being careful to substitute an appropriate password for *`<password>`*.

```
-- Create the role

CREATE ROLE C##CISSCANROLE CONTAINER=ALL;

-- Grant necessary privileges to the role

GRANT CREATE SESSION TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON V_$PARAMETER TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_TAB_PRIVS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_TABLES TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_PROFILES TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_SYS_PRIVS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_STMT_AUDIT_OPTS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_ROLE_PRIVS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_OBJ_AUDIT_OPTS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_PRIV_AUDIT_OPTS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_PROXIES TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_USERS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_ROLES TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_USERS_WITH_DEFPWD TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_DB_LINKS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON V_$INSTANCE to C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON V_$DATABASE to C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON V_$PDBS to C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON V_$SYSTEM_PARAMETER TO C##CISSCANROLE CONTAINER=ALL;
GRANT AUDIT_VIEWER TO C##CISSCANROLE CONTAINER=ALL;

-- Create the user and assign the user to the role

CREATE USER C##CISSCAN IDENTIFIED BY <password> CONTAINER=ALL;
GRANT C##CISSCANROLE TO C##CISSCAN CONTAINER=ALL;
ALTER USER C##CISSCAN SET CONTAINER_DATA=ALL CONTAINER=CURRENT;

-- THE NEXT FOUR GRANTS MUST BE PERFORMED BY SYS:

CREATE VIEW SYS.X_$KSPPI AS SELECT * FROM SYS.X$KSPPI;
CREATE VIEW SYS.X_$KSPPCV AS SELECT * FROM SYS.X$KSPPCV;
GRANT SELECT ON SYS.X_$KSPPI to C##CISSCANROLE;
GRANT SELECT ON SYS.X_$KSPPCV to C##CISSCANROLE;
```

If you rely on similar roles and/or users, but they are not named `C##CISSCANROLE` or `C##CISSCAN`, or if you have roles or users named `C##CISSCANROLE` or `C##CISSCAN` intended to be used for different purposes, be aware that some recommendations herein explicitly name `C##CISSCANROLE` and `C##CISSCAN`.

These are:

- 4.4 Ensure No Users Are Assigned the `DEFAULT` Profile

- 5.1.3.2 Ensure 'ALL' Is Revoked from Unauthorized `GRANTEE` on `DBA_%`

Note: 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.

8 Appendix: Establishing a Unified Audit Policy

This document has been authored with the expectation that an audit policy has been created for Unified Auditing. If you do not already have a Unified Audit Policy, you can use this as an example for your database.

NON-MULTI-TENANT Create this in non-container or Pluggable Database (if you do not have access to the container database).

```
-- Create policy and add actions
CREATE AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY PRIVILEGES CREATE USER;
```

MULTI-TENANT Create this in the container database.

```
CREATE AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY PRIVILEGES CREATE USER
CONTAINER=ALL;
```

All of the rest of the commands can be run in either container or pluggable databases.

```
AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS ALTER USER;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS DROP USER;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS CREATE ROLE;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS ALTER ROLE;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS DROP ROLE;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS GRANT;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS REVOKE;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS CREATE PROFILE;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS ALTER PROFILE;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS DROP PROFILE;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS CREATE DATABASE LINK;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS ALTER DATABASE LINK;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS DROP DATABASE LINK;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS CREATE SYNONYM;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS ALTER SYNONYM;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS DROP SYNONYM;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD PRIVILEGES SELECT ANY
DICTIONARY;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS ALL ON
AUDSYS.AUD$UNIFIED;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS CREATE PROCEDURE,
CREATE FUNCTION, CREATE PACKAGE, CREATE PACKAGE BODY;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS ALTER PROCEDURE,
ALTER FUNCTION, ALTER PACKAGE, ALTER PACKAGE BODY;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS DROP PROCEDURE, DROP
FUNCTION, DROP PACKAGE, DROP PACKAGE BODY;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS ALTER SYSTEM;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS CREATE TRIGGER;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS ALTER TRIGGER;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS DROP TRIGGER;
ALTER AUDIT POLICY CIS_UNIFIED_AUDIT_POLICY ADD ACTIONS LOGON, LOGOFF;
```

Note: Different organizations may wish to follow the instructions in this appendix in different ways. Such a decision is ultimately left up to the implementing organization for example, you may wish to have different unified auditing policies.

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 (Unscored)	☐	☐
2	**Oracle Parameter Settings**		
2.1	**Listener Settings**		
2.1.1	Ensure 'SECURE_CONTROL_<listener_name>' Is Set In 'listener.ora' (Automated)	☐	☐
2.1.2	Ensure 'extproc' Is Not Present in 'listener.ora' (Automated)	☐	☐
2.1.3	Ensure 'ADMIN_RESTRICTIONS_<listener_name>' Is Set to 'ON' (Automated)	☐	☐
2.1.4	Ensure 'SECURE_REGISTER_<listener_name>' Is Set to 'TCPS' or 'IPC' (Automated)	☐	☐
2.2	**Database Settings**		
2.2.1	Ensure 'AUDIT_SYS_OPERATIONS' Is Set to 'TRUE' (Automated)	☐	☐
2.2.2	Ensure 'AUDIT_TRAIL' Is Set to 'DB', 'XML', 'OS', 'DB,EXTENDED', or 'XML,EXTENDED' (Automated)	☐	☐
2.2.3	Ensure 'GLOBAL_NAMES' Is Set to 'TRUE' (Automated)	☐	☐
2.2.4	Ensure 'O7_DICTIONARY_ACCESSIBILITY' Is Set to 'FALSE' (Automated)	☐	☐
2.2.5	Ensure 'OS_ROLES' Is Set to 'FALSE' (Automated)	☐	☐
2.2.6	Ensure 'REMOTE_LISTENER' Is Empty (Automated)	☐	☐
2.2.7	Ensure 'REMOTE_LOGIN_PASSWORDFILE' Is Set to 'NONE' (Automated)	☐	☐
2.2.8	Ensure 'REMOTE_OS_AUTHENT' Is Set to 'FALSE' (Automated)	☐	☐
2.2.9	Ensure 'REMOTE_OS_ROLES' Is Set to 'FALSE' (Automated)	☐	☐
2.2.10	Ensure 'SEC_CASE_SENSITIVE_LOGON' Is Set to 'TRUE' (Automated)	☐	☐
2.2.11	Ensure 'SEC_MAX_FAILED_LOGIN_ATTEMPTS' Is '3' or Less (Automated)	☐	☐
2.2.12	Ensure 'SEC_PROTOCOL_ERROR_FURTHER_ACTION' Is Set to '(DROP,3)' (Automated)	☐	☐
2.2.13	Ensure 'SEC_PROTOCOL_ERROR_TRACE_ACTION' Is Set to 'LOG' (Automated)	☐	☐
2.2.14	Ensure 'SEC_RETURN_SERVER_RELEASE_BANNER' Is Set to 'FALSE' (Automated)	☐	☐

2.2.15	Ensure 'SQL92_SECURITY' Is Set to 'TRUE' (Automated)	☐	☐
2.2.16	Ensure '_trace_files_public' Is Set to 'FALSE' (Automated)	☐	☐
2.2.17	Ensure 'RESOURCE_LIMIT' Is Set to 'TRUE' (Automated)	☐	☐
3	**Oracle Connection and Login Restrictions**		
3.1	Ensure 'FAILED_LOGIN_ATTEMPTS' Is Less than or Equal to '5' (Automated)	☐	☐
3.2	Ensure 'PASSWORD_LOCK_TIME' Is Greater than or Equal to '1' (Automated)	☐	☐
3.3	Ensure 'PASSWORD_LIFE_TIME' Is Less than or Equal to '90' (Automated)	☐	☐
3.4	Ensure 'PASSWORD_REUSE_MAX' Is Greater than or Equal to '20' (Automated)	☐	☐
3.5	Ensure 'PASSWORD_REUSE_TIME' Is Greater than or Equal to '365' (Automated)	☐	☐
3.6	Ensure 'PASSWORD_GRACE_TIME' Is Less than or Equal to '5' (Automated)	☐	☐
3.7	Ensure 'PASSWORD_VERIFY_FUNCTION' Is Set for All Profiles (Automated)	☐	☐
3.8	Ensure 'SESSIONS_PER_USER' Is Less than or Equal to '10' (Automated)	☐	☐
3.9	Ensure 'INACTIVE_ACCOUNT_TIME' Is Less than or Equal to '120' (Automated)	☐	☐
4	**Users**		
4.1	Ensure All Default Passwords Are Changed (Automated)	☐	☐
4.2	Ensure All Sample Data And Users Have Been Removed (Automated)	☐	☐
4.3	Ensure 'DBA_USERS.AUTHENTICATION_TYPE' Is Not Set to 'EXTERNAL' for Any User (Automated)	☐	☐
4.4	Ensure No Users Are Assigned the 'DEFAULT' Profile (Automated)	☐	☐
4.5	Ensure 'SYS.USER$MIG' Has Been Dropped (Automated)	☐	☐
4.6	Ensure No Public Database Links Exist (Automated)	☐	☐
5	**Privileges & Grants & ACLs**		
5.1	**Excessive Table, View and Package Privileges**		
5.1.1	**Public Privileges**		
5.1.1.1	Ensure 'EXECUTE' is revoked from 'PUBLIC' on "Network" Packages (Automated)	☐	☐
5.1.1.2	Ensure 'EXECUTE' is revoked from 'PUBLIC' on "File System" Packages (Automated)	☐	☐
5.1.1.3	Ensure 'EXECUTE' is revoked from 'PUBLIC' on "Encryption" Packages (Automated)	☐	☐
5.1.1.4	Ensure 'EXECUTE' is revoked from 'PUBLIC' on "Java" Packages (Automated)	☐	☐

5.1.1.5	Ensure 'EXECUTE' is revoked from 'PUBLIC' on "Job Scheduler" Packages (Automated)	☐	☐
5.1.1.6	Ensure 'EXECUTE' is revoked from 'PUBLIC' on "SQL Injection Helper" Packages (Automated)	☐	☐
5.1.2	**Non-Default Privileges**		
5.1.2.1	Ensure 'EXECUTE' is not granted to 'PUBLIC' on "Non-default" Packages (Automated)	☐	☐
5.1.3	**Other Privileges**		
5.1.3.1	Ensure 'ALL' Is Revoked from Unauthorized 'GRANTEE' on 'AUD$' (Automated)	☐	☐
5.1.3.2	Ensure 'ALL' Is Revoked from Unauthorized 'GRANTEE' on 'DBA_%' (Automated)	☐	☐
5.1.3.3	Ensure 'ALL' Is Revoked on 'Sensitive' Tables (Automated)	☐	☐
5.2	**Excessive System Privileges**		
5.2.1	Ensure '%ANY%' Is Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐
5.2.2	Ensure 'DBA_SYS_PRIVS.%' Is Revoked from Unauthorized 'GRANTEE' with 'ADMIN_OPTION' Set to 'YES' (Automated)	☐	☐
5.2.3	Ensure 'EXECUTE ANY PROCEDURE' Is Revoked from 'OUTLN' (Automated)	☐	☐
5.2.4	Ensure 'EXECUTE ANY PROCEDURE' Is Revoked from 'DBSNMP' (Automated)	☐	☐
5.2.5	Ensure 'SELECT ANY DICTIONARY' Is Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐
5.2.6	Ensure 'SELECT ANY TABLE' Is Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐
5.2.7	Ensure 'AUDIT SYSTEM' Is Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐
5.2.8	Ensure 'EXEMPT ACCESS POLICY' Is Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐
5.2.9	Ensure 'BECOME USER' Is Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐
5.2.10	Ensure 'CREATE PROCEDURE' Is Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐
5.2.11	Ensure 'ALTER SYSTEM' Is Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐
5.2.12	Ensure 'CREATE ANY LIBRARY' Is Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐
5.2.13	Ensure 'CREATE LIBRARY' Is Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐
5.2.14	Ensure 'GRANT ANY OBJECT PRIVILEGE' Is Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐
5.2.15	Ensure 'GRANT ANY ROLE' Is Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐

5.2.16	Ensure 'GRANT ANY PRIVILEGE' Is Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐
5.3	**Excessive Role Privileges**		
5.3.1	Ensure 'SELECT_CATALOG_ROLE' Is Revoked from Unauthorized 'GRANTEE' **(Automated)**	☐	☐
5.3.2	Ensure 'EXECUTE_CATALOG_ROLE' Is Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐
5.3.3	Ensure 'DBA' Is Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐
6	**Audit/Logging Policies and Procedures**		
6.1	**Traditional Auditing**		
6.1.1	Ensure the 'USER' Audit Option Is Enabled (Automated)	☐	☐
6.1.2	Ensure the 'ROLE' Audit Option Is Enabled (Automated)	☐	☐
6.1.3	Ensure the 'SYSTEM GRANT' Audit Option Is Enabled (Automated)	☐	☐
6.1.4	Ensure the 'PROFILE' Audit Option Is Enabled (Automated)	☐	☐
6.1.5	Ensure the 'DATABASE LINK' Audit Option Is Enabled (Automated)	☐	☐
6.1.6	Ensure the 'PUBLIC DATABASE LINK' Audit Option Is Enabled (Automated)	☐	☐
6.1.7	Ensure the 'PUBLIC SYNONYM' Audit Option Is Enabled (Automated)	☐	☐
6.1.8	Ensure the 'SYNONYM' Audit Option Is Enabled (Automated)	☐	☐
6.1.9	Ensure the 'DIRECTORY' Audit Option Is Enabled (Automated)	☐	☐
6.1.10	Ensure the 'SELECT ANY DICTIONARY' Audit Option Is Enabled (Automated)	☐	☐
6.1.11	Ensure the 'GRANT ANY OBJECT PRIVILEGE' Audit Option Is Enabled (Automated)	☐	☐
6.1.12	Ensure the 'GRANT ANY PRIVILEGE' Audit Option Is Enabled (Automated)	☐	☐
6.1.13	Ensure the 'DROP ANY PROCEDURE' Audit Option Is Enabled (Automated)	☐	☐
6.1.14	Ensure the 'ALL' Audit Option on 'SYS.AUD$' Is Enabled (Automated)	☐	☐
6.1.15	Ensure the 'PROCEDURE' Audit Option Is Enabled (Automated)	☐	☐
6.1.16	Ensure the 'ALTER SYSTEM' Audit Option Is Enabled (Automated)	☐	☐
6.1.17	Ensure the 'TRIGGER' Audit Option Is Enabled (Automated)	☐	☐
6.1.18	Ensure the 'CREATE SESSION' Audit Option Is Enabled (Automated)	☐	☐
6.2	**Unified Auditing**		

6.2.1	Ensure the 'CREATE USER' Action Audit Is Enabled (Automated)	☐	☐
6.2.2	Ensure the 'ALTER USER' Action Audit Is Enabled (Automated)	☐	☐
6.2.3	Ensure the 'DROP USER' Audit Option Is Enabled (Automated)	☐	☐
6.2.4	Ensure the 'CREATE ROLE' Action Audit Is Enabled (Automated)	☐	☐
6.2.5	Ensure the 'ALTER ROLE' Action Audit Is Enabled (Automated)	☐	☐
6.2.6	Ensure the 'DROP ROLE' Action Audit Is Enabled (Automated)	☐	☐
6.2.7	Ensure the 'GRANT' Action Audit Is Enabled (Automated)	☐	☐
6.2.8	Ensure the 'REVOKE' Action Audit Is Enabled (Automated)	☐	☐
6.2.9	Ensure the 'CREATE PROFILE' Action Audit Is Enabled (Automated)	☐	☐
6.2.10	Ensure the 'ALTER PROFILE' Action Audit Is Enabled (Automated)	☐	☐
6.2.11	Ensure the 'DROP PROFILE' Action Audit Is Enabled (Automated)	☐	☐
6.2.12	Ensure the 'CREATE DATABASE LINK' Action Audit Is Enabled (Automated)	☐	☐
6.2.13	Ensure the 'ALTER DATABASE LINK' Action Audit Is Enabled (Automated)	☐	☐
6.2.14	Ensure the 'DROP DATABASE LINK' Action Audit Is Enabled (Automated)	☐	☐
6.2.15	Ensure the 'CREATE SYNONYM' Action Audit Is Enabled (Automated)	☐	☐
6.2.16	Ensure the 'ALTER SYNONYM' Action Audit Is Enabled (Automated)	☐	☐
6.2.17	Ensure the 'DROP SYNONYM' Action Audit Is Enabled (Automated)	☐	☐
6.2.18	Ensure the 'SELECT ANY DICTIONARY' Privilege Audit Is Enabled (Automated)	☐	☐
6.2.19	Ensure the 'AUDSYS.AUD$UNIFIED' Access Audit Is Enabled (Automated)	☐	☐
6.2.20	Ensure the 'CREATE PROCEDURE/FUNCTION/PACKAGE/PACKAGE BODY' Action Audit Is Enabled (Automated)	☐	☐
6.2.21	Ensure the 'ALTER PROCEDURE/FUNCTION/PACKAGE/PACKAGE BODY' Action Audit Is Enabled (Automated)	☐	☐
6.2.22	Ensure the 'DROP PROCEDURE/FUNCTION/PACKAGE/PACKAGE BODY' Action Audit Is Enabled (Automated)	☐	☐

6.2.23	Ensure the 'ALTER SYSTEM' Privilege Audit Is Enabled (Automated)	☐	☐
6.2.24	Ensure the 'CREATE TRIGGER' Action Audit Is Enabled (Automated)	☐	☐
6.2.25	Ensure the 'ALTER TRIGGER' Action Audit IS Enabled (Automated)	☐	☐
6.2.26	Ensure the 'DROP TRIGGER' Action Audit Is Enabled (Automated)	☐	☐
6.2.27	Ensure the 'LOGON' AND 'LOGOFF' Actions Audit Is Enabled (Automated)	☐	☐
7	**Appendix: Establishing an Audit/Scan User**		
8	**Appendix: Establishing a Unified Audit Policy**		

Appendix: Change History

Date	Version	Changes for this version
Jul 16, 2020	1.0.0	Initial Release