

CIS Oracle Database 26ai Benchmark

v1.0.0 - 03-27-2026

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/>

For information on referencing and/or citing CIS Benchmarks in 3rd party documentation (including using portions of Benchmark Recommendations) please contact CIS Legal (legalnotices@cisecurity.org) and request guidance on copyright usage.

NOTE: It is **NEVER** acceptable to host a CIS Benchmark in **ANY** format (PDF, etc.) on a 3rd party (non-CIS owned) site.

Table of Contents

Terms of Use	1
Table of Contents	2
Overview	7
Important Usage Information	7
Key Stakeholders.....	7
Apply the Correct Version of a Benchmark	8
Exceptions	8
Remediation	9
Summary.....	9
Target Technology Details	10
Intended Audience	10
Consensus Guidance.....	11
Typographical Conventions	12
Recommendation Definitions	13
Title.....	13
Assessment Status	13
Automated	13
Manual.....	13
Profile.....	13
Description	13
Rationale Statement.....	13
Impact Statement.....	14
Audit Procedure.....	14
Remediation Procedure	14
Default Value	14
References	14
CIS Critical Security Controls® (CIS Controls®).....	14
Additional Information	14
Profile Definitions	15
Acknowledgements.....	16
Recommendations	17
1 Oracle Database Installation and Patching Requirements	17
1.1 Ensure That Appropriate Version/Patches For Oracle Software Are Installed (Manual)....	18
2 Oracle Parameter Settings	20
2.1 Listener Settings.....	20
2.1.1 Ensure 'extproc' Is Not Present In 'listener.ora' (Automated).....	21

2.1.2 Ensure 'ACCEPT_MD5_CERTS' Is Configured Correctly (Automated)	23
2.1.3 Ensure 'ACCEPT_SHA1_CERTS' Is Configured Correctly (Automated)	25
2.1.4 Ensure 'ALLOWED_WEAK_CERT_ALGORITHMS' Is NOT Set. (Automated)	27
2.2 SQLNET.ORA Settings	29
2.2.1 Ensure 'ACCEPT_MD5_CERTS' Is NOT SET (Automated)	29
2.2.2 Ensure 'ACCEPT_SHA1_CERTS' Is NOT Set (Automated)	31
2.2.3 Ensure 'ALLOWED_WEAK_CERT_ALGORITHMS' Is NOT Set (Automated)	33
2.2.4 Ensure 'SQLNET.ALLOWED_LOGON_VERSION_CLIENT' Is Set To 12a (Automated)	36
2.2.5 Ensure 'SQLNET.ALLOWED_LOGON_VERSION_SERVER' Is Set To 12a (Automated)	38
2.2.6 Ensure 'SQLNET.ENCRYPTION_CLIENT' Is Set To 'REQUIRED' (Automated)	40
2.2.7 Ensure 'SQLNET.ENCRYPTION_SERVER' Is Set To 'REQUIRED' (Automated)	43
2.2.8 Ensure 'SQLNET.ENCRYPTION_TYPES_CLIENT' Is Set To 'AES256' (Automated)	46
2.2.9 Ensure 'SQLNET.ENCRYPTION_TYPES_SERVER' Is Set To AES256 (Manual)	49
2.2.10 Ensure 'SQLNET.CRYPTO_CHECKSUM_CLIENT' Is Set To 'REQUIRED' (Automated)	52
2.2.11 Ensure 'SQLNET.CRYPTO_CHECKSUM_SERVER' Is Set To 'REQUIRED' (Automated)	55
2.2.12 Ensure 'SSL_CERT_REVOCATION' Is Set To 'REQUIRED' (Automated)	58
2.3 Database Settings	60
2.3.1 Ensure 'BACKGROUND_CORE_DUMP' Is Not Set To 'Full' (Automated)	61
2.3.2 Ensure 'SHADOW_CORE_DUMP' Is Not Set To 'Full' (Automated)	63
2.3.3 Ensure 'MLE_PROG_LANGUAGES' Is Set To 'OFF' (Automated)	65
2.3.4 Ensure 'ALLOW_GROUP_ACCESS_TO_SGA' Is Set To 'FALSE' (Automated)	67
2.3.5 Review Undocumented (Underscore) Parameters Not Set To 'DEFAULT' Values (Manual)	69
2.3.6 Ensure 'OS_ROLES' Is Set To 'FALSE' (Automated)	71
2.3.7 Ensure 'REMOTE_OS_ROLES' Is Set To 'FALSE' (Automated)	73
2.3.8 Ensure 'SEC_MAX_FAILED_LOGIN_ATTEMPTS' Is Set To '3' Or Less (Automated)	75
2.3.9 Ensure 'SEC_PROTOCOL_ERROR_FURTHER_ACTION' Is Set To '(DROP,3)' (Automated)	77
2.3.10 Ensure 'SEC_PROTOCOL_ERROR_TRACE_ACTION' Is Set To 'LOG' (Automated)	78
2.3.11 Ensure 'SEC_RETURN_SERVER_RELEASE_BANNER' Is Set To 'FALSE' (Automated)	80
2.3.12 Ensure 'REMOTE_LOGIN_PASSWORDFILE' Is Set To 'NONE' (Automated)	82
2.3.13 Ensure 'REMOTE_LISTENER' Is Empty (Automated)	84
2.3.14 Ensure 'RESOURCE_LIMIT' Is Set To 'TRUE' (Automated)	86
3 Oracle Connection and Login Restrictions	88
3.1 Ensure 'FAILED_LOGIN_ATTEMPTS' Is Less Than Or Equal To '5' (Automated)	89
3.2 Ensure 'PASSWORD_LOCK_TIME' Is Greater Than Or Equal To '1' (Automated)	91
3.3 Ensure 'PASSWORD_LIFE_TIME + PASSWORD_GRACE_TIME' Is Less Than Or Equal To '365' (Automated)	93
3.4 Ensure 'PASSWORD_REUSE_MAX' Is Set To 'UNLIMITED' (Automated)	95
3.5 Ensure 'PASSWORD_VERIFY_FUNCTION' Is Set For All Profiles (Automated)	97
3.6 Ensure 'PASSWORD_VERIFY_FUNCTION' Is Configured Correctly (Manual)	99
3.7 Ensure 'PASSWORD_ROLLOVER_TIME' Is set to '0' (Automated)	101
3.8 Ensure 'INACTIVE_ACCOUNT_TIME' Is Less than or Equal to '120' (Automated)	103
4 Users	105
4.1 Ensure All Default Passwords Are Changed (Automated)	106
4.2 Ensure No Custom 'ORACLE_MAINTAINED' Users Exist (Automated)	108
4.3 Review The Users Created Through Real Application Security (Manual)	110
4.4 Ensure Old Password Versions Are Not Used (Automated)	112
4.5 Ensure The Latest Version of The Password File Is Used (Automated)	114
4.6 Ensure That Users In Different RAC Instances Are Identical In PW Files (Automated)	116

4.7 Ensure No Public Database Links Exist (Automated)	117
4.8 Ensure That Database Link Passwords Are Using The Latest Encryption (Automated) ..	119
5 Unified Auditing	121
5.1 Ensure All Auditable System Actions Commands Are Audited (Automated)	123
5.2 Ensure the 'LOGON' AND 'LOGOFF' Actions Audit Is Enabled (Automated)	126
5.3 Ensure Critical Packages Are Audited (Automated)	128
5.4 Ensure All Export Activities Are Audited (Automated)	131
5.5 Ensure The Use Of SYS* Privileges Is Audited (Automated)	134
6 Privileges & Grants & ACLs	137
6.1 Excessive System Privileges	138
6.1.1 Ensure '%ANY%' Is Revoked From Unauthorized 'GRANTEE' (Automated)	139
6.1.2 Ensure Admin Privileges Are Revoked From Unauthorized 'GRANTEE' (Automated) ..	141
6.1.3 Ensure 'IMPORT' And 'EXPORT' 'FULL DATABASE' Is Revoked From Unauthorized 'GRANTEE' (Automated)	143
6.1.4 Ensure 'CREATE EXTERNAL JOB' Is Revoked From Unauthorized 'GRANTEE' (Automated)	145
6.1.5 Ensure 'BECOME USER' Is Revoked From Unauthorized 'GRANTEE' (Automated) ...	147
6.1.6 Ensure 'TEXT DATASTORE ACCESS' Is Revoked From Unauthorized 'GRANTEE' (Automated)	149
6.1.7 Ensure 'CREATE', 'ALTER', And 'DROP' 'PUBLIC DATABASE LINK' Is Revoked From Unauthorized 'GRANTEE' (Automated)	151
6.1.8 Ensure 'LOGMINING' Is Revoked From Unauthorized 'GRANTEE' (Automated)	154
6.1.9 Ensure 'ALTER SYSTEM' Is Revoked From Unauthorized 'GRANTEE' (Automated) ..	156
6.1.10 Ensure 'CREATE LIBRARY' Is Revoked From Unauthorized 'GRANTEE' (Automated)	158
6.1.11 Ensure All 'SYSTEM' Privileges Are Revoked from Unauthorized 'GRANTEE' (Automated)	160
6.2 Excessive Role Privileges	162
6.2.1 Ensure 'DBA' Is Revoked from Unauthorized 'GRANTEE' (Automated)	163
6.2.2 Ensure 'EXP_FULL_DATABASE' Is Revoked From Unauthorized 'GRANTEE' (Automated)	165
6.2.3 Ensure 'IMP_FULL_DATABASE' Is Revoked From Unauthorized 'GRANTEE' (Automated)	167
6.2.4 Ensure 'DATAPUMP_EXP_FULL_DATABASE' Is Revoked From Unauthorized 'GRANTEE' (Automated)	169
6.2.5 Ensure 'DATAPUMP_IMP_FULL_DATABASE' is Revoked From Unauthorized 'GRANTEE' (Automated)	171
6.2.6 Ensure 'DV_ADMIN' Is Revoked From Unauthorized 'GRANTEE' (Automated)	173
6.2.7 Ensure 'DV_AUDIT_CLEANUP' Is Revoked From Unauthorized 'GRANTEE' (Automated)	175
6.2.8 Ensure 'OLAP_DBA' Is Revoked From Unauthorized 'GRANTEE' (Automated)	177
6.2.9 Ensure 'LBAC_DBA' Is Revoked From Unauthorized 'GRANTEE' (Automated)	179
6.2.10 Ensure 'JAVA_ADMIN' Is Revoked From Unauthorized 'GRANTEE' (Automated)	181
6.2.11 Ensure 'JAVASYSPRIVS' Is Revoked From Unauthorized 'GRANTEE' (Automated) ..	183
6.2.12 Ensure 'LOGSTDBY_ADMINISTRATOR' Is Revoked From Unauthorized 'GRANTEE' (Automated)	185
6.2.13 Ensure 'SQL_FIREWALL_ADMIN' Is Revoked From Unauthorized 'GRANTEE' (Automated)	187
6.2.14 Ensure 'MAINTPLAN_APP' Is Revoked From Unauthorized 'GRANTEE' (Automated)	189
6.2.15 Ensure 'JAVADEBUGPRIV' Is Revoked From Unauthorized 'GRANTEE' (Automated)	191
6.2.16 Ensure 'DV_PATCH_ADMIN' Is Revoked From Unauthorized 'GRANTEE' (Automated)	193

6.2.17 Ensure 'DV_POLICY_OWNER' Is Revoked From Unauthorized 'GRANTEE' (Automated)	195
6.2.18 Ensure AUDIT_ADMIN' Is Revoked From Unauthorized 'GRANTEE' (Automated)	197
6.2.19 Ensure 'AUDIT_VIEWER' Is Revoked From Unauthorized 'GRANTEE' (Automated)	199
6.2.20 Ensure 'PDB_DBA' Is Revoked From Unauthorized 'GRANTEE' (Automated)	201
6.2.21 Ensure 'SELECT_CATALOG_ROLE' Is Revoked From Unauthorized 'GRANTEE' (Automated)	203
6.2.22 Ensure 'EXECUTE_CATALOG_ROLE' Is Revoked from Unauthorized 'GRANTEE' (Automated)	205
6.3 Excessive Schema Privileges	207
6.3.1 Ensure 'CDB_SCHEMA_PRIVS' Does Not Have Unauthorized Privileges (Manual) ...	208
6.4 Excessive Object Privileges	210
6.4.1 Ensure 'ALL' Is Revoked On 'Sensitive' Tables (Automated)	211
6.5 Excessive Column Privileges	213
6.5.1 Ensure 'DBA_COL_PRIVS' Is Revoked from Unauthorized 'GRANTEE' (Manual)	214
6.6 Excessive Proxy Privileges	216
6.6.1 Ensure Proxy User Privileges Are Revoked from Unauthorized 'GRANTEE' (Manual)	217
6.7 Excessive Java Privileges	219
6.7.1 Ensure Custom Java Privileges Are Revoked from Unauthorized 'GRANTEE' (Manual)	220
6.8 Excessive Directory Privileges	222
6.8.1 Ensure Directory Object Access Is Revoked From Unauthorized 'GRANTEE' (Manual)	223
6.8.2 Review Directory Objects Privileges (Manual)	225
6.8.3 Review External Tables With Preprocessor (Manual)	227
6.8.4 Review External Tables (Manual)	229
7 Appendix: Establishing an Audit/Scan User	231
8 Appendix: Establishing a Unified Audit Policy	233
8.1 All DDL Auditable System Actions Policy	233
8.1.1 CDB All DDL Auditable System Actions Policy	233
8.1.2 PDB All DDL Auditable System Actions Policy	234
8.2 Logon/Logoff Audit Policy	235
8.2.1 CDB Logon/Logoff Audit Policy	235
8.2.2 PDB Logon/Logoff Audit Policy	236
8.3 Audit Usage Of Critical Packages	236
8.3.1 CDB Critical Packages Audit Policy	236
8.3.2 PDB Critical Packages Audit Policy	237
8.4 Audit All Export Activities	237
8.4.1 CDB Export Audit Policy	237
8.4.2 PDB Export Audit Policy	238
8.5 Audit 'SYS*-Privileges'	238
8.5.1 CDB 'SYS*-Privileges' Audit Policy	238
8.5.2 PDB 'SYS*-Privileges' Audit Policy	238
9 Appendix: Comprehensive SQL To Identify Direct And Indirect System Privileges Granted To Users And Roles.	240
Appendix: Summary Table	242
Appendix: Change History	251

Overview

All CIS Benchmarks™ (Benchmarks) focus on technical configuration settings used to maintain and/or increase the security of the addressed technology, and they should be used in **conjunction** with other essential cyber hygiene tasks like:

- Monitoring the base operating system and applications for vulnerabilities and quickly updating with the latest security patches.
- End-point protection (Antivirus software, Endpoint Detection and Response (EDR), etc.).
- Logging and monitoring user and system activity.

In the end, the Benchmarks are designed to be a key **component** of a comprehensive cybersecurity program.

Important Usage Information

All Benchmarks are available free for non-commercial use from the [CIS Website](#). They can be used to manually assess and remediate systems and applications. In lieu of manual assessment and remediation, there are several tools available to assist with assessment:

- [CIS Configuration Assessment Tool \(CIS-CAT® Pro Assessor\)](#)
- [CIS Benchmarks™ Certified 3rd Party Tooling](#)

These tools make the hardening process much more scalable for large numbers of systems and applications.

NOTE: Some tooling focuses only on the Benchmark Recommendations that can be fully automated (skipping ones marked **Manual**). It is important that **ALL** Recommendations (**Automated** and **Manual**) be addressed since all are important for properly securing systems and are typically in scope for audits.

Key Stakeholders

Cybersecurity is a collaborative effort, and cross functional cooperation is imperative within an organization to discuss, test, and deploy Benchmarks in an effective and efficient way. The Benchmarks are developed to be best practice configuration guidelines applicable to a wide range of use cases. In some organizations, exceptions to specific Recommendations will be needed, and this team should work to prioritize the problematic Recommendations based on several factors like risk, time, cost, and labor. These exceptions should be properly categorized and documented for auditing purposes.

Apply the Correct Version of a Benchmark

Benchmarks are developed and tested for a specific set of products and versions and applying an incorrect Benchmark to a system can cause the resulting pass/fail score to be incorrect. This is due to the assessment of settings that do not apply to the target systems. To assure the correct Benchmark is being assessed:

- **Deploy the Benchmark applicable to the way settings are managed in the environment:** An example of this is the Microsoft Windows family of Benchmarks, which have separate Benchmarks for Group Policy, Intune, and Stand-alone systems based upon how system management is deployed. Applying the wrong Benchmark in this case will give invalid results.
- **Use the most recent version of a Benchmark:** This is true for all Benchmarks, but especially true for cloud technologies. Cloud technologies change frequently and using an older version of a Benchmark may have invalid methods for auditing and remediation.

Exceptions

The guidance items in the Benchmarks are called recommendations and not requirements, and exceptions to some of them are expected and acceptable. The Benchmarks strive to be a secure baseline, or starting point, for a specific technology, with known issues identified during Benchmark development are documented in the Impact section of each Recommendation. In addition, organizational, system specific requirements, or local site policy may require changes as well, or an exception to a Recommendation or group of Recommendations (e.g. A Benchmark could Recommend that a Web server not be installed on the system, but if a system's primary purpose is to function as a Webserver, there should be a documented exception to this Recommendation for that specific server).

In the end, exceptions to some Benchmark Recommendations are common and acceptable, and should be handled as follows:

- The reasons for the exception should be reviewed cross-functionally and be well documented for audit purposes.
- A plan should be developed for mitigating, or eliminating, the exception in the future, if applicable.
- If the organization decides to accept the risk of this exception (not work toward mitigation or elimination), this should be documented for audit purposes.

It is the responsibility of the organization to determine their overall security policy, and which settings are applicable to their unique needs based on the overall risk profile for the organization.

Remediation

CIS has developed [Build Kits](#) for many technologies to assist in the automation of hardening systems. Build Kits are designed to correspond to Benchmark's "Remediation" section, which provides the manual remediation steps necessary to make that Recommendation compliant to the Benchmark.

When remediating systems (changing configuration settings on deployed systems as per the Benchmark's Recommendations), please approach this with caution and test thoroughly.

The following is a reasonable remediation approach to follow:

- CIS Build Kits, or internally developed remediation methods should never be applied to production systems without proper testing.
- Proper testing consists of the following:
 - Understand the configuration (including installed applications) of the targeted systems. Various parts of the organization may need different configurations (e.g., software developers vs standard office workers).
 - Read the Impact section of the given Recommendation to help determine if there might be an issue with the targeted systems.
 - Test the configuration changes with representative lab system(s). If issues arise during testing, they can be resolved prior to deploying to any production systems.
 - When testing is complete, initially deploy to a small sub-set of production systems and monitor closely for issues. If there are issues, they can be resolved prior to deploying more broadly.
 - When the initial deployment above is completed successfully, iteratively deploy to additional systems and monitor closely for issues. Repeat this process until the full deployment is complete.

Summary

Using the Benchmarks Certified tools, working as a team with key stakeholders, being selective with exceptions, and being careful with remediation deployment, it is possible to harden large numbers of deployed systems in a cost effective, efficient, and safe manner.

NOTE: As previously stated, the PDF versions of the CIS Benchmarks™ are available for free, non-commercial use on the [CIS Website](#). All other formats of the CIS Benchmarks™ (MS Word, Excel, and [Build Kits](#)) are available for CIS [SecureSuite®](#) members.

CIS-CAT® Pro is also available to CIS [SecureSuite®](#) members.

Target Technology Details

This document is intended to address the recommended security settings for Oracle Database 26ai. This guide was tested against Oracle Database 26ai installed with and without pluggable database support. Future Oracle Database 26ai 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 26ai.

Consensus Guidance

This CIS Benchmark™ was created using a consensus review process comprised of a global community of subject matter experts. The process combines real world experience with data-based information to create technology specific guidance to assist users to secure their environments. 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
<code>Stylized Monospace font</code>	Used for blocks of code, command, and script examples. Text should be interpreted exactly as presented.
<code>Monospace font</code>	Used for inline code, commands, UI/Menu selections or examples. Text should be interpreted exactly as presented.
<Monospace font in brackets>	Text set in angle brackets denote a variable requiring substitution for a real value.
<i>Italic font</i>	Used to reference other relevant settings, CIS Benchmarks and/or Benchmark Communities. Also, used to denote the title of a book, article, or other publication.
Bold font	Additional information or caveats things like Notes , Warnings , or Cautions (usually just the word itself and the rest of the text normal).

Recommendation Definitions

The following defines the various components included in a CIS recommendation as applicable. If any of the components are not applicable it will be noted, or the component will not be included in the recommendation.

Title

Concise description for the recommendation's intended configuration.

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

A collection of recommendations for securing a technology or a supporting platform. Most benchmarks include at least a Level 1 and Level 2 Profile. Level 2 extends Level 1 recommendations and is not a standalone profile. The Profile Definitions section in the benchmark provides the definitions as they pertain to the recommendations included for the technology.

Description

Detailed information pertaining to the setting with which the recommendation is concerned. In some cases, the description will include the recommended value.

Rationale Statement

Detailed reasoning for the recommendation to provide the user a clear and concise understanding on the importance of the recommendation.

Impact Statement

Any security, functionality, or operational consequences that can result from following the recommendation.

Audit Procedure

Systematic instructions for determining if the target system complies with the recommendation.

Remediation Procedure

Systematic instructions for applying recommendations to the target system to bring it into compliance according to the recommendation.

Default Value

Default value for the given setting in this recommendation, if known. If not known, either not configured or not defined will be applied.

References

Additional documentation relative to the recommendation.

CIS Critical Security Controls® (CIS Controls®)

The mapping between a recommendation and the CIS Controls is organized by CIS Controls version, Safeguard, and Implementation Group (IG). The Benchmark in its entirety addresses the CIS Controls safeguards of (v7) "5.1 - Establish Secure Configurations" and (v8) "4.1 - Establish and Maintain a Secure Configuration Process" so individual recommendations will not be mapped to these safeguards.

Additional Information

Supplementary information that does not correspond to any other field but may be useful to the user.

Profile Definitions

The following configuration profiles are defined by this Benchmark:

- **Level 1 - RDBMS**

Items in this profile apply to Oracle Database 26ai 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 On Linux Host OS**

This profile extends the “RDBMS” profile. Items in this profile apply to RDBMS running on a Linux Host operating system with Oracle Database 26ai 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 On Windows Server Host OS**

This profile extends the “RDBMS” profile. Items in this profile apply to RDBMS running on a Windows Server operating system with Oracle Database 26ai and intend to:

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

Acknowledgements

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

Contributor

Emad Al-Mousa

Editor

Nelly Chng

Alexander Kornbrust

Jay Mehta

Tim Harrison, Center for Internet Security, New York

Russ Lowenthal

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 That Appropriate Version/Patches For Oracle Software Are Installed (Manual)

Profile Applicability:

- Level 1 - RDBMS

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_number>\s*.*$"
```

For example, on Windows systems:

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

Remediation:

Download and apply the latest quarterly Critical Patch Update patches.

References:

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

Additional Information:

The following SQL may be used to determine the patches that have been applied:

```
select *  
from cdb_registry_sqlpatch  
order by action_time ;
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>2.2 Ensure Authorized Software is Currently Supported</u> Ensure that only currently supported software is designated as authorized in the software inventory for enterprise assets. If software is unsupported, yet necessary for the fulfillment of the enterprise's mission, document an exception detailing mitigating controls and residual risk acceptance. For any unsupported software without an exception documentation, designate as unauthorized. Review the software list to verify software support at least monthly, or more frequently.			
v7	<u>2.2 Ensure Software is Supported by Vendor</u> 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 'extproc' Is Not Present In 'listener.ora' (Automated)

Profile Applicability:

- Level 1 - RDBMS On Linux Host OS
- Level 1 - RDBMS On Windows Server Host OS

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:

On the Linux platform, execute the following shell command to audit this recommendation.

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

On the Windows platform, execute the following command to audit this recommendation.

```
findstr -i extproc %ORACLE_HOME%\network\admin\listener.ora
```

To be compliant with this recommendation, the output of this command should be **NULL**.

Assumption: The audit command above assumes that **listener.ora** is located in the default path **\$ORACLE_HOME/network/admin/listener.ora**. If this is not the case, please specify the correct location. For example, you may be using the **TNS_ADMIN** environment variable to point to the location of **SQLNET.ORA** and **TNSNAMES.ORA**.

Remediation:

To remediate this recommendation, remove **extproc** from the **listener.ora** file.

Instead of relying on the **EXTPROC** feature in Oracle to access external libraries, you can use a **database directory** to directly access the library files stored within the database server's file system, allowing you to execute code from those libraries within your PL/SQL code without going through a separate external process.

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netag/configuring-and-administering-oracle-net-listener.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.4 <u>Implement and Manage a Firewall on Servers</u> Implement and manage a firewall on servers, where supported. Example implementations include a virtual firewall, operating system firewall, or a third-party firewall agent.			
v7	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.			

2.1.2 Ensure 'ACCEPT_MD5_CERTS' Is Configured Correctly (Automated)

Profile Applicability:

- Level 1 - RDBMS On Linux Host OS
- Level 1 - RDBMS On Windows Server Host OS

Description:

The setting **ACCEPT_MD5_CERTS** specifies whether Oracle accepts certificates signed with the MD5 algorithm.

Note: See Additional Information regarding the deprecation of this setting.

Rationale:

Weak algorithms such as MD5 and SHA1 have known vulnerabilities that make them susceptible to attacks. Allowing their use can compromise data integrity and authentication, potentially exposing systems to risks. Transitioning to stronger algorithms, such as SHA-2, is recommended.

Impact:

Applications that use MD5-signed certificates must be updated to use certificates signed with a stronger, more secure algorithm such as SHA-2.

Audit:

On the Linux platform, execute the following shell command to audit this recommendation.

```
grep -Ei "^[^#]*ACCEPT_MD5_CERTS\s*=\s*TRUE\s*$"
"$ORACLE_HOME/network/admin/listener.ora"
```

On the Windows platform, execute the following PowerShell command to audit this recommendation.

```
Select-String -Path $Env:ORACLE_HOME\network\admin\listener.ora -Pattern
"^\\s*ACCEPT_MD5_CERTS\s*=\s*TRUE\s*$" -CaseSensitive:$false
```

To be compliant with this recommendation, the output of this command should be **NULL**.

The audit command above assumes that **listener.ora** is located in the default path **\$ORACLE_HOME/network/admin/listener.ora**. If this is not the case, please specify the correct location. For example, you may be using the **TNS_ADMIN** environment variable to point to the location of **SQLNET.ORA** and **TNSNAMES.ORA**.

Remediation:

To remediate this recommendation, remove `ACCEPT_MD5_CERTS` from `listener.ora` or set `ACCEPT_MD5_CERTS` to the value `FALSE`.

In addition to `listener.ora`, this parameter must also be set in `sqlnet.ora`.

Default Value:

The default value for `ACCEPT_MD5_CERTS` is `FALSE`.

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netrf/parameters-for-the-sqlnet.ora.html#GUID-84DC4C50-A550-4770-9210-1EECB4B5DB27>
2. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/using-the-orapki-utility-to-manage-pki-elements.html#GUID-93CE62ED-97B1-43B3-8900-64107F8F274C>
3. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netrf/parameters-for-the-sqlnet.ora.html#GUID-9240F605-720C-4C57-93E3-DD26CBEA1410>

Additional Information:

Starting in Oracle Database 26ai, the `ALLOW_MD5_CERTS` and `ALLOW_SHA1_CERTS` `sqlnet.ora` parameters are deprecated.

If `ALLOWED_WEAK_CERT_ALGORITHMS` is set, then Oracle Database ignores `ALLOW_MD5_CERTS` and `ALLOW_SHA1_CERTS`. If

`ALLOWED_WEAK_CERT_ALGORITHMS` is not set, then Oracle Database checks and uses the `ALLOW_MD5_CERTS` and `ALLOW_SHA1_CERTS` settings.[1]

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 <u>Encrypt Sensitive Data in Transit</u> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	14.4 <u>Encrypt All Sensitive Information in Transit</u> Encrypt all sensitive information in transit.			

2.1.3 Ensure 'ACCEPT_SHA1_CERTS' Is Configured Correctly (Automated)

Profile Applicability:

- Level 1 - RDBMS On Linux Host OS
- Level 1 - RDBMS On Windows Server Host OS

Description:

The setting **ACCEPT_SHA1_CERTS** specifies whether Oracle accepts certificates signed with the SHA1 algorithm.

Note: See Additional Information regarding the deprecation of this setting.

Rationale:

Weak algorithms such as MD5 and SHA1 have known vulnerabilities that make them susceptible to attacks. Allowing their use can compromise data integrity and authentication, potentially exposing systems to risks. Transitioning to stronger algorithms, such as SHA-2, is recommended. The SHA1 algorithm has been deprecated by NIST in 2011.

Impact:

Applications that use SHA-1-signed certificates must be updated to use certificates signed with a stronger, more secure algorithm such as SHA-2.

Audit:

On the Linux platform, execute the following shell command to audit this recommendation.

```
grep -Ei "^[^#]*ACCEPT_SHA1_CERTS\s*=\s*TRUE\s*$"
"$ORACLE_HOME/network/admin/listener.ora"
```

On the Windows platform, execute the following command to audit this recommendation.

```
Select-String -Path $Env:ORACLE_HOME\network\admin\listener.ora -Pattern
"^\\s*ACCEPT_SHA1_CERTS\s*=\s*TRUE\s*$" -CaseSensitive:$false
```

To be compliant with this recommendation, the output of this command should be **NULL**.

The audit command above assumes that **listener.ora** is located in the default path **\$ORACLE_HOME/network/admin/listener.ora**. If this is not the case, please specify the correct location. For example, you may be using the **TNS_ADMIN** environment variable to point to the location of **SQLNET.ORA** and **TNSNAMES.ORA**.

Remediation:

To remediate this recommendation, remove **ACCEPT_SHA1_CERTS** from **listener.ora**, or set the **ACCEPT_SHA1_CERTS** to the value **FALSE**.

In addition to **listener.ora**, this parameter must also be set to **FALSE** in **sqlnet.ora**.

Default Value:

The default value for **ACCEPT_SHA1_CERTS** is **FALSE**.

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netrf/parameters-for-the-sqlnet.ora.html#GUID-84DC4C50-A550-4770-9210-1EECB4B5DB27>
2. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseq/using-the-orapki-utility-to-manage-pki-elements.html#GUID-93CE62ED-97B1-43B3-8900-64107F8F274C>
3. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netrf/parameters-for-the-sqlnet.ora.html#GUID-9240F605-720C-4C57-93E3-DD26CBEA1410>

Additional Information:

Starting in Oracle Database 26ai, the **ALLOW_MD5_CERTS** and **ALLOW_SHA1_CERTS** **sqlnet.ora** parameters are deprecated.

If **ALLOWED_WEAK_CERT_ALGORITHMS** is set, then Oracle Database ignores **ALLOW_MD5_CERTS** and **ALLOW_SHA1_CERTS**. If

ALLOWED_WEAK_CERT_ALGORITHMS is not set, then Oracle Database checks and uses the **ALLOW_MD5_CERTS** and **ALLOW_SHA1_CERTS** settings.[1]

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 <u>Encrypt Sensitive Data in Transit</u> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	14.4 <u>Encrypt All Sensitive Information in Transit</u> Encrypt all sensitive information in transit.			

2.1.4 Ensure 'ALLOWED_WEAK_CERT_ALGORITHMS' Is NOT Set. (Automated)

Profile Applicability:

- Level 1 - RDBMS On Linux Host OS
- Level 1 - RDBMS On Windows Server Host OS

Description:

The **ALLOWED_WEAK_CERT_ALGORITHMS** setting determines whether Oracle accepts certificates signed with the SHA1 or MD5 or both algorithms.

Rationale:

Weak algorithms such as MD5 and SHA1 have known vulnerabilities that make them susceptible to attacks. Allowing their use can compromise data integrity and authentication, potentially exposing systems to risks. Transitioning to stronger algorithms, such as SHA-2, is recommended.

Impact:

Applications that use MD5 or SHA-1-signed certificates must be updated to use certificates signed with a stronger, more secure algorithm such as SHA-2.

Audit:

To audit this recommendation:

On the Linux platform, execute the following shell command to audit this recommendation.

```
filePath="$ORACLE_HOME/network/admin/listener.ora"
if grep -qi "^s*ALLOWED_WEAK_CERT_ALGORITHMS\s*=\s*(NONE)\s*$" "$filePath";
then
    # This is compliant.
    :
elif grep -qi "^#.*ALLOWED_WEAK_CERT_ALGORITHMS" "$filePath"; then
    echo "The parameter is commented out."
elif grep -qi "^s*ALLOWED_WEAK_CERT_ALGORITHMS" "$filePath"; then
    grep -i "^s*ALLOWED_WEAK_CERT_ALGORITHMS" "$filePath"
else
    echo "The parameter is not found in the file."
fi
```

On the Windows platform, execute the following command to audit this recommendation.

```
$filePath = "$Env:ORACLE_HOME\network\admin\listener.ora"
$content = Get-Content $filePath
$lineFound = $false
foreach ($line in $content) {
```

```

$line = $line.Trim()
if ($line -match "(?i)^ALLOWED_WEAK_CERT_ALGORITHMS") {
    $lineFound = $true
    if ($line -match "(?i)^ALLOWED_WEAK_CERT_ALGORITHMS\s*=\s*\ (NONE\)\s*$")
    {
        # This is compliant.
    } else {
        Write-Host $line
    }
}
}
if (!$lineFound) {
    Write-Host "The setting is not found in the file."
}

```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

The audit command above assumes that **listener.ora** is located in the default path **\$ORACLE_HOME/network/admin/listener.ora**. If this is not the case, please specify the correct location. For example, you may be using the **TNS_ADMIN** environment variable to point to the location of **SQLNET.ORA** and **TNSNAMES.ORA**.

Remediation:

To remediate this recommendation, set the **ALLOWED_WEAK_CERT_ALGORITHMS** parameter to **NONE**.

Default Value:

SHA1

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netrf/parameters-for-the-sqlnet.ora.html#GUID-84DC4C50-A550-4770-9210-1EECB4B5DB27>
2. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/database-security-guide.pdf>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 <u>Encrypt Sensitive Data in Transit</u> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	14.4 <u>Encrypt All Sensitive Information in Transit</u> Encrypt all sensitive information in transit.			

2.2 SQLNET.ORA Settings

2.2.1 Ensure 'ACCEPT_MD5_CERTS' Is NOT SET (Automated)

Profile Applicability:

- Level 1 - RDBMS On Linux Host OS
- Level 1 - RDBMS On Windows Server Host OS

Description:

The setting **ACCEPT_MD5_CERTS** specifies whether Oracle accepts certificates signed with the MD5 algorithm.

Note: See Additional Information regarding the deprecation of this setting.

Rationale:

Weak algorithms such as MD5 and SHA-1 have known vulnerabilities that make them susceptible to attacks. Allowing their use can compromise data integrity and authentication, potentially exposing systems to risks. Transitioning to stronger algorithms, such as SHA-2, is recommended.

Impact:

Applications that use MD5-signed certificates must be updated to use certificates signed with a stronger, more secure algorithm such as SHA-2.

Audit:

On the Linux platform, execute the following shell command to audit this recommendation.

```
grep -Ei "^[^#]*ACCEPT_MD5_CERTS\s*=\s*TRUE\s*$"  
"$ORACLE_HOME/network/admin/sqlnet.ora"
```

On the Windows platform, execute the following Powershell command to audit this recommendation.

```
Select-String -Path $Env:ORACLE_HOME\network\admin\sqlnet.ora -Pattern  
"^\\s*ACCEPT_MD5_CERTS\s*=\s*TRUE\s*$" -CaseSensitive:$false
```

To be compliant with this recommendation, the output of this command should be **NULL**.

The audit command above assumes that **sqlnet.ora** is located in the default path **\$ORACLE_HOME/network/admin/sqlnet.ora**. If this is not the case, please specify the correct location. For example, you might be using the **TNS_ADMIN** environment variable to point to the location of **sqlnet.ora**. Alternatively, **sqlnet.ora** could be located in the **\$ORACLE_BASE_HOME/network/admin** or **/opt/oracle/<release_number>/network/admin** directories.

Remediation:

To remediate this recommendation, set the **ACCEPT_MD5_CERTS** to the value **FALSE** or remove **ACCEPT_MD5_CERTS** from **sqlnet.ora**.

In addition to **sqlnet.ora**, this parameter must also be set to **FALSE** in **listener.ora**.

Default Value:

FALSE

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netrf/parameters-for-the-sqlnet.ora.html#GUID-84DC4C50-A550-4770-9210-1EECB4B5DB27>
2. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/using-the-orapki-utility-to-manage-pki-elements.html#GUID-93CE62ED-97B1-43B3-8900-64107F8F274C>
3. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netrf/parameters-for-the-sqlnet.ora.html#GUID-9240F605-720C-4C57-93E3-DD26CBEA1410>

Additional Information:

Starting in Oracle Database 26ai, the **ALLOW_MD5_CERTS** and **ALLOW_SHA1_CERTS** **sqlnet.ora** parameters are deprecated.

If **ALLOWED_WEAK_CERT_ALGORITHMS** is set, then Oracle Database ignores **ALLOW_MD5_CERTS** and **ALLOW_SHA1_CERTS**. If

ALLOWED_WEAK_CERT_ALGORITHMS is not set, then Oracle Database checks and uses the **ALLOW_MD5_CERTS** and **ALLOW_SHA1_CERTS** settings.[1]

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 <u>Encrypt Sensitive Data in Transit</u> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	14.4 <u>Encrypt All Sensitive Information in Transit</u> Encrypt all sensitive information in transit.			

2.2.2 Ensure 'ACCEPT_SHA1_CERTS' Is NOT Set (Automated)

Profile Applicability:

- Level 1 - RDBMS On Linux Host OS
- Level 1 - RDBMS On Windows Server Host OS

Description:

The setting **ACCEPT_SHA1_CERTS** specifies whether Oracle accepts SHA-1 signed certificates.

Note: See Additional Information regarding the deprecation of this setting.

Rationale:

Weak algorithms such as MD5 and SHA-1 have known vulnerabilities that make them susceptible to attacks. Allowing their use can compromise data integrity and authentication, potentially exposing systems to risks. Transitioning to stronger algorithms, such as SHA-2, is recommended. The SHA1 algorithm has been deprecated by NIST in 2011.

Impact:

Applications that use SHA-1-signed certificates must be updated to use certificates signed with a stronger, more secure algorithm such as SHA-2.

Audit:

On the Linux platform, execute the following shell command to audit this recommendation.

```
grep -Ei "^[^#]*ACCEPT_SHA1_CERTS\s*=\s*TRUE\s*$"  
"$ORACLE_HOME/network/admin/sqlnet.ora"
```

On the Windows platform, execute the following command to audit this recommendation.

```
Select-String -Path $Env:ORACLE_HOME\network\admin\sqlnet.ora -Pattern  
"^\\s*ACCEPT_SHA1_CERTS\s*=\s*TRUE\s*$" -CaseSensitive:$false
```

To be compliant with this recommendation, the output of this command should be **NULL**.

The audit command above assumes that **sqlnet.ora** is located in the default path **\$ORACLE_HOME/network/admin/sqlnet.ora**. If this is not the case, please specify the correct location. For example, you might be using the **TNS_ADMIN** environment variable to point to the location of **sqlnet.ora**. Alternatively, **sqlnet.ora** could be located in the **\$ORACLE_BASE_HOME/network/admin** or **/opt/oracle/<release_number>/network/admin** directories.

Remediation:

To remediate this recommendation, set the **ACCEPT_SHA1_CERTS** to the value **FALSE** or remove **ACCEPT_SHA1_CERTS** from **sqlnet.ora**.

In addition to **sqlnet.ora**, this parameter must also be set to **FALSE** in **listener.ora**.

Default Value:

FALSE

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netrf/parameters-for-the-sqlnet.ora.html#GUID-84DC4C50-A550-4770-9210-1EECB4B5DB27>
2. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/using-the-orapki-utility-to-manage-pki-elements.html#GUID-93CE62ED-97B1-43B3-8900-64107F8F274C>
3. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netrf/parameters-for-the-sqlnet.ora.html#GUID-9240F605-720C-4C57-93E3-DD26CBEA1410>

Additional Information:

Starting in Oracle Database 26ai, the **ALLOW_MD5_CERTS** and **ALLOW_SHA1_CERTS** **sqlnet.ora** parameters are deprecated.

If **ALLOWED_WEAK_CERT_ALGORITHMS** is set, then Oracle Database ignores **ALLOW_MD5_CERTS** and **ALLOW_SHA1_CERTS**. If

ALLOWED_WEAK_CERT_ALGORITHMS is not set, then Oracle Database checks and uses the **ALLOW_MD5_CERTS** and **ALLOW_SHA1_CERTS** settings.[1]

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 <u>Encrypt Sensitive Data in Transit</u> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	14.4 <u>Encrypt All Sensitive Information in Transit</u> Encrypt all sensitive information in transit.			

2.2.3 Ensure 'ALLOWED_WEAK_CERT_ALGORITHMS' Is NOT Set (Automated)

Profile Applicability:

- Level 1 - RDBMS On Linux Host OS
- Level 1 - RDBMS On Windows Server Host OS

Description:

The setting **ALLOWED_WEAK_CERT_ALGORITHMS** determines whether Oracle accepts certificates signed with the SHA-1 or MD5 or both algorithms.

Rationale:

Weak algorithms such as MD5 and SHA-1 have known vulnerabilities that make them susceptible to attacks. Allowing their use can compromise data integrity and authentication, potentially exposing systems to risks. Transitioning to stronger algorithms, such as SHA-2, is recommended.

Impact:

Applications that use MD5 or SHA-1-signed certificates must be updated to use certificates signed with a stronger, more secure algorithm such as SHA-2.

Audit:

To audit this recommendation: On the Linux platform, execute the following shell command to audit this recommendation.

```
filePath="$ORACLE_HOME/network/admin/sqlnet.ora"
if grep -qi "^\\s*ALLOWED_WEAK_CERT_ALGORITHMS\\s*=\\s*(NONE)\\s*$" "$filePath";
then
    # This is compliant.
    :
elif grep -qi "^#.*ALLOWED_WEAK_CERT_ALGORITHMS" "$filePath"; then
    echo "The parameter is commented out."
elif grep -qi "^\\s*ALLOWED_WEAK_CERT_ALGORITHMS" "$filePath"; then
    grep -i "^\\s*ALLOWED_WEAK_CERT_ALGORITHMS" "$filePath"
else
    echo "The parameter is not found in the file."
fi
```

On the Windows platform, execute the following command to audit this recommendation.

```
$filePath = "$Env:ORACLE_HOME\network\admin\sqlnet.ora"
$content = Get-Content $filePath
$lineFound = $false
foreach ($line in $content) {
    $line = $line.Trim()
    if ($line -match "(?i)^ALLOWED_WEAK_CERT_ALGORITHMS") {
```

```

$lineFound = $true
if ($line -match "(?i)^ALLOWED_WEAK_CERT_ALGORITHMS\s*=\s*\ (NONE\)\s*$")
{
    # This is compliant.
} else {
    Write-Host $line
}
}
}
if (!$lineFound) {
    Write-Host "The setting is not found in the file."
}

```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

The audit command above assumes that **sqlnet.ora** is located in the default path **\$ORACLE_HOME/network/admin/sqlnet.ora**. If this is not the case, please specify the correct location. For example, you might be using the **TNS_ADMIN** environment variable to point to the location of **sqlnet.ora**. Alternatively, **sqlnet.ora** could be located in the **\$ORACLE_BASE_HOME/network/admin** or **/opt/oracle/<release_number>/network/admin** directories.

Remediation:

To remediate this recommendation, set the **ALLOWED_WEAK_CERT_ALGORITHMS** parameter to **NONE**.

Default Value:

NULL

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netrf/parameters-for-the-sqlnet.ora.html#GUID-B2908ADF-0973-44A9-9B34-587A3D605BED>

Additional Information:

Starting in Oracle Database 26ai, the **ALLOW_MD5_CERTS** and **ALLOW_SHA1_CERTS** **sqlnet.ora** parameters are deprecated.

If **ALLOWED_WEAK_CERT_ALGORITHMS** is set, then Oracle Database ignores **ALLOW_MD5_CERTS** and **ALLOW_SHA1_CERTS**. If **ALLOWED_WEAK_CERT_ALGORITHMS** is not set, then Oracle Database checks and uses the **ALLOW_MD5_CERTS** and **ALLOW_SHA1_CERTS** settings.[1]

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 <u>Encrypt Sensitive Data in Transit</u> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	14.4 <u>Encrypt All Sensitive Information in Transit</u> Encrypt all sensitive information in transit.			

2.2.4 Ensure 'SQLNET.ALLOWED_LOGON_VERSION_CLIENT' Is Set To 12a (Automated)

Profile Applicability:

- Level 1 - RDBMS On Linux Host OS
- Level 1 - RDBMS On Windows Server Host OS

Description:

This setting **SQLNET.ALLOWED_LOGON_VERSION_CLIENT** configures the minimum authentication protocols clients can use to connect to database instances. Please note that the term **VERSION** in the parameter name refers to the version of the authentication protocol, not the version of the Oracle Database release.

Rationale:

Allowing deprecated or weaker authentication protocols can expose the database to security vulnerabilities, increasing the risk of unauthorized access, data loss, or breaches. Ensuring that clients use secure protocols improves the overall security posture of the database environment.

Impact:

Setting this parameter to **12a** may prevent some clients from connecting to the database, leading to authentication failures. Specifically, clients may encounter the error **ORA-28040: The database does not accept your client's authentication protocol; login denied.**

Audit:

On the Linux platform, execute the following shell command to audit this recommendation.

```
filePath="$ORACLE_HOME/network/admin/sqlnet.ora"
if grep -qi "^s*SQLNET\..ALLOWED_LOGON_VERSION_CLIENT\s*=\s*12a\s*$"
"$filePath"; then
    # This is compliant.
    :
elif grep -qi "^s*#.*SQLNET\..ALLOWED_LOGON_VERSION_CLIENT" "$filePath"; then
    echo "The parameter is commented out."
elif grep -qi "SQLNET\..ALLOWED_LOGON_VERSION_CLIENT" "$filePath"; then
    grep -i "SQLNET\..ALLOWED_LOGON_VERSION_CLIENT" "$filePath"
else
    echo "The parameter is not found in the file."
fi
```

On the Windows platform, execute the following command to audit this recommendation.

```
$filePath = "$Env:ORACLE_HOME\network\admin\sqlnet.ora"
```

```

$content = Get-Content $filePath
$lineFound = $false
foreach ($line in $content) {
    $line = $line.Trim()
    if ($line -match "(?i)^SQLNET\.\ALLOWED_LOGON_VERSION_CLIENT") {
        $lineFound = $true
        if ($line -match "(?i)^SQLNET\.\ALLOWED_LOGON_VERSION_CLIENT\s*=\s*12a$")
        {
            # This is compliant.
        } else {
            Write-Host $line
        }
    }
}
if (!$lineFound) {
    Write-Host "The setting is not found in the file."
}

```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

The audit command above assumes that **sqlnet.ora** is located in the default path **\$ORACLE_HOME/network/admin/sqlnet.ora**. If this is not the case, please specify the correct location. For example, you might be using the **TNS_ADMIN** environment variable to point to the location of **sqlnet.ora**. Alternatively, **sqlnet.ora** could be located in the **\$ORACLE_BASE_HOME/network/admin** or **/opt/oracle/<release_number>/network/admin** directories.

Remediation:

To remediate this recommendation, set **SQLNET.ALLOWED_LOGON_VERSION_CLIENT** to **12a**.

Default Value:

12

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netrf/parameters-for-the-sqlnet.ora.html#GUID-B2908ADF-0973-44A9-9B34-587A3D605BED>

2.2.5 Ensure

'SQLNET.ALLOWED_LOGON_VERSION_SERVER' Is Set To 12a (Automated)

Profile Applicability:

- Level 1 - RDBMS On Linux Host OS
- Level 1 - RDBMS On Windows Server Host OS

Description:

This setting **SQLNET.ALLOWED_LOGON_VERSION_SERVER** configures the minimum authentication protocols clients can use to connect to database instances. Please note that the term **VERSION** in the parameter name refers to the version of the authentication protocol, not the version of the Oracle Database release.

Rationale:

Allowing deprecated or weaker authentication protocols can expose the database to security vulnerabilities, increasing the risk of unauthorized access, data loss, or breaches. Ensuring that clients use secure protocols improves the overall security posture of the database environment.

Impact:

Setting this parameter to **12a** may prevent some clients from connecting to the database, leading to authentication failures. Specifically, clients may encounter the error **ORA-28040: The database does not accept your client's authentication protocol; login denied**

Audit:

On the Linux platform, execute the following shell command to audit this recommendation.

```
filePath="$ORACLE_HOME/network/admin/sqlnet.ora"
if grep -qi "^s*SQLNET\..ALLOWED_LOGON_VERSION_SERVER\s*=\s*12a\s*$"
"$filePath"; then
    # This is compliant.
    :
elif grep -qi "^s*#.*SQLNET\..ALLOWED_LOGON_VERSION_SERVER" "$filePath"; then
    echo "The parameter is commented out."
elif grep -qi "SQLNET\..ALLOWED_LOGON_VERSION_SERVER" "$filePath"; then
    grep -i "SQLNET\..ALLOWED_LOGON_VERSION_SERVER" "$filePath"
else
    echo "The parameter is not found in the file."
fi
```

On the Windows platform, execute the following command to audit this recommendation.

```

$filePath = "$Env:ORACLE_HOME\network\admin\sqlnet.ora"
$content = Get-Content $filePath
$lineFound = $false
foreach ($line in $content) {
    $line = $line.Trim()
    if ($line -match "(?i)^SQLNET\.\ALLOWED_LOGON_VERSION_SERVER") {
        $lineFound = $true
        if ($line -match "(?i)^SQLNET\.\ALLOWED_LOGON_VERSION_SERVER\s*=\s*12a$")
        {
            # This is compliant.
        } else {
            Write-Host $line
        }
    }
}
if (!$lineFound) {
    Write-Host "The setting is not found in the file."
}

```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

The audit command above assumes that **sqlnet.ora** is located in the default path **\$ORACLE_HOME/network/admin/sqlnet.ora**. If this is not the case, please specify the correct location. For example, you might be using the **TNS_ADMIN** environment variable to point to the location of **sqlnet.ora**. Alternatively, **sqlnet.ora** could be located in the **\$ORACLE_BASE_HOME/network/admin** or **/opt/oracle/<release_number>/network/admin** directories.

Remediation:

To remediate this recommendation, set **SQLNET.ALLOWED_LOGON_VERSION_SERVER** to **12a**.

Default Value:

12

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netrf/parameters-for-the-sqlnet.ora.html#GUID-1FA9D26C-4D97-4D1C-AB47-1EC234D924AA>

2.2.6 Ensure 'SQLNET.ENCRYPTION_CLIENT Is Set To 'REQUIRED' (Automated)

Profile Applicability:

- Level 1 - RDBMS On Linux Host OS
- Level 1 - RDBMS On Windows Server Host OS

Description:

The **SQLNET.ENCRYPTION_CLIENT** parameter determines whether the client side of a database connection enforces network encryption. When set to **REQUIRED**, this parameter mandates that all data transmitted between the client and the database is encrypted, preventing the client from connecting if encryption cannot be established. This setting ensures that sensitive data remains protected during transmission.

If you are using TLS, this is not a required check. Oracle database network encryption configured through TLS/SSL is also an acceptable mechanism and may be implemented in lieu of this setting.

Rationale:

Network encryption is crucial for safeguarding data transmitted over networks. Without encryption, data, including sensitive information like credentials, financial data, and personal details, is vulnerable to interception and potential compromise. Setting **SQLNET.ENCRYPTION_CLIENT** to **REQUIRED** ensures that all client connections to the database use encryption, aligning with best practices for data protection and regulatory compliance requirements.

Impact:

Clients or applications that do not support encryption will NOT be able to connect to the database, which may necessitate updates to older client configurations.

Audit:

On the Linux platform, execute the following shell command to audit this recommendation.

```
filePath="$ORACLE_HOME/network/admin/sqlnet.ora"
if grep -qi "^s*SQLNET\..ENCRYPTION_CLIENT\s*=\s*REQUIRED\s*$" "$filePath";
then
    # This is compliant.
    :
elif grep -qi "^#.*SQLNET\..ENCRYPTION_CLIENT" "$filePath"; then
    echo "The parameter is commented out."
elif grep -qi "SQLNET\..ENCRYPTION_CLIENT" "$filePath"; then
    grep -i "SQLNET\..ENCRYPTION_CLIENT" "$filePath"
else
    echo "The parameter is not found in the file."
```

fi

On the Windows platform, execute the following command to audit this recommendation.

```
$filePath = "$Env:ORACLE_HOME\network\admin\sqlnet.ora"
$content = Get-Content $filePath
$lineFound = $false
foreach ($line in $content) {
    $line = $line.Trim()
    if ($line -match "(?i)^SQLNET\..ENCRYPTION_CLIENT") {
        $lineFound = $true
        if ($line -match "(?i)^SQLNET\..ENCRYPTION_CLIENT\s*=\s*REQUIRED$") {
            # This is compliant.
        } else {
            Write-Host $line
        }
    }
}
if (!$lineFound) {
    Write-Host "The parameter is not found in the file."
}
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance. The audit command above assumes that **sqlnet.ora** is located in the default path **\$ORACLE_HOME/network/admin/sqlnet.ora**. If this is not the case, please specify the correct location. For example, you might be using the **TNS_ADMIN** environment variable to point to the location of **sqlnet.ora**. Alternatively, **sqlnet.ora** could be located in the **\$ORACLE_BASE_HOME/network/admin** or **/opt/oracle/<release_number>/network/admin** directories.

Remediation:

To remediate this recommendation, set **SQLNET.ENCRYPTION_CLIENT** to **REQUIRED**.

Default Value:

ACCEPTED

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netrf/parameters-for-the-sqlnet.ora.html#GUID-F4A86AFC-4600-405F-88E7-DC79213FEC19>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 <u>Encrypt Sensitive Data in Transit</u> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	14.4 <u>Encrypt All Sensitive Information in Transit</u> Encrypt all sensitive information in transit.			

2.2.7 Ensure 'SQLNET.ENCRYPTION_SERVER' Is Set To 'REQUIRED' (Automated)

Profile Applicability:

- Level 1 - RDBMS On Linux Host OS
- Level 1 - RDBMS On Windows Server Host OS

Description:

The **SQLNET.ENCRYPTION_SERVER** parameter determines whether the server side of a database connection enforces network encryption. When set to **REQUIRED**, this parameter mandates that all data transmitted between the client and the database is encrypted, preventing the server from connecting if encryption cannot be established. This setting ensures that sensitive data remains protected during transmission.

If you are using TLS, this is not a required check. Oracle database network encryption configured through TLS/SSL is also an acceptable mechanism and may be implemented in lieu of this setting.

Caution: Modifying this parameter changes how the database creates and tests password verifiers. Please refer to recommendation 4.4.

Rationale:

Network encryption is crucial for safeguarding data transmitted over networks. Without encryption, data, including sensitive information like credentials, financial data, and personal details, is vulnerable to interception and potential compromise. Setting **SQLNET.ENCRYPTION_SERVER** to **REQUIRED** ensures that all client connections to the database use encryption, aligning with best practices for data protection and regulatory compliance requirements.

Impact:

Clients or applications that do not support encryption will NOT be able to connect to the database, which may necessitate updates to older client configurations.

Audit:

On the Linux platform, execute the following shell command to audit this recommendation.

```
filePath="$ORACLE_HOME/network/admin/sqlnet.ora"
if grep -qi "^s*SQLNET\..ENCRYPTION_SERVER\s*=\s*REQUIRED\s*$" "$filePath";
then
    # This is compliant.
    :
elif grep -qi "^#.*SQLNET\..ENCRYPTION_SERVER" "$filePath"; then
    echo "The parameter is commented out."
elif grep -qi "SQLNET\..ENCRYPTION_SERVER" "$filePath"; then
```

```
grep -i "SQLNET\|.ENCRYPTION_SERVER" "$filePath"
else
echo "The parameter is not found in the file."
fi
```

On the Windows platform, execute the following command to audit this recommendation.

```
$filePath = "$Env:ORACLE_HOME\network\admin\sqlnet.ora"
$content = Get-Content $filePath
$lineFound = $false
foreach ($line in $content) {
    $line = $line.Trim()
    if ($line -match "(?i)^SQLNET\|.ENCRYPTION_SERVER") {
        $lineFound = $true
        if ($line -match "(?i)^SQLNET\|.ENCRYPTION_SERVER\s*=\s*REQUIRED$") {
            # This is compliant.
        } else {
            Write-Host $line
        }
    }
}
if (!$lineFound) {
    Write-Host "The parameter is not found in the file."
}
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

The audit command above assumes that **sqlnet.ora** is located in the default path **\$ORACLE_HOME/network/admin/sqlnet.ora**. If this is not the case, please specify the correct location. For example, you might be using the **TNS_ADMIN** environment variable to point to the location of sqlnet.ora. Alternatively, sqlnet.ora could be located in the **\$ORACLE_BASE_HOME/network/admin** or **/opt/oracle/<release_number>/network/admin** directories.

Remediation:

To remediate this recommendation, set **SQLNET.ENCRYPTION_SERVER** to **REQUIRED**.

Default Value:

ACCEPTED

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netrf/parameters-for-the-sqlnet.ora.html#GUID-AADC66D0-7574-4AD4-8408-5CD7D2AABA2A>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 <u>Encrypt Sensitive Data in Transit</u> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	14.4 <u>Encrypt All Sensitive Information in Transit</u> Encrypt all sensitive information in transit.			

2.2.8 Ensure 'SQLNET.ENCRYPTION_TYPES_CLIENT' Is Set To 'AES256' (Automated)

Profile Applicability:

- Level 1 - RDBMS On Linux Host OS
- Level 1 - RDBMS On Windows Server Host OS

Description:

The **SQLNET.ENCRYPTION_TYPES_CLIENT** parameter specifies the encryption algorithms that the client can use for database connections. This setting allows you to ensure that all data transmitted between the client and the database is encrypted using strong, secure algorithms.

Rationale:

Limiting the client to use only strong encryption algorithms helps to protect data integrity and confidentiality during transit. Using stronger algorithms reduces the risk of interception or tampering by unauthorized parties.

Impact:

If the AES256 algorithm is not available or installed on either the client or server, connections will be terminated. This may result in the error **ORA-12650: No common encryption or data integrity algorithm**, preventing SQL clients from connecting to the database.

Audit:

On the Linux platform, execute the following shell command to audit this recommendation.

```
filePath="$ORACLE_HOME/network/admin/sqlnet.ora"
if grep -qi "^s*SQLNET\..ENCRYPTION_TYPES_CLIENT\s*=\s*(AES256)" "$filePath";
then
    # This is compliant.
    :
elif grep -qi "^#.*SQLNET\..ENCRYPTION_TYPES_CLIENT" "$filePath"; then
    echo "The parameter is commented out."
elif grep -qi "^#.*SQLNET\..ENCRYPTION_TYPES_CLIENT\s*=\s*,.*" "$filePath";
then
    echo "Multiple encryption types are specified."
elif grep -qi "SQLNET\..ENCRYPTION_TYPES_CLIENT" "$filePath"; then
    grep -i "SQLNET\..ENCRYPTION_TYPES_CLIENT" "$filePath"
else
    echo "The parameter is not found in the file."
fi
```

On the Windows platform, execute the following command to audit this recommendation.

```

$filePath = "$Env:ORACLE_HOME\network\admin\sqlnet.ora"
$content = Get-Content $filePath
$lineFound = $false
foreach ($line in $content) {
    $line = $line.Trim()
    if ($line -match "(?i)^SQLNET\..ENCRYPTION_TYPES_CLIENT") {
        $lineFound = $true
        if ($line -match
"(?i)^SQLNET\..ENCRYPTION_TYPES_CLIENT\s*=\s*\s*(AES256\)$") {
            # This is compliant.
        } elseif ($line -match "(?i)^SQLNET\..ENCRYPTION_TYPES_CLIENT\s*=.*,.*") {
            Write-Host "Multiple encryption types are specified."
        } else {
            Write-Host $line
        }
    }
}
if (!$lineFound) {
    Write-Host "The parameter is not found in the file."
}

```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

The audit command above assumes that **sqlnet.ora** is located in the default path **\$ORACLE_HOME/network/admin/sqlnet.ora**. If this is not the case, please specify the correct location. For example, you might be using the **TNS_ADMIN** environment variable to point to the location of **sqlnet.ora**. Alternatively, **sqlnet.ora** could be located in the **\$ORACLE_BASE_HOME/network/admin** or **/opt/oracle/<release_number>/network/admin** directories.

Remediation:

To remediate this recommendation, set **SQLNET.ENCRYPTION_TYPES_CLIENT** to **AES256**.

```
SQLNET.ENCRYPTION_TYPES_CLIENT=(AES256)
```

Default Value:

All available algorithms

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netrf/parameters-for-the-sqlnet.ora.html#GUID-9E9E6054-DD98-4B51-93A4-3802B5AA779F>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 <u>Encrypt Sensitive Data in Transit</u> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	14.4 <u>Encrypt All Sensitive Information in Transit</u> Encrypt all sensitive information in transit.			

2.2.9 Ensure 'SQLNET.ENCRYPTION_TYPES_SERVER' Is Set To AES256 (Manual)

Profile Applicability:

- Level 1 - RDBMS On Linux Host OS
- Level 1 - RDBMS On Windows Server Host OS

Description:

The **SQLNET.ENCRYPTION_TYPES_SERVER** parameter specifies the encryption algorithms that the server enforces for database connections. This setting allows you to ensure that all data transmitted between the client and the database is encrypted using strong, secure algorithms.

Rationale:

Limiting the server to use only strong encryption algorithms helps to protect data integrity and confidentiality during transit. Using stronger algorithms reduces the risk of interception or tampering by unauthorized parties.

Impact:

If the **AES256** algorithm is not available or installed on either the client or server, connections will be terminated. This may result in the error **ORA-12650: No common encryption or data integrity algorithm**, preventing SQL clients from connecting to the database.

Audit:

On the Linux platform, execute the following shell command to audit this recommendation.

```
filePath="$ORACLE_HOME/network/admin/sqlnet.ora"
if grep -qi "^s*SQLNET\..ENCRYPTION_TYPES_SERVER\s*=\s*(AES256)" "$filePath";
then
    # This is compliant.
    :
elif grep -qi "^#.*SQLNET\..ENCRYPTION_TYPES_SERVER" "$filePath"; then
    echo "The parameter is commented out."
elif grep -qi "^#.*SQLNET\..ENCRYPTION_TYPES_SERVER\s*=\s*,.*" "$filePath";
then
    echo "Multiple encryption types are specified."
elif grep -qi "SQLNET\..ENCRYPTION_TYPES_SERVER" "$filePath"; then
    grep -i "SQLNET\..ENCRYPTION_TYPES_SERVER" "$filePath"
else
    echo "The parameter is not found in the file."
fi
```

On the Windows platform, execute the following command to audit this recommendation.

```

$filePath = "$Env:ORACLE_HOME\network\admin\sqlnet.ora"
$content = Get-Content $filePath
$lineFound = $false
foreach ($line in $content) {
    $line = $line.Trim()
    if ($line -match "(?i)^SQLNET\..ENCRYPTION_TYPES_SERVER") {
        $lineFound = $true
        if ($line -match
"(?i)^SQLNET\..ENCRYPTION_TYPES_SERVER\s*=\s*\s*(AES256\)$") {
            # This is compliant.
        } elseif ($line -match "(?i)^SQLNET\..ENCRYPTION_TYPES_SERVER\s*=\s*.*,.*") {
            Write-Host "Multiple encryption types are specified."
        } else {
            Write-Host $line
        }
    }
}
if (!$lineFound) {
    Write-Host "The line 'SQLNET.ENCRYPTION_TYPES_SERVER' is not found in the
file."
}

```

To be compliant with this recommendation, the output of this command should be NULL. Lack of results indicates compliance. The audit command above assumes that **sqlnet.ora** is located in the default path **\$ORACLE_HOME/network/admin/sqlnet.ora**. If this is not the case, please specify the correct location. For example, you might be using the **TNS_ADMIN** environment variable to point to the location of **sqlnet.ora**. Alternatively, **sqlnet.ora** could be located in the **\$ORACLE_BASE_HOME/network/admin** or **/opt/oracle/<release_number>/network/admin** directories.

Remediation:

To remediate this recommendation, set **SQLNET.ENCRYPTION_TYPES_SERVER** to **AES256**.

```
SQLNET.ENCRYPTION_TYPES_SERVER=(AES256)
```

Default Value:

All available algorithms

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netrf/parameters-for-the-sqlnet.ora.html#GUID-5A5BFE73-BA12-4035-A600-834CA9E38427>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 <u>Encrypt Sensitive Data in Transit</u> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	14.4 <u>Encrypt All Sensitive Information in Transit</u> Encrypt all sensitive information in transit.			

2.2.10 Ensure 'SQLNET.CRYPTO_CHECKSUM_CLIENT' Is Set To 'REQUIRED' (Automated)

Profile Applicability:

- Level 1 - RDBMS On Linux Host OS
- Level 1 - RDBMS On Windows Server Host OS

Description:

The **SQLNET.CRYPTO_CHECKSUM_CLIENT** parameter specifies the checksum behavior for the client when connecting to a server. This setting enables the client to enforce cryptographic checksums, which verify the integrity of data transmitted during client-server interactions.

Oracle networking already performs checksumming, so additional cryptographic checksumming is usually of limited value. A setting of **REQUIRED** at the server requires that incoming connections are encrypted with AES256.

Oracle database network encryption configured through TLS/SSL is also an acceptable mechanism and may be implemented in lieu of this setting.

Rationale:

Enabling cryptographic checksums for client connections ensures a higher degree of data integrity.

Impact:

If **SQLNET.CRYPTO_CHECKSUM_CLIENT** is set to **REQUIRED**, the client will be unable to connect to servers that do not require cryptographic checksums, resulting in connection failures. This may cause errors such as **ORA-12650: No common encryption or data integrity algorithm** if the server does not meet the client's checksum requirements.

Audit:

On the Linux platform, execute the following shell command to audit this recommendation.

```
filePath="$ORACLE_HOME/network/admin/sqlnet.ora"
if grep -qi "^\\s*SQLNET\\.CRYPTO_CHECKSUM_CLIENT\\s*=\\s*REQUIRED\\s*$"
"$filePath"; then
    # This is compliant.
    :
elif grep -qi "^#.*SQLNET\\.CRYPTO_CHECKSUM_CLIENT" "$filePath"; then
    echo "The parameter is commented out."
elif grep -qi "SQLNET\\.CRYPTO_CHECKSUM_CLIENT" "$filePath"; then
    grep -i "SQLNET\\.CRYPTO_CHECKSUM_CLIENT" "$filePath"
else
```

```
    echo "The parameter is not found in the file."
fi
```

On the Windows platform, execute the following command to audit this recommendation.

```
$filePath = "$Env:ORACLE_HOME\network\admin\sqlnet.ora"
$content = Get-Content $filePath
$lineFound = $false
foreach ($line in $content) {
    $line = $line.Trim()
    if ($line -match "(?i)^SQLNET\..CRYPTO_CHECKSUM_CLIENT") {
        $lineFound = $true
        if ($line -match "(?i)^SQLNET\..CRYPTO_CHECKSUM_CLIENT\s*=\s*REQUIRED$") {
            # This is compliant.
        } else {
            Write-Host $line
        }
    }
}
if (!$lineFound) {
    Write-Host "The setting is not found in the file."
}
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance. The audit command above assumes that **sqlnet.ora** is located in the default path **\$ORACLE_HOME/network/admin/sqlnet.ora**. If this is not the case, please specify the correct location. For example, you might be using the **TNS_ADMIN** environment variable to point to the location of **sqlnet.ora**. Alternatively, **sqlnet.ora** could be located in the **\$ORACLE_BASE_HOME/network/admin** or **/opt/oracle/<release_number>/network/admin** directories.

Remediation:

To remediate this recommendation, set **SQLNET.CRYPTO_CHECKSUM_CLIENT** to **REQUIRED**.

```
SQLNET.CRYPTO_CHECKSUM_CLIENT=REQUIRED
```

Default Value:

accepted

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netrf/parameters-for-the-sqlnet.ora.html#GUID-D76832AE-FFFA-47D1-9EF6-46D95C78004C>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 <u>Encrypt Sensitive Data in Transit</u> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	14.4 <u>Encrypt All Sensitive Information in Transit</u> Encrypt all sensitive information in transit.			

2.2.11 Ensure 'SQLNET.CRYPTO_CHECKSUM_SERVER' Is Set To 'REQUIRED' (Automated)

Profile Applicability:

- Level 1 - RDBMS On Linux Host OS
- Level 1 - RDBMS On Windows Server Host OS

Description:

The **SQLNET.CRYPTO_CHECKSUM_SERVER** parameter specifies the checksum behavior for the server when connecting to a client. This setting enables the server to enforce cryptographic checksums, which verify the integrity of data transmitted during client-server interactions.

Oracle networking already performs checksumming, so additional cryptographic checksumming is usually of limited value. A setting of **REQUIRED** at the server requires that the outgoing connections are encrypted with AES256.

Oracle database network encryption configured through TLS/SSL is also an acceptable mechanism and may be implemented in lieu of this setting.

Rationale:

Enabling cryptographic checksums for server connections ensures a higher degree of data integrity.

Impact:

If **SQLNET.CRYPTO_CHECKSUM_SERVER** is set to **REQUIRED**, the client will be unable to connect to servers that do not require cryptographic checksums, resulting in connection failures. This may cause errors such as **ORA-12650: No common encryption or data integrity algorithm** if the server does not meet the client's checksum requirements.

Audit:

On the Linux platform, execute the following shell command to audit this recommendation.

```
filePath="$ORACLE_HOME/network/admin/sqlnet.ora"
if grep -qi "^\\s*SQLNET\\.CRYPTO_CHECKSUM_SERVER\\s*=\\s*REQUIRED\\s*$"
"$filePath"; then
    # This is compliant.
    :
elif grep -qi "^#.*SQLNET\\.CRYPTO_CHECKSUM_SERVER" "$filePath"; then
    echo "The parameter is commented out."
elif grep -qi "SQLNET\\.CRYPTO_CHECKSUM_SERVER" "$filePath"; then
    grep -i "SQLNET\\.CRYPTO_CHECKSUM_SERVER" "$filePath"
else
```

```
    echo "The parameter is not found in the file."
fi
```

On the Windows platform, execute the following command to audit this recommendation.

```
$filePath = "$Env:ORACLE_HOME\network\admin\sqlnet.ora"
$content = Get-Content $filePath
$lineFound = $false
foreach ($line in $content) {
    $line = $line.Trim()
    if ($line -match "(?i)^SQLNET\.CRYPTO_CHECKSUM_SERVER") {
        $lineFound = $true
        if ($line -match "(?i)^SQLNET\.CRYPTO_CHECKSUM_SERVER\s*=\s*REQUIRED$") {
            # This is compliant.
        } else {
            Write-Host $line
        }
    }
}
if (!$lineFound) {
    Write-Host "The setting is not found in the file."
}
```

If the output shows **FALSE** or **NONE**, the control is not configured correctly.

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance. The audit command above assumes that **sqlnet.ora** is located in the default path **\$ORACLE_HOME/network/admin/sqlnet.ora**. If this is not the case, please specify the correct location. For example, you might be using the **TNS_ADMIN** environment variable to point to the location of **sqlnet.ora**. Alternatively, **sqlnet.ora** could be located in the **\$ORACLE_BASE_HOME/network/admin** or **/opt/oracle/<release_number>/network/admin** directories.

Remediation:

To remediate this recommendation, set **SQLNET.CRYPTO_CHECKSUM_SERVER** to **REQUIRED**.

```
SQLNET.CRYPTO_CHECKSUM_SERVER=REQUIRED
```

Default Value:

accepted

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netrf/parameters-for-the-sqlnet.ora.html#GUID-FE083F54-FF01-4D2C-90E7-BEA527FF5696>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 <u>Encrypt Sensitive Data in Transit</u> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	14.4 <u>Encrypt All Sensitive Information in Transit</u> Encrypt all sensitive information in transit.			

2.2.12 Ensure 'SSL_CERT_REVOCATION' Is Set To 'REQUIRED' (Automated)

Profile Applicability:

- Level 1 - RDBMS On Linux Host OS
- Level 1 - RDBMS On Windows Server Host OS

Description:

The **SSL_CERT_REVOCATION** parameter in Oracle's **sqlnet.ora** file specifies whether the system should check the revocation status of SSL certificates during authentication. This check ensures that invalid, revoked, or compromised certificates cannot be used to establish secure connections.

Rationale:

Enabling SSL certificate revocation checking helps maintain the integrity and security of SSL connections by ensuring that only valid certificates are accepted. Without this validation, compromised certificates could allow unauthorized entities to impersonate legitimate servers or clients, potentially exposing sensitive data to interception or attacks.

Impact:

Enabling this setting without proper certificate infrastructure (e.g., CRL or OCSP) could result in failed connections if the revocation status cannot be determined.

Audit:

On the Linux platform, execute the following shell command to audit this recommendation.

```
filePath="$ORACLE_HOME/network/admin/sqlnet.ora"
if grep -qi "^s*SQLNET\.SSL_CERT_REVOCATION\s*=\s*REQUIRED\s*$" "$filePath";
then
    # This is compliant.
    :
elif grep -qi "^#.*SQLNET\.SSL_CERT_REVOCATION" "$filePath"; then
    echo "The parameter is commented out."
elif grep -qi "SQLNET\.SSL_CERT_REVOCATION" "$filePath"; then
    grep -i "SQLNET\.SSL_CERT_REVOCATION" "$filePath"
else
    echo "The parameter is not found in the file."
fi
```

On the Windows platform, execute the following command to audit this recommendation.

```
$filePath = "$Env:ORACLE_HOME\network\admin\sqlnet.ora"
$content = Get-Content $filePath
```

```

$lineFound = $false
foreach ($line in $content) {
    $line = $line.Trim()
    if ($line -match "(?i)^SQLNET\.SSL_CERT_REVOCATION") {
        $lineFound = $true
        if ($line -match "(?i)^SQLNET\.SSL_CERT_REVOCATION\s*=\s*REQUIRED$") {
            # This is compliant.
        } else {
            Write-Host $line
        }
    }
}
if (!$lineFound) {
    Write-Host "The setting is not found in the file."
}

```

If the output shows **FALSE** or **NONE**, the control is not configured correctly.

Remediation:

To remediate this recommendation, set **SSL_CERT_REVOCATION** to **required**.

```
SQLNET.SSL_CERT_REVOCATION=REQUIRED
```

Default Value:

none

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/netrf/parameters-for-the-sqlnet.ora.html#GUID-412678FF-2A41-4B92-8522-11E6A14A6671>
2. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-transport-layer-security-encryption.html#GUID-C176D190-CFF8-45C1-BAA0-34388008A56A>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.2 Establish an Access Revoking Process Establish and follow a process, preferably automated, for revoking access to enterprise assets, through disabling accounts immediately upon termination, rights revocation, or role change of a user. Disabling accounts, instead of deleting accounts, may be necessary to preserve audit trails.			
v7	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.			

2.3 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.

2.3.1 Ensure 'BACKGROUND_CORE_DUMP' Is Not Set To 'Full' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **BACKGROUND_CORE_DUMP** parameter in Oracle's **init.ora** file specifies the level of detail captured in core dumps generated by background processes during an exception. Setting this parameter to **FULL** enables the creation of detailed SGA dumps, which may include sensitive information.

Rationale:

Allowing **BACKGROUND_CORE_DUMP** to be set to **FULL** increases the risk of exposing sensitive data, such as encryption keys, passwords, or other confidential information, in the core dump files. These files could be accessed by unauthorized users, leading to data breaches. Limiting the detail in core dumps reduces this risk while still providing enough information for diagnostics.

The use of TDE doesn't prevent, or encrypt the dumps generated by the background processes.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT DECODE(V.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID_TO_CON_NAME(V.CON_ID)) AS CONTAINERNAME, UPPER(V.VALUE)
FROM GV$SYSTEM_PARAMETER V
WHERE UPPER(NAME) = 'BACKGROUND_CORE_DUMP'
AND UPPER(VALUE) = 'FULL'
ORDER BY CON_ID;
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

Remediation:

To remediate this recommendation, execute the following SQL statement.

```
ALTER SYSTEM SET BACKGROUND_CORE_DUMP='partial' SCOPE=BOTH;
```

Note: This parameter is not modifiable at the PDB level. You must modify this parameter at the CDB level.

Default Value:

partial

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/26/refrn/BACKGROUND_CORE_DUMP.html#GUID-A7118B00-4814-453E-872E-B68E4CB192F6

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.5 <u>Securely Dispose of Data</u> Securely dispose of data as outlined in the enterprise's data management process. Ensure the disposal process and method are commensurate with the data sensitivity.			
v7	13.2 <u>Remove Sensitive Data or Systems Not Regularly Accessed by Organization</u> Remove sensitive data or systems not regularly accessed by the organization from the network. These systems shall only be used as stand alone systems (disconnected from the network) by the business unit needing to occasionally use the system or completely virtualized and powered off until needed.			

2.3.2 Ensure 'SHADOW_CORE_DUMP' Is Not Set To 'Full' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The setting **SHADOW_CORE_DUMP** determines whether SGA is included in the core dump for foreground(client) processes.

Rationale:

The non-default value of **full** presents a security concern due to the potential for inclusion of sensitive data in the dump file, even when TDE-tablespace is in use.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT DECODE(V.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID TO CON_NAME(V.CON_ID)) AS CONTAINERNAME,
       UPPER(NAME), UPPER(V.VALUE)
FROM GV$SYSTEM_PARAMETER V
WHERE UPPER(NAME) = 'SHADOW_CORE_DUMP'
AND UPPER(VALUE) = 'FULL'
ORDER BY CON_ID;
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

Remediation:

To remediate this recommendation, execute the following SQL statement.

```
ALTER SYSTEM SET SHADOW_CORE_DUMP='partial' SCOPE=BOTH;
```

Or

```
ALTER SYSTEM SET SHADOW_CORE_DUMP='none' SCOPE=BOTH;
```

Default Value:

On Linux platform, the default value is **partial**. On Windows, the default value is **none**.

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/26/refrn/SHADOW_CORE_DUMP.html#GUID-9DB6F1E1-22F7-4A69-ACE6-F5865625F5B2

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.5 <u>Securely Dispose of Data</u> Securely dispose of data as outlined in the enterprise's data management process. Ensure the disposal process and method are commensurate with the data sensitivity.			
v7	13.2 <u>Remove Sensitive Data or Systems Not Regularly Accessed by Organization</u> Remove sensitive data or systems not regularly accessed by the organization from the network. These systems shall only be used as stand alone systems (disconnected from the network) by the business unit needing to occasionally use the system or completely virtualized and powered off until needed.			

2.3.3 Ensure 'MLE_PROG_LANGUAGES' Is Set To 'OFF' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **MLE_PROG_LANGUAGES** will enable Oracle Database Multilingual Engine (MLE), which is currently supporting Java Script. This parameter is available starting with Oracle Database 26ai.

Rationale:

To minimize attack surface, set the parameter to **OFF**, if Java Script will not be running against the Oracle database system.

Impact:

Java Script code will not be running anymore, MLE feature will be disabled.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT DECODE(V.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID_TO_CON_NAME(V.CON_ID)) AS CONTAINERNAME, UPPER(V.VALUE)
FROM GV$SYSTEM_PARAMETER V
WHERE UPPER(NAME) = 'MLE_PROG_LANGUAGES'
AND UPPER(VALUE) != 'OFF'
ORDER BY CON_ID;
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

Remediation:

To remediate this recommendation, execute the following SQL statement.

```
ALTER SYSTEM SET MLE_PROG_LANGUAGES='OFF' SCOPE=BOTH;
```

Default Value:

ALL

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/26/refrn/MLE_PROG_LANGUAGES.html#GUID-22596210-9BAB-41CF-B6B7-A95B77C0DD72

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.5 <u>Use Up-to-Date and Trusted Third-Party Software Components</u> Use up-to-date and trusted third-party software components. When possible, choose established and proven frameworks and libraries that provide adequate security. Acquire these components from trusted sources or evaluate the software for vulnerabilities before use.			
v7	18.4 <u>Only Use Up-to-date And Trusted Third-Party Components</u> Only use up-to-date and trusted third-party components for the software developed by the organization.			

2.3.4 Ensure 'ALLOW_GROUP_ACCESS_TO_SGA' Is Set To 'FALSE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **ALLOW_GROUP_ACCESS_TO_SGA** controls Linux group access to shared memory on Linux platforms. By default, database shared memory is created with owner access only.

Rationale:

This is to minimize the attack surface, as setting **ALLOW_GROUP_ACCESS_TO_SGA** to true will allow linux accounts with the same group membership of the oracle software account owner to access the shared memory and thus the ability to dump data from memory.

Impact:

No impact on database operations, but it may impact third party software that requires access to Oracle SGA memory such as security monitoring software.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT DECODE(V.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID TO_CON_NAME(V.CON_ID)) AS CONTAINERNAME, UPPER(V.VALUE)
FROM GV$SYSTEM_PARAMETER V
WHERE UPPER(NAME) = 'ALLOW_GROUP_ACCESS_TO_SGA'
AND UPPER(VALUE) != 'FALSE'
ORDER BY CON_ID;
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

Remediation:

To remediate this setting, execute the following SQL statement.

```
ALTER SYSTEM SET ALLOW_GROUP_ACCESS_TO_SGA='FALSE' SCOPE=SPFILE;
```

Note: This parameter is not modifiable at the PDB level. You must modify this parameter at the CDB level.

Default Value:

FALSE

References:

1. https://docs.oracle.com/en//database/oracle/oracle-database/26/refrn/ALLOW_GROUP_ACCESS_TO_SGA.html#GUID-A6BABC97-7EF8-4E1A-89C9-CF81EDCFB4A2

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

2.3.5 Review Undocumented (Underscore) Parameters Not Set To 'DEFAULT' Values (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

Oracle allows the use of undocumented (also known as hidden) parameters, which are primarily intended for internal or diagnostic purposes. Undocumented parameter values should not be set or changed from their default values, unless explicitly instructed by Oracle Support for internal or diagnostic purposes.

Rationale:

As not documented, the impact of these parameters is unknown or unpredictable. Reviewing such parameters ensures that they are not inadvertently affecting database operation, security, stability, or performance. They must be reviewed carefully to avoid unintended consequences, compliance violations, or operational issues.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT DECODE(V.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID_TO_CON_NAME(V.CON_ID)) AS CONTAINERNAME,
       UPPER(NAME), UPPER(V.VALUE), V.UPDATE_COMMENT
FROM GV$SYSTEM_PARAMETER V
WHERE SUBSTR(NAME,1,1) = '_'
AND ISDEFAULT = 'FALSE';
```

To be compliant with this recommendation, the output of this command should be **NULL**. If the output of this command is not **NULL**, then review the output for applicability of undocumented parameters for your organization.

Remediation:

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

```
ALTER SYSTEM SET <undocumented_parameter>=<Default value> SCOPE=SPFILE;
```

Default Value:

Undocumented parameters do not always have officially defined default values. Oracle does not guarantee backward compatibility for them across versions.

Additional Information:

Please note that some parameters such as `_instance_recovery_bloom_filter_size` or `_dmm_blas_library` could be ignored in 23.5.

2.3.6 Ensure 'OS_ROLES' Is Set To 'FALSE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

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:

To assess this recommendation, execute the following SQL statement.

```
SELECT DECODE(V.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID_TO_CON_NAME(V.CON_ID)) AS CONTAINERNAME,
              UPPER(NAME), UPPER(V.VALUE)
FROM GV$SYSTEM_PARAMETER V
WHERE UPPER(NAME) = 'OS_ROLES'
AND UPPER(VALUE) != 'FALSE'
ORDER BY CON_ID;
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

Remediation:

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

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

Note: This parameter is not modifiable at the PDB level. You must modify this parameter at the CDB level.

Default Value:

FALSE

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/26/refrn/OS_ROLES.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

2.3.7 Ensure 'REMOTE_OS_ROLES' Is Set To 'FALSE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

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 DECODE(V.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID TO CON_NAME(V.CON_ID)) AS CONTAINERNAME,
       UPPER(NAME), UPPER(V.VALUE)
FROM GV$SYSTEM_PARAMETER V
WHERE UPPER(NAME) = 'REMOTE_OS_ROLES'
AND UPPER(VALUE) != 'FALSE'
ORDER BY CON_ID;
```

To be compliant with this recommendation, the output of this command should be `NULL`. Lack of results indicates compliance.

Remediation:

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

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

Note: This parameter is not modifiable at the PDB level. You must modify this parameter at the CDB level.

Default Value:

`FALSE`

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/26/refrn/REMOTE_OS_ROLES.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.6 <u>Centralize Account Management</u> Centralize account management through a directory or identity service.			
v7	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.			

2.3.8 Ensure 'SEC_MAX_FAILED_LOGIN_ATTEMPTS' Is Set To '3' Or Less (Automated)

Profile Applicability:

- Level 1 - RDBMS

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 DECODE(V.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID_TO_CON_NAME(V.CON_ID)) AS CONTAINERNAME,
       UPPER(NAME),UPPER(V.VALUE)
FROM GV$SYSTEM_PARAMETER V
WHERE UPPER(NAME)='SEC_MAX_FAILED_LOGIN_ATTEMPTS'
AND TO_NUMBER(DECODE(VALUE,'1',1,'2',2,'3',3,9999)) > 3;
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

Remediation:

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

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

Note: This parameter is not modifiable at the PDB level. You must modify this parameter at the CDB level.

Default Value:

3

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/26/refrn/SEC_MAX_FAILED_LOGIN_ATTEMPTS.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.2 <u>Establish an Access Revoking Process</u> Establish and follow a process, preferably automated, for revoking access to enterprise assets, through disabling accounts immediately upon termination, rights revocation, or role change of a user. Disabling accounts, instead of deleting accounts, may be necessary to preserve audit trails.			
v7	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.			

2.3.9 Ensure 'SEC_PROTOCOL_ERROR_FURTHER_ACTION' Is Set To '(DROP,3)' (Automated)

Profile Applicability:

- Level 1 - RDBMS

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)** or **(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 DECODE(V.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID_TO_CON_NAME(V.CON_ID)) AS CONTAINERNAME, UPPER(V.VALUE)
FROM GV$SYSTEM_PARAMETER V
WHERE UPPER(NAME) = 'SEC_PROTOCOL_ERROR_FURTHER_ACTION'
AND UPPER(VALUE) NOT IN ('(DROP,3)', '(DROP, 3)', 'DROP,3', 'DROP, 3' )
ORDER BY CON_ID;
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

Remediation:

To remediate this setting, execute the following SQL statement.

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

Default Value:

(DROP,3)

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/26/refrn/SEC_PROTOCOL_ERROR_FURTHER_ACTION.html

2.3.10 Ensure 'SEC_PROTOCOL_ERROR_TRACE_ACTION' Is Set To 'LOG' (Automated)

Profile Applicability:

- Level 1 - RDBMS

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 DECODE(V.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID_TO_CON_NAME(V.CON_ID)) AS CONTAINERNAME, UPPER(V.VALUE)
FROM GV$SYSTEM_PARAMETER V
WHERE UPPER(NAME) = 'SEC_PROTOCOL_ERROR_TRACE_ACTION'
AND UPPER(VALUE) != 'LOG'
ORDER BY CON_ID;
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

Remediation:

To remediate this setting, execute the following SQL statement.

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

Note: This parameter is not modifiable at the PDB level. You must modify this parameter at the CDB level.

Default Value:

TRACE

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/26/refrn/SEC_PROTOCOL_ERROR_TRACE_ACTION.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 <u>Collect Audit Logs</u> Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.			
v8	8.5 <u>Collect Detailed Audit Logs</u> Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.			
v7	6.2 <u>Activate audit logging</u> Ensure that local logging has been enabled on all systems and networking devices.			
v7	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.			

2.3.11 Ensure 'SEC_RETURN_SERVER_RELEASE_BANNER' Is Set To 'FALSE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

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 DECODE(V.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID TO_CON_NAME(V.CON_ID)) AS CONTAINERNAME, UPPER(V.VALUE)
FROM GV$SYSTEM_PARAMETER V
WHERE UPPER(NAME) = 'SEC_RETURN_SERVER_RELEASE_BANNER'
AND UPPER(VALUE) != 'FALSE'
ORDER BY CON_ID;
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

Remediation:

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

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

Note: This parameter is not modifiable at the PDB level. You must modify this parameter at the CDB level.

Default Value:

FALSE

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/26/refrn/SEC_RETURN_SERVER_RELEASE_BANNER.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

2.3.12 Ensure 'REMOTE_LOGIN_PASSWORDFILE' Is Set To 'NONE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

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 DECODE (V.CON_ID,
               0, 'ENTIRE-' || SYS_CONTEXT ('USERENV', 'DB_NAME'),
               1, 'ROOTONLY-' || SYS_CONTEXT ('USERENV', 'DB_NAME'),
               CON_ID_TO_CON_NAME (V.CON_ID)) AS CONTAINERNAME,
       UPPER (V.VALUE)
FROM GV$SYSTEM_PARAMETER V
WHERE UPPER (NAME) = 'REMOTE_LOGIN_PASSWORDFILE'
AND (
    ((SELECT COUNT(*) FROM V$ARCHIVE_DEST WHERE STATUS = 'VALID' AND
      TARGET = 'STANDBY') = 0 AND UPPER (VALUE) != 'NONE')
    OR
    ((SELECT COUNT(*) FROM V$ARCHIVE_DEST WHERE STATUS = 'VALID' AND
      TARGET = 'STANDBY') > 0 AND UPPER (VALUE) != 'EXCLUSIVE')
);
```

To be compliant with this recommendation, the output of this command should be `NULL`. Lack of results indicates compliance.

Remediation:

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

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

Default Value:

`EXCLUSIVE`

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/26/refrn/REMOTE_LOGIN_PASSWORDFILE.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.6 <u>Centralize Account Management</u> Centralize account management through a directory or identity service.			
v7	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.			

2.3.13 Ensure 'REMOTE_LISTENER' Is Empty (Automated)

Profile Applicability:

- Level 1 - RDBMS

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:

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 DECODE(V.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID_TO_CON_NAME(V.CON_ID)) AS CONTAINERNAME, UPPER(V.VALUE)
FROM GV$SYSTEM_PARAMETER V
WHERE UPPER(NAME) = 'REMOTE_LISTENER'
AND VALUE IS NOT NULL
ORDER BY CON_ID;
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

Remediation:

To remediate this setting, execute the following SQL statement.

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

Default Value:

There is no default value.

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/26/refrn/REMOTE_LISTENER.html

Additional Information:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software</u> Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.			
v7	<u>9.2 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.			

2.3.14 Ensure 'RESOURCE_LIMIT' Is Set To 'TRUE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

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:

To assess this recommendation, execute the following SQL statement.

```
SELECT DECODE(V.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID_TO_CON_NAME(V.CON_ID)) AS CONTAINERNAME, UPPER(V.VALUE)
FROM GV$SYSTEM_PARAMETER V
WHERE UPPER(NAME) = 'RESOURCE_LIMIT'
AND UPPER(VALUE) != 'TRUE'
ORDER BY CON_ID;
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

Remediation:

To remediate this setting, execute the following SQL statement.

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

Default Value:

FALSE

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/26/refrn/RESOURCE_LIMIT.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

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

Description:

The **FAILED_LOGIN_ATTEMPTS** setting defines the number of consecutive failed login attempts allowed before a user's account is locked. While different profiles can have customized settings based on organizational needs, a minimum recommended value of **5** should be enforced.

Organizations may choose a higher value for specific use cases, such as application data source accounts, to prevent unnecessary lockouts while maintaining security. However, increasing this threshold should be carefully reviewed to ensure that the system's security posture is maintained and accounts are protected against unauthorized or brute-force login attempts.

Rationale:

Repeated failed login attempts may indicate the initiation of a brute-force attack. Therefore, this value should be configured based on the organization's security requirements, balancing account protection with operational needs.

Audit:

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 DECODE(P.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID_TO_CON_NAME(P.CON_ID)) AS CONTAINERNAME,
       P.PROFILE, P.RESOURCE_NAME, P.LIMIT
FROM CDB_PROFILES P
WHERE P.RESOURCE_NAME = 'FAILED_LOGIN_ATTEMPTS'
AND 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
ORDER BY CON_ID, PROFILE, RESOURCE_NAME;
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates 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;
```

Default Value:

10

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/sqlrf/CREATE-PROFILE.html#d387296e851>

Additional Information:

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.

The **SEC_MAX_FAILED_LOGIN_ATTEMPTS** prevents multiple failed login attempts by a single connection. The parameter differs from the limit set on user profiles and applied to failed login attempts to a single user account. Limiting failed authentication attempts by a single connection helps protect against Denial of Service (DoS) attacks and authentication attempts against multiple user accounts.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.2 Establish an Access Revoking Process Establish and follow a process, preferably automated, for revoking access to enterprise assets, through disabling accounts immediately upon termination, rights revocation, or role change of a user. Disabling accounts, instead of deleting accounts, may be necessary to preserve audit trails.			
v7	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

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:

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 CON_ID, DECODE(P.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
                     1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
                     CON_ID_TO_CON_NAME(P.CON_ID)) AS CONTAINERNAME,
       P.PROFILE, P.RESOURCE_NAME, P.LIMIT
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'
ORDER BY CON_ID, PROFILE, RESOURCE_NAME;
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates 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;
```

Default Value:

1

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/sqlrf/CREATE-PROFILE.html#d387296e968>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.2 <u>Establish an Access Revoking Process</u> Establish and follow a process, preferably automated, for revoking access to enterprise assets, through disabling accounts immediately upon termination, rights revocation, or role change of a user. Disabling accounts, instead of deleting accounts, may be necessary to preserve audit trails.			
v7	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.			

3.3 Ensure 'PASSWORD_LIFE_TIME + PASSWORD_GRACE_TIME' Is Less Than Or Equal To '365' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **PASSWORD_LIFE_TIME** setting determines how long a password may be used before the user is required to change it. The suggested value is 365 days or less. Note that recent NIST guidelines recommend using longer lifetimes for human actor passwords.

Rationale:

Allowing passwords to remain unchanged for long periods makes the success of attacks leveraging stolen passwords more likely.

Audit:

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.

```
WITH PWD_LIFE_TIME
AS
    (SELECT CON_ID, DECODE(P.CON_ID,0,'ENTIRE-
'||SYS_CONTEXT('USERENV','DB_NAME'),
    1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
    CON_ID_TO_CON_NAME(P.CON_ID)) AS CONTAINERNAME,
    P.PROFILE, P.RESOURCE_NAME,
    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)) LIMIT
    FROM CDB_PROFILES P
    WHERE P.RESOURCE_NAME = 'PASSWORD_LIFE_TIME'),
PWD_GRACE_TIME AS
    (SELECT CON_ID, DECODE(P.CON_ID,0,'ENTIRE-
'||SYS_CONTEXT('USERENV','DB_NAME'),
    1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
    CON_ID_TO_CON_NAME(P.CON_ID)) AS CONTAINERNAME,
    P.PROFILE, P.RESOURCE_NAME,
    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)) LIMIT
FROM CDB_PROFILES P
WHERE P.RESOURCE_NAME = 'PASSWORD_GRACE_TIME')
SELECT L.CONTAINERNAME, L.PROFILE, L.RESOURCE_NAME, L.LIMIT, G.RESOURCE_NAME,
G.LIMIT
FROM PWD_LIFE_TIME L, PWD_GRACE_TIME G
WHERE L.CON_ID = G.CON_ID
AND L.PROFILE = G.PROFILE
AND L.LIMIT + G.LIMIT > 365
ORDER BY L.CON_ID, L.PROFILE, L.RESOURCE_NAME;

```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates 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 365;
OR
ALTER PROFILE <profile_name> LIMIT PASSWORD_GRACE_TIME <NEW_VALUE>;

```

Default Value:

180

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/sqlrf/CREATE-PROFILE.html#d387296e861>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.2 Establish an Access Revoking Process Establish and follow a process, preferably automated, for revoking access to enterprise assets, through disabling accounts immediately upon termination, rights revocation, or role change of a user. Disabling accounts, instead of deleting accounts, may be necessary to preserve audit trails.			

3.4 Ensure 'PASSWORD_REUSE_MAX' Is Set To 'UNLIMITED' (Automated)

Profile Applicability:

- Level 1 - RDBMS

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 UNLIMITED.

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:

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 DECODE(P.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID_TO_CON_NAME(P.CON_ID)) AS CONTAINERNAME,
       P.PROFILE, P.RESOURCE_NAME, P.LIMIT
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)) < 9999
AND P.RESOURCE_NAME = 'PASSWORD_REUSE_MAX'
ORDER BY CON_ID, PROFILE, RESOURCE_NAME;
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates 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 'UNLIMITED';
```

Default Value:

UNLIMITED

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/sqlrf/CREATE-PROFILE.html#d387296e891>

Additional Information:

The above restriction should be applied along with the **PASSWORD_REUSE_TIME** setting, which should also be set to UNLIMITED.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.2 Use Unique Passwords Use unique passwords for all enterprise assets. Best practice implementation includes, at a minimum, an 8-character password for accounts using MFA and a 14-character password for accounts not using MFA.			
v7	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_VERIFY_FUNCTION' Is Set For All Profiles (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **PASSWORD_VERIFY_FUNCTION** enforces password complexity checks when a database account's password is changed. It should be set for all profiles.

Note that this setting does not apply to administrative roles (like SYSDBA, SYSDB or SYSOPER) authenticated by the Oracle password file. From 12.2, Oracle provides password complexity rules for password file users (Doc 2294754.1)

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:

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 DECODE(P.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID_TO_CON_NAME(P.CON_ID)) AS CONTAINERNAME,
       P.PROFILE, P.RESOURCE_NAME, P.LIMIT
FROM SYS.CDB_PROFILES P
WHERE RESOURCE_NAME='PASSWORD_VERIFY_FUNCTION'
AND P.LIMIT = 'NULL'
ORDER BY CON_ID, PROFILE, RESOURCE_NAME;
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

Remediation:

Use the password verification function, or create a custom password verification function which fulfills the password requirements of the organization.

Oracle supplies two password verification functions with the database **ora12c_verify_function** and **ora12c_strong_verify_function**. You may also create your own function if your organization's standards are different from the functions Oracle supplies. For a sample of a password verification function that you can customize to meet your needs, see **\$ORACLE_HOME/rdbms/admin/catpvf.sql**. In most cases, we recommend that **ora12c_strong_verify_function** be used.

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/sqlrf/CREATE-PROFILE.html#d387296e997>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.2 Use Unique Passwords Use unique passwords for all enterprise assets. Best practice implementation includes, at a minimum, an 8-character password for accounts using MFA and a 14-character password for accounts not using MFA.			
v7	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_VERIFY_FUNCTION' Is Configured Correctly (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The setting **PASSWORD_VERIFY_FUNCTION** is used to enable password complexity verification. If using a custom verification function, it's important to ensure that the complexity function meets your organization's policy for password complexity. Review the code to ensure that the password verification function meets the password complexity rules that have been set for your organization.

Rationale:

Having strong password management for users will protect against attackers' brute force techniques. This is important especially if external authentication is not possible to implement due to application requirements or restrictions.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT DECODE(CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID_TO_CON_NAME(CON_ID)) AS CONTAINERNAME,
       LINE, TEXT
FROM CDB_SOURCE
WHERE (CON_ID,NAME) IN ( SELECT DISTINCT CON_ID, LIMIT
                        FROM SYS.CDB_PROFILES
                        WHERE RESOURCE_NAME='PASSWORD_VERIFY_FUNCTION'
                        AND (LIMIT IS NOT NULL OR LIMIT != 'DEFAULT'))
ORDER BY CON_ID, NAME, LINE ;
```

To be compliant with this recommendation, review the code to ensure that the function meets your organization's password complexity requirements..

Remediation:

If you discover profiles using verification functions that do not meet your organization's standards, you may remediate this setting by executing the following SQL statement for each **PROFILE** returned by the audit procedure. This will cause accounts assigned those profiles to use the **ora12c_strong_verify_function**. The next time that account's password is changed, the new function will validate that they meet standards.

```
ALTER PROFILE <profile_name> LIMIT
PASSWORD_VERIFY_FUNCTION ORA12C_STRONG_VERIFY_FUNCTION;
```

Default Value:

NULL

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-authentication.html#GUID-F48598E4-2D72-4A3B-8904-AD6D2C67D715>
2. <https://docs.oracle.com/en/database/oracle/oracle-database/26/sqlrf/CREATE-PROFILE.html#d387296e997>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.2 Use Unique Passwords Use unique passwords for all enterprise assets. Best practice implementation includes, at a minimum, an 8-character password for accounts using MFA and a 14-character password for accounts not using MFA.			
v7	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.			

3.7 Ensure 'PASSWORD_ROLLOVER_TIME' Is set to '0' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

This setting PASSWORD_ROLLOVER_TIME enables the gradual database password rollover time allowing both the new and old password to be used for a set period of time after a password is changed. This capability is not normally appropriate for human actors, but is frequently valuable for application and batch processing service accounts.

Rationale:

With the need to change passwords at some frequency and the goal to limit downtime for applications operating across multiple tiers it is necessary to have an overlap period where both the new and old passwords are accepted. This allows time for all tiers supporting the application to be updated to use the new password.

Impact:

Configuring this setting could result in application account lockouts which may impact the service(s) provided by the application. To minimize the impact, you may set this setting to a sufficiently higher value for you to update applications and batch processes to use new passwords after a password change.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT DECODE(P.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID_TO_CON_NAME(P.CON_ID)) AS CONTAINERNAME,
       P.PROFILE, P.RESOURCE_NAME, P.LIMIT
FROM SYS.CDB_PROFILES P
WHERE TO_NUMBER(DECODE(P.LIMIT,'DEFAULT',(SELECT LIMIT
                                         FROM CDB_PROFILES
                                         WHERE PROFILE='DEFAULT'
                                         AND RESOURCE_NAME='PASSWORD_ROLLOVER_TIME'
                                         AND CON_ID = P.CON_ID),
                    P.LIMIT)) > 0
AND RESOURCE_NAME='PASSWORD_ROLLOVER_TIME'
ORDER BY CON_ID, PROFILE, RESOURCE_NAME;
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

Remediation:

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

```
ALTER PROFILE DEFAULT LIMIT PASSWORD_ROLLOVER_TIME 0;
```

Default Value:

0 or NULL

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-authentication.html#d23067e2021>

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

Profile Applicability:

- Level 1 - RDBMS

Description:

The **INACTIVE_ACCOUNT_TIME** parameter in Oracle specifies the number of days an account can remain inactive before it is automatically locked. This setting helps mitigate security risks by reducing the exposure of unused accounts, which could be exploited for unauthorized access.

Rationale:

Inactive user accounts pose security risks as they can be potentially targeted by the attackers for unauthorized access. Setting a reasonable threshold for inactivity helps enforce security best practices by ensuring that unused accounts are locked before they can be potentially targeted for unauthorized access. A value of **120** days or less strikes a balance between security and operational flexibility.

Impact:

If an account is locked due to inactivity, administrators may need to manually unlock it when required. Organizations should communicate this policy to users to avoid disruptions.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT DECODE(P.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID_TO_CON_NAME(P.CON_ID)) AS CONTAINERNAME,
       P.PROFILE, P.RESOURCE_NAME, P.LIMIT
FROM SYS.CDB_PROFILES P
WHERE RESOURCE_NAME='INACTIVE_ACCOUNT_TIME'
AND TO_NUMBER(DECODE(P.LIMIT,
                     'DEFAULT',(SELECT 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
ORDER BY CON_ID, PROFILE, RESOURCE_NAME;
```

To be compliant with this recommendation, the output of this command should be **NULL**. Lack of results indicates compliance.

Remediation:

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

```
ALTER PROFILE DEFAULT LIMIT INACTIVE_ACCOUNT_TIME 120
```

Default Value:

The default value may vary based on Oracle versions and configurations.

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-authentication.html#d23067e2021>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.3 <u>Disable Dormant Accounts</u> Delete or disable any dormant accounts after a period of 45 days of inactivity, where supported.			
v7	16.9 <u>Disable Dormant Accounts</u> 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

Description:

Default passwords should not be used by Oracle database users. Leaving default passwords unchanged can create significant security vulnerabilities by allowing unauthorized access to the database.

Rationale:

Default passwords are widely known and documented. If left unchanged, any malicious user with database access can authenticate using these default credentials, potentially leading to unauthorized data access, privilege escalation, or database compromise.

Audit:

To check for accounts still using default passwords, execute the following SQL statement:

```
SELECT DECODE(C.CON_ID,0,'ENTIRE-'||SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID_TO_CON_NAME(C.CON_ID)) AS CONTAINERNAME,
       A.USERNAME, C.ACCOUNT_STATUS
FROM CDB_USERS_WITH_DEFPWD A, CDB_USERS C
WHERE A.USERNAME = C.USERNAME
ORDER BY C.CON_ID;
```

Lack of results implies compliance.

Remediation:

To reset the account with **NO AUTHENTICATION**, execute the following SQL statement in CDB:

```
ALTER USER <USERNAME> NO AUTHENTICATION;
```

Notes:

- As per Oracle Support Document 2173962.1, newly created database may list **SYS** and **SYSTEM** in **CBA_USERS_WITH_DEFPWD** even if they were set with non-default passwords. Running **ALTER USER** with the same password will correctly recognize these accounts as non-default.
- If **remote_password_file** is set to **NONE**, changing the **SYS** password via **ALTER USER** is not possible. In this case, update **remote_password_file** to **EXCLUSIVE** before modifying the **SYS** password.

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/keeping-your-oracle-database-secure.html>
2. <https://support.oracle.com/epmos/faces/DocumentDisplay?id=2173962.1>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>4.7 Manage Default Accounts on Enterprise Assets and Software</u> Manage default accounts on enterprise assets and software, such as root, administrator, and other pre-configured vendor accounts. Example implementations can include: disabling default accounts or making them unusable.			
v7	<u>4.2 Change Default Passwords</u> Before deploying any new asset, change all default passwords to have values consistent with administrative level accounts.			

4.2 Ensure No Custom 'ORACLE_MAINTAINED' Users Exist (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

Oracle-maintained accounts should not be created. The **ORACLE_MAINTAINED** flag is used to identify such accounts. Only users provided by Oracle should have this flag set to **Y**. Custom database users should not be assigned this designation.

Rationale:

Oracle-maintained accounts are used by internal tools, database packages, procedures, and third-party software to make critical decisions. For example:

- **Data Pump** often excludes objects marked as **ORACLE_MAINTAINED**.
- **Oracle upgrades and patches** may modify or delete objects marked with this flag.
- **Custom schemas** should not use this flag, as it can lead to unintended system behavior, including unauthorized modifications during maintenance activities.

Some organizations attempt to create custom **ORACLE_MAINTAINED** users to bypass the **C##-prefix requirement** for common users. However, misusing this flag can introduce security risks and operational issues. It can impact proper functioning and integrity of Oracle databases.

Impact:

Dropping a user may result in data loss or impact application availability, requiring thorough assessment before taking action. You may need to export all objects owned by common users in the CDB and all PDBs before making changes.

```
DROP USER <USERNAME>;
```

Audit:

To assess this recommendation, execute the following SQL statement:

```
SELECT CON_ID_TO_CON_NAME(C.CON_ID) AS CONTAINERNAME, USERNAME,
FROM CDB_USERS C
WHERE ORACLE_MAINTAINED = 'Y'
AND USERNAME NOT IN (
    'ANONYMOUS', 'APEX_LISTENER', 'APEX_PUBLIC_USER', 'APEX_REST_PUBLIC_USER',
    'APEX_230200', 'APEX_240100', 'APEX_PUBLIC_ROUTER', 'APPQOSSYS', 'AUDSYS',
    'CTXSYS', 'DBSFUSER', 'DBSNMP', 'DGPDB_INT', 'DIP', 'DVF', 'DVSYS',
    'FLOWS_FILES', 'GGSHARED CAP', 'GGSYS', 'GSMADMIN_INTERNAL',
    'GSMCATUSER',
```

```
'GSMROOTUSER', 'GSMUSER', 'LBACSYS', 'MDDATA', 'MDSYS', 'OJVM SYS',
'OLAPSYS',
'OUTLN', 'REMOTE_SCHEDULER_AGENT', 'SYS', 'SYSBACKUP', 'SYSDG',
'SYSKM',
'SYSRAC', 'SYSTEM', 'SYS$UMF', 'VECSYS', 'WMSYS', 'XDB', 'XS$NULL'
)
ORDER BY 1;
```

A lack of results implies compliance.

Remediation:

To remediate this setting, perform the following steps:

```
<Export the User>
DROP USER <USERNAME>;
<Recreate the user as non-Oracle-maintained user using import utilities>
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.5 <u>Establish and Maintain an Inventory of Service Accounts</u> Establish and maintain an inventory of service accounts. The inventory, at a minimum, must contain department owner, review date, and purpose. Perform service account reviews to validate that all active accounts are authorized, on a recurring schedule at a minimum quarterly, or more frequently.			
v7	16.6 <u>Maintain an Inventory of Accounts</u> Maintain an inventory of all accounts organized by authentication system.			

4.3 Review The Users Created Through Real Application Security (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

Oracle Real Application Security (RAS) introduces an alternative user model that allows users to log in directly (e.g., with SQL*Plus) when **DIRECT_LOGON = YES**. RAS users do not appear in the **DBA_USERS** or **CDB_USERS** views and are not created using the standard **CREATE USER** command, making them difficult to detect. To ensure proper security monitoring, the **CDB_XS_USERS** view should be reviewed regularly.

Rationale:

RAS users are not visible in traditional **DBA_USERS** or **CDB_USERS** views. Privileges for RAS users are stored and displayed separately from standard database users. Without regular monitoring, unauthorized or unintended RAS users may remain undetected. Security assessments must include RAS-related views, such as **CDB_XS_USERS** and **DBA_XS_USERS**, to analyze users and their authorizations.

Impact:

Dropping a RAS user may result in data loss or impact application availability, requiring thorough assessment before taking action. Some applications or scripts may rely on RAS users for specific tasks. Removing it may break these applications, requiring modifications or redesign.

Audit:

To assess this recommendation, execute the following statement:

```
SELECT DECODE(C.CON_ID,0,'ENTIRE-'|| SYS_CONTEXT('USERENV','DB_NAME'),
              1,'ROOTONLY-'||SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID TO CON_NAME(C.CON_ID)) AS CONTAINERNAME,
       C.NAME, C.DIRECT_LOGON_USER, C.STATUS, ACCOUNT_STATUS
FROM CDB_XS_USERS C
WHERE C.NAME NOT IN ('XSGUEST');
```

Review the list of RAS users. If any unauthorized user exists, this is a fail.

Remediation:

Remove RAS-Users that are not needed.

```
BEGIN
  SYS.XS_PRINCIPAL.DELETE_PRINCIPAL('<rasuser>');
END;
```

Default Value:

Empty

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbfsg/predefined-objects-in-real-application-security.html#GUID-7B011292-C771-4B88-8CFA-0CA30782773C>
2. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbfsg/real-application-security-hr-demo.html#GUID-FA992EC6-83D5-4465-86D8-334EAA195C41>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.5 <u>Establish and Maintain an Inventory of Service Accounts</u> Establish and maintain an inventory of service accounts. The inventory, at a minimum, must contain department owner, review date, and purpose. Perform service account reviews to validate that all active accounts are authorized, on a recurring schedule at a minimum quarterly, or more frequently.			
v7	16.6 <u>Maintain an Inventory of Accounts</u> Maintain an inventory of all accounts organized by authentication system.			

4.4 Ensure Old Password Versions Are Not Used (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

No password versions (hashes) prior to **12c** should be allowed for user authentication.

Rationale:

Oracle 12c and later versions enforce stronger password hashing algorithms and complexity policies, enhancing security. Older password versions (pre-12c) are considered weak and do not meet modern security standards. Using outdated password versions increases the risk of password compromise.

Impact:

Resetting user passwords without proper migration planning may result in application downtime or loss of access.

Users with passwords hashed in an older format will be required to reset their passwords. Limiting authentication to 12c password versions may impact connectivity from older clients that rely on deprecated authentication mechanisms. Applications using pre-12c authentication methods must be updated to support 12c or later password hashing mechanisms.

This checks for all users except SYS because SYS either uses bequeath (no password) for local connections or the remote login password file for remote connections. Strength of the remote login password file is examined in recommendation 4.5. Non SYS users have the option of either logging in normally (via **SYS.USER\$**) or with administrative rights (with **SYSDBA/SYSOPER/...**).

Audit:

To assess this recommendation, execute the following SQL statement:

```
SELECT DECODE(C.CON_ID,
              0, 'ENTIRE-' || SYS_CONTEXT('USERENV','DB_NAME'),
              1, 'ROOTONLY-' || SYS_CONTEXT('USERENV','DB_NAME'),
              CON_ID_TO_CON_NAME(C.CON_ID)) AS CONTAINERNAME,
       C.USERNAME,
       C.PASSWORD_VERSIONS
FROM CDB_USERS C
WHERE NOT REGEXP_LIKE(C.PASSWORD_VERSIONS, '^s*12C\s*$')
and username not in ('SYS')
order by USERNAME;
```

A lack of results implies compliance.

Remediation:

Follow the process "Finding and Resetting User Passwords That Use the 10G Password Version" as outlined in the Oracle Database Upgrade Guide to reset affected user passwords and enforce 12c password versioning.

- Identify users with old password versions.
- Ensure that `SQLNET.ALLOWED_LOGON_VERSION_SERVER` is set to 12a.
- Reset their passwords using the `ALTER USER` command.

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/upgrd/recommended-and-best-practices-complete-upgrading-oracle-database.html#GUID-D7B09DFE-F55D-449A-8F8A-174D89936304>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 <u>Encrypt Sensitive Data at Rest</u> Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●
v7	14.8 <u>Encrypt Sensitive Information at Rest</u> Encrypt all sensitive information at rest using a tool that requires a secondary authentication mechanism not integrated into the operating system, in order to access the information.			●

4.5 Ensure The Latest Version of The Password File Is Used (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The database password file is used to authenticate users with administrative privileges (**SYSDBA**, **SYSOPER**, etc.) from a remote machine. It enables secure remote database management and contains a list of privileged users with their corresponding encrypted passwords.

Starting with Oracle 12.2, Oracle enforces **stronger password hashing algorithms** in password files to improve security. It is recommended to ensure that the password file format is updated to **12.2** to leverage enhanced security mechanisms.

Rationale:

Ensuring that the password file format is 12.2 provides the following benefits:

- Enhanced security: Passwords are stored using a stronger hashing algorithm.
- Compliance with best practices: The latest format ensures compatibility with the recommended security settings.
- Ability to grant administrative privileges (**SYSOPER**, **SYSBACKUP**, etc.) to global users securely.

Impact:

In environments using **ASM**, **RAC**, or **RMAN**, upgrading the password file format may require additional considerations.

Existing users in the password file must already have passwords that comply with **12.2 format** before upgrading.

Any changes to the password file format should be thoroughly tested in a non-production environment before implementation.

Audit:

To assess this recommendation, execute the following statement:

```
SELECT inst_id, format
FROM GV$PASSWORDFILE_INFO
WHERE FORMAT != '12.2';
```

A lack of results implies compliance.

Remediation:

To update the password file to the latest format (12.2), follow these steps:

- Ensure that all users in the password file have passwords meeting 12.2 complexity requirements.
- Use the **orapwd** utility to create a new password file in 12.2 format:

```
orapwd file=orapwd122 FORMAT=12.2
```

- If the password file name or location has recently changed and the changes are not reflected, refresh the metadata cache:

```
ALTER SYSTEM FLUSH PASSWORDFILE_METADATA_CACHE;
```

- Validate the password file format using the audit query again.

Default Value:

12.2

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/26/refrn/V-PASSWORDFILE_INFO.html#GUID-726CF0F6-14A4-465B-B46B-E2AC8CFFD27A

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 Encrypt Sensitive Data at Rest Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●
v7	14.8 Encrypt Sensitive Information at Rest Encrypt all sensitive information at rest using a tool that requires a secondary authentication mechanism not integrated into the operating system, in order to access the information.			●

4.6 Ensure That Users In Different RAC Instances Are Identical In PW Files (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

In Oracle Real Application Clusters (RAC) environments, each instance maintains its own password file for authenticating users with administrative privileges (**SYSDBA**, **SYSOPER**, **SYSBACKUP**, etc.). If password files are not synchronized across all RAC nodes, certain administrative users may exist on only one node, causing authentication failures when attempting to log in from other nodes.

To ensure seamless remote administration and high availability, all RAC instances must have identical password files.

Rationale:

Maintaining identical password files across all RAC nodes ensures consistent authentication for administrative users across all nodes and remote management of the database without login failures.

Impact:

If password files are not synchronized, privileged users may be unable to authenticate on certain RAC nodes.

Audit:

To assess this recommendation, execute the following SQL statement:

```
select username
from GV$PWFILERS_USERS
group by username
having count(distinct inst_id) < (select count(distinct inst_id) from
GV$PWFILERS_USERS);
```

If any users are not identical across RAC Instances, this is a fail.

Remediation:

After changes to the password file, all nodes should be synchronized.

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/26/refrn/V-PWFILERS_USERS.html#GUID-C8D29599-13BD-493A-BE23-0F16BDDA7725

4.7 Ensure No Public Database Links Exist (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

Public database links provide a mechanism for establishing connections between databases, enabling data access across multiple environments.

Rationale:

Public database links pose a security risk because they allow any database user to execute queries or modify data on a remote database, depending on the credentials stored in the link. This can lead to unauthorized access, data breaches, and compliance violations.

Impact:

Applications relying on public database links may fail after removal. Alternative connection methods (e.g., private database links) may need to be implemented.

Audit:

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 DECODE(V.CON_ID, 0, 'ENTIRE-' || SYS_CONTEXT('USERENV','DB_NAME'),
                  1, 'ROOTONLY-' || SYS_CONTEXT('USERENV','DB_NAME'),
                  CON_ID_TO_CON_NAME(V.CON_ID)) AS CONTAINERNAME,
       DB_LINK, HOST
FROM CDB_DB_LINKS V
WHERE OWNER = 'PUBLIC';
```

Lack of results implies compliance.

Remediation:

To remove a public database link, execute the following SQL command:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

4.8 Ensure That Database Link Passwords Are Using The Latest Encryption (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

In Oracle databases prior to version 26ai, encrypted passwords for database links are stored in the **PASSWORDX** column, which can be decrypted. Oracle 26ai enhances security by storing encrypted database link passwords in the **SPARE1** column, making decryption significantly more difficult.

Rationale:

Using the latest encryption method for database link passwords reduces the risk of credential exposure. Attackers who gain access to **PASSWORDX** can potentially decrypt and misuse stored credentials for unauthorized access.

Impact:

If a database link password is stored using an older encryption method, it could be decrypted and exploited, posing a security risk.

Audit:

To verify if database link passwords are still using the **PASSWORDX** column for encryption, execute the following SQL statement:

```
SELECT DECODE(v.con_id, 0, 'Entire-' || SYS_CONTEXT('USERENV', 'DB_NAME'),
               1, 'RootOnly-' || SYS_CONTEXT('USERENV', 'DB_NAME'),
               CON_ID_TO_CON_NAME(v.con_id)) AS container_name,
       name AS db_link,
       host,
       userid
FROM CONTAINERS("SYS"."LINK$") v
WHERE passwordx IS NOT NULL;
```

Lack of results implies compliance.

Remediation:

After upgrading to Oracle 26ai, drop and recreate the database link to ensure that passwords are encrypted using the latest method.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 <u>Encrypt Sensitive Data at Rest</u> Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.			
v7	14.8 <u>Encrypt Sensitive Information at Rest</u> Encrypt all sensitive information at rest using a tool that requires a secondary authentication mechanism not integrated into the operating system, in order to access the information.			

5 Unified Auditing

The ability to audit database activity is one of the most important security features of databases. Decisions need to be made about the scope of the audit, as the audit incurs costs - for the storage of the audit log and for the performance impact on the audited operations - and perhaps even for the database or system in general. Then there are the additional costs of managing (storing, backing up, securing) and reviewing the data in the audit trail.

Enabling auditing in an Oracle database may introduce performance overhead, particularly in high-transaction environments. Depending on the volume of audited activities, it can lead to increased storage usage, slower query execution, and potential contention for system resources. In extreme cases, improper audit configurations may contribute to database instability or unexpected service disruptions.

Before enabling auditing, it is strongly recommended to perform thorough testing in a non-production environment, ensure sufficient storage allocation, and monitor performance impact. Additionally, review Oracle's documentation and best practices to implement auditing in a controlled and efficient manner.

This section deals with Oracle Unified Auditing, as from Oracle 26ai only this type of auditing is supported by Oracle.

If the Oracle database was upgraded from Oracle 19c to Oracle 26ai, traditional auditing can still exist, but no new audit rules can be added.

There are 2 types of policies under Oracle Unified Auditing:

- Common Audit Policies
- Local Audit Policies

The Common Audit Policy in the CDB is defined with the option **CONTAINER=ALL** and applies automatically in the CDB and all PDBs. A common audit policy applies to all containers. When you create a common audit policy, prefix the name with **C##** or **c##** (for example, **c##all_select_pol**). This policy can only contain actions, system privileges, common roles, and common objects. You can apply a common audit policy only to common users. Attempts to enforce a common audit policy for a local user across all containers result in an error.

Local audit policies must be installed in the CDB as well as in each PDB and then apply to all types of database users (local and common). A local audit policy applies to a single PDB. You can enforce local audit policies for local and common users in this PDB only. Attempts to enforce local audit policies across all containers result in an error. In order to log all user activities, a local audit policy must therefore be installed in the CDB and each PDB.

Depending on the application purpose, the unified audit trail can either be read in each PDB or the common unified audit trail of all PDBs and the CDB together from the CDB.

As of 26ai, audit data can be collected from the following sources with corresponding Oracle Unified Audit Policies:

- Audit records (including SYS* audit records) from unified audit policies and AUDIT settings
- Fine-grained audit records from the **DBMS_FGA PL/SQL** package
- Oracle Database Real Application Security audit records
- Oracle Recovery Manager audit records
- Oracle Database Vault audit records
- Oracle Label Security audit records
- Oracle Machine Learning for SQL records
- Oracle Data Pump
- Oracle **SQL*Loader** Direct Load
- Oracle **XML DB HTTP** and **FTP** protocol messages

The following audit data is collected by default in the following section:

- Oracle Mandatory Auditing
- Oracle DDL, System
- Oracle Control
- System Privileges
- **SYSDBA**, **SYSRAC**, **SYSDG**, **SYSKM** and **SYSOPER** activities
- Oracle Logon / Logoff
- Oracle Datapump
- Powerful Oracle Packages

The following query can be used to query all enabled unified audit policies.

```
select CON_ID TO_CON_NAME(c.con_id) as containername, c.* from  
containers(AUDIT_UNIFIED_ENABLED_POLICIES) c;
```

5.1 Ensure All Auditable System Actions Commands Are Audited (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

Oracle defines (*) the various command categories such as **DDL**, **DML**, **TCL**, etc. partly deviating from the ANSI standard.

However, these command categories are incomplete and only represent a part of the auditable commands.

For this reason, the term “Auditable System Actions” is used in the following. This type of definition allows a flexible and future-oriented possibility to add new commands and commands that will be added to future patch sets.

According to the Oracle definition* of Data Definition Language (DDL) statements, these tasks can be performed by definition:

- Create, modify and delete schema objects
- Granting and revoking privileges and roles
- Analyze information about a table, index or cluster
- Setting up auditing options
- Adding comments to the data dictionary

<https://docs.oracle.com/en/database/oracle/oracle-database/23/sqlrf/Types-of-SQL-Statements.html>

Enabling this unified action audit causes logging of all **DDL** commands, 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 perform these **DDL** commands, 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 these **DDL** commands.

Impact:

Auditing all **DDL** can result in rapid growth of the audit trail, particularly in active or dynamic environments. This may lead to storage management challenges and require regular maintenance (e.g., purging or archiving old audit records). The audit trail may consume excessive space, leading to issues such as system slowdowns or errors when storage limits are reached if it is not carefully managed.

Audit:

To assess this recommendation, execute the following SQL statement:

```
WITH PRIVS(PRIVILEGE) AS (
  SELECT DISTINCT A.PRIVILEGE AS NAME FROM DBA_SYS_PRIVS A
    WHERE A.PRIVILEGE NOT IN (SELECT NAME FROM AUDITABLE_SYSTEM_ACTIONS WHERE
      COMPONENT='Standard')
    AND PRIVILEGE NOT IN ('INHERIT ANY PRIVILEGES')
  UNION
    SELECT A.NAME FROM AUDITABLE_SYSTEM_ACTIONS A
      WHERE COMPONENT='Standard'
      AND A.NAME NOT IN
('ALL', 'SELECT', 'DELETE', 'INSERT', 'UPDATE', 'EXECUTE', 'LOGON', 'LOGOFF')
),
CIS_AUDIT(CONTAINERNAME, CON_ID, AUDIT_OPTION) AS (
  SELECT C.NAME, C.CON_ID, LL.PRIVILEGE FROM V$CONTAINERS C, PRIVS LL
    WHERE C.OPEN_MODE='READ WRITE'
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUD.CON_ID, AUDIT_OPTION
  FROM CONTAINERS(AUDIT_UNIFIED_POLICIES) AUD
    WHERE AUD.AUDIT_OPTION IN (SELECT PRIVILEGE FROM PRIVS )
    AND AUD.AUDIT_OPTION_TYPE IN ('SYSTEM PRIVILEGE', 'STANDARD ACTION')
    AND EXISTS (SELECT ENABLED.*
      FROM CONTAINERS(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
          AND ENABLED.CON_ID = AUD.CON_ID)
)
SELECT C.CONTAINERNAME, C.AUDIT_OPTION
FROM CIS_AUDIT C
WHERE NOT EXISTS (
  SELECT 1 FROM AUDIT_ENABLED E WHERE E.AUDIT_OPTION = C.AUDIT_OPTION AND
  E.CON_ID = C.CON_ID
)
ORDER BY 1, 2;
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this recommendation:

```
ALTER AUDIT POLICY CIS_LOCAL_SYSTEM_ACTIONS
```

```
ADD  
ACTIONS  
<DDL>;
```

or

```
ALTER AUDIT POLICY CIS_LOCAL_SYSTEM_ACTIONS  
ADD  
PRIVILEGES  
<DDL>;
```

Note: If you do not have **CIS_CDB_DDL_ACTIONS** or **CIS_PDB_DDL_ACTIONS** policy, please create one using the **CREATE AUDIT POLICY** statement. Refer to Section 8.1 where a PL/SQL block is provided to help create or modify the audit policy to remediate this item in both container and pluggable database.

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/sqlrf/Types-of-SQL-Statements.html#GUID-FD9A8CB4-6B9A-44E5-B114-EFB8DA76FC88>
2. <https://docs.oracle.com/en/database/oracle/oracle-database/26/sqlrf/AUDIT-Unified-Auditing.html>

Additional Information:

Check for **EXCEPTION**, as it may disable auditing for certain users.

```
SELECT POLICY_NAME, ENABLED_OPTION, ENTITY_NAME  
FROM AUDIT_UNIFIED_ENABLED_POLICIES  
WHERE ENABLED_OPTION = 'EXCEPT USER';
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 <u>Collect Detailed Audit Logs</u> Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.			
v7	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.			

5.2 Ensure the 'LOGON' AND 'LOGOFF' Actions Audit Is Enabled (Automated)

Profile Applicability:

- Level 1 - RDBMS

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** and **LOGOFF BY CLEANUP** 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**, **LOGOFF** and **LOGOFF BY CLEANUP**.

Audit:

To assess this recommendation, execute the following SQL statement.

```
WITH
CIS_AUDIT(CONTAINERNAME, CON_ID, AUDIT_OPTION) AS (
SELECT C.NAME, C.CON_ID, LL.COLUMN_VALUE AS AUDIT_OPTION FROM V$CONTAINERS C,
TABLE(DBMSOUTPUT_LINESARRAY('LOGON','LOGOFF')) LL
WHERE C.OPEN_MODE='READ WRITE'
),
AUDIT_ENABLED AS
( SELECT DISTINCT AUD.CON_ID, AUDIT_OPTION
  FROM CONTAINERS(AUDIT_UNIFIED_POLICIES) AUD
  WHERE AUD.AUDIT_OPTION IN ('LOGON','LOGOFF')
    AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
    AND EXISTS (SELECT ENABLED.*
                FROM CONTAINERS(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
                  AND ENABLED.CON_ID = AUD.CON_ID)
)
SELECT C.CONTAINERNAME, C.AUDIT_OPTION
FROM CIS_AUDIT C
WHERE NOT EXISTS (
```

```

SELECT 1 FROM AUDIT_ENABLED E WHERE E.AUDIT_OPTION = C.AUDIT_OPTION AND
E.CON_ID = C.CON_ID
)
ORDER BY 1, 2;

```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this setting.

Run the following query in the CDB and in each PDB:

```

CREATE AUDIT POLICY CIS_CDB_LOGON_LOGOFF
ACTIONS
LOGON, LOGOFF
ACTIONS
COMPONENT=PROTOCOL HTTP, FTP, AUTHENTICATION;

AUDIT POLICY CIS_CDB_LOGON_LOGOFF;

```

Note: If you do not have **CIS_CDB_LOGON_LOGOFF**, please create one using the **CREATE AUDIT POLICY** statement. Refer to Section 8.2 where a PL/SQL block is provided to help create or modify the audit policy to remediate this item in both container and pluggable database.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.			
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.			
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.			
v7	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.			

5.3 Ensure Critical Packages Are Audited (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

Auditing critical packages in Oracle is essential to ensure database security, maintain accountability, and support compliance with regulatory requirements. These packages provide powerful functionality, such as scheduling jobs, performing cryptographic operations, or handling web-based utilities where, if misused or exploited, can compromise the integrity, confidentiality, and availability of the database.

Auditing these critical packages is recommended; however, it should be enabled based on the specific needs and requirements of the organization:

- `SYS.DBMS_AW`
- `SYS.DBMS_CRYPTO`
- `SYS.DBMS_FGA`
- `SYS.DBMS_JAVA_TEST`
- `SYS.DBMS_JOB`
- `SYS.DBMS_LOGMNR`
- `SYS.DBMS_NETWORK_ACL_ADMIN`
- `SYS.DBMS_REDACT`
- `SYS.DBMS_REDEFINITION`
- `SYS.DBMS_RLS`
- `SYS.DBMS_SCHEDULER`
- `SYS.DBMS_SQL_TRANSLATOR`
- `SYS.DBMS_SYS_SQL`
- `SYS.DBMS_TSDP_MANAGE`
- `SYS.DBMS_TSDP_PROTECT`
- `SYS.DBMS_XMLGEN`
- `SYS.DBMS_XMLSTORE`
- `SYS.OWA_UTIL`

Rationale:

Auditing these packages helps monitor their usage, detect unauthorized access, and prevent potential misuse or security breaches. This practice also supports incident response and forensic investigations by maintaining a detailed record of package invocation, ensuring that critical operations are tracked and controlled in high-security environments.

Impact:

Auditing system packages generate additional logs, which can impact system performance, especially in high-transaction environments. Auditing can introduce slight delays in query execution due to the extra logging steps.

Audit:

To assess this recommendation, execute the following SQL statement.

```
WITH
CIS_AUDIT(NAME, CON_ID, AUDIT_OBJECT_NAME) AS (
SELECT C.NAME, C.CON_ID, LL.* FROM V$CONTAINERS C,
TABLE(DBMSOUTPUT_LINESARRAY('DBMS_AW','DBMS_CRYPTO','DBMS_FGA',
'DBMS_JAVA_TEST','DBMS_JOB','DBMS_LOGMNR',
'DBMS_NETWORK_ACL_ADMIN','DBMS_REDACT','DBMS_REDEFINITION','DBMS_RLS',
'DBMS_SCHEDULER','DBMS_SQL_TRANSLATOR','DBMS_SYS_SQL','DBMS_TSDP_MANAGE',
'DBMS_TSDP_PROTECT','DBMS_XMLGEN','DBMS_XMLSTORE','OWA_UTIL'))
) LL WHERE C.OPEN_MODE='READ WRITE'
),
AUDIT_ENABLED AS
( SELECT DISTINCT CON_ID_TO_CON_NAME(AUD.CON_ID) AS CONTAINERNAME,
AUDIT_OPTION,
AUD.OBJECT_NAME, AUD.POLICY_NAME, AUD.CON_ID
FROM CONTAINERS(AUDIT_UNIFIED_POLICIES) AUD
WHERE AUD.OBJECT_NAME IN
('DBMS_AW','DBMS_CRYPTO','DBMS_FGA','DBMS_JAVA_TEST','DBMS_JOB','DBMS_LOGMNR'
,
'DBMS_NETWORK_ACL_ADMIN','DBMS_REDACT','DBMS_REDEFINITION','DBMS_RLS',
'DBMS_SCHEDULER','DBMS_SQL_TRANSLATOR','DBMS_SYS_SQL','DBMS_TSDP_MANAGE',
'DBMS_TSDP_PROTECT','DBMS_XMLGEN','DBMS_XMLSTORE','OWA_UTIL')
AND AUD.AUDIT_OPTION_TYPE = 'OBJECT ACTION'
AND EXISTS (SELECT CON_ID_TO_CON_NAME(ENABLED.CON_ID) AS CONTAINERNAME,
ENABLED.*
FROM CONTAINERS(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
AND ENABLED.CON_ID = AUD.CON_ID)
)
SELECT C.NAME, C.AUDIT_OBJECT_NAME
FROM CIS_AUDIT C
WHERE NOT EXISTS (
SELECT 1 FROM AUDIT_ENABLED E WHERE E.OBJECT_NAME = C.AUDIT_OBJECT_NAME AND
E.CON_ID = C.CON_ID
)
ORDER BY 1, 2;
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement to remediate this recommendation:

```

CREATE AUDIT POLICY CIS_CDB_CRITICAL_PACKAGES
ACTIONS
EXECUTE ON SYS.DBMS_AW,
EXECUTE ON SYS.DBMS_CRYPTO,
EXECUTE ON SYS.DBMS_FGA,
EXECUTE ON SYS.DBMS_JAVA_TEST,
EXECUTE ON SYS.DBMS_JOB,
EXECUTE ON SYS.DBMS_LOGMNR,
EXECUTE ON SYS.DBMS_NETWORK_ACL_ADMIN,
EXECUTE ON SYS.DBMS_REDACT,
EXECUTE ON SYS.DBMS_REDEFINITION,
EXECUTE ON SYS.DBMS_RLS,
EXECUTE ON SYS.DBMS_SCHEDULER,
EXECUTE ON SYS.DBMS_SQL_TRANSLATOR,
EXECUTE ON SYS.DBMS_SYS_SQL,
EXECUTE ON SYS.DBMS_TSDP_MANAGE,
EXECUTE ON SYS.DBMS_TSDP_PROTECT,
EXECUTE ON SYS.DBMS_XMLGEN,
EXECUTE ON SYS.DBMS_XMLSTORE,
EXECUTE ON SYS.OWA_UTIL
ONLY TOPLEVEL;

AUDIT POLICY CIS_CDB_CRITICAL_PACKAGES;

```

Note: If you do not have **CIS_CDB_CRITICAL_PACKAGES**, please create one using the **CREATE AUDIT POLICY** statement. Refer to Section 8.3 where a PL/SQL block is provided to help create or modify the audit policy to remediate this item in both container and pluggable database.

Default Value:

These packages are not audited by default.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 <u>Collect Detailed Audit Logs</u> Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.			
v7	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.			

5.4 Ensure All Export Activities Are Audited (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

Auditing all export activities in Oracle is crucial for several security and compliance reasons such as data leakage prevention, accountability and traceability as well as compliance with regulations.

Rationale:

Auditing export activities helps to detect unauthorized use of export tools or **RMAN** to exfiltrate sensitive data during a security breach. Comprehensive auditing of these activities can help detect and mitigate such threats promptly. By implementing detailed auditing for export and **RMAN** activities, organizations can strengthen their security posture, safeguard sensitive data, and ensure compliance with regulatory and internal policies.

Impact:

Auditing export operations can introduce additional I/O and CPU usage, especially in large-volume exports.

Audit:

To assess this recommendation, execute the following SQL statement.

```
WITH
CIS_AUDIT(NAME, CON_ID, AUDIT_OPTION) AS (
SELECT C.NAME, C.CON_ID, 'EXPORT' AS AUDIT_OPTION FROM V$CONTAINERS C WHERE
C.OPEN_MODE='READ WRITE'
),
AUDIT_ENABLED AS
( SELECT DISTINCT CON_ID_TO_CON_NAME(AUD.CON_ID) AS CONTAINERNAME,
AUDIT_OPTION, AUD.POLICY_NAME, AUD.CON_ID
FROM CONTAINERS(AUDIT_UNIFIED_POLICIES) AUD
WHERE AUD.AUDIT_OPTION IN ('EXPORT')
AND AUD.AUDIT_OPTION_TYPE = 'DATAPUMP ACTION'
AND EXISTS (SELECT CON_ID_TO_CON_NAME(ENABLED.CON_ID) AS
CONTAINERNAME, ENABLED.*
FROM CONTAINERS(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
AND ENABLED.CON_ID = AUD.CON_ID)
)
SELECT C.NAME, C.AUDIT_OPTION
FROM CIS_AUDIT C
```

```
WHERE NOT EXISTS (  
    SELECT 1 FROM AUDIT_ENABLED E WHERE E.AUDIT_OPTION = C.AUDIT_OPTION AND  
    E.CON_ID = C.CON_ID  
)  
ORDER BY 1, 2;
```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement in the CDB and in each PDB to remediate this recommendation:

```
ALTER AUDIT POLICY CIS_CDB_EXPORT  
ADD  
ACTIONS  
COMPONENT=datapump  
EXPORT;
```

Note: If you do not have **CIS_CDB_EXPORT**, please create one using the **CREATE AUDIT POLICY** statement. Refer to Section 8.4 where a PL/SQL block is provided to help create or modify the audit policy to remediate this item in both container and pluggable database.

Default Value:

Export **datapump** activities are not audited by default.

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-audit-policies.html#GUID-AA781864-5756-464E-AFB6-675625AF0EF5>

Additional Information:

RMAN activities are subject to mandatory auditing as part of the **ORA\$MANDATORY** audit policy, which is listed in the **UNIFIED_AUDIT_POLITICES** column of the **UNIFIED_AUDIT_TRAIL** data dictionary view. This policy cannot be disabled.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 <u>Collect Detailed Audit Logs</u> Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.			
v7	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.			

5.5 Ensure The Use Of SYS* Privileges Is Audited (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The usage of the **SYS*** (**SYS**, **SYSKM**, **SYSBACKUP**, **SYSRAC**, **SYSDG**, **PUBLICSYS**, **SYSKM**, **SYSBACKUP**, **SYSRAC**, **SYSDG**, **PUBLIC**) privileges should always be audited.

Under Oracle classic auditing, this was the case with the parameter **AUDIT_SYS_OPERATIONS = TRUE**. To achieve the same behavior with Unified Auditing, it is necessary to audit all users who use **SYS*** privileges.

As the **SYS*** privileges have the special feature of being used first, it is necessary to audit them separately, as otherwise they will not be taken into account by the other audit policies.

Example: A user with **SYSDBA** privileges accesses a table (e.g. **HR.EMP**). In this case, not the **SELECT ANY TABLE** privilege, but the **SYSDBA** privilege is used. As a result, audit rule 5.1 does not fire.

With unified auditing, it must always be ensured that the privileges are used for object access in the following order (if available):

1. **SYS*** privileges (e.g. **SYSDBA**)
2. Direct access rights (e.g. direct object grant or public grant)
3. **ANY** rights

Rationale:

All users using the **SYS***-Privilege should be audited to avoid that the highest privileged user is not audited.

Impact:

SYS* performs critical system-level operations, and auditing every action can add considerable performance overhead. **SYS** operations generate a large volume of audit logs, especially in high-transaction environments. Too much logging can make it difficult to identify critical security events among routine activities especially in Oracle Dataguard environments.

Audit:

To assess this recommendation, execute the following SQL statement.

```
WITH  
CIS_AUDIT(CONTAINERNAME, CON_ID, AUDIT_OPTION, ENTITY_NAME_EXPECTED) AS (
```

```

SELECT C.NAME, C.CON_ID, 'ALL' AS AUDIT_OPTION, LL.COLUMN_VALUE AS
ENTITY_NAME_EXPECTED FROM V$CONTAINERS C,
TABLE(DBMSOUTPUT_LINESARRAY('SYS','SYSBACKUP','SYSDG','SYSKM','SYSRAC','PUBLIC')) LL
WHERE C.OPEN_MODE='READ WRITE'
),
AUDIT_ENABLED AS (
SELECT DISTINCT AUD.CON_ID, AUD.AUDIT_OPTION, ENABLED.ENTITY_NAME
FROM CONTAINERS(AUDIT_UNIFIED_POLICIES) AUD,
CONTAINERS(AUDIT_UNIFIED_ENABLED_POLICIES) ENABLED
WHERE ENABLED.CON_ID = AUD.CON_ID
AND ENABLED.POLICY_NAME = AUD.POLICY_NAME
AND AUD.AUDIT_OPTION IN ('ALL')
AND AUD.AUDIT_OPTION_TYPE = 'STANDARD ACTION'
AND ENABLED.SUCCESS = 'YES'
AND ENABLED.FAILURE = 'YES'
AND ENABLED.ENABLED_OPTION = 'BY USER'
AND ENABLED.ENTITY_NAME IN
('SYS','SYSBACKUP','SYSDG','SYSKM','SYSRAC','PUBLIC')
)
SELECT C.CONTAINERNAME, C.AUDIT_OPTION, C.ENTITY_NAME_EXPECTED
FROM CIS_AUDIT C
WHERE NOT EXISTS (
SELECT 1 FROM AUDIT_ENABLED E WHERE E.AUDIT_OPTION = C.AUDIT_OPTION AND
E.ENTITY_NAME = C.ENTITY_NAME_EXPECTED AND E.CON_ID = C.CON_ID
) ORDER BY 1, 3;

```

Lack of results implies compliance.

Remediation:

Execute the following SQL statement in the CDB and in each PDB to remediate this recommendation:

```

CREATE AUDIT POLICY CIS_CDB_ALL_ACTIONS_BY_PRIVILEGED_USERS
ACTIONS ALL
WHEN q'! (SYS_CONTEXT('USERENV','CLIENT_PROGRAM_NAME') NOT IN ('emagent') AND
INSTR(UPPER(SYS_CONTEXT('USERENV','CLIENT_PROGRAM_NAME')), 'PERL') = 0 AND
INSTR(UPPER(SYS_CONTEXT('USERENV','CLIENT_PROGRAM_NAME')), 'RMAN') = 0 AND
INSTR(UPPER(SYS_CONTEXT('USERENV','CLIENT_PROGRAM_NAME')), 'OMS') = 0)!'
EVALUATE PER SESSION
ONLY TOPLEVEL;

AUDIT POLICY CIS_CDB_ALL_ACTIONS_BY_PRIVILEGED_USERS BY SYS, SYSKM,
SYSBACKUP, SYSRAC, SYSDG, PUBLIC;

```

Note: If you do not have **CIS_CDB_ALL_ACTIONS_BY_PRIVILEGED_USERS**, please create one using the **CREATE AUDIT POLICY** statement. Refer to Section 8.5 where a PL/SQL block is provided to help create or modify the audit policy to remediate this item in both container and pluggable database.

For legacy reasons the **SYSOPER** privilege is using the name **PUBLIC**. The used privilege in the unified audit log is **SYSOPR**. **Emagent**, **OMS**, **RMAN** and **Perl** have been excluded to prevent an excessive number of events and issues with spillover files.

Default Value:

Empty

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/sqlrf/CREATE-AUDIT-POLICY-Unified-Auditing.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 <u>Collect Detailed Audit Logs</u> Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.			
v7	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 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.

6.1 Excessive System Privileges

The recommendations within this section revoke excessive system privileges.

The following audit SQL identifies both direct and indirect grants of the system privileges to non **ORACLE_MAINTAINED** users and roles. Specifically, it detects instances where:

- A non-**ORACLE_MAINTAINED** user or role is directly granted the system privilege.
- A non-**ORACLE_MAINTAINED** user or role is indirectly granted the system privilege through an Oracle-maintained role.

Any matches require validation to ensure compliance with security policies.

Please be aware that these audit SQLs do not detect users who have been granted the privilege through a proxy user. Therefore, additional manual checks may be necessary to ensure comprehensive coverage.

There is a comprehensive audit SQL statement in the appendix section provided by Alexander Kornbrust that can be used to identify the list of direct and indirect privileges granted to users and roles.

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

Profile Applicability:

- Level 1 - RDBMS

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:

System privileges in Oracle can be granted directly to users or roles, and can also be inherited through role hierarchies, either directly or recursively. To remediate and revoke system privileges from unauthorized users or roles, it is essential to trace the chain of privilege grants by examining the Granted Privilege and How Granted columns.

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.

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    PRIVILEGE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_SYS_PRIVS
    WHERE PRIVILEGE LIKE '%ANY%'
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_SYS_PRIVS P, SYS.CDB_ROLES R WHERE P.PRIVILEGE LIKE '%ANY%' AND
    P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
)
```

```

SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;

```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

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

In the case of a grant via a role :

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.1.2 Ensure Admin Privileges Are Revoked from Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

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:

System privileges in Oracle can be granted directly to users or roles, and can also be inherited through role hierarchies, either directly or recursively. To remediate and revoke system privileges from unauthorized users or roles, it is essential to trace the chain of privilege grants by examining the Granted Privilege and How Granted columns.

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.

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    PRIVILEGE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_SYS_PRIVS
    WHERE ADMIN_OPTION = 'YES'
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_SYS_PRIVS P, SYS.CDB_ROLES R WHERE P.ADMIN_OPTION = 'Y' AND P.GRANTEE
    = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
```

```
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

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

If needed, grant the privilege to user without admin option:

```
GRANT <SYSTEM PRIVILEGE> TO <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/keeping-your-oracle-database-secure.html#GUID-1574C821-528C-4874-AD6B-92762DBEB400>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.			
v7	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.			

6.1.3 Ensure 'IMPORT' And 'EXPORT' 'FULL DATABASE' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **EXPORT FULL DATABASE** and **IMPORT FULL DATABASE** privileges in Oracle Database allow users to perform full database exports and imports, which can lead to data exfiltration, unauthorized modifications, and compliance violations if granted to unauthorized users.

Rationale:

An unauthorized user with **EXPORT FULL DATABASE** privilege could create a full backup of the database and move it to another location. An unauthorized user with **IMPORT FULL DATABASE** privilege can overwrite, inject, or alter critical database data.

Audit:

System privileges in Oracle can be granted directly to users or roles, and can also be inherited through role hierarchies, either directly or recursively. To remediate and revoke system privileges from unauthorized users or roles, it is essential to trace the chain of privilege grants by examining the Granted Privilege and How Granted columns.

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.

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    PRIVILEGE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_SYS_PRIVS
    WHERE PRIVILEGE IN ('IMPORT FULL DATABASE','EXPORT FULL DATABASE')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_SYS_PRIVS P, SYS.CDB_ROLES R WHERE P.PRIVILEGE IN ('IMPORT FULL
    DATABASE','EXPORT FULL DATABASE') AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR
    GRANTEE = GRANTED_ROLE
)
```

```

SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;

```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```

REVOKE IMPORT FULL DATABASE FROM <grantee>;
REVOKE EXPORT FULL DATABASE FROM <grantee>;

```

In the case of a grant via a role:

```

REVOKE <rolename> FROM <grantee>;

```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html#GUID-58D04BBE-A40D-4699-A2D7-1AB40F532A6D>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.1.4 Ensure 'CREATE EXTERNAL JOB' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **CREATE EXTERNAL JOB** privilege in Oracle allows a user to create external jobs that run at the OS level using Oracle Scheduler.

Rationale:

If this privilege is granted to unauthorized users, it can lead to privilege escalation, system compromise, and security risks. It allows unauthorized users to perform actions like modifying system files, executing scripts, or starting/stopping services.

Audit:

System privileges in Oracle can be granted directly to users or roles, and can also be inherited through role hierarchies, either directly or recursively. To remediate and revoke system privileges from unauthorized users or roles, it is essential to trace the chain of privilege grants by examining the Granted Privilege and How Granted columns.

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.

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS  
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'  
    UNION  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS  
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'  
),  
DIRECT_PRIVS AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,  
    PRIVILEGE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED  
    FROM CDB_SYS_PRIVS  
    WHERE PRIVILEGE IN ('CREATE EXTERNAL JOB')  
),  
INDIRECT_PRIVS AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,  
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED  
    FROM CDB_ROLE_PRIVS  
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM  
    SYS.CDB_SYS_PRIVS P, SYS.CDB_ROLES R WHERE P.PRIVILEGE IN ('CREATE EXTERNAL  
    JOB') AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE  
)  
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED  
FROM (  
    SELECT * FROM DIRECT_PRIVS  
    UNION  
    SELECT * FROM INDIRECT_PRIVS
```

```

SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;

```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE CREATE EXTERNAL JOB FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/keeping-your-oracle-database-secure.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.			
v7	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.			

6.1.5 Ensure 'BECOME USER' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **BECOME USER** system privilege allows a user to switch identities to another database user during the execution of a session, which can lead to serious security risks if granted to unauthorized users.

Rationale:

A user with the **BECOME USER** system privilege can impersonate another user and access data they are not authorized to see, bypassing access controls and security policies. Oracle Data Pump Import utilities **impdp** and **imp** uses **BECOME USER** system privilege to assume the identity of another user to perform operations that cannot be directly performed by a third party (for example, loading objects such as object privilege grants). In an Oracle Database Vault environment, Database Vault provides several levels of required authorization that affect grants of **BECOME USER**. This capability should be restricted according to the needs of the organization.

Audit:

System privileges in Oracle can be granted directly to users or roles, and can also be inherited through role hierarchies, either directly or recursively. To remediate and revoke system privileges from unauthorized users or roles, it is essential to trace the chain of privilege grants by examining the Granted Privilege and How Granted columns.

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.

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS  
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'  
    UNION  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS  
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'  
),  
DIRECT_PRIVS AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,  
    PRIVILEGE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED  
    FROM CDB_SYS_PRIVS  
    WHERE PRIVILEGE IN ('BECOME USER')  
),  
INDIRECT_PRIVS AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,  
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED  
    FROM CDB_ROLE_PRIVS
```

```

START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
SYS.CDB_SYS_PRIVS P, SYS.CDB_ROLES R WHERE P.PRIVILEGE IN ('BECOME USER') AND
P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;

```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

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

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/keeping-your-oracle-database-secure.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.1.6 Ensure 'TEXT DATASTORE ACCESS' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **TEXT DATASTORE ACCESS** system privilege should be restricted due to the potential security risks associated with Oracle Text and its ability to access external files and data sources.

Rationale:

Granting **TEXT DATASTORE ACCESS** gives the user the privilege to index either an arbitrary file in the file system in the case of **FILE** datastore and an arbitrary URL in the case of **URL** datastore and is not recommended.

Audit:

System privileges in Oracle can be granted directly to users or roles, and can also be inherited through role hierarchies, either directly or recursively. To remediate and revoke system privileges from unauthorized users or roles, it is essential to trace the chain of privilege grants by examining the Granted Privilege and How Granted columns.

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.

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    PRIVILEGE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_SYS_PRIVS
    WHERE PRIVILEGE IN ('TEXT DATASTORE ACCESS')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_SYS_PRIVS P, SYS.CDB_ROLES R WHERE P.PRIVILEGE IN ('TEXT DATASTORE
    ACCESS') AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
```

```

SELECT * FROM DIRECT_PRIVS
UNION
SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;

```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE TEXT DATASTORE ACCESS FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/ccref/oracle-text-indexing-elements.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.1.7 Ensure 'CREATE', 'ALTER', And 'DROP' 'PUBLIC DATABASE LINK' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

Public database links in Oracle allow database users to connect to remote databases without needing explicit credentials each time.

Rationale:

If an unauthorized user creates a public database link, all users in the database can use it to connect to the remote database. This can lead to data exposure and unauthorized access to sensitive information. Likewise, if an unauthorized user modifies or drops a public database link, this can result in pointing an existing database link to a malicious remote database, queries could return manipulated or incorrect data, leading to data corruption.

Audit:

System privileges in Oracle can be granted directly to users or roles, and can also be inherited through role hierarchies, either directly or recursively. To remediate and revoke system privileges from unauthorized users or roles, it is essential to trace the chain of privilege grants by examining the Granted Privilege and How Granted columns.

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.

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS  
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'  
    UNION  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS  
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'  
),  
DIRECT_PRIVS AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,  
    PRIVILEGE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED  
    FROM CDB_SYS_PRIVS  
    WHERE PRIVILEGE IN ('CREATE PUBLIC DATABASE LINK','ALTER PUBLIC DATABASE  
    LINK','DROP PUBLIC DATABASE LINK')  
),  
INDIRECT_PRIVS AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,  
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED  
    FROM CDB_ROLE_PRIVS
```

```

START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
SYS.CDB_SYS_PRIVS P, SYS.CDB_ROLES R WHERE P.PRIVILEGE IN ('CREATE PUBLIC
DATABASE LINK','ALTER PUBLIC DATABASE LINK','DROP PUBLIC DATABASE LINK') AND
P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;

```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```

REVOKE CREATE PUBLIC DATABASE LINK FROM <grantee>;
REVOKE ALTER PUBLIC DATABASE LINK FROM <grantee>;
REVOKE DROP PUBLIC DATABASE LINK FROM <grantee>;

```

In the case of a grant via a role:

```

REVOKE <rolename> FROM <grantee>;

```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/keeping-your-oracle-database-secure.html#GUID-5167857B-7CEC-423B-8A6F-64569B3A661D>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.1.8 Ensure 'LOGMINING' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **LOGMINING** system privilege in Oracle Database is a powerful privilege that allows users to query online and archived database redo log files through a SQL interface.

Rationale:

Redo log files contain information about the history of activity on a database, including sensitive data like credit card numbers or passwords. Allowing unauthorized access to log mining could expose this sensitive data.

Audit:

System privileges in Oracle can be granted directly to users or roles, and can also be inherited through role hierarchies, either directly or recursively. To remediate and revoke system privileges from unauthorized users or roles, it is essential to trace the chain of privilege grants by examining the Granted Privilege and How Granted columns.

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.

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    PRIVILEGE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_SYS_PRIVS
    WHERE PRIVILEGE IN ('LOGMINING')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_SYS_PRIVS P, SYS.CDB_ROLES R WHERE P.PRIVILEGE IN ('LOGMINING') AND
    P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
```

```

SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;

```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE LOGMINING FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/sutil/oracle-logminer-utility.html#GUID-D857AF96-AC24-4CA1-B620-8EA3DF30D72E>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.1.9 Ensure 'ALTER SYSTEM' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

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:

System privileges in Oracle can be granted directly to users or roles, and can also be inherited through role hierarchies, either directly or recursively. To remediate and revoke system privileges from unauthorized users or roles, it is essential to trace the chain of privilege grants by examining the Granted Privilege and How Granted columns.

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.

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    PRIVILEGE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_SYS_PRIVS
    WHERE PRIVILEGE IN ('ALTER SYSTEM')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_SYS_PRIVS P, SYS.CDB_ROLES R WHERE P.PRIVILEGE IN ('ALTER SYSTEM')
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
```

```

SELECT * FROM DIRECT_PRIVS
UNION
SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;

```

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/26/dbseg/keeping-your-oracle-database-secure.html#GUID-1574C821-528C-4874-AD6B-92762DBEB400>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.1.10 Ensure 'CREATE LIBRARY' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The Oracle database **CREATE LIBRARY** privilege allows the designated user to create objects that are associated with 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:

System privileges in Oracle can be granted directly to users or roles, and can also be inherited through role hierarchies, either directly or recursively. To remediate and revoke system privileges from unauthorized users or roles, it is essential to trace the chain of privilege grants by examining the Granted Privilege and How Granted columns.

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.

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS  
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'  
    UNION  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS  
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'  
),  
DIRECT_PRIVS AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,  
    PRIVILEGE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED  
    FROM CDB_SYS_PRIVS  
    WHERE PRIVILEGE IN ('CREATE LIBRARY')  
),  
INDIRECT_PRIVS AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,  
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED  
    FROM CDB_ROLE_PRIVS  
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM  
    SYS.CDB_SYS_PRIVS P, SYS.CDB_ROLES R WHERE P.PRIVILEGE IN ('CREATE LIBRARY')  
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE  
)  
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED  
FROM (  
    SELECT * FROM DIRECT_PRIVS  
    UNION  
    SELECT * FROM INDIRECT_PRIVS
```

```

SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;

```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

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

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/keeping-your-oracle-database-secure.html#GUID-1574C821-528C-4874-AD6B-92762DBEB400>

Additional Information:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.1.11 Ensure All 'SYSTEM' Privileges Are Revoked from Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

System privileges grant access to sensitive operations, such as creating or modifying database structures, managing user accounts, and accessing sensitive data.

Rationale:

Unauthorized users with system privileges can perform actions that bypass normal security controls, potentially leading to data breaches, tampering, or destruction.

Audit:

System privileges in Oracle can be granted directly to users or roles, and can also be inherited through role hierarchies, either directly or recursively. To remediate and revoke system privileges from unauthorized users or roles, it is essential to trace the chain of privilege grants by examining the Granted Privilege and How Granted columns.

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.

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS  
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'  
    UNION  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS  
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'  
),  
DIRECT_PRIVS AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,  
    PRIVILEGE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED  
    FROM CDB_SYS_PRIVS  
),  
INDIRECT_PRIVS AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,  
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED  
    FROM CDB_ROLE_PRIVS  
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM  
    SYS.CDB_SYS_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTEE = R.ROLE) CONNECT BY  
    PRIOR GRANTEE = GRANTED_ROLE  
)  
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED  
FROM (  
    SELECT * FROM DIRECT_PRIVS  
    UNION  
    SELECT * FROM INDIRECT_PRIVS  
    ) COMBINED_PRIVS
```

```
WHERE
  GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

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

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/keeping-your-oracle-database-secure.html#GUID-1574C821-528C-4874-AD6B-92762DBEB400>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.			
v7	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.			

6.2 Excessive Role Privileges

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

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

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DBA** role provides **full administrative privileges** over the database, allowing grantees to perform any operation. Unauthorized users with this role can access and modify critical database configurations and objects.

Rationale:

Granting **DBA** privileges to unauthorized users increases the risk of data breaches, unauthorized modifications, and privilege escalation attacks. Access to this role should be **limited** to authorized administrators.

Audit:

Run the following query to check for unauthorized grants:

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    WHERE GRANTED_ROLE IN ('DBA')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN ('DBA')
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

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

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.2.2 Ensure 'EXP_FULL_DATABASE' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **EXP_FULL_DATABASE** privilege allows a user to export all objects in the database using Oracle export utility. Unauthorized access to this privilege may result in data leakage.

Rationale:

Unauthorized users with this privilege can perform **full database exports**, leading to potential **data breaches**.

Audit:

Run the following query to check for unauthorized grants:

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    WHERE GRANTED_ROLE IN ('EXP_FULL_DATABASE')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN
    ('EXP_FULL_DATABASE')
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE EXP_FULL_DATABASE FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.2.3 Ensure 'IMP_FULL_DATABASE' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **IMP_FULL_DATABASE** privilege allows users to import database objects using **Oracle Import utility**. Unauthorized access to this privilege can lead to **data corruption**.

Rationale:

A user with **IMP_FULL_DATABASE** can **restore** data, potentially leading to data corruption or security violations.

Audit:

Run the following query to check for unauthorized grants:

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    WHERE GRANTED_ROLE IN ('IMP_FULL_DATABASE')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN
    ('IMP_FULL_DATABASE')
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE IMP_FULL_DATABASE FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.2.4 Ensure 'DATAPUMP_EXP_FULL_DATABASE' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DATAPUMP_EXP_FULL_DATABASE** privilege allows a user to export all objects in the database using Oracle data pump export utility. Unauthorized access to this privilege may result in data leakage.

Rationale:

Users with this privilege can perform **full database** exports, leading to potential **data breaches**.

Audit:

Run the following query to check for unauthorized grants:

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    WHERE GRANTED_ROLE IN ('DATAPUMP_EXP_FULL_DATABASE')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN
    ('DATAPUMP_EXP_FULL_DATABASE')
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE DATAPUMP_EXP_FULL_DATABASE FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.2.5 Ensure 'DATAPUMP_IMP_FULL_DATABASE' is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DATAPUMP_IMP_FULL_DATABASE** privilege allows users to import database objects using **Oracle Data Pump Import utility**. Unauthorized access to this privilege can lead to **data corruption**.

Rationale:

A user with **DATAPUMP_IMP_FULL_DATABASE** can restore data, potentially leading to data corruption or security violations.

Audit:

Run the following query to check for unauthorized grants:

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    WHERE GRANTED_ROLE IN ('DATAPUMP_IMP_FULL_DATABASE')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN
    ('DATAPUMP_IMP_FULL_DATABASE')
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE DATAPUMP_IMP_FULL_DATABASE FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.2.6 Ensure 'DV_ADMIN' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DV_ADMIN** role allows users to configure **Database Vault policies** and manage security controls. Unauthorized access can **weaken security**.

Rationale:

This role should be **restricted** to authorized administrators, as unauthorized use can lead to alteration or deletion of established **security policies**.

Audit:

Run the following query to check for unauthorized grants:

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    WHERE GRANTED_ROLE IN ('DV_ADMIN')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN ('DV_ADMIN')
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE DV_ADMIN FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.2.7 Ensure 'DV_AUDIT_CLEANUP' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DV_AUDIT_CLEANUP** role allows grantees to delete or purge data vault audit records. Unauthorized users with this role can temper with audit records, resulting in violation of security compliance policies.

Rationale:

Unauthorized deletion of audit records compromises security policies and violates compliance requirements. Unauthorized deletion can also be used to hide unauthorized activities.

Audit:

Run the following query to check for unauthorized grants:

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    WHERE GRANTED_ROLE IN ('DV_AUDIT_CLEANUP')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN
    ('DV_AUDIT_CLEANUP')
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
```

```
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE DV_AUDIT_CLEANUP FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.2.8 Ensure 'OLAP_DBA' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **OLAP_DBA** role allows grantees to perform Oracle OLAP administration tasks. Unauthorized access can impact analytical workspaces.

Rationale:

Users with this role can **alter OLAP settings**, potentially impacting business intelligence applications.

Audit:

Run the following query to check for unauthorized grants:

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    WHERE GRANTED_ROLE IN ('OLAP_DBA')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN ('OLAP_DBA')
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE OLAP_DBA FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.2.9 Ensure 'LBAC_DBA' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **LBAC_DBA** role allows users to administer **Label-Based Access Control (LBAC)**. Unauthorized users can modify security labels.

Rationale:

Granting this role to unauthorized users can **impact data access restrictions** through security labels.

Audit:

Run the following query to check for unauthorized grants:

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    WHERE GRANTED_ROLE IN ('LBAC_DBA')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN ('LBAC_DBA')
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE LBAC_DBA FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.2.10 Ensure 'JAVA_ADMIN' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **JAVA_ADMIN** role allows users to manage Java objects within the database. Unauthorized users with this role may leverage this role to compromise Java-based security mechanisms.

Rationale:

A user with this role can **alter, modify, or execute Java objects**, potentially introducing security risks.

Audit:

Run the following query to check for unauthorized grants:

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    WHERE GRANTED_ROLE IN ('JAVA_ADMIN')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN ('JAVA_ADMIN')
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE JAVA_ADMIN FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	14.7 <u>Enforce Access Control to Data through Automated Tools</u> Use an automated tool, such as host-based Data Loss Prevention, to enforce access controls to data even when data is copied off a system.			

6.2.11 Ensure 'JAVASYSPRIVS' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **JAVASYSPRIV** privilege grants Java system-level permissions within the database. Unauthorized access could allow unrestricted Java execution.

Rationale:

A user with this role can run Java programs that may compromise database security and integrity.

Audit:

Run the following query to check for unauthorized grants:

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
  SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
  GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
  UNION
  SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
  GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
  SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
  GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
  FROM CDB_ROLE_PRIVS
  WHERE GRANTED_ROLE IN ('JAVAUSERPRIV')
),
INDIRECT_PRIVS AS (
  SELECT DISTINCT CON_ID TO CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
  GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
  FROM CDB_ROLE_PRIVS
  START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
  SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN
  ('JAVAUSERPRIV')
  AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
  SELECT * FROM DIRECT_PRIVS
  UNION
  SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
  GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE JAVAUSERPRIV FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.2.12 Ensure 'LOGSTDBY_ADMINISTRATOR' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **LOGSTDBY_ADMINISTRATOR** role provides privileges to administer logical standby databases. Unauthorized grantees can perform operations that can disrupt logical standby database.

Rationale:

Unauthorized changes may **impact logical standby database replication** and lead to **data inconsistency**.

Audit:

Run the following query to check for unauthorized grants:

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    WHERE GRANTED_ROLE IN ('LOGSTDBY_ADMINISTRATOR')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN
    ('LOGSTDBY_ADMINISTRATOR')
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE LOGSTDBY_ADMINISTRATOR FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.2.13 Ensure 'SQL_FIREWALL_ADMIN' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SQL_FIREWALL_ADMIN** role grants administrative privileges to the SQL Firewall, allowing grantees to configure security policies through firewall rules. Unauthorized users can bypass firewall rules, potentially allowing execution of malicious code, including SQL injection attacks.

Rationale:

An unauthorized user with this role can **modify** or **disable firewall rules**, exposing the database to **SQL injection** or unauthorized access.

Audit:

To assess this recommendation, execute the following SQL statement.

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    WHERE GRANTED_ROLE IN ('SQL_FIREWALL_ADMIN')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN
    ('SQL_FIREWALL_ADMIN')
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
```

```
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE SQL_FIREWALL_ADMIN FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>
2. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/using-sql-firewall.html#GUID-47339B36-F95A-4371-AC87-F8EF2C799455>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.2.14 Ensure 'MAINTPLAN_APP' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **MAINTPLAN_APP** role grants access to database maintenance plans. Unauthorized use can lead to disruptions in Oracle database operation and service.

Rationale:

This role should only be assigned to database administrators to prevent accidental or intentional maintenance plan modifications. A user with this role can alter or remove scheduled maintenance jobs, affecting database availability.

Audit:

Run the following query to check for unauthorized grants:

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS  
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'  
    UNION  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS  
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'  
),  
DIRECT_PRIVS AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,  
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED  
    FROM CDB_ROLE_PRIVS  
    WHERE GRANTED_ROLE IN ('MAINTPLAN_APP')  
),  
INDIRECT_PRIVS AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,  
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED  
    FROM CDB_ROLE_PRIVS  
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM  
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN  
    ('MAINTPLAN_APP')  
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE  
),  
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED  
FROM (  
    SELECT * FROM DIRECT_PRIVS  
    UNION  
    SELECT * FROM INDIRECT_PRIVS  
) COMBINED_PRIVS  
WHERE  
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)  
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE MAINTPLAN_APP FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.2.15 Ensure 'JAVADEBUGPRIV' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **JAVADEBUGPRIV** privilege allows debugging of Java stored procedures within the Oracle database. Debugging privileges can potentially be exploited to inspect or alter Java code execution. Revoking this privilege from unauthorized users helps prevent unauthorized debugging of Java stored procedures and reduces the risk of exposing internal Java code logic.

Rationale:

Unauthorized granting of the **JAVADEBUGPRIV** privilege may expose Java procedures to unintended debugging, which can lead to security risks such as unauthorized access or code manipulation.

Audit:

Run the following query to check for unauthorized grants:

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS  
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'  
    UNION  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS  
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'  
),  
DIRECT_PRIVS AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,  
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED  
    FROM CDB_ROLE_PRIVS  
    WHERE GRANTED_ROLE IN ('JAVADEBUGPRIV')  
),  
INDIRECT_PRIVS AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,  
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED  
    FROM CDB_ROLE_PRIVS  
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM  
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN  
    ('JAVADEBUGPRIV')  
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE  
),  
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED  
FROM (  
    SELECT * FROM DIRECT_PRIVS  
    UNION  
    SELECT * FROM INDIRECT_PRIVS  
    ) COMBINED_PRIVS
```

```
WHERE
  GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE JAVADEBUGPRIV FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.2.16 Ensure 'DV_PATCH_ADMIN' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DV_PATCH_ADMIN** privilege allows users to apply patches to the database vault component of the Oracle database. This privilege grants the ability to update/upgrade a critical component of the database.

Rationale:

Users with **DV_PATCH_ADMIN** can modify or apply patches that may impact the Data Vault component and hence the security of the database. Restricting this privilege helps maintain database security.

Audit:

Run the following query to check for unauthorized grants:

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    WHERE GRANTED_ROLE IN ('DV_PATCH_ADMIN')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN
    ('DV_PATCH_ADMIN')
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
```

```
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE DV_PATCH_ADMIN FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.2.17 Ensure 'DV_POLICY_OWNER' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DV_POLICY_OWNER** privilege allows a user to own and manage security policies within Oracle Database Vault. This role provides control over security policies and rules governing database access.

Rationale:

Users with **DV_POLICY_OWNER** can modify security policies, potentially weakening database security. Revoking this privilege from unauthorized users ensures that only security administrators manage security policies.

Audit:

Run the following query to check for unauthorized grants:

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    WHERE GRANTED_ROLE IN ('DV_POLICY_OWNER')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN
    ('DV_POLICY_OWNER')
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
```

```
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE DV_POLICY_OWNER FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.2.18 Ensure AUDIT_ADMIN' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **AUDIT_ADMIN** privilege allows a user to manage database auditing policies and audit records. This role grants control over auditing configurations and access to audit trails.

Rationale:

Users with **AUDIT_ADMIN** can modify audit settings or delete audit records, potentially concealing unauthorized activities. Revoking this privilege from unauthorized users ensures that only designated administrators control audit policies, preventing audit log tampering.

Audit:

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.

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS  
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'  
    UNION  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS  
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'  
),  
DIRECT_PRIVS AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,  
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED  
    FROM CDB_ROLE_PRIVS  
    WHERE GRANTED_ROLE IN ('AUDIT_ADMIN')  
),  
INDIRECT_PRIVS AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,  
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED  
    FROM CDB_ROLE_PRIVS  
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM  
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN ('AUDIT_ADMIN')  
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE  
)  
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED  
FROM (  
    SELECT * FROM DIRECT_PRIVS  
    UNION  
    SELECT * FROM INDIRECT_PRIVS  
    ) COMBINED_PRIVS  
WHERE
```

```
GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE AUDIT_ADMIN FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.2.19 Ensure 'AUDIT_VIEWER' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **AUDIT_VIEWER** privilege allows users to view audit records without modifying them. This role is designed for read-only access to audit logs.

Rationale:

Unauthorized access to audit logs can expose sensitive information and compromise audit integrity. Revoking this privilege from unauthorized users prevents unnecessary exposure of audit records and enhances data security.

Audit:

Run the following query to check for unauthorized grants:

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    WHERE GRANTED_ROLE IN ('AUDIT_VIEWER')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN
    ('AUDIT_VIEWER')
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE AUDIT_VIEWER FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.2.20 Ensure 'PDB_DBA' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **PDB_DBA** privilege allows users to manage Pluggable Databases (PDBs) within a Container Database (CDB). This privilege grants the ability to create, drop, and modify PDBs.

Rationale:

Users with **PDB_DBA** can modify or delete PDBs, impacting database availability and security. Revoking this privilege from unauthorized users ensures that only authorized administrators manage PDBs, ensuring database security.

Audit:

Run the following query to check for unauthorized grants:

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    WHERE GRANTED_ROLE IN ('PDB_DBA')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN ('PDB_DBA')
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

```
REVOKE PDB_DBA FROM <grantee>;
```

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.2.21 Ensure 'SELECT_CATALOG_ROLE' Is Revoked From Unauthorized 'GRANTEE' (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The Oracle database **SELECT_CATALOG_ROLE** provides **SELECT** privileges on two-thirds of 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 multiple but not all dictionary data.

Audit:

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.

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS  
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'  
    UNION  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS  
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'  
),  
DIRECT_PRIVS AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,  
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED  
    FROM CDB_ROLE_PRIVS  
    WHERE GRANTED_ROLE IN ('SELECT_CATALOG_ROLE')  
),  
INDIRECT_PRIVS AS (  
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,  
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED  
    FROM CDB_ROLE_PRIVS  
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM  
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN  
    ('SELECT_CATALOG_ROLE')  
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE  
),  
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED  
FROM (  
    SELECT * FROM DIRECT_PRIVS  
    UNION  
    SELECT * FROM INDIRECT_PRIVS  
    ) COMBINED_PRIVS  
WHERE  
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
```

```
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

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

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

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

Profile Applicability:

- Level 1 - RDBMS

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.

Audit:

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.

```
WITH GRANTEES_NOT_ORACLEMAINTAINED AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, ROLE AS
    GRANTEE FROM CDB_ROLES WHERE ORACLE_MAINTAINED = 'N'
    UNION
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, USERNAME AS
    GRANTEE FROM CDB_USERS WHERE ORACLE_MAINTAINED = 'N'
),
DIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Direct Grant' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    WHERE GRANTED_ROLE IN ('EXECUTE_CATALOG_ROLE')
),
INDIRECT_PRIVS AS (
    SELECT DISTINCT CON_ID_TO_CON_NAME(CON_ID) AS CONTAINERNAME, GRANTEE,
    GRANTED_ROLE AS GRANTED_PRIVILEGE, 'Privileges Through Role' AS HOW_GRANTED
    FROM CDB_ROLE_PRIVS
    START WITH GRANTED_ROLE IN ( SELECT DISTINCT P.GRANTEE FROM
    SYS.CDB_ROLE_PRIVS P, SYS.CDB_ROLES R WHERE P.GRANTED_ROLE IN
    ('EXECUTE_CATALOG_ROLE')
    AND P.GRANTEE = R.ROLE) CONNECT BY PRIOR GRANTEE = GRANTED_ROLE
)
SELECT CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE, HOW_GRANTED
FROM (
    SELECT * FROM DIRECT_PRIVS
    UNION
    SELECT * FROM INDIRECT_PRIVS
) COMBINED_PRIVS
WHERE
    GRANTEE IN (SELECT GRANTEE FROM GRANTEES_NOT_ORACLEMAINTAINED)
```

```
ORDER BY CONTAINERNAME, GRANTEE, GRANTED_PRIVILEGE;
```

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. Please ensure proper impact analysis is done before revoking the privilege from a role.

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

In the case of a grant via a role:

```
REVOKE <rolename> FROM <grantee>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.3 Excessive Schema Privileges

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

6.3.1 Ensure 'CDB_SCHEMA_PRIVS' Does Not Have Unauthorized Privileges (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

In Oracle 26ai, the new Schema Privileges provide fine-grained control over which users can create objects in a particular schema without requiring extensive system privileges such as **CREATE ANY TABLE**. This improves security by allowing administrators to specifically grant **CREATE**, **ALTER**, **DROP** and **SELECT** privileges at the schema level instead of granting global access.

Rationale:

The use of Schema privileges improves security by simplifying authorization for database objects, especially for schemas that frequently add new objects. Instead of granting broad system-level (*** ANY**) privileges that apply to the entire database, privileges can now be granted at the individual schema level.

Schema privileges may inadvertently grant a technical database user excessive rights, potentially violating the principle of least privilege.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT CON_ID_TO_CON_NAME(c.con_id) AS containername, c.*  
FROM CDB_SCHEMA_PRIVS c;
```

Review the results for any unauthorized privileges. If any unauthorized privileges exist, this is a fail.

Remediation:

To remediate this recommendation, revoke privileges that are no longer required by executing the following SQL statement.

```
REVOKE <SCHEMA_PRIVILEGE> ON SCHEMA <USERNAME> FROM <GRANTEE>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/dbseg/configuring-privilege-and-role-authorization.html#GUID-483D04AF-BC5B-4B3D-9D9A-1D2C3CE8F12F>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.4 Excessive Object Privileges

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

6.4.1 Ensure 'ALL' Is Revoked On 'Sensitive' Tables (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

Some tables contain critical information such as password hashes, encrypted passwords, etc. No user other than **SYS** should have table privileges on these tables.

Rationale:

Unauthorized users with access to these tables can perform actions at the operating system level that bypass normal security controls, potentially leading to data breaches, tampering, or destruction.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT CON_ID_TO_CON_NAME(c.con_id) AS containername, c.*
FROM CDB_TAB_PRIVS c
WHERE c.owner='SYS' AND c.table_name IN
('CDB_LOCAL_ADMINAUTH$', 'DEFAULT_PWD$', 'ENC$', 'HISTGRM$', 'HIST_HEAD$', 'LINK$',
'PDB_SYNC$', 'SCHEDULER$ _CREDENTIAL', 'USER$', 'USER_HISTORY$', 'XS$VERIFIERS');
```

If any sensitive tables have users, who do **NOT** have an accepted business need, with **ALL** privileges, this is a fail.

Remediation:

To remediate this recommendation, revoke privileges that are no longer required by executing the following SQL statement.

```
REVOKE <privilege> ON <table> <directory_name>;
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.5 Excessive Column Privileges

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

6.5.1 Ensure 'DBA_COL_PRIVS' Is Revoked from Unauthorized 'GRANTEE' (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DBA_COL_PRIVS** view provides DBAs a view to manage all column level privileges granted to users and roles.

Rationale:

Granting **DBA_COL_PRIVS** privileges to unauthorized users increases the risk of data breaches, unauthorized modifications, and privilege escalation attacks. Access to this should be limited to authorized DBAs.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT CON_ID TO_CON_NAME(C.CON_ID) AS CONTAINERNAME,O.ORACLE_MAINTAINED, C.*  
FROM CDB_COL_PRIVS C  
JOIN CDB_OBJECTS O ON C.CON_ID = O.CON_ID AND C.OWNER=O.OWNER  
AND O.OBJECT_NAME=C.TABLE_NAME AND O.ORACLE_MAINTAINED='N';
```

If any unauthorized users are granted **DBA_COL_PRIVS**, this is a fail.

Remediation:

To remediate this recommendation, execute the following SQL statement.

```
REVOKE UPDATE ON <TABLE> FROM <GRANTEE>;
```

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/26/refrn/DBA_COL_PRIVS.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.6 Excessive Proxy Privileges

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

6.6.1 Ensure Proxy User Privileges Are Revoked from Unauthorized 'GRANTEE' (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

Oracle Database supports proxy user authentication, which allows a client user to connect to the database as a proxy user.

Rationale:

An Oracle user with the right to connect to the database as a proxy user inherits the authorizations and roles of the client.

For example, the user **DUMMYUSER** can log on as **SYSTEM** and then use its DBA role:

```
ALTER USER SYSTEM GRANT CONNECT THROUGH DUMMYUSER;
```

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT CON_ID TO_CON_NAME(J.CON_ID) AS CONTAINERNAME, J.* FROM CDB_PROXIES J;
```

If any unauthorized users are granted proxy user privileges, this is a fail.

Remediation:

To remediate this recommendation, execute the following SQL statement.

```
ALTER USER <CLIENT_USER> REVOKE CONNECT THROUGH <PROXY_USER>;
```

References:

1. https://docs.oracle.com/en/database/oracle/oracle-database/26/refrn/DBA_PROXIES.html#GUID-1F0BFB9F-8A91-41CD-953F-B3EADB17E0AD

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.7 Excessive Java Privileges

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

6.7.1 Ensure Custom Java Privileges Are Revoked from Unauthorized 'GRANTEE' (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

Oracle supports Java embedded in the database. Privileges exist for this Java, which are maintained in separate privilege tables.

Rationale:

Unauthorized users with certain Java privileges can perform actions at the operating system level that bypass normal security controls, potentially leading to data breaches, tampering, or destruction.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT CON_ID TO_CON_NAME(J.CON_ID) AS CONTAINERNAME, 'CUSTOM JAVA PRIVILEGE'
AS "JAVA PRIVILEGE", J.*
FROM CDB_JAVA_POLICY J WHERE DBMS_UTILITY.GET_HASH_VALUE(
(KIND||GRANTEE||TYPE_SCHEMA||TYPE_NAME||NAME||ACTION||ENABLED), 2,
2010304050) NOT IN ( 800515347, 151129288, 976537527, 30494973, 1937158100,
364905785, 1985378421, 1309631600, 1029870508, 1546167066, 184142192,
859260823, 1514899866, 1402070492, 470712301, 856789430, 152768586,
1516266150, 742382950, 835887237, 27519048, 127800042, 585523424, 116384647,
595031329, 519425340, 565011516, 104142482, 365736720, 105705833, 382398907,
71790781, 62579200, 202664153, 1299118788, 105413428, 545541759, 1119910297,
1712531359, 950804353, 809152653, 63495589, 1621785741, 888624743,
1298694530, 1094834124, 1254650837, 1289482879, 236689110, 449271147,
356128445, 1546786406, 374027524, 1161638561, 1892729955, 972739099,
539210486, 1634287789, 2000803097, 741404310, 1879138467, 623589740,
498687020, 782135231, 1194762317, 884905623, 1896250536, 970596985,
1082897628, 448525273, 1242306455, 23750295, 1550194092, 1772592918,
1603737720, 142136119, 289269359, 1315794462, 96990011, 1095201623,
20727542, 1973515175, 540292561, 343336248, 179666672, 9440468, 1825382930,
1855646886, 598800695, 631787483, 1467839181, 665909584, 1461437618,
1030723320, 311348090, 1398805244, 179119810, 72116459, 1979563728, 57733337,
54818063, 1367689664, 1508476194, 1523274446, 313964240, 1328352206,
689371984, 1299489851, 121628881, 696051281, 1374172231, 42876420,
1612485481, 1577690416, 551128519, 1695954190, 826909143, 983212206,
1658826990, 1414117567, 980793982, 742010474, 230612329, 245501173,
424452954, 1700638178, 1789496901, 1202501131, 1731444159, 335317020,
253453448, 1936475065, 777837297, 462854008, 368852509, 129848269, 549321505,
301832111, 1323272794, 92418107, 715497686, 1854322866, 1423215774,
1973681135, 375128879, 553190962, 907068071, 638021877, 1788624774,
1662401506, 1362996185, 478866415, 1598256716, 1469155910, 334914317,
870540049, 768584438, 1885395547, 755918725, 350200414, 134114043, 43446111,
486709639, 83488831, 139365274, 1260703408, 467391551, 701952766, 230475370,
736630396, 112581665, 376052929, 471626899, 542637624, 680632989, 1103049052,
```

```
1347282494, 1559487896, 1592685230, 1654109637, 1735090119, 1831375678,
1886462769, 1912735089, 1959933679, 1234282401, 1772747954, 1514447620,
201492720, 167207852, 434486659, 1169110020, 1649953039, 2008561561,
1333987409, 1192825952, 1740977823, 1501827236, 934263545, 1081144386,
310891775, 1279875884, 1539747260, 1630393794, 1764828081, 336339792,
503085247, 639778710, 764636870, 919569250, 993349195, 124317783, 426721579,
632709131, 703662335, 866241111, 1367484647, 1523591677, 1902848780);
```

If any unauthorized users are granted Java privileges, this is a fail.

Remediation:

To remediate this recommendation, execute the following SQL statement.

```
begin
  DBMS_JAVA.disable_permission(<number>);
  DBMS_JAVA.delete_permission(<number>);
end;
```

Note: Custom Java Privileges can be revoked via the **SEQ**(uence) number.

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/jjdev/DBMS-JAVA-package.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.8 Excessive Directory Privileges

This section contains recommendations that revoke powerful directory privileges where they are likely not needed.

6.8.1 Ensure Directory Object Access Is Revoked From Unauthorized 'GRANTEE' (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

A directory object specifies an alias for a directory on the server file system where external files and data are located.

Rationale:

Users with access to these directories can perform actions at the operating system level that bypass normal security controls, potentially leading to data breaches, tampering, or destruction.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT CON_ID TO_CON_NAME(C.CON_ID) AS CONTAINERNAME, O.ORACLE_MAINTAINED,
C.OWNER, C.DIRECTORY_NAME, C.DIRECTORY_PATH
FROM CDB_DIRECTORIES C
JOIN CDB_OBJECTS O ON C.CON_ID = O.CON_ID AND C.OWNER=O.OWNER AND
O.OBJECT_NAME=C.DIRECTORY_NAME AND OBJECT_TYPE='DIRECTORY';
```

If any unauthorized users are granted directory object access, this is a fail.

Note: The flag `oracle_maintained='N'` indicates whether the directory object was created by Oracle or someone else.

Remediation:

To remediate this recommendation, execute the following SQL statement to delete directories which are no longer required.

```
DROP DIRECTORY <DIRECTORY_NAME>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/sqlrf/CREATE-DIRECTORY.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.8.2 Review Directory Objects Privileges (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

A directory object specifies an alias for a directory on the server file system where external files and data are located.

Rationale:

Unauthorized users with access to these directories can perform actions at the operating system level that bypass normal security controls, potentially leading to data breaches, tampering, or destruction.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT CON_ID_TO_CON_NAME(D.CON_ID) AS CONTAINERNAME,O.ORACLE_MAINTAINED,  
D.DIRECTORY_NAME,D.DIRECTORY_PATH,  
T.GRANTEE, T.GRANTOR, T.PRIVILEGE, T.GRANTABLE, T.HIERARCHY, T.COMMON,  
T.INHERITED, D.CON_ID  
--T.GRANTEE, T.PRIVILEGE, T.GRANTABLE  
FROM CDB_DIRECTORIES D  
LEFT JOIN CDB_TAB_PRIVS T ON D.CON_ID = T.CON_ID AND D.OWNER=T.OWNER AND  
D.DIRECTORY_NAME=T.TABLE_NAME AND T.TYPE='DIRECTORY'  
JOIN CDB_OBJECTS O ON D.CON_ID = O.CON_ID AND D.OWNER=O.OWNER AND  
O.OBJECT_NAME=D.DIRECTORY_NAME AND O.OBJECT_TYPE='DIRECTORY';
```

The flag **oracle_maintained='N'** indicates whether the directory object was created by Oracle or someone else.

The non-Oracle objects should be checked carefully for possible problems.

Remediation:

Delete directories that are no longer required

```
DROP DIRECTORY <DIRECTORY_NAME>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/sqlrf/CREATE-DIRECTORY.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.8.3 Review External Tables With Preprocessor (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

A directory object specifies an alias for a directory on the server file system where external files and data are located.

Rationale:

Unauthorized users with access to these directories can perform actions at the operating system level that bypass normal security controls, potentially leading to data breaches, tampering, or destruction.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT CON_ID_TO_CON_NAME(C.CON_ID) AS CONTAINERNAME,O.ORACLE_MAINTAINED,
REGEXP_SUBSTR(SUBSTR(ACCESS_PARAMETERS, INSTR(ACCESS_PARAMETERS,
'PREPROCESSOR ')), '^(.*)$', 1, 1, NULL, 1) AS FILENAME, C.*
FROM CDB_EXTERNAL_TABLES C
JOIN CDB_OBJECTS O ON C.CON_ID = O.CON_ID AND C.OWNER=O.OWNER AND
O.OBJECT_NAME=C.TABLE_NAME AND O.OBJECT_TYPE='TABLE';
```

The flag `oracle_maintained='N'` indicates whether the directory object was created by Oracle or someone else.

The non-Oracle objects should be checked carefully for possible problems.

Remediation:

Delete directories that are no longer required

```
DROP DIRECTORY <DIRECTORY_NAME>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/sqlrf/CREATE-DIRECTORY.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

6.8.4 Review External Tables (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

A directory object specifies an alias for a directory on the server file system where external files and data are located.

Rationale:

Unauthorized users with access to these directories can perform actions at the operating system level that bypass normal security controls, potentially leading to data breaches, tampering, or destruction.

Audit:

To assess this recommendation, execute the following SQL statement.

```
SELECT CON_ID_TO_CON_NAME(C.CON_ID) AS CONTAINERNAME,O.ORACLE_MAINTAINED, C.*  
FROM CDB_EXTERNAL_TABLES C  
JOIN CDB_OBJECTS O  
ON C.CON_ID = O.CON_ID AND C.OWNER=O.OWNER  
AND O.OBJECT_NAME=C.TABLE_NAME  
AND O.OBJECT_TYPE='TABLE';
```

The flag `oracle_maintained='N'` indicates whether the directory object was created by Oracle or someone else.

The non-Oracle objects should be checked carefully for possible problems.

Remediation:

Delete directories that are no longer required

```
DROP DIRECTORY <DIRECTORY_NAME>;
```

References:

1. <https://docs.oracle.com/en/database/oracle/oracle-database/26/sqlrf/CREATE-DIRECTORY.html>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.			
v7	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.			

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.

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>*.

```
-- For assessing the container and all its pluggable databases
-- 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 cdb_registry_sqlpatch TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON GV_$SYSTEM_PARAMETER TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON V_$DATABASE TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_PROFILES TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_SOURCE TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_USERS_WITH_DEFPWD TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_USERS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_XS_USERS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON GV_$PASSWORDFILE_INFO TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON GV_$PWFILERS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_DB_LINKS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON SYS.LINK$ TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON DBA_SYS_PRIVS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON V_$CONTAINERS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON AUDITABLE_SYSTEM_ACTIONS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON AUDIT_UNIFIED_POLICIES TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON AUDIT_UNIFIED_ENABLED_POLICIES TO C##CISSCANROLE
CONTAINER=ALL;
GRANT SELECT ON CDB_ROLES TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_SYS_PRIVS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_ROLE_PRIVS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_SCHEMA_PRIVS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_TAB_PRIVS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_COL_PRIVS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_OBJECTS TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_PROXIES TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_JAVA_POLICY TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_DIRECTORIES TO C##CISSCANROLE CONTAINER=ALL;
GRANT SELECT ON CDB_EXTERNAL_TABLES 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;
```

```

-- For assessing only pluggable database
-- Create the role
CREATE ROLE CISPDBSCANROLE;
-- Grant necessary privileges to the role
GRANT CREATE SESSION TO CISPDBSCANROLE;
GRANT SELECT ON cdb_registry_sqlpatch TO CISPDBSCANROLE;
GRANT SELECT ON GV_$SYSTEM_PARAMETER TO CISPDBSCANROLE;
GRANT SELECT ON V_$DATABASE TO CISPDBSCANROLE;
GRANT SELECT ON CDB_PROFILES TO CISPDBSCANROLE;
GRANT SELECT ON CDB_SOURCE TO CISPDBSCANROLE;
GRANT SELECT ON CDB_USERS_WITH_DEFPWD TO CISPDBSCANROLE;
GRANT SELECT ON CDB_USERS TO CISPDBSCANROLE;
GRANT SELECT ON CDB_XS_USERS TO CISPDBSCANROLE;
GRANT SELECT ON GV_$PASSWORDFILE_INFO TO CISPDBSCANROLE;
GRANT SELECT ON GV_$PWFIL_USERS TO CISPDBSCANROLE;
GRANT SELECT ON CDB_DB_LINKS TO CISPDBSCANROLE;
GRANT SELECT ON SYS.LINK$ TO CISPDBSCANROLE;
GRANT SELECT ON DBA_SYS_PRIVS TO CISPDBSCANROLE;
GRANT SELECT ON V_$CONTAINERS TO CISPDBSCANROLE;
GRANT SELECT ON AUDITABLE_SYSTEM_ACTIONS TO CISPDBSCANROLE;
GRANT SELECT ON AUDIT_UNIFIED_POLICIES TO CISPDBSCANROLE;
GRANT SELECT ON AUDIT_UNIFIED_ENABLED_POLICIES TO CISPDBSCANROLE;
GRANT SELECT ON CDB_ROLES TO CISPDBSCANROLE;
GRANT SELECT ON CDB_SYS_PRIVS TO CISPDBSCANROLE;
GRANT SELECT ON CDB_ROLE_PRIVS TO CISPDBSCANROLE;
GRANT SELECT ON CDB_SCHEMA_PRIVS TO CISPDBSCANROLE;
GRANT SELECT ON CDB_TAB_PRIVS TO CISPDBSCANROLE;
GRANT SELECT ON CDB_COL_PRIVS TO CISPDBSCANROLE;
GRANT SELECT ON CDB_OBJECTS TO CISPDBSCANROLE;
GRANT SELECT ON CDB_PROXIES TO CISPDBSCANROLE;
GRANT SELECT ON CDB_JAVA_POLICY TO CISPDBSCANROLE;
GRANT SELECT ON CDB_DIRECTORIES TO CISPDBSCANROLE;
GRANT SELECT ON CDB_EXTERNAL_TABLES TO CISPDBSCANROLE;
-- Create the user and assign the user to the role
CREATE USER CISPDBSCAN IDENTIFIED BY <password>;
GRANT CISPDBSCANROLE TO CISPDBSCAN;

```

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.

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 section as an example for your database.

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.

8.1 All DDL Auditable System Actions Policy

8.1.1 CDB All DDL Auditable System Actions Policy

This policy must be executed in **CDB\$ROOT**.

```
declare
    do_create boolean := true;
    do_sql varchar2(1000);
begin
    for privs in (select name from auditable_system_actions
                  where component='Standard'
                  and name not in ('CREATE PMEM FILESTORE', 'ALTER PMEM FILESTORE', 'DROP
PMEM
FILESTORE', 'LOAD', 'LOGON', 'LOGOFF', 'INSERT', 'UPDATE', 'DELETE', 'SELECT', 'ALL',
'CALL', 'EXECUTE', 'COMMIT', 'ROLLBACK', 'CREATE TABLE'))
    loop
        if do_create then
            --nonPDB      do_sql := 'create audit policy CIS_CDB_DDL_ACTIONS actions '
|| privs.name ;
            -- PDB
            do_sql := 'create audit policy CIS_CDB_DDL_ACTIONS actions ' ||
privs.name || ' CONTAINER=ALL';
            do_create := false;
        else
            do_sql := 'alter audit policy CIS_CDB_DDL_ACTIONS add actions ' ||
privs.name;
        end if;
        begin
            execute immediate do_sql;
        exception
            when others then
                dbms_output.put_line('skipping action "' || privs.name || '" due
to: ' || SQLERRM);
            end;
        end loop;
    end;
/

declare
    -- do_create boolean := true;
    do_sql varchar2(1000);
begin
```

```

        for privs in (select distinct a.privilege as name from dba_sys_privs a
                        where a.privilege not in (select name from auditable_system_actions
                        where component='Standard')
                        and privilege not in ('INHERIT ANY PRIVILEGES','SELECT ANY
DICTIONARY','SELECT ANY TABLE','CREATE TABLE','CREATE ANY TABLE','CREATE
SESSION'))
        ) loop
            -- if do_create then
            -- do_sql := 'create audit policy CIS_CDB_DDL_ACTIONS privileges ' ||
privs.name || ' CONTAINER=ALL';
            dbms_output.put_line(privs.name || ' added');
            do_create := false;
            -- else
            do_sql := 'alter audit policy CIS_CDB_DDL_ACTIONS add privileges ' ||
privs.name;
            dbms_output.put_line(privs.name || ' added');
            -- end if;
            begin
                execute immediate do_sql;
            exception
                when others then
                    dbms_output.put_line('skipping privilege "' || privs.name || '" due
to: ' || SQLERRM);
            end;
        end loop;
    end;
/

audit policy CIS_CDB_DDL_ACTIONS;

```

8.1.2 PDB All DDL Auditable System Actions Policy

This policy must be repeated in every PDB.

```

declare
    do_create boolean := true;
    do_sql varchar2(1000);
begin
    for privs in (select name from auditable_system_actions
                  where component='Standard'
                  and name not in ('CREATE PMEM FILESTORE','ALTER PMEM FILESTORE','DROP
PMEM
FILESTORE','LOAD','LOGON','LOGOFF','INSERT','UPDATE','DELETE','SELECT','ALL',
'CALL','EXECUTE','COMMIT','ROLLBACK','CREATE TABLE'))
    ) loop
        if do_create then
            --nonPDB do_sql := 'create audit policy CIS_PDB_DDL_ACTIONS actions '
|| privs.name ;
            -- PDB
            do_sql := 'create audit policy CIS_PDB_DDL_ACTIONS actions ' ||
privs.name || ' CONTAINER=CURRENT';
            do_create := false;
            -- else
            do_sql := 'alter audit policy CIS_PDB_DDL_ACTIONS add actions ' ||
privs.name;
            -- end if;
        end if;
    end loop;
end;

```

```

begin
    execute immediate do_sql;
exception
    when others then
        dbms_output.put_line('skipping action "' || privs.name || '" due
to: ' || SQLERRM);
    end;
end loop;
end;
/

declare
-- do_create boolean := true;
do_sql varchar2(1000);
begin
    for privs in (select distinct a.privilege as name from dba_sys_privs a
        where a.privilege not in (select name from auditable_system_actions
where component='Standard')
        and privilege not in ('INHERIT ANY PRIVILEGES','SELECT ANY
DICTIONARY','SELECT ANY TABLE','CREATE TABLE','CREATE ANY TABLE','CREATE
SESSION')
    ) loop
        -- if do_create then
        --     do_sql := 'create audit policy CIS_PDB_DDL_ACTIONS privileges ' ||
privs.name || ' CONTAINER=CURRENT';
        --     dbms_output.put_line(privs.name || ' added');
        --     do_create := false;
        --     else
        do_sql := 'alter audit policy CIS_PDB_DDL_ACTIONS add privileges ' ||
privs.name;
        dbms_output.put_line(privs.name || ' added');
        --     end if;
        begin
            execute immediate do_sql;
        exception
            when others then
                dbms_output.put_line('skipping privilege "' || privs.name || '" due
to: ' || SQLERRM);
            end;
        end loop;
    end;
/

audit policy CIS_PDB_DDL_ACTIONS;

```

8.2 Logon/Logoff Audit Policy

8.2.1 CDB Logon/Logoff Audit Policy

This policy must be executed in **CDB\$ROOT**.

```

CREATE AUDIT POLICY CIS_CDB_LOGON_LOGOFF
ACTIONS
LOGON, LOGOFF
ACTIONS

```

```

COMPONENT=PROTOCOL HTTP, FTP, AUTHENTICATION
ONLY TOPLEVEL
CONTAINER=ALL;

AUDIT POLICY CIS_CDB_LOGON_LOGOFF;

COMMENT ON AUDIT POLICY CIS_CDB_LOGON_LOGOFF IS 'Audit policy for LOGON and
LOGOFF Events';

```

8.2.2 PDB Logon/Logoff Audit Policy

This policy must be repeated in every PDB.

```

CREATE AUDIT POLICY CIS_PDB_LOGON_LOGOFF
ACTIONS
LOGON, LOGOFF
ACTIONS
COMPONENT=PROTOCOL HTTP, FTP, AUTHENTICATION
ONLY TOPLEVEL
CONTAINER=CURRENT;

AUDIT POLICY CIS_PDB_LOGON_LOGOFF;

COMMENT ON AUDIT POLICY CIS_PDB_LOGON_LOGOFF IS 'Audit policy for LOGON and
LOGOFF Events';

```

8.3 Audit Usage Of Critical Packages

8.3.1 CDB Critical Packages Audit Policy

This policy must be executed in **CDB\$ROOT**.

```

CREATE AUDIT POLICY CIS_CDB_CRITICAL_PACKAGES
ACTIONS
EXECUTE ON SYS.DBMS_AW,
EXECUTE ON SYS.DBMS_CRYPTO,
EXECUTE ON SYS.DBMS_FGA,
EXECUTE ON SYS.DBMS_JAVA_TEST,
EXECUTE ON SYS.DBMS_JOB,
EXECUTE ON SYS.DBMS_LOGMNR,
EXECUTE ON SYS.DBMS_NETWORK_ACL_ADMIN,
EXECUTE ON SYS.DBMS_REDACT,
EXECUTE ON SYS.DBMS_REDEFINITION,
EXECUTE ON SYS.DBMS_RLS,
EXECUTE ON SYS.DBMS_SCHEDULER,
EXECUTE ON SYS.DBMS_SQL_TRANSLATOR,
EXECUTE ON SYS.DBMS_TSDP_MANAGE,
EXECUTE ON SYS.DBMS_TSDP_PROTECT,
EXECUTE ON SYS.DBMS_XMLGEN,
EXECUTE ON SYS.DBMS_XMLSTORE,
EXECUTE ON SYS.OWA_UTIL
ONLY TOPLEVEL
CONTAINER=ALL;

```

```
AUDIT POLICY CIS_CDB_CRITICAL_PACKAGES;  
  
COMMENT ON AUDIT POLICY CIS_CDB_CRITICAL_PACKAGES IS 'Audit policy for  
critical Oracle Packages';
```

8.3.2 PDB Critical Packages Audit Policy

This policy must be repeated in every PDB.

```
CREATE AUDIT POLICY CIS_PDB_CRITICAL_PACKAGES  
ACTIONS  
EXECUTE ON SYS.DBMS_AW,  
EXECUTE ON SYS.DBMS_CRYPTO,  
EXECUTE ON SYS.DBMS_FGA,  
EXECUTE ON SYS.DBMS_JAVA_TEST,  
EXECUTE ON SYS.DBMS_JOB,  
EXECUTE ON SYS.DBMS_LOGMNR,  
EXECUTE ON SYS.DBMS_NETWORK_ACL_ADMIN,  
EXECUTE ON SYS.DBMS_REDACT,  
EXECUTE ON SYS.DBMS_REDEFINITION,  
EXECUTE ON SYS.DBMS_RLS,  
EXECUTE ON SYS.DBMS_SCHEDULER,  
EXECUTE ON SYS.DBMS_SQL_TRANSLATOR,  
EXECUTE ON SYS.DBMS_TSDP_MANAGE,  
EXECUTE ON SYS.DBMS_TSDP_PROTECT,  
EXECUTE ON SYS.DBMS_XMLGEN,  
EXECUTE ON SYS.DBMS_XMLSTORE,  
EXECUTE ON SYS.OWA_UTIL  
ONLY TOPLEVEL  
CONTAINER=current;  
  
AUDIT POLICY CIS_PDB_CRITICAL_PACKAGES;  
  
COMMENT ON AUDIT POLICY CIS_PDB_CRITICAL_PACKAGES IS 'Audit policy for  
critical Oracle Packages';
```

8.4 Audit All Export Activities

8.4.1 CDB Export Audit Policy

This policy must be executed in **CDB\$ROOT**.

```
CREATE AUDIT POLICY CIS_CDB_EXPORT  
ACTIONS  
COMPONENT=DATAPUMP ALL  
ACTIONS  
COMPONENT=DIRECT_LOAD ALL  
CONTAINER=ALL;  
  
AUDIT POLICY CIS_CDB_EXPORT;  
  
COMMENT ON AUDIT POLICY CIS_CDB_EXPORT IS 'Audit policy Export and direct  
Load';
```

8.4.2 PDB Export Audit Policy

This policy must be repeated in every PDB.

```
CREATE AUDIT POLICY CIS_PDB_EXPORT
ACTIONS
COMPONENT=DATAPUMP ALL
ACTIONS
COMPONENT=DIRECT_LOAD ALL
CONTAINER=CURRENT;

AUDIT POLICY CIS_PDB_EXPORT;

COMMENT ON AUDIT POLICY CIS_PDB_EXPORT IS 'Audit policy Export and direct
Load';
```

8.5 Audit 'SYS*-Privileges'

This policy audits all **SYS** roles, such as **SYSDBA**, **SYSKM**, **SYSBACKUP**, **SYSRAC**, **SYSDBG** and **SYSOPER** (displayed as **PUBLIC** in the **Audit.log**).

The policy must be activated (and deactivated) for all roles.

8.5.1 CDB 'SYS*-Privileges' Audit Policy

This policy must be executed in **CDB\$ROOT**.

```
CREATE AUDIT POLICY CIS_CDB_ALL_ACTIONS_BY_PRIVILEGED_USERS
ACTIONS ALL
WHEN q'! (SYS_CONTEXT('USERENV','CLIENT_PROGRAM_NAME') NOT IN ('emagent') AND
INSTR(UPPER(SYS_CONTEXT('USERENV','CLIENT_PROGRAM_NAME')), 'PERL') = 0 AND
INSTR(UPPER(SYS_CONTEXT('USERENV','CLIENT_PROGRAM_NAME')), 'RMAN') = 0 AND
INSTR(UPPER(SYS_CONTEXT('USERENV','CLIENT_PROGRAM_NAME')), 'OMS') = 0)!'
EVALUATE PER SESSION
ONLY TOPLEVEL;

AUDIT POLICY CIS_CDB_ALL_ACTIONS_BY_PRIVILEGED_USERS BY SYS, SYSKM,
SYSBACKUP, SYSRAC, SYSDBG, PUBLIC;

COMMENT ON AUDIT POLICY CIS_CDB_ALL_ACTIONS_BY_PRIVILEGED_USERS IS 'Audit
policy for privileged users';
```

8.5.2 PDB 'SYS*-Privileges' Audit Policy

This policy must be repeated in every PDB.

```
CREATE AUDIT POLICY CIS_PDB_ALL_ACTIONS_BY_PRIVILEGED_USERS
ACTIONS ALL
WHEN q'! (SYS_CONTEXT('USERENV','CLIENT_PROGRAM_NAME') NOT IN ('emagent') AND
INSTR(UPPER(SYS_CONTEXT('USERENV','CLIENT_PROGRAM_NAME')), 'PERL') = 0 AND
INSTR(UPPER(SYS_CONTEXT('USERENV','CLIENT_PROGRAM_NAME')), 'RMAN') = 0 AND
INSTR(UPPER(SYS_CONTEXT('USERENV','CLIENT_PROGRAM_NAME')), 'OMS') = 0)!'
```

```
EVALUATE PER SESSION  
ONLY TOPLEVEL;
```

```
AUDIT POLICY CIS_PDB_ALL_ACTIONS_BY_PRIVILEGED_USERS BY SYS, SYSKM,  
SYSBACKUP, SYSRAC, SYSDG, PUBLIC;
```

```
COMMENT ON AUDIT POLICY CIS_PDB_ALL_ACTIONS_BY_PRIVILEGED_USERS IS 'Audit  
policy for privileged users';
```

9 Appendix: Comprehensive SQL To Identify Direct And Indirect System Privileges Granted To Users And Roles.

This section has been kindly authored by Alexander Kornbrust.

```
SELECT
con_id,
PRIVILEGE,
OBJ_OWNER,
OBJ_NAME,
USERNAME,
LISTAGG(GRANT_TARGET, ',') WITHIN GROUP (ORDER BY GRANT_TARGET) AS
GRANT_SOURCES, -- Lists the sources of the permission
MAX(ADMIN_OR_GRANT_OPT) AS ADMIN_OR_GRANT_OPT, -- MAX acts as a Boolean OR by
picking 'YES' over 'NO'
MAX(HIERARCHY_OPT) AS HIERARCHY_OPT, -- MAX acts as a Boolean OR by picking
'YES' over 'NO',
oracle_maintained
FROM (
-- Gets all roles a user has, even inherited ones
WITH ALL_ROLES_FOR_CDB_USER AS (
SELECT DISTINCT p.con_id,CONNECT_BY_ROOT GRANTEE AS GRANTED_USER,
GRANTED_ROLE, u.oracle_maintained
FROM CDB_ROLE_PRIVS p right outer join CDB_USERS u on (p.con_id=u.con_id and
p.grantee=u.username)
CONNECT BY GRANTEE = PRIOR GRANTED_ROLE
)
SELECT
con_id,
PRIVILEGE,
OBJ_OWNER,
OBJ_NAME,
USERNAME,
REPLACE(GRANT_TARGET, USERNAME, 'Direct to user') AS GRANT_TARGET,
ADMIN_OR_GRANT_OPT,
HIERARCHY_OPT,
oracle_maintained
FROM (
-- System privileges granted directly to users
SELECT distinct u.con_id,PRIVILEGE, NULL AS OBJ_OWNER, NULL AS OBJ_NAME,
GRANTEE AS USERNAME, GRANTEE AS GRANT_TARGET, ADMIN_OPTION AS
ADMIN_OR_GRANT_OPT, NULL AS HIERARCHY_OPT, oracle_maintained
FROM CDB_SYS_PRIVS p join CDB_USERS u on (p.con_id=u.con_id and
p.grantee=u.username)
UNION ALL
-- System privileges granted users through roles
SELECT CDB_SYS_PRIVS.con_id, PRIVILEGE, NULL AS OBJ_OWNER, NULL AS OBJ_NAME,
ALL_ROLES_FOR_CDB_USER.GRANTED_USER AS USERNAME, GRANTEE AS GRANT_TARGET,
ADMIN_OPTION AS ADMIN_OR_GRANT_OPT, NULL AS HIERARCHY_OPT,oracle_maintained
FROM CDB_SYS_PRIVS
JOIN ALL_ROLES_FOR_CDB_USER ON (ALL_ROLES_FOR_CDB_USER.GRANTED_ROLE =
CDB_SYS_PRIVS.GRANTEE and ALL_ROLES_FOR_CDB_USER.CON_ID =
CDB_SYS_PRIVS.CON_ID)
```

```
) ALL_USER_PRIVS
-- Adjust your filter here
WHERE privilege in ('EXEMPT ACCESS POLICY','EXEMPT REDACTION POLICY')
and oracle_maintained='N'
) DISTINCT_USER_PRIVS

GROUP BY
con_id,
PRIVILEGE,
OBJ_OWNER,
OBJ_NAME,
USERNAME,
oracle_maintained;
```

Appendix: Summary Table

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
1	Oracle Database Installation and Patching Requirements		
1.1	Ensure That Appropriate Version/Patches For Oracle Software Are Installed (Manual)	☐	☐
2	Oracle Parameter Settings		
2.1	Listener Settings		
2.1.1	Ensure 'extproc' Is Not Present In 'listener.ora' (Automated)	☐	☐
2.1.2	Ensure 'ACCEPT_MD5_CERTS' Is Configured Correctly (Automated)	☐	☐
2.1.3	Ensure 'ACCEPT_SHA1_CERTS' Is Configured Correctly (Automated)	☐	☐
2.1.4	Ensure 'ALLOWED_WEAK_CERT_ALGORITHMS' Is NOT Set. (Automated)	☐	☐
2.2	SQLNET.ORA Settings		
2.2.1	Ensure 'ACCEPT_MD5_CERTS' Is NOT SET (Automated)	☐	☐
2.2.2	Ensure 'ACCEPT_SHA1_CERTS' Is NOT Set (Automated)	☐	☐
2.2.3	Ensure 'ALLOWED_WEAK_CERT_ALGORITHMS' Is NOT Set (Automated)	☐	☐
2.2.4	Ensure 'SQLNET.ALLOWED_LOGON_VERSION_CLIENT' Is Set To 12a (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
2.2.5	Ensure 'SQLNET.ALLOWED_LOGON_VERSION_SERVER' Is Set To 12a (Automated)	☐	☐
2.2.6	Ensure 'SQLNET.ENCRYPTION_CLIENT' Is Set To 'REQUIRED' (Automated)	☐	☐
2.2.7	Ensure 'SQLNET.ENCRYPTION_SERVER' Is Set To 'REQUIRED' (Automated)	☐	☐
2.2.8	Ensure 'SQLNET.ENCRYPTION_TYPES_CLIENT' Is Set To 'AES256' (Automated)	☐	☐
2.2.9	Ensure 'SQLNET.ENCRYPTION_TYPES_SERVER' Is Set To AES256 (Manual)	☐	☐
2.2.10	Ensure 'SQLNET.CRYPTO_CHECKSUM_CLIENT' Is Set To 'REQUIRED' (Automated)	☐	☐
2.2.11	Ensure 'SQLNET.CRYPTO_CHECKSUM_SERVER' Is Set To 'REQUIRED' (Automated)	☐	☐
2.2.12	Ensure 'SSL_CERT_REVOCATION' Is Set To 'REQUIRED' (Automated)	☐	☐
2.3	Database Settings		
2.3.1	Ensure 'BACKGROUND_CORE_DUMP' Is Not Set To 'Full' (Automated)	☐	☐
2.3.2	Ensure 'SHADOW_CORE_DUMP' Is Not Set To 'Full' (Automated)	☐	☐
2.3.3	Ensure 'MLE_PROG_LANGUAGES' Is Set To 'OFF' (Automated)	☐	☐
2.3.4	Ensure 'ALLOW_GROUP_ACCESS_TO_SGA' Is Set To 'FALSE' (Automated)	☐	☐
2.3.5	Review Undocumented (Underscore) Parameters Not Set To 'DEFAULT' Values (Manual)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
2.3.6	Ensure 'OS_ROLES' Is Set To 'FALSE' (Automated)	☐	☐
2.3.7	Ensure 'REMOTE_OS_ROLES' Is Set To 'FALSE' (Automated)	☐	☐
2.3.8	Ensure 'SEC_MAX_FAILED_LOGIN_ATTEMPTS' Is Set To '3' Or Less (Automated)	☐	☐
2.3.9	Ensure 'SEC_PROTOCOL_ERROR_FURTHER_ACTION' Is Set To '(DROP,3)' (Automated)	☐	☐
2.3.10	Ensure 'SEC_PROTOCOL_ERROR_TRACE_ACTION' Is Set To 'LOG' (Automated)	☐	☐
2.3.11	Ensure 'SEC_RETURN_SERVER_RELEASE_BANNER' Is Set To 'FALSE' (Automated)	☐	☐
2.3.12	Ensure 'REMOTE_LOGIN_PASSWORDFILE' Is Set To 'NONE' (Automated)	☐	☐
2.3.13	Ensure 'REMOTE_LISTENER' Is Empty (Automated)	☐	☐
2.3.14	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 + PASSWORD_GRACE_TIME' Is Less Than Or Equal To '365' (Automated)	☐	☐
3.4	Ensure 'PASSWORD_REUSE_MAX' Is Set To 'UNLIMITED' (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
3.5	Ensure 'PASSWORD_VERIFY_FUNCTION' Is Set For All Profiles (Automated)	☐	☐
3.6	Ensure 'PASSWORD_VERIFY_FUNCTION' Is Configured Correctly (Manual)	☐	☐
3.7	Ensure 'PASSWORD_ROLLOVER_TIME' Is set to '0' (Automated)	☐	☐
3.8	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 No Custom 'ORACLE_MAINTAINED' Users Exist (Automated)	☐	☐
4.3	Review The Users Created Through Real Application Security (Manual)	☐	☐
4.4	Ensure Old Password Versions Are Not Used (Automated)	☐	☐
4.5	Ensure The Latest Version of The Password File Is Used (Automated)	☐	☐
4.6	Ensure That Users In Different RAC Instances Are Identical In PW Files (Automated)	☐	☐
4.7	Ensure No Public Database Links Exist (Automated)	☐	☐
4.8	Ensure That Database Link Passwords Are Using The Latest Encryption (Automated)	☐	☐
5	Unified Auditing		
5.1	Ensure All Auditable System Actions Commands Are Audited (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
5.2	Ensure the 'LOGON' AND 'LOGOFF' Actions Audit Is Enabled (Automated)	☐	☐
5.3	Ensure Critical Packages Are Audited (Automated)	☐	☐
5.4	Ensure All Export Activities Are Audited (Automated)	☐	☐
5.5	Ensure The Use Of SYS* Privileges Is Audited (Automated)	☐	☐
6	Privileges & Grants & ACLs		
6.1	Excessive System Privileges		
6.1.1	Ensure '%ANY%' Is Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐
6.1.2	Ensure Admin Privileges Are Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐
6.1.3	Ensure 'IMPORT' And 'EXPORT' 'FULL DATABASE' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.1.4	Ensure 'CREATE EXTERNAL JOB' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.1.5	Ensure 'BECOME USER' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.1.6	Ensure 'TEXT DATASTORE ACCESS' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.1.7	Ensure 'CREATE', 'ALTER', And 'DROP' 'PUBLIC DATABASE LINK' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.1.8	Ensure 'LOGMINING' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.1.9	Ensure 'ALTER SYSTEM' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
6.1.10	Ensure 'CREATE LIBRARY' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.1.11	Ensure All 'SYSTEM' Privileges Are Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2	Excessive Role Privileges		
6.2.1	Ensure 'DBA' Is Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.2	Ensure 'EXP_FULL_DATABASE' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.3	Ensure 'IMP_FULL_DATABASE' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.4	Ensure 'DATAPUMP_EXP_FULL_DATABASE' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.5	Ensure 'DATAPUMP_IMP_FULL_DATABASE' is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.6	Ensure 'DV_ADMIN' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.7	Ensure 'DV_AUDIT_CLEANUP' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.8	Ensure 'OLAP_DBA' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.9	Ensure 'LBAC_DBA' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.10	Ensure 'JAVA_ADMIN' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.11	Ensure 'JAVASYSPRIVS' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
6.2.12	Ensure 'LOGSTDBY_ADMINISTRATOR' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.13	Ensure 'SQL_FIREWALL_ADMIN' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.14	Ensure 'MAINTPLAN_APP' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.15	Ensure 'JAVADEBUGPRIV' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.16	Ensure 'DV_PATCH_ADMIN' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.17	Ensure 'DV_POLICY_OWNER' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.18	Ensure AUDIT_ADMIN' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.19	Ensure 'AUDIT_VIEWER' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.20	Ensure 'PDB_DBA' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.21	Ensure 'SELECT_CATALOG_ROLE' Is Revoked From Unauthorized 'GRANTEE' (Automated)	☐	☐
6.2.22	Ensure 'EXECUTE_CATALOG_ROLE' Is Revoked from Unauthorized 'GRANTEE' (Automated)	☐	☐
6.3	Excessive Schema Privileges		
6.3.1	Ensure 'CDB_SCHEMA_PRIVS' Does Not Have Unauthorized Privileges (Manual)	☐	☐
6.4	Excessive Object Privileges		
6.4.1	Ensure 'ALL' Is Revoked On 'Sensitive' Tables (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
6.5	Excessive Column Privileges		
6.5.1	Ensure 'DBA_COL_PRIVS' Is Revoked from Unauthorized 'GRANTEE' (Manual)	☐	☐
6.6	Excessive Proxy Privileges		
6.6.1	Ensure Proxy User Privileges Are Revoked from Unauthorized 'GRANTEE' (Manual)	☐	☐
6.7	Excessive Java Privileges		
6.7.1	Ensure Custom Java Privileges Are Revoked from Unauthorized 'GRANTEE' (Manual)	☐	☐
6.8	Excessive Directory Privileges		
6.8.1	Ensure Directory Object Access Is Revoked From Unauthorized 'GRANTEE' (Manual)	☐	☐
6.8.2	Review Directory Objects Privileges (Manual)	☐	☐
6.8.3	Review External Tables With Preprocessor (Manual)	☐	☐
6.8.4	Review External Tables (Manual)	☐	☐
7	Appendix: Establishing an Audit/Scan User		
8	Appendix: Establishing a Unified Audit Policy		
8.1	All DDL Auditable System Actions Policy		
8.1.1	CDB All DDL Auditable System Actions Policy		
8.1.2	PDB All DDL Auditable System Actions Policy		
8.2	Logon/Logoff Audit Policy		
8.2.1	CDB Logon/Logoff Audit Policy		
8.2.2	PDB Logon/Logoff Audit Policy		

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
8.3	Audit Usage Of Critical Packages		
8.3.1	CDB Critical Packages Audit Policy		
8.3.2	PDB Critical Packages Audit Policy		
8.4	Audit All Export Activities		
8.4.1	CDB Export Audit Policy		
8.4.2	PDB Export Audit Policy		
8.5	Audit 'SYS*-Privileges'		
8.5.1	CDB 'SYS*-Privileges' Audit Policy		
8.5.2	PDB 'SYS*-Privileges' Audit Policy		
9	Appendix: Comprehensive SQL To Identify Direct And Indirect System Privileges Granted To Users And Roles.		

Appendix: Change History

Date	Version	Changes for this version
27 Mar 2026	1.0.0	Initial Release