

CIS IBM Db2 12.1 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	8
Important Usage Information	8
Key Stakeholders.....	8
Apply the Correct Version of a Benchmark	9
Exceptions	9
Remediation	10
Summary.....	10
Target Technology Details	11
Intended Audience	11
Consensus Guidance.....	12
Typographical Conventions	13
Recommendation Definitions	14
Title.....	14
Assessment Status	14
Automated	14
Manual.....	14
Profile.....	14
Description	14
Rationale Statement.....	14
Impact Statement.....	15
Audit Procedure.....	15
Remediation Procedure	15
Default Value	15
References	15
CIS Critical Security Controls® (CIS Controls®).....	15
Additional Information	15
Profile Definitions	16
Acknowledgements.....	17
Recommendations	18
1 Staying Current.....	18
1.1 General Considerations	18
1.1.1 Install Available Updates (Manual).....	18
2 Securing the Server Environment.....	20
2.1 Prevent Database Users from Logging into the Operating System (Manual).....	20

3 Securing the Server Instance.....	24
3.1 Database Manager Configuration Parameters	24
3.1.1 Require Explicit Authorization for Cataloging (CATALOG_NOAUTH) (Automated).....	25
3.1.2 Secure Permissions for Default Database File Path (DFTDBPATH) (Automated).....	27
3.1.3 Set Diagnostic Logging to Capture Errors and Warnings (DIAGLEVEL) (Automated) ...	30
3.1.4 Secure Permissions for All Diagnostic Logs (DIAGPATH) (Automated)	32
3.1.5 Secure Permissions for Alternate Diagnostic Log Path (ALT_DIAGPATH) (Automated)	35
3.1.6 Set Maximum Connection Limits (MAX_CONNECTIONS and MAX_COORDAGENTS) (Automated)	38
3.1.7 Set Administrative Notification Level (NOTIFYLEVEL) (Automated).....	41
3.1.8 Secure the Java Development Kit Installation Path (JDK_PATH) (Automated)	43
3.1.9 Secure the Python Runtime Path (PYTHON_PATH) (Automated).....	46
3.1.10 Secure the R Runtime Path (R_PATH) (Automated).....	49
3.1.11 Secure the Communication Buffer Exit Library (COMM_EXIT_LIST) (Automated)	52
3.2 Db2 Registry Values	54
3.2.1 Specify Secure Remote Shell Command (DB2RSHCMD) (Automated)	55
3.2.2 Turn Off Remote Command Legacy Mode (DB2RCMD_LEGACY_MODE) (Automated)	57
3.2.3 Disable Grants During Restore (DB2_RESTORE_GRANT_ADMIN_AUTHORITIES) (Automated)	59
3.2.4 Enable Extended Security (DB2_EXTSECURITY) (Automated)	61
3.2.5 Limit OS Privileges of Fenced Mode Process (DB2_LIMIT_FENCED_GROUP) (Automated)	63
3.3 General Considerations	65
3.3.1 Secure Db2 Runtime Library (Manual).....	65
3.3.2 Secure the Database Container Directory (Manual)	67
3.3.3 Set umask Value in the Db2 Instance Owner's .profile (Automated).....	68
4 Securing the Database.....	69
4.1 Database Configuration Parameters	70
4.1.1 Creating the Database Without PUBLIC Grants (RESTRICTIVE) (Automated).....	71
4.1.2 Set Failed Archive Retry Delay (ARCHRETRYDELAY) (Automated)	74
4.1.3 Auto-restart After Abnormal Termination (AUTORESTART) (Automated).....	76
4.1.4 Secure Permissions for the Primary Archive Log Location (LOGARCHMETH1) (Automated)	78
4.1.5 Secure Permissions for the Secondary Archive Log Location (LOGARCHMETH2) (Automated)	81
4.1.6 Secure Permissions for the Tertiary Archive Log Location (FAILARCHPATH) (Automated)	84
4.1.7 Secure Permissions for the Log Mirror Location (MIRRORLOGPATH) (Automated)	87
4.1.8 Secure Permissions for the Log Overflow Location (OVERFLOWLOGPATH) (Manual)	90
4.1.9 Establish Retention Set Size for Backups (NUM_DB_BACKUPS) (Manual)	93
4.1.10 Set Archive Log Failover Retry Limit (NUMARCHRETRY) (Automated).....	95
4.1.11 Set Maximum Number of Applications (MAXAPPLS) (Automated)	97
4.1.12 Ensure a Secure Connect Procedure is Used (CONNECT_PROC) (Manual)	99
4.1.13 Specify a Secure Location for External Tables (EXTBL_LOCATION) (Manual)	101
4.2 Secure the Database Catalog Views.....	103
4.2.1 Restrict Access to SYSCAT.AUDITPOLICIES (Automated).....	104
4.2.2 Restrict Access to SYSCAT.AUDITUSE (Automated).....	106
4.2.3 Restrict Access to SYSCAT.COLAUTH (Automated)	108
4.2.4 Restrict Access to SYSCAT.COLDIST (Automated)	110
4.2.5 Restrict Access to SYSCAT.COLGROUPDIST (Automated)	112
4.2.6 Restrict Access to SYSCAT.COLUMNS (Automated)	114
4.2.7 Restrict Access to SYSCAT.CONTEXTATTRIBUTES (Automated)	116
4.2.8 Restrict Access to SYSCAT.CONTEXTS (Automated)	118
4.2.9 Restrict Access to SYSCAT.CONTROLDEP (Automated)	120

4.2.10 Restrict Access to SYSCAT.CONTROLS (Automated)	122
4.2.11 Restrict Access to SYSCAT.DBAUTH (Automated)	124
4.2.12 Restrict Access to SYSCAT.EVENTS (Automated)	126
4.2.13 Restrict Access to SYSCAT.EVENTTABLES (Automated)	128
4.2.14 Restrict Access to SYSCAT.EXTERNALTABLEOPTIONS (Automated)	130
4.2.15 Restrict Access to SYSCAT.INDEXAUTH (Automated)	132
4.2.16 Restrict Access to SYSCAT.MODULEAUTH (Automated)	134
4.2.17 Restrict Access to SYSCAT.PACKAGEAUTH (Automated)	136
4.2.18 Restrict Access to SYSCAT.PACKAGES (Automated)	138
4.2.19 Restrict Access to SYSCAT.PASSTHRUAUTH (Automated)	140
4.2.20 Restrict Access to SYSCAT.ROLEAUTH (Automated)	142
4.2.21 Restrict Access to SYSCAT.ROLES (Automated)	144
4.2.22 Restrict Access to SYSCAT.ROUTINEAUTH (Automated)	146
4.2.23 Restrict Access to SYSCAT.ROUTINES (Automated)	148
4.2.24 Restrict Access to SYSCAT.SECURITYLABELACCESS (Automated)	150
4.2.25 Restrict Access to SYSCAT.SECURITYLABELCOMPONENTELEMENTS (Automated)	152
4.2.26 Restrict Access to SYSCAT.SECURITYLABELCOMPONENTS (Automated)	154
4.2.27 Restrict Access to SYSCAT.SECURITYLABELS (Automated)	156
4.2.28 Restrict Access to SYSCAT.SECURITYPOLICIES (Automated)	158
4.2.29 Restrict Access to SYSCAT.SECURITYPOLICYCOMPONENTRULES (Automated)	160
4.2.30 Restrict Access to SYSCAT.SECURITYPOLICYEXEMPTIONS (Automated)	162
4.2.31 Restrict Access to SYSCAT.SERVEROPTIONS (Automated)	164
4.2.32 Restrict Access to SYSCAT.SCHEMAAUTH (Automated)	166
4.2.33 Restrict Access to SYSCAT.SCHEMATA (Automated)	168
4.2.34 Restrict Access to SYSCAT.SEQUENCEAUTH (Automated)	170
4.2.35 Restrict Access to SYSCAT.STATEMENTS (Automated)	172
4.2.36 Restrict Access to SYSCAT.STATEMENTTEXTS (Automated)	174
4.2.37 Restrict Access to SYSCAT.SURROGATEAUTHIDS (Automated)	176
4.2.38 Restrict Access to SYSCAT.TBAUTH (Automated)	178
4.2.39 Restrict Access to SYSCAT.TBSPACEAUTH (Automated)	180
4.2.40 Restrict Access to SYSCAT.USEROPTIONS (Automated)	182
4.2.41 Restrict Access to SYSCAT.VARIABLEAUTH (Automated)	184
4.2.42 Restrict Access to SYSCAT.VARIABLES (Automated)	186
4.2.43 Restrict Access to SYSCAT.WORKLOADAUTH (Automated)	188
4.2.44 Restrict Access to SYSCAT.WRAPOPTIONS (Automated)	190
4.2.45 Restrict Access to SYSCAT.XSROBJECTAUTH (Automated)	192
4.2.46 Restrict Access to SYSSTAT.COLDIST (Automated)	194
4.2.47 Restrict Access to SYSSTAT.COLGROUPDIST (Automated)	196
4.2.48 Restrict Access to SYSSTAT.COLUMNS (Automated)	198
4.2.49 Restrict Access to SYSCAT.AUDITEXCEPTIONS (Manual)	200
4.2.50 Restrict Access to SYSCAT.BUFFERPOOLAUTH (Automated)	202
4.3 Secure the Database Catalog Tables	204
4.3.1 Restrict Access to SYSIBM.SYSAUDITPOLICIES (Automated)	205
4.3.2 Restrict Access to SYSIBM.SYSAUDITUSE (Automated)	207
4.3.3 Restrict Access to SYSIBM.SYSCOLAUTH (Automated)	209
4.3.4 Restrict Access to SYSIBM.SYSCOLDIST (Automated)	211
4.3.5 Restrict Access to SYSIBM.SYSCOLGROUPDIST (Automated)	213
4.3.6 Restrict Access to SYSIBM.SYSCOLUMNS (Automated)	215
4.3.7 Restrict Access to SYSIBM.SYSCONTEXTATTRIBUTES (Automated)	217
4.3.8 Restrict Access to SYSIBM.SYSCONTEXTS (Automated)	219
4.3.9 Restrict Access to SYSIBM.SYSDEPENDENCIES (Automated)	221
4.3.10 Restrict Access to SYSIBM.SYSCONTROLS (Automated)	223
4.3.11 Restrict Access to SYSIBM.SYSDBAUTH (Automated)	225
4.3.12 Restrict Access to SYSIBM.SYSEVENTS (Automated)	227
4.3.13 Restrict Access to SYSIBM.SYSEVENTTABLES (Automated)	229

4.3.14 Restrict Access to SYSIBM.SYSEXTTAB (Automated).....	231
4.3.15 Restrict Access to SYSIBM.SYSINDEXAUTH (Automated).....	233
4.3.16 Restrict Access to SYSIBM.SYSMODULEAUTH (Automated)	235
4.3.17 Restrict Access to SYSIBM.SYSPASSTHROUGH (Automated)	237
4.3.18 Restrict Access to SYSIBM.SYSPLANAUTH (Automated)	239
4.3.19 Restrict Access to SYSIBM.SYSPLAN (Automated)	241
4.3.20 Restrict Access to SYSIBM.SYSROLEAUTH (Automated)	243
4.3.21 Restrict Access to SYSIBM.SYSROLES (Automated)	245
4.3.22 Restrict Access to SYSIBM.SYSROUTINEAUTH (Automated)	247
4.3.23 Restrict Access to SYSIBM.SYSROUTINES (Automated)	249
4.3.24 Restrict Access to SYSIBM.ROUTINES_S (Automated).....	251
4.3.25 Restrict Access to SYSIBM.SYSSCHEMAAUTH (Automated)	253
4.3.26 Restrict Access to SYSIBM.SYSSCHEMATA (Automated)	255
4.3.27 Restrict Access to SYSIBM.SYSSECURITYLABELACCESS (Automated)	257
4.3.28 Restrict Access to SYSIBM.SYSSECURITYLABELCOMPONENTELEMENTS (Automated)	259
4.3.29 Restrict Access to SYSIBM.SYSSECURITYLABELCOMPONENTS (Automated)	261
4.3.30 Restrict Access to SYSIBM.SYSSECURITYLABELS (Automated).....	263
4.3.31 Restrict Access to SYSIBM.SYSSECURITYPOLICIES (Automated).....	265
4.3.32 Restrict Access to SYSIBM.SYSSECURITYPOLICYCOMPONENTRULES (Automated)	267
4.3.33 Restrict Access to SYSIBM.SYSSECURITYPOLICYEXEMPTIONS (Automated)	269
4.3.34 Restrict Access to SYSIBM.SYSSERVEROPTIONS (Automated)	271
4.3.35 Restrict Access to SYSIBM.SYSSEQUENCEAUTH (Automated)	273
4.3.36 Restrict Access to SYSIBM.SYSSTATEMENTTEXTS (Automated)	275
4.3.37 Restrict Access to SYSIBM.SYSSTMT (Automated).....	277
4.3.38 Restrict Access to SYSIBM.SYSSURROGATEAUTHIDS (Automated).....	279
4.3.39 Restrict Access to SYSIBM.SYSTABAUTH (Automated).....	281
4.3.40 Restrict Access to SYSIBM.SYSTBSPACEAUTH (Automated)	283
4.3.41 Restrict Access to SYSIBM.SYSUSEROPTIONS (Automated)	285
4.3.42 Restrict Access to SYSIBM.SYSVARIABLEAUTH (Automated)	287
4.3.43 Restrict Access to SYSIBM.SYSVARIABLES (Automated)	289
4.3.44 Restrict Access to SYSIBM.SYSWORKLOADAUTH (Automated).....	291
4.3.45 Restrict Access to SYSIBM.SYSWRAPOPTIONS (Automated)	293
4.3.46 Restrict Access to SYSIBM.SYSXSROBJECTAUTH (Automated)	295
4.3.47 Restrict Access to SYSIBM.SYSAUDITEXCEPTIONS (Automated)	297
4.3.48 Restrict Access to SYSIBM.SYSBUFFERPOOLAUTH (Automated)	299
4.4 Secure the Database Administrative Views and Routines	301
4.4.1 Restrict Access to SYSIBMADM.AUTHORIZATIONIDS (Automated)	302
4.4.2 Restrict Access to SYSIBMADM.OBJECTOWNERS (Automated)	304
4.4.3 Restrict Access to SYSIBMADM.PRIVILEGES (Automated)	306
4.4.4 Restrict Access to SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID (Automated)	308
4.4.5 Restrict Access to SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID (Automated)	310
4.4.6 Restrict Access to SYSPROC.AUTH_LIST_GROUPS_FOR_AUTHID (Automated) ...	312
4.5 General Database Considerations	314
4.5.1 Restrict Access to Tablespace (Automated)	314
4.5.2 Remove Unused Schemas (Automated).....	316
4.5.3 Review System Tablespaces (Automated)	317
5 Authentication Considerations.....	318
5.1 Specify a Secure Connection Authentication Type (SRVCON_AUTH) (Manual).....	318
5.2 Specify a Secure Authentication Type (AUTHENTICATION) (Manual).....	321
5.3 Database Manager Configuration Parameter: ALTERNATE_AUTH_ENC (Manual)	324
5.4 Database Manager Configuration Parameter: TRUST_ALLCLNTS (Manual).....	326
5.5 Database Manager Configuration Parameter: TRUST_CLNTAUTH (Automated)	328

5.6 Secure Permissions for All Authentication Plugins (Manual)	330
5.7 DB2_GRP_LOOKUP Registry Variable (Windows only) (Manual)	332
5.8 DB2DOMAINLIST Registry Variable (Windows only) (Manual)	333
5.9 DB2AUTH Registry Variable (Manual)	335
5.10 DB2CHGPWD_EEE Registry Variable (Manual)	337
6 Authorization Considerations.....	338
6.1 Secure Database Authorities.....	338
6.1.1 Secure SYSADM Authority (Manual)	338
6.1.2 Secure SYSCTRL Authority (Manual)	340
6.1.3 Secure SYSMANT Authority (Manual)	342
6.1.4 Secure SYSMON Authority (Manual)	344
6.1.5 Secure SECADM Authority (Manual)	346
6.1.6 Secure DBADM Authority (Manual).....	348
6.1.7 Secure SQLADM Authority (Manual)	350
6.1.8 Secure DATAACCESS Authority (Manual)	352
6.1.9 Secure ACCESSCTRL Authority (Manual)	354
6.1.10 Secure WLMADM Authority (Manual)	356
6.1.11 Secure CREATAB Authority (Manual)	358
6.1.12 Secure BINDADD Authority (Manual)	360
6.1.13 Secure CONNECT Authority (Manual).....	362
6.1.14 Secure LOAD Authority (Manual).....	364
6.1.15 Secure EXTERNALROUTINE Authority (Manual)	366
6.1.16 Secure QUIESCECONNECT Authority (Manual)	368
6.1.17 Secure SETSESSIONUSER Privilege (Manual).....	370
6.1.18 Secure SCHEMAADM Authority (Manual)	372
6.1.19 Secure Schema ACCESSCTRL Authority (Manual).....	374
6.1.20 Secure Schema DATAACCESS Authority (Manual).....	376
6.2 General Authorization	378
6.2.1 Review Users, Groups, and Roles (Manual)	378
6.2.2 Review Roles (Manual)	381
6.2.3 Review Role Members (Manual)	383
6.2.4 Nested Roles (Manual).....	385
6.2.5 Review Roles Granted to PUBLIC (Manual)	387
6.2.6 Review Role Grantees with WITH ADMIN OPTION (Manual).....	389
6.3 Row and Column Access Control.....	391
6.3.1 Review Organization's Policies Against Db2 RCAC Policies (Manual)	391
6.3.2 Review Row Permission Logic According to Policy (Manual).....	393
6.3.3 Review Column Mask Logic According to Policy (Manual).....	395
6.4 Trusted context Considerations	397
6.4.1 Ensure Trusted Contexts are Enabled (Manual).....	397
6.4.2 Do Not Allow Trusted Context to Switch Users Without Authentication (Manual)	398
7 Audit Considerations	400
7.1 General Audit Considerations.....	400
7.1.1 Disable the Audit Buffer (Automated).....	400
7.1.2 Disable Limited Audit of Applications (DB2_LIMIT_AUDIT_APPS) (Manual)	402
7.1.3 Ensure Audit Policies are Enabled Within the Database (Manual)	403
7.1.4 Ensure Audit is Enabled Within the Instance (Automated)	405
8 Encryption Considerations	407
8.1 Encryption of Data in Motion.....	407
8.1.1 Configure a Server-side Key Store for TLS (SSL_SVR_KEYDB) (Manual)	407
8.1.2 Configure a Server-side Stash File for TLS (SSL_SVR_STASH) (Manual)	409
8.1.3 Configure an Endpoint Certificate (SSL_SVR_LABEL) (Manual)	411
8.1.4 Configure the Service Name for TLS (SSL_SVCENAME) (Manual)	413
8.1.5 Configure a Secure TLS Version (SSL_VERSIONS) (Manual)	415

8.1.6 Configure Secure TLS Cipher Suites (SSL_CIPHERSPECS) (Manual)	417
8.1.7 Unset the Service Name for Plaintext Communication (SVCENAME) (Manual)	421
8.1.8 Configure a Client-side Key Store for TLS (SSL_CLNT_KEYDB) (Manual)	423
8.1.9 Configure a Client-side Stash File for TLS (SSL_CLNT_STASH) (Manual)	425
8.1.10 Enable TLS Communication Between HADR Primary and Standby Instances (HADR_SSL_LABEL) (Manual)	427
8.1.11 Enable Remote TLS Connections to Db2 (DB2COMM) (Manual)	429
8.2 Encryption of Data at Rest	431
8.2.1 Encrypt the Database (Manual)	432
8.2.2 Do Not Use Encryption Algorithms that are Not Secure (Manual)	434
8.2.3 Secure the Configuration File (Automated)	436
8.2.4 Secure the Stash File (Manual)	437
8.2.5 Backup the Stash File (Manual)	439
8.2.6 Create a Strong Password (Manual)	440
8.2.7 Backup Your Keystore (Manual)	441
8.2.8 Backup Your Password In Case Stash File is Inaccessible or Corrupted (Manual)	442
8.2.9 Rotate the Master Key (Manual)	443
8.2.10 Turn Off ALLOW_KEY_INSERT_WITHOUT_KEYSTORE_BACKUP (Manual)	445
8.2.11 Keep Master Key Labels Unique (Manual)	446
8.2.12 Retain All Master Keys (Manual)	447
8.2.13 Set CFG Values in a Single Command (Manual)	448
8.2.14 Key Rotation in HADR Environment (Manual)	449
9 Additional Considerations	451
9.1 Leverage the Least Privilege Principle (Manual)	452
9.2 Enable Backup Redundancy (Manual)	453
9.3 Protecting Backups (Manual)	454
Appendix: Summary Table	455
Appendix: Change History	470

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, CIS IBM Db2 12.1 Benchmark, provides prescriptive guidance for establishing a secure configuration posture for Db2 versions 12.x running on Linux and Windows.

The content of this document applies to all offerings based on the products Db2 12.1 and above. Where known differences exist in the guidance for the above products or for one of the supported platforms for these products, specific notes are provided.

The Db2 product is also referred to as Db2 for Linux, Unix, and Windows or Db2 LUW.

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

Intended Audience

This document 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 Db2 on its supported platforms.

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.
<code><Monospace font in brackets></code>	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 the RDBMS proper and intend to:

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

- **Level 2 - RDBMS (extends Level 1 - RDBMS)**

This profile extends the 'Level 1 - RDBMS' profile. Items in this profile exhibit one or more of the following characteristics:

- Are intended for environments or use cases where security is paramount
- Acts as defense in depth measure
- May negatively inhibit the utility or performance of the technology

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:

Author

Greg Stager

Editor

Tim Harrison, Center for Internet Security, New York

Recommendations

1 Staying Current

1.1 General Considerations

1.1.1 Install Available Updates (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The Db2 product is identified by a **<Version>.<Release>.<Modpack>.<Fixpack>** scheme. For example, 12.1.2.0 refers to version 12, release 1, modpack 2 fixpack 0. Modification packs contain new features as well as defect fixes and are indicated by a change in the modification and fix pack numbers in the product identifier. Fix packs contain only defect fixes and are indicated by a change in just the fix pack number in the product identifier (e.g. Db2 12.1.2.3 would be fix pack 3 for Db2 12.1 mod 2). Cumulative service builds (CSBs) contain fixes made available since the last modification or fix pack released and are identified by an **special_NNNNN** suffix, where **NNNNN** is the number of the CSB.

Periodically, IBM releases updates for its Db2 12.1 products in the form of modification packs, fix packs, and cumulative service builds. All updates are cumulative and contain the contents of the previous updates provided.

It is recommended that you review the available updates for Db2 12.1 on a regular and frequent basis and, optionally, subscribe for notification of critical Db2 fixes.

Rationale:

Being aware of the available updates and critical fixes helps you evaluate which Db2 update is the minimum level that you should use for your next system update. It will also help you understand the relative urgency of acquiring the latest Db2 fix pack to help protect the database from known vulnerabilities and reduce downtime that may otherwise result from functional defects.

Audit:

Perform the following Db2 commands to obtain the version:

```
$ db2level

DB21085I  This instance or install (instance name, where applicable:
"db2inst1") uses "64" bits and DB2 code release "SQL12012" with level
identifier "02030110".
```

Informational tokens are "DB2 v12.1.2.0", "special_64532", "DYN2508011629AMD64_64532", and Fix Pack "0".

Remediation:

Apply the latest fix pack as offered from IBM.

References:

1. <https://www-01.ibm.com/support/docview.wss?uid=ibm10718119>
2. <https://www.ibm.com/support/mynotifications>
3. <https://www.ibm.com/support/pages/published-security-vulnerabilities-db2-linux-unix-and-windows-including-special-build-information>
4. <https://www-01.ibm.com/support/docview.wss?rs=71&uid=swg27007053>
5. https://www.ibm.com/developerworks/community/blogs/IMSupport/entry/All_about_ifixes_interim_fixes?lang=en
6. <https://www.ibm.com/support/pages/node/7175778>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	7.4 Perform Automated Application Patch Management Perform application updates on enterprise assets through automated patch management on a monthly, or more frequent, basis.			
v7	3.5 Deploy Automated Software Patch Management Tools Deploy automated software update tools in order to ensure that third-party software on all systems is running the most recent security updates provided by the software vendor.			

2 Securing the Server Environment

2.1 Prevent Database Users from Logging into the Operating System (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

Db2's default authentication mechanism (SERVER) uses the operating system for authentication. This necessitates that users who require access to the database can be authenticated by the operating system. A by-product of this is that those users will be able to log into a shell in the OS of the database server, such as through ssh. The scope of the problem is greater if the OS has been configured to use an LDAP server for authentication, as that would likely contain more than just database users. Unless explicitly authorized, database users should not be able to log into the OS and action is required to prevent this.

- For Windows, the recommendation is based on the CIS Benchmark for Windows Server 2019, section *"2.2.23 Ensure 'Deny access to this computer from the network' to include 'Guests, Local account and member of Administrators group'"*.

Rationale:

Only authorized users should be able to log into the operating system that is running the Db2 database server. This will reduce the attack exposure of the system by preventing database users from accessing operating system resources or taking advantage of operating system flaws.

Impact:

The recommendation in this section affects who can log into the operating system of the server where Db2 is installed. Care must be taken to ensure that appropriate settings are made and system administrators continue to have the ability to login to the system.

Audit:

RHEL

Run the following commands as root:

```
$ grep pam_access /etc/pam.d/system-auth
```

If the command does not return a line from the file, this is a Fail, and the remediation should be followed.

AIX

In the file `/etc/security/user`, the default stanza should have the following:

```
rlogin = false
```

This ensure that by default a user is not able to remotely log into the system unless their user has been explicitly configured with `rlogin = true`. If the `rlogin` value in the default stanza is not false, that is a finding, and the remediation should be followed.

Windows

To establish the recommended configuration via Group Policy, navigate the following UI path:

```
Computer Configuration\Policies\Windows Settings\Security Settings\Local Policies\User Rights Assignment\Access this computer from the network
```

Ensure these two values are present in the setting, and if not, this is a Fail, and the remediation should be followed:

1. Guests
2. Local account and member of Administrators group.

Remediation:

The steps to accomplish this differ per OS and even the level of OS.

RHEL 8

Run the following commands as root:

1. Modify the file `/etc/security/access.conf`

Add users who are allowed to log into the OS with a `+`

```
+ : <user1> <user2>
```

Add a deny all rule as the last rule to prevent other users not explicitly allowed to log in, which would apply to database users:

```
- : ALL : ALL
```

2. Run this command to update the auth configuration:

```
$ authselect enable-feature with-pamaccess
```

Using these steps will ensure that `account required pam_access.so` is placed into the file `/etc/pam.d/system-auth` and the access list will be enforced for OS login.

AIX

1. Ensure appropriate users are explicitly listed as being allowed to log into the system. It is important that you have users listed with privilege to `rlogin` so that

you do not lock yourself out of the system. Modify the file `/etc/security/user`, and for each user allowed to log in, find their stanza and add the following line:

```
rlogin = true
```

2. Modify the default stanza to indicate the default value is that users are not able to login

```
rlogin = false
```

3. Ensure the **DB2LOGINRESTRICTIONS** registry variable is set to a value of **LOCAL** (the default if not specified) or **NONE**. Setting a value of **NETWORK** will return an error during Db2 authentication for any user with **rlogin** set to **false**. You can check this value as the instance owner by issuing the command:

```
$ db2set | grep DB2LOGINRESTRICTIONS
```

4. To change the value, use the command:

```
db2set DB2LOGINRESTRICTIONS=LOCAL
```

Windows

Follow these steps:

To establish the recommended configuration via Group Policy, configure the following UI path:

```
Computer Configuration\Policies\Windows Settings\Security Settings\Local Policies\User Rights Assignment\Access this computer from the network
```

Ensure these two values are present in the setting, and if not follow the remediation:

1. Guests
2. Local account and member of Administrators group.

Set the **DB2_WINDOWS_LOGON_TYPE** to **DEFINITION**. This setting controls how Db2 authenticates users when they connect. Local users must hold the right "Allow log on locally" and not be part of the "Deny log on locally". Domain users must hold the right "Access this computer from the network" and not be part of "Deny access to this computer from the network". This setting will ensure that local users are authenticated when connecting to Db2 according to their ability to log on locally, and not through the default value of accessing this computer from the network.

Issue the following command

```
db2set DB2_WINDOW_LOGON_TYPE=DEFINITION
```

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 Securing the Server Instance

3.1 Database Manager Configuration Parameters

This section provides guidance on various database manager configuration parameters at the instance level. Function specific parameters (for example specific to authentication) can be found in their appropriate sections.

3.1.1 Require Explicit Authorization for Cataloging (CATALOG_NOAUTH) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

Db2 can be configured to allow users that do not possess the **SYSADM** authority to **catalog** and **uncatalog** databases and nodes. It is recommended that the **CATALOG_NOAUTH** parameter be set to **NO**.

Rationale:

Cataloging a database is the process of registering a database from a remote client to allow remote call and access. Setting **CATALOG_NOAUTH** to **YES** bypasses all permissions checks and allows anyone to **catalog** and **uncatalog** databases.

Audit:

Perform the following to determine if authorization is explicitly required to **catalog** and **uncatalog** databases and nodes:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the value of **CATALOG_NOAUTH** in the output:

```
Cataloging allowed without authority (CATALOG_NOAUTH) = NO
```

Ensure **CATALOG_NOAUTH** is **NO**.

Remediation:

Perform the following to require explicit authorization to **catalog** and **uncatalog** databases and nodes.

1. Attach to the Db2 instance

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => update database manager configuration using catalog_noauth no
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=parameters-catalog-noauth-cataloging-allowed-without-authority>

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.			

3.1.2 Secure Permissions for Default Database File Path (DFTDBPATH) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DFTDBPATH** parameter contains the default file path used to create Db2 databases. It is recommended that the permissions for this directory be set to full access for Db2 administrators and read and execute access only for all other accounts. It is also recommended that this directory be owned by the Db2 Administrator.

Rationale:

Restricting access to the directory used as the default file path through permissions will help ensure that the confidentiality, integrity, and availability of the files there are protected.

Audit:

For Windows and Linux:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command from the Db2 command window:

```
db2 => get database manager configuration
```

3. Locate this value in the output to find the default file path:

```
Default database path (DFTDBPATH) = <valid directory>
```

Additional steps for Windows:

1. Connect to the Db2 host
2. Right-click over the directory used for the default file path
3. Choose **Properties**
4. Select the **Security** tab
5. Review and verify the privileges for all accounts.
6. Review and verify that the Db2 Administrator is the owner of the directory.

Additional steps for Linux:

1. Connect to the Db2 host

2. Change to the directory used as the default file path
3. Review and verify the permissions for the directory for all users; also ensure that the Db2 Administrator is the owner.

```
$ ls -al
```

Remediation:

For Windows and Linux:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command to change the default file path, if necessary:

```
db2 => update database manager configuration using dftdbpath <valid directory>
```

Additional steps for Windows:

1. Connect to the Db2 host
2. Right-click over the directory used as the default file path
3. Choose **Properties**
4. Select the **Security** tab
5. Assign ownership of the directory to the Db2 Administrator
6. Grant all Db2 administrator accounts the **Full Control** authority
7. Grant only read and execute privileges to all other users (revoke all other privileges)

Additional steps for Linux:

1. Connect to the Db2 host
2. Change to the directory used as the default file path
3. Assign the Db2 Administrator to be the owner of the directory using the **chown** command
4. Change the permissions for the directory

```
$ chmod -R 755
```

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.			

3.1.3 Set Diagnostic Logging to Capture Errors and Warnings (DIAGLEVEL) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DIAGLEVEL** parameter specifies the type of diagnostic errors that will be recorded in the **db2diag.log** file. It is recommended that the **DIAGLEVEL** parameter be set to at least **3**.

Rationale:

The recommended **DIAGLEVEL** setting is **3**, but any value greater than **3** is also acceptable. A value of at least **3** will allow the Db2 instance to capture all errors and warnings that occur on the system.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the DIAGLEVEL value in the output:

```
Diagnostic error capture level (DIAGLEVEL) = 3
```

Ensure **DIAGLEVEL** is greater than or equal to **3**.

Remediation:

1. Attach to the Db2 instance

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => update database manager configuration using diaglevel 3
```

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.			

3.1.4 Secure Permissions for All Diagnostic Logs (DIAGPATH) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DIAGPATH** parameter specifies the location of the diagnostic files for the Db2 instance. The directory at this location should be secured so that users have read and execute privileges only (no write privileges). All Db2 administrators should have full access to the directory.

Rationale:

Securing the directory will ensure that the confidentiality, integrity, and availability of the diagnostic files contained in the directory are preserved.

Audit:

For both Windows and Linux

Perform the following Db2 commands to obtain the location of the directory:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the DIAGPATH value in the output:

```
Diagnostic data directory path (DIAGPATH) = <valid directory>
```

Additional steps for Windows:

1. Connect to the Db2 host
2. Right-click over the diagnostic log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Review the access for all accounts

Additional steps for Linux:

1. Connect to the Db2 host

2. Change to the diagnostic log directory
3. Review the permissions of the directory

```
$ ls -al
```

Remediation:

For Windows and Linux

To change the directory for the diagnostic logs:

1. Attach to the Db2 instance

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => update database manager configuration using diagpath <valid directory>
```

Additional steps for Windows:

1. Connect to the Db2 host
2. Right-click over the diagnostic log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Grant the *Full Control* authority to all Db2 administrator accounts
6. Grant only read and execute privileges to all other accounts (revoke any other privileges)

Additional steps for Linux:

1. Connect to the Db2 host
2. Change to the diagnostic log directory
3. Change the permissions of the directory

```
$ chmod -R 3777 .
```

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.			

3.1.5 Secure Permissions for Alternate Diagnostic Log Path (ALT_DIAGPATH) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **ALT_DIAGPATH** parameter specifies an alternative location of the diagnostic files for the Db2 instance if the primary location specified by **DIAGPATH** is unavailable. The directory at this location should be secured so that users have read and execute privileges only (no write privileges). All Db2 administrators should have full access to the directory.

Rationale:

Securing the directory will ensure that the confidentiality, integrity, and availability of the diagnostic files contained in the directory are preserved.

Audit:

For both Windows and Linux

Perform the following Db2 commands to obtain the location of the directory:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the **ALT_DIAGPATH** value in the output:

```
Alternate diagnostic data directory path (ALT_DIAGPATH) = <valid directory>
```

Additional steps for Windows:

1. Connect to the Db2 host
2. Right-click over the diagnostic log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Review the access for all accounts

Additional steps for Linux:

1. Connect to the Db2 host
2. Change to the diagnostic log directory
3. Review the permissions of the directory

```
$ ls -al
```

Remediation:

For Windows and Linux

To change the directory for the diagnostic logs:

1. Attach to the Db2 instance

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => update database manager configuration using ALT_DIAGPATH <valid directory>
```

Additional steps for Windows:

1. Connect to the Db2 host
2. Right-click over the diagnostic log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Grant the *Full Control* authority to all Db2 administrator accounts
6. Grant only read and execute privileges to all other accounts (revoke any other privileges)

Additional steps for Linux:

1. Connect to the Db2 host
2. Change to the diagnostic log directory
3. Change the permissions of the directory

```
$ chmod -R 3777 .
```

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.			

3.1.6 Set Maximum Connection Limits (MAX_CONNECTIONS and MAX_COORDAGENTS) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **MAX_CONNECTIONS** parameter indicates the maximum number of client connections allowed per member. It is recommended that this parameter be set without the **AUTOMATIC** setting. The **AUTOMATIC** setting means for the value to grow unconstrained, and a value of **-1** means to use the same value as **MAX_COORDAGENTS**.

The **MAX_COORDAGENTS** parameter equals the maximum number of agents needed to perform connections to the database or attachments to the instance. The **AUTOMATIC** setting means for the value to grow unconstrained.

The **MAX_COORDAGENTS** parameter should be set to a fixed value without the **AUTOMATIC** setting. The exact value is highly dependent on business requirements for simultaneous connections. For example, if only a single application with a connection pool of 10 connections will connect to the database server, a much smaller value may be appropriate than a database server that expects to have hundreds of simultaneous connections. These examples will use a value of 200.

Ensure that dependent parameters, such as **MAXAPPLS**, are set less than or equal to the **MAX_CONNECTIONS** parameter. As instance parameters, **MAX_CONNECTIONS** and **MAX_COORDAGENTS** govern all databases within the instance, and thus the summation of **MAXAPPLS** value for all databases must be considered.

Rationale:

By default, Db2 allows an unlimited number of users to access the Db2 instance. In addition to giving access to the Db2 instance to authorized users only, it is recommended to set a limit to the number of users allowed to access a Db2 instance. This helps prevent denial of service conditions should an authorized process malfunction and attempt many simultaneous connections.

Audit:

Perform the following Db2 commands to obtain the value(s) for these settings:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the **MAX_COORDAGENTS** values in the output:

```
Max number of coordinating agents (MAX_COORDAGENTS) = AUTOMATIC(200)
```

The value of **MAX_COORDAGENTS** should be set without **AUTOMATIC**.

4. Locate the **MAX_CONNECTIONS** values in the output:

```
Max number of client connections (MAX_CONNECTIONS) =  
AUTOMATIC(MAX_COORDAGENTS)
```

The value of **MAX_COORDAGENTS** for the **MAX_CONNECTIONS** parameter signifies a value of -1 and indicates it uses the value of **MAX_COORDAGENTS**. If the value of **MAX_CONNECTIONS** shows **AUTOMATIC**, this is a Fail, and the remediation steps should be followed.

Remediation:

The default value for **MAX_COORDAGENTS** is **AUTOMATIC(200)**. Allowable range is **1** to **64,000**. The recommended value is **200**, without the **AUTOMATIC** setting. The value of **200** is used as an example and is dependent on workload as discussed in the Description.

The default value for **MAX_CONNECTIONS** is set to **AUTOMATIC(-1)**. Allowable range is **1** to **64,000**, or **-1** for matching the value of **MAX_COORDAGENTS**. The recommended value is **-1** without **AUTOMATIC**. It is also acceptable to have a value for **MAX_CONNECTIONS** that is greater than **MAX_COORDAGENTS**, such as **300**, in order to turn on the Concentrator feature.

Generally, both **MAX_COORDAGENTS** and **MAX_CONNECTIONS** should be configured within the same statement, otherwise the error **SQL6112N** may be encountered. The following command will set the **MAX_COORDAGENTS** to **200**, as well as set the **MAX_CONNECTIONS** to **-1**.

1. Attach to the Db2 instance

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => update database manager configuration using max_coordagents 200  
max_connections -1
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=parameters-max-connections-maximum-number-client-connections>

2. <https://www.ibm.com/docs/en/db2/12.1?topic=parameters-max-coordagents-maximum-number-coordinating-agents>

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.			

3.1.7 Set Administrative Notification Level (NOTIFYLEVEL) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **NOTIFYLEVEL** parameter specifies the type of administration notification messages that are written to the administration notification log.

It is recommended that this parameter be set greater than or equal to **3**.

A setting of **3**, which includes settings **1** & **2**, will log all fatal errors, failing services, system integrity, as well as system health.

Rationale:

The system should be monitoring all Health Monitor alarms, warnings, and attentions. This may give an indication of any malicious usage on the Db2 instance.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the NOTIFYLEVEL value in the output:

```
Notify Level (NOTIFYLEVEL) = 3
```

Remediation:

1. Attach to the Db2 instance

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => update database manager configuration using notifylevel 3
```

Default Value:

The default value of **NOTIFYLEVEL** is 3.

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.			
v7	6.2 <u>Activate audit logging</u> Ensure that local logging has been enabled on all systems and networking devices.			

3.1.8 Secure the Java Development Kit Installation Path (JDK_PATH) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **JDK_PATH** parameter contains the directory under which the Software Developer's Kit (SDK) for Java™ is installed. The Java SDK is used for running Java stored procedures and user-defined functions. It is recommended that the owner of this directory is bin on Linux and AIX, and a member of the Db2 administration group on Windows. The directory should have read and execute permission for all users, but only write permission for the owner.

Rationale:

Restricting access to the Java JDK will help ensure that only an authorized JDK is used for running Java routines within Db2.

Audit:

For Windows and Linux:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command from the Db2 command window:

```
db2 => get database manager configuration
```

3. Locate this value in the output to find the JDK path:

```
Default database path (JDK_PATH) = <valid directory>
```

Additional steps for Windows:

1. Connect to the Db2 host
2. Right-click over the directory used for the JDK path
3. Choose *Properties*
4. Select the *Security* tab
5. Review and verify the privileges for all accounts.
6. Review and verify that the Db2 Administrator is the owner of the directory.

Additional steps for Linux:

1. Connect to the Db2 host
2. Change to the directory used as the JDK path
3. Review and verify the permissions for the directory for all users; also ensure that bin is the owner.

```
$ ls -ald
```

Remediation:

For Windows and Linux:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command to change the JDK path, if necessary:

```
db2 => update database manager configuration using  
      jdk_path <valid directory>
```

Additional steps for Windows:

1. Connect to the Db2 host
2. Right-click over the directory used as the JDK path
3. Choose *Properties*
4. Select the *Security* tab
5. Assign ownership of the directory to the Db2 Administrator
6. Grant all Db2 administrator accounts the *Full Control* authority
7. Grant only read and execute privileges to all other users (revoke all other privileges)

Additional steps for Linux:

1. Connect to the Db2 host as root
2. Change to the directory used as the JDK path
3. Assign bin to be the owner of the directory using the chown command
4. Change the permissions for the directory

```
$ chmod -R 755
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=parameters-jdk-path-software-developer>

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.			

3.1.9 Secure the Python Runtime Path (PYTHON_PATH) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **PYTHON_PATH** parameter contains the directory under which the Python runtime is installed. It is recommended that the owner of this directory is bin on Linux and AIX, and a member of the Db2 administration group on Windows. The directory should have read and execute permission for all users, but only write permission for the owner.

Rationale:

Restricting access to the python runtime will help ensure that only an authorized runtime is used for running Python routines within Db2.

Audit:

For Windows and Linux:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command from the Db2 command window:

```
db2 => get database manager configuration
```

3. Locate this value in the output to find the Python path:

```
Default database path (PYTHON_PATH) = <valid directory>
```

Additional steps for Windows:

1. Connect to the Db2 host
2. Right-click over the directory used for the Python path
3. Choose *Properties*
4. Select the *Security* tab
5. Review and verify the privileges for all accounts.
6. Review and verify that the Db2 Administrator is the owner of the directory.

Additional steps for Linux:

1. Connect to the Db2 host
2. Change to the directory used as the Python path

3. Review and verify the permissions for the directory for all users; also ensure that bin is the owner.

```
$ ls -ald
```

Remediation:

For Windows and Linux:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command to change the Python path, if necessary:

```
db2 => update database manager configuration using  
python_path <valid directory>
```

Additional steps for Windows:

1. Connect to the Db2 host
2. Right-click over the directory used as the Python path
3. Choose *Properties*
4. Select the *Security* tab
5. Assign ownership of the directory to the Db2 Administrator
6. Grant all Db2 administrator accounts the *Full Control* authority
7. Grant only read and execute privileges to all other users (revoke all other privileges)

Additional steps for Linux:

1. Connect to the Db2 host as root
2. Change to the directory used as the Python path
3. Assign bin to be the owner of the directory using the chown command
4. Change the permissions for the directory

```
$ chmod -R 755
```


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.			

3.1.10 Secure the R Runtime Path (R_PATH) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **R_PATH** parameter contains the directory under which the R runtime is installed. The R runtime is used for running R stored procedures and user-defined functions. It is recommended that the owner of this directory is bin on Linux and AIX, and a member of the Db2 administration group on Windows. The directory should have read and execute permission for all users, but only write permission for the owner.

Rationale:

Restricting access to the R runtime will help ensure that only an authorized runtime is used for running R routines within Db2.

Audit:

For Windows and Linux:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command from the Db2 command window:

```
db2 => get database manager configuration
```

3. Locate this value in the output to find the R path:

```
Default database path (R_PATH) = <valid directory>
```

Additional steps for Windows:

1. Connect to the Db2 host
2. Right-click over the directory used for the R path
3. Choose *Properties*
4. Select the *Security* tab
5. Review and verify the privileges for all accounts.
6. Review and verify that the Db2 Administrator is the owner of the directory.

Additional steps for Linux:

1. Connect to the Db2 host
2. Change to the directory used as the R path

3. Review and verify the permissions for the directory for all users; also ensure that bin is the owner.

```
$ ls -ald
```

Remediation:

For Windows and Linux:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command to change the R path, if necessary:

```
db2 => update database manager configuration using  
r_path <valid directory>
```

Additional steps for Windows:

1. Connect to the Db2 host
2. Right-click over the directory used as the R path
3. Choose *Properties*
4. Select the *Security* tab
5. Assign ownership of the directory to the Db2 Administrator
6. Grant all Db2 administrator accounts the *Full Control* authority
7. Grant only read and execute privileges to all other users (revoke all other privileges)

Additional steps for Linux:

1. Connect to the Db2 host as root
2. Change to the directory used as the R path
3. Assign bin to be the owner of the directory using the chown command
4. Change the permissions for the directory

```
$ chmod -R 755
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=parameters-r-path-r-runtime-directory>

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.			

3.1.11 Secure the Communication Buffer Exit Library (COMM_EXIT_LIST) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

A communication exit library is a dynamically loaded library that vendor applications use to examine communication buffers. The **COMM_EXIT_LIST** parameter specifies the list of communication buffer exist libraries. The permissions on the libraries should be secured so that users other than the instance owner do not have write privileges.

Rationale:

If a malicious user has write access to a communication exit library, they can overwrite it with their own thereby receiving all of the communication buffers that Db2 receives over the network. Securing the libraries will prevent a loss of confidentiality of data.

Audit:

Steps for Linux:

- 32-bit and 64-bit server side user authentication plugins are found in **\$DB2PATH/security32/plugin/commexit** and **\$DB2PATH/security64/plugin/commexit** directories respectively

Review the permissions of the plugins that are in use:

```
ls -al
```

Steps for Windows:

- 32-bit and 64-bit server side user authentication plugins are found in **\$DB2PATH\security\plugin\commexit\<instance name>**

Review the permissions of the plugins that are in use:

1. Right-click over the plugin file
2. Choose properties
3. Select the Security tab
4. Review the access for all accounts

Remediation:

To change permissions of a file on Linux:

```
chmod 755 <file>
```

To change permissions of a file on Windows:

1. Right-click on the file
2. Choose properties
3. Select the Security tab
4. Grant the Full Control authority to all Db2 administrator accounts
5. Grant only read and execute privileges to all other accounts (revoke any other privileges)

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>3.3 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	<u>14.6 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.			

3.2 Db2 Registry Values

This section provides guidance on various registry variable configuration parameters at the instance level. Function specific parameters (for example specific to authentication) can be found in their appropriate sections.

3.2.1 Specify Secure Remote Shell Command (DB2RSHCMD) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DB2RSHCMD** registry variable specifies the remote shell command to use when starting remote database partitions and with the **db2_all** script to run utilities and commands on all database partitions. It is recommended that a value be used which encrypts the data sent between partitions, such as ssh.

Rationale:

The traditional **rsh** command sends all its data, including passwords, in plaintext between partitions. An attacker who can read network traffic may have access to these passwords and other data. Specifying a remote shell command, such as ssh encrypts the data sent over the network.

Impact:

Enabling this value without configuring public and private keys may result in an inability to start or stop Db2 across all partitions without manually issuing **db2start/db2stop** individually on each partition.

Audit:

Verify that the **DB2RSHCMD** registry variable is set to a value such as ssh that encrypts the data by running the following command:

```
db2set -all | grep DB2RSHCMD
```

The above command should not yield a value, or should yield a value of ssh.

Remediation:

1. Follow the guidance on this page to create public and private keys for ssh:

<https://www.ibm.com/docs/en/db2/12.1?topic=installation-enabling-execution-remote-commands>

2. Run the following command to set the **DB2_RSHCMD** registry variable to ssh:

```
db2set DB2RSHCMD=ssh
```

Default Value:

The default value is **ssh**.

References:

1. https://www.ibm.com/docs/en/db2/12.1?topic=variables-communications#r0005660_C_DB2RSHCMD
2. <https://www.ibm.com/docs/en/db2/12.1?topic=installation-enabling-execution-remote-commands>

CIS Controls:

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

3.2.2 Turn Off Remote Command Legacy Mode (DB2RCMD_LEGACY_MODE) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DB2RCMD_LEGACY_MODE** registry variable determines whether the Db2 Remote Command Service runs with enhanced security or not. It is recommended that legacy mode not be enabled.

This registry variable only applies to Db2 Servers running on Windows.

Rationale:

Legacy mode requires the Db2 service account to have privileges to impersonate the client account.

Audit:

Verify that the **DB2RCMD_LEGACY_MODE** registry variable is set to off by running the following command:

```
db2set -all | grep DB2RCMD_LEGACY_MODE
```

The above command should not yield a value or should yield a value of **OFF**.

Remediation:

Run the following command to set the **DB2RCMD_LEGACY_MODE** registry variable to **OFF**:

```
db2set DB2RCMD_LEGACY_MODE=OFF
```

Default Value:

The default value of **DB2RCMD_LEGACY_MODE** is **OFF**.

References:

1. [https://www.ibm.com/docs/en/db2/12.1?topic=variables-system-environment#r0005658 %5C S DB2RCMD LEGACY MODE](https://www.ibm.com/docs/en/db2/12.1?topic=variables-system-environment#r0005658_%5C_S_DB2RCMD_LEGACY_MODE)

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.			

3.2.3 Disable Grants During Restore (DB2_RESTORE_GRANT_ADMIN_AUTHORITIES) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DB2_RESTORE_GRANT_ADMIN_AUTHORITIES** registry variable determines whether the authorization ID of the user performing a restore is granted administrative authorities (**SECADM**, **DBADM**, **DATAACCESS**, and **ACCESSCTRL** authorities) on the restored database. It is typically used when restoring a database on a server where the original database creator account does not exist. It is recommended that this variable not be set except when specifically performing a restore where you wish these privileges to be granted so they are not accidentally granted.

Rationale:

Use of this registry variable may grant administrative authorities accidentally if the value is left on during normal operations and a restore is run.

Audit:

Verify the value of the **DB2_RESTORE_GRANT_ADMIN_AUTHORITIES** registry variable by running the following command:

```
db2set -all | grep DB2_RESTORE_GRANT_ADMIN_AUTHORITIES
```

The above command should not yield a value or should yield a value of **OFF**.

Remediation:

Run the following command to set the **DB2_RESTORE_GRANT_ADMIN_AUTHORITIES** registry variable to **OFF**:

```
db2set DB2_RESTORE_GRANT_ADMIN_AUTHORITIES=OFF
```

Default Value:

The default value of **DB2_RESTORE_GRANT_ADMIN_AUTHORITIES** is **OFF**.

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=variables-system-environment#r0005658> \ S DB2 RESTORE GRANT ADMIN AUTHORITIES

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.			

3.2.4 Enable Extended Security (DB2_EXTSECURITY) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DB2_EXTSECURITY** registry variable determines whether extended security is enabled on Windows. Extended security enables file permissions for all Db2 objects so they can be accessed by users in **DB2ADMNS** and **DB2USERS** groups. It is recommended that extended security be enabled.

This registry variable only applies to Db2 Servers running on Windows.

Rationale:

Extended security provides file permission protection for Db2 objects when running in a multi-user environment.

Impact:

Before enabling extended security for an existing installation, you should ensure you have added users to the **DB2ADMNS** and **DB2USERS** groups so that you are able to access Db2 once the permission have been applied. Once extended security is enabled, it is not recommended to be directly disabled, instead a full re-install should be used.

Audit:

Verify that extended security is enabled by running the following command:

```
db2set -all | grep DB2_EXTSECURITY
```

The above command should yield a value of **ON**.

Remediation:

Run the following command to enable extended security (consult documentation for additional options as appropriate):

```
db2extsec
```

References:

1. https://www.ibm.com/docs/en/db2/12.1?topic=variables-miscellaneous#r0005669_M_DB2_EXTSECURITY
2. <https://www.ibm.com/docs/en/db2/12.1?topic=security-extended-windows-using-db2admns-db2users-groups>
3. <https://www.ibm.com/docs/en/db2/12.1?topic=commands-db2extsec-set-permissions-db2-objects>

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.			

3.2.5 Limit OS Privileges of Fenced Mode Process (DB2_LIMIT_FENCED_GROUP) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DB2_LIMIT_FENCED_GROUP** registry variable allows restricting the operating system privileges of the fenced mode process (**db2fmp**) to the privileges assigned to the **DB2USERS** group.

This variable only has effect if extended security is enabled (**DB2_EXTSEC**) and the Db2 Service Account is not LocalSystem.

This registry variable only applies to Db2 Servers running on Windows.

Rationale:

By default, the fenced mode process has access to both the **DB2ADMNS** and **DB2USERS** groups.

Audit:

Verify the value of the **DB2_LIMIT_FENCED_GROUP** registry variable by running the following command:

```
db2set -all | grep DB2_LIMIT_FENCED_GROUP
```

The above command should yield a value of **ON**.

Remediation:

Run the following command to set the **DB2_LIMIT_FENCED_GROUP** registry variable to **ON**:

```
db2set DB2_LIMIT_FENCED_GROUP=ON
```

Default Value:

The default value of **DB2_LIMIT_FENCED_GROUP** is **OFF**.

References:

1. https://www.ibm.com/docs/en/db2/12.1?topic=variables-miscellaneous#r0005669_M_DB2_LIMIT_FENCED_GROUP
2. <https://www.ibm.com/docs/en/db2/12.1?topic=windows-restricting-operating-system-privileges-db2fmp-process>

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.			

3.3 General Considerations

3.3.1 Secure Db2 Runtime Library (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

A Db2 software installation will place all executables under the default `<DB2PATH>\sqllib` directory. This directory needs to be secured so it grants only the necessary access to authorized users and administrators.

Rationale:

The Db2 runtime is comprised of files that are executed as part of the Db2 service. If these resources are not secured, an attacker may alter them to execute arbitrary code.

Audit:

Perform the following to obtain the value for this setting:

For Windows:

1. Connect to the Db2 host
2. Right-click on the `NODE000x\sqlbdir` directory
3. Choose *Properties*
4. Select the *Security* tab
5. Determine the permissions for DB administrator accounts and all other accounts

For Linux:

1. Connect to the Db2 host
2. Change to the `NODE000x/sqlbdir` directory
3. Determine the permissions for the directory

```
$ ls -al
```

Remediation:

For Windows:

1. Connect to the Db2 host
2. Right-click on the `NODE000x\sqlbdir` directory
3. Choose *Properties*
4. Select the *Security* tab
5. Select all DB administrator accounts and grant them the *Full Control* authority
6. Select all other accounts and revoke all privileges other than *Read* and *Execute*

For Linux:

1. Connect to the Db2 host
2. Change to the `/NODE000x/sqldbdir` directory
3. Change the permission level of the directory to this recommended value

```
$ chmod -R 755
```

Default Value:

Linux

- `$DB2PATH/NODE000x/sqldbdir` is owned by the Db2 administrator with read, write, and execute access.

Windows

- `$DB2PATH\NODE000x\sqldbdir` owned by the Db2 administrator with read, write, and execute access.

The database instance `db2inst1` located in `/home/NODE000x` needs the following permissions:

```
drwxrwxr-x 11 db2inst1 db2grp1 4096 Aug 08 1:34 NODE0000
```

All lower directories need the same settings:

`/db2`, `/db2/data`, `/db2/data/db2inst1`, `/db2/data/db2inst1/db2inst1` and `/db2/data/db2inst1/db2inst1/NODE0000` would need the same settings `drwxrwxr-x`.

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.			

3.3.2 Secure the Database Container Directory (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

A Db2 database container is the physical storage of the data.

Rationale:

The containers are needed for the database to operate properly. The loss of the containers can cause down time. Also, allowing excessive access to the containers may help an attacker to gain access to their contents. Therefore, secure the location(s) of the containers by restricting the access and ownership. Allow only the instance owner to have access to the tablespace containers.

Audit:

Review all users that have access to the directory of the containers. On Linux and AIX ensure that only the instance owner has access to the directory of the containers and the container files themselves. On Windows only administrators, and if extended security is enabled, members of the **DB2ADMINS** group should have access to the directory of the containers and the container files. No other users should have access.

Remediation:

On Linux and AIX, set the privileges for the directory of the containers so that only the instance owner has full access, and all other users have no access.

On Windows, set the privileges for the directory of the containers and the container files so that only administrators, and if extended security is enabled members of the DB2ADMINS group, have full access, and all other users have no access.

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.			

3.3.3 Set `umask` Value in the Db2 Instance Owner's `.profile` (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The Db2 instance owner's `.profile` file in Linux sets the environment variables and the settings for the user. This file is specific to the Korn shell and BASH shell, and other shells may have a different file.

Rationale:

The `umask` value should be set to `022` for the owner of the Db2 software at all times to ensure files are not created with unnecessary privileges.

Audit:

1. Ensure that the `umask 022` setting exists in the `.profile`.

```
$ grep umask ~/.profile
umask 022
```

2. Ensure that `umask 022` is currently enforced when logged in as the instance owner:

```
$ umask
022
```

Remediation:

Add `umask 022` to the `.profile` file.

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 Securing the Database

4.1 Database Configuration Parameters

This section provides guidance on various database configuration parameters. Function specific parameters (for example specific to authentication) can be found in their appropriate sections.

4.1.1 Creating the Database Without PUBLIC Grants (RESTRICTIVE) (Automated)

Profile Applicability:

- Level 2 - RDBMS (extends Level 1 - RDBMS)

Description:

This parameter indicates whether the database was created with the **RESTRICTIVE** clause in the **CREATE DATABASE** statement. When creating a database, the use of the **RESTRICTIVE** clause will cause certain privileges to be revoked from **PUBLIC**.

Rationale:

Impact:

Allowing the default privileges granted to the group **PUBLIC** to remain in tack can have negative impacts on the database as well as undermine measures put in place to limit access to authorized users.

Audit:

Perform the following Db2 commands to determine if the database was created with the **RESTRICTIVE** clause.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select value from sysibmadm.dbcfg where name = 'restrict_access'
```

If the value is **YES** then this is a Pass. Otherwise the database was not created with the **RESTRICTIVE** clause.

Remediation:

There is no remediation for this parameter due to the fact that the placement of the **RESTRICTIVE** clause happens within the **CREATE DATABASE** statement. Unless your backup strategies allow for a complete overhaul of your environment where you are able to recreate the database with the **RESTRICTIVE** clause, we do not recommend changing this parameter. However, if you would like to align your database configuration to that which the **RESTRICTIVE** clause would provide, please ensure the following:

1. **SYSCAT.DBAUTH** – Ensure **PUBLIC** is **NOT** granted the following authorities:
 - **CREATETAB**

- BINDADD
 - CONNECT
 - IMPLICIT_SCHEMA
2. SYSCAT.TABAUTH – Ensure PUBLIC is **NOT** granted the following privileges:
 - SELECT on all SYSCAT and SYSIBM tables
 - SELECT and UPDATE on all SYSSTAT tables
 - SELECT on the following views in schema SYSIBMADM:
 - ALL_*
 - USER_*
 - ROLE_*
 - SESSION_*
 - DICTIONARY
 - TAB
 3. SYSCAT.ROUTINEAUTH – Ensure PUBLIC is **NOT** granted the following privileges:
 - EXECUTE with GRANT on all procedures in schema SQLJ
 - EXECUTE with GRANT on all functions and procedures in schema SYSFUN
 - EXECUTE with GRANT on all functions and procedures in schema SYSPROC
 - EXECUTE on all table functions in schema SYSIBM
 - EXECUTE on all other procedures in schema SYSIBM
 4. SYSCAT.MODULEAUTH – Ensure PUBLIC is **NOT** granted the following privileges:
 - EXECUTE on the following modules in schema SYSIBMADM:
 - DBMS_DDL
 - DBMS_JOB
 - DBMS_LOB
 - DBMS_OUTPUT
 - DBMS_SQL
 - DBMS_STANDARD
 - DBMS_UTILITY
 5. SYSCAT.PACKAGEAUTH – Ensure PUBLIC is **NOT** granted the following privileges:
 - BIND on all packages created in the NULLID schema
 - EXECUTE on all packages created in the NULLID schema
 6. SYSCAT.SCHEMAAUTH – Ensure PUBLIC is **NOT** granted the following privileges:
 - CREATEIN on schema SQLJ
 - CREATEIN on schema NULLID
 7. SYSCAT.TBSPACEAUTH – Ensure PUBLIC is **NOT** granted the USE privilege on table space USERSPACE1.
 8. SYSCAT.WORKLOADAUTH – Ensure PUBLIC is **NOT** granted the USAGE privilege on SYSDEFAULTUSERWORKLOAD.
 9. SYSCAT.VARIABLEAUTH – Ensure PUBLIC is **NOT** granted the READ privilege on schema global variables in the SYSIBM schema.

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=commands-create-database>

2. <https://www.ibm.com/docs/en/db2/12.1?topic=ownership-default-privileges-granted-creating-database>

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.			

4.1.2 Set Failed Archive Retry Delay (ARCHRETRYDELAY) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **ARCHRETRYDELAY** parameter specifies the number of seconds the Db2 service will wait before it reattempts to archive log files after a failure. It is recommended that this parameter be set anywhere in the range of **10-30**. You do not want the delay to be so short that the database ends up in a denial of service scenario, but you don't want the delay to be too long if an outside attack happens at the same time.

Rationale:

Ensure that the value is non-zero, otherwise archive logging will not retry after the first failure. A denial of service attack can render the database without an archive log if this setting is not set. An archive log will ensure that all transactions can safely be restored or logged for auditing.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => get database configuration
```

3. Locate the **ARCHRETRYDELAY** value in the output:

```
Log archive retry Delay (secs) (ARCHRETRYDELAY) = 20
```

Remediation:

1. Connect to the Db2 database

```
db2 => connect to <dbname>
```

2. To successfully set the **ARCHRETRYDELAY** within the **10-30** range, run the following command:

```
db2 => update database configuration using archretrydelay *nn* (where *nn* is a range between 10-30)
```

Default Value:

The default value for **ARCHRETRYDELAY** is **20**.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.10 <u>Retain Audit Logs</u> Retain audit logs across enterprise assets for a minimum of 90 days.			

4.1.3 Auto-restart After Abnormal Termination (AUTORESTART) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **AUTORESTART** parameter specifies if the database manager will start a crash recovery automatically when the first user tries to connect to a database which has previously terminated abnormally.

If the parameter is set to **OFF** an explicit **RESTART DATABASE** command has to be issued to initiate a crash recovery.

It is recommended that this parameter is set to **ON** (which is also the default).

Rationale:

Setting the database to auto-restart will reduce the downtime of the database.

Audit:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => get database configuration
```

3. Locate the **AUTORESTART** value in the output:

```
Auto restart enabled (AUTORESTART) = ON
```

Remediation:

1. Connect to the Db2 database

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => update database configuration using autorestart on
```

Default Value:

The default value for **AUTORESTART** is **ON**.

4.1.4 Secure Permissions for the Primary Archive Log Location (LOGARCHMETH1) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **LOGARCHMETH1** parameter specifies the type of media and the location used as the primary destination of archived logs. It is recommended that the directory used for the archived logs be set to full access for Db2 administrator accounts and read and execute for all other accounts.

Rationale:

Restricting access to the contents of the primary archive log directory will help ensure that the confidentiality, integrity, and availability of archive logs are protected. Although there are many ways to ensure that your primary logs will be archived, we recommend using the value of **DISK** as part of the **LOGARCHMETH1** parameter. This will properly ensure that the primary logs are archived. A finding of **OFF** is not acceptable.

Audit:

For Windows and Linux:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => get database configuration
```

3. Locate this value in the output to find the primary archive log directory:

```
Default database path (LOGARCHMETH1) = <valid directory>
```

Additional steps for Windows:

1. Connect to the Db2 host
2. Right-click on the primary archive log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Review and verify the privileges for all accounts

Additional steps for Linux:

1. Connect to the Db2 host
2. Change to the primary archive log directory
3. Review and verify the permissions for the directory for all users.

```
$ ls -al
```

Remediation:

For Windows and Linux:

1. Attach to the Db2 instance.
2. Run the following command to change the primary archive log directory, if necessary:

```
db2 => update database configuration using  
logarchmeth1 <valid directory>
```

Additional steps for Windows (assuming that the `logarchmeth1` parameter includes `DISK`):

1. Connect to the Db2 host
2. Right-click on the primary archive log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Grant all Db2 administrator accounts the *Full Control* authority
6. Grant all other accounts read and execute privileges only (revoke all other privileges)

Additional steps for Linux (assuming that the `logarchmeth1` parameter includes `DISK`):

1. Connect to the Db2 host
2. Change to the primary archive log directory
3. Change the permissions for the directory

```
$ chmod -R 755
```


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.			

4.1.5 Secure Permissions for the Secondary Archive Log Location (LOGARCHMETH2) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **LOGARCHMETH2** parameter specifies the type of media and the location used as the secondary destination for archived logs. It is recommended that the directory used for the archived logs be set to full access for Db2 administrator accounts and read and execute only for all other accounts.

Rationale:

Restricting access to the contents of the secondary archive log directory will help ensure that the confidentiality, integrity, and availability of archive logs are protected. Although there are many ways to ensure that your logs will be archived, we recommend using the value of **DISK** as part of the **LOGARCHMETH2** parameter. This will properly ensure that the logs are archived. A finding of **OFF** is not acceptable.

Audit:

For Windows and Linux:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => get database configuration
```

3. Locate this value in the output to find the secondary archive log directory:

```
Default database path (LOGARCHMETH2) = <valid directory>
```

Additional steps for Windows:

1. Connect to the Db2 host
2. Right-click on the secondary archive log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Review and verify the privileges for all accounts

Additional steps for Linux:

1. Connect to the Db2 host
2. Change to the secondary archive log directory
3. Review and verify the permissions for the directory for all users

```
$ ls -al
```

Remediation:

For Windows and Linux:

1. Attach to the Db2 instance.
2. Run the following command to change the secondary archive log directory, if necessary:

```
db2 => update database configuration using  
logarchmeth2 <valid directory>
```

Additional steps for Windows (assuming that the **logarchmeth2** parameter includes **DISK**):

1. Connect to the Db2 host
2. Right-click on the secondary archive log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Grant all Db2 administrator accounts the *Full Control* authority
6. Grant all other accounts read and execute privileges only (revoke all other privileges)

Additional steps for Linux (assuming that the **logarchmeth2** parameter includes **DISK**):

1. Connect to the Db2 host
2. Change to the secondary archive log directory
3. Change the permissions for the directory

```
$ chmod -R 755
```

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.			

4.1.6 Secure Permissions for the Tertiary Archive Log Location (FAILARCHPATH) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **FAILARCHPATH** parameter specifies the type of media and the location used as the tertiary destination of archived logs. It is recommended that the directory used for the archived logs be set to full access for Db2 administrator accounts and read and execute only for all other accounts.

Rationale:

Restricting access to the contents of the tertiary archive log directory will help ensure that the confidentiality, integrity, and availability of archive logs are protected. Although there are many ways to ensure that your logs will be archived, we recommend using the value of **DISK** as part of the **FAILARCHPATH** parameter. This will properly ensure that the logs are archived. A finding of **OFF** is not acceptable.

Audit:

For Windows and Linux:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => get database configuration
```

3. Locate this value in the output to find the tertiary archive log directory:

```
Default database path (FAILARCHPATH) = <valid directory>
```

Additional steps for Windows:

1. Connect to the Db2 host
2. Right-click on the tertiary archive log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Review and verify the privileges for all accounts

Additional steps for Linux:

1. Connect to the Db2 host
2. Change to the tertiary archive log directory
3. Review and verify the permissions for the directory for all users.

```
$ ls -al
```

Remediation:

For Windows and Linux:

1. Attach to the Db2 instance.
2. Run the following command to change the tertiary archive log directory, if necessary:

```
db2 => update database configuration using failarchpath
```

Additional steps for Windows (assuming that the **failarchpath** parameter includes **DISK**):

1. Connect to the Db2 host
2. Right-click on the tertiary archive log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Grant all Db2 administrator accounts the *Full Control* authority
6. Grant all other accounts read and execute privileges only (revoke all other privileges)

For Linux (assuming that the **failarchpath** parameter includes **DISK**):

1. Connect to the Db2 host
2. Change to the tertiary archive log directory
3. Change the permissions for the directory

```
$ chmod -R 755
```

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.			

4.1.7 Secure Permissions for the Log Mirror Location (MIRRORLOGPATH) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **MIRRORLOGPATH** parameter specifies the type of media and the location used to store the mirror copy of the logs. It is recommended that the directory used for the mirror copy of the logs be set to full access for Db2 administrator accounts and read and execute only for all other accounts.

Rationale:

A mirror log path should not be empty and it should be a valid path. The mirror log path stores a second copy of the active log files. Access to the directory pointed to by that path should be restricted through permissions to help ensure that the confidentiality, integrity, and availability of the mirror logs are protected.

Audit:

For Windows and Linux

Perform the following Db2 commands to obtain the directory location:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => get database configuration
```

3. Locate the **MIRRORLOGPATH** value in the output:

```
Mirror log path (MIRRORLOGPATH) = C:\DB2MIRRORLOGS
```

Additional steps for Windows:

1. Connect to the Db2 host
2. Right-click on the mirror log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Review and verify the privileges for all accounts

Additional steps for Linux:

1. Connect to the Db2 host
2. Change to the mirror log directory
3. Review and verify the permissions for the directory for all users.

```
$ ls -al
```

Remediation:

For Windows and Linux:

1. Connect to the Db2 database

```
db2 => connect to <dbname>
```

2. Run the following command to change the mirror log directory, if necessary:

```
db2 => update database configuration using mirrorlogpath <valid directory>
```

Additional steps for Windows:

1. Connect to the Db2 host
2. Right-click on the mirror log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Grant all Db2 administrator accounts the *Full Control* authority
6. Grant all other accounts read and execute privileges only (revoke all other privileges)

Additional steps for Linux:

1. Connect to the Db2 host
2. Change to the mirror log directory
3. Change the permissions for the directory

```
$ chmod -R 755
```

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.			

4.1.8 Secure Permissions for the Log Overflow Location (OVERFLOWLOGPATH) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **OVERFLOWLOGPATH** parameter specifies a location for Db2® databases to find log files needed for a **rollforward** operation, as well as where to store active log files retrieved from the archive. It also gives a location for finding and storing log files needed for using **db2ReadLog** API. It is recommended that the directory used be set to full access for Db2 administrator accounts and read and execute only for all other accounts.

Rationale:

The overflow log path can contain log files containing user data. Access to the directory pointed to by that path should be restricted through permissions to help ensure that the confidentiality, integrity, and availability of the logs are protected.

Audit:

For Windows and Linux

Perform the following Db2 commands to obtain the directory location:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => get database configuration
```

3. Locate the **OVERFLOWLOGPATH** value in the output:

```
Overflow log path (OVERFLOWLOGPATH) =
```

Additional steps for Windows:

1. Connect to the Db2 host
2. Right-click on the overflow log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Review and verify the privileges for all accounts

Additional steps for Linux:

1. Connect to the Db2 host
2. Change to the overflow log directory
3. Review and verify the permissions for the directory for all users.

```
$ ls -al
```

Remediation:

For Windows and Linux:

1. Connect to the Db2 database

```
db2 => connect to <dbname>
```

2. Run the following command to change the mirror log directory, if necessary:

```
db2 => update database configuration using overflowlogpath <valid  
directory>
```

Additional steps for Windows:

1. Connect to the Db2 host
2. Right-click on the overflow archive log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Grant all Db2 administrator accounts the *Full Control* authority
6. Grant all other accounts read and execute privileges only (revoke all other privileges)

Additional steps for Linux:

1. Connect to the Db2 host
2. Change to the overflow log directory
3. Change the permissions for the directory

```
$ chmod -R 755
```

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.			

4.1.9 Establish Retention Set Size for Backups (NUM_DB_BACKUPS) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **NUM_DB_BACKUPS** parameter specifies the number of backups to retain for a database before marking the oldest backup as deleted. It is recommended that this parameter be set to at least **12**.

Rationale:

Retain multiple copies of the database backup to ensure that the database can recover from an unexpected failure. This parameter should not be set to **0**. Multiple backups should be kept ensuring that all logs and transactions can be used for auditing.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => get database configuration
```

3. Locate the **NUM_DB_BACKUPS** value in the output:

```
Number of database backups to retain (NUM_DB_BACKUPS) = 12
```

Remediation:

1. Connect to the Db2 database

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => update database configuration using num_db_backups 12
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	11.2 <u>Perform Automated Backups</u> Perform automated backups of in-scope enterprise assets. Run backups weekly, or more frequently, based on the sensitivity of the data.			
v7	10.1 <u>Ensure Regular Automated Back Ups</u> Ensure that all system data is automatically backed up on regular basis.			

4.1.10 Set Archive Log Failover Retry Limit (NUMARCHRETRY) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **NUMARCHRETRY** parameter determines how many times a database will try to archive the log file to the primary or the secondary archive destination before trying the failover directory. It is recommended that this parameter be set to **5**.

Rationale:

Establishing a failover retry time limit will ensure that the database will always have a means to recover from an abnormal termination. This parameter should not be set to **0**.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => get database configuration
```

3. Locate the **NUMARCHRETRY** value in the output:

```
Number of log archive retries on error (NUMARCHRETRY) = 5
```

Remediation:

1. Connect to the Db2 database

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => update database configuration using numarchretry 5
```


CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.3 <u>Ensure Adequate Audit Log Storage</u> Ensure that logging destinations maintain adequate storage to comply with the enterprise's audit log management process.			
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.			
v7	6.4 <u>Ensure adequate storage for logs</u> Ensure that all systems that store logs have adequate storage space for the logs generated.			

4.1.11 Set Maximum Number of Applications (MAXAPPLS) (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **MAXAPPLS** parameter specifies the maximum number of concurrent applications that can be connected (both local and remote) to a database. In an instance with a single database, it is recommended that this value be set to **AUTOMATIC**. While this indicates that any number of connections should be allowed, an upper limit of **MAX_CONNECTIONS** database manager configuration parameter is still enforced. If there are multiple databases within the instance, then the sum of **MAXAPPLS** for each database should be less than or equal to **MAX_CONNECTIONS**.

When **AUTOMATIC** is used, **MAXAPPLS** can also have a parameter. The value is not used to determine the maximum number of connections, but rather for dependent parameters, such as **PCKCACHESZ** and **CATALOG_CACHESZ** that can derive their value from **MAXAPPLS**. In such a case the value specified with **AUTOMATIC** should represent the expected number of connections.

Rationale:

By default, Db2 allows an unlimited number of users to access the Db2 instance. In addition to giving access to the Db2 instance to authorized users only, it is recommended to set a limit to the number of users allowed to access a Db2 instance. This helps prevent denial of service conditions should an authorized process malfunction and attempt many simultaneous connections.

Impact:

Increasing the value of this parameter without lowering the **MAXLOCKS** parameter or increasing the **LOCKLIST** parameter could cause you to reach the database limit on locks (**LOCKLIST**) rather than the application limit and as a result cause pervasive lock escalation problems.

Audit:

Perform the following Db2 commands to obtain the value of the **MAXAPPLS** parameter:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => get database configuration
```

3. Locate the **MAXAPPLS** value in the output:

```
Max Number of Active Applications (MAXAPPLS) = AUTOMATIC(40)
```

Remediation:

1. Connect to the Db2 database

```
db2 => connect to <dbname>
```

2. If a single database is used in the instance, run the following command:

```
db2 => update database configuration using maxappls AUTOMATIC
```

3. If multiple databases are used in the instance, determine appropriate values for each database such that the sum of **MAXAPPLS** values equals the **MAX_CONNECTIONS** database manager configuration parameter value (for example **100** for each of 2 databases when **MAX_CONNECTIONS** is **200**), and run the following command:

```
db2 => update database configuration using maxappls 200
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=parameters-maxappls-maximum-number-active-applications>

4.1.12 Ensure a Secure Connect Procedure is Used (CONNECT_PROC) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **CONNECT_PROC** stored procedure runs as part of every connection to the database and allows customization of the application environment, such as setting special registers. It could for example, set the **CURRENT_PATH** special register which controls the search path for finding functions and procedures to execute. Only an authorized procedure should be used.

Rationale:

The **CONNECT_PROC** procedure could be used to modify the application environment within the connection causing unexpected behavior.

Audit:

Perform the following Db2 commands to determine if a **CONNECT_PROC** procedure is being used:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => get database configuration
```

3. Locate the **CONNECT_PROC** value in the output:

```
Connect procedure (CONNECT_PROC) =
```

If the value is not set, this is a Pass. If a value is set, it should be reviewed to ensure it is an appropriate procedure to be run for every connection.

Remediation:

To turn off the connect proc, perform the following commands:

1. Connect to the Db2 database

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => update database configuration using  
       connect_proc NULL IMMEDIATE
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=parameters-connect-proc-connect-procedure-name-database>
2. <https://www.ibm.com/docs/en/db2/12.1?topic=databases-customizing-application-environment-using-connect-procedure>

4.1.13 Specify a Secure Location for External Tables (EXTBL_LOCATION) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **EXTBL_LOCATION** database configuration parameter provides an allow list of paths that external tables may access for local files, for both reading and writing. It is recommended that this value be set to appropriate paths with the understanding that confidential data may reside in this directory. Specifying appropriate paths is part of an organizations standard operating procedures (SOP).

Rationale:

External tables can read and write data to the paths configured within the **EXTBL_LOCATION** configuration parameter. To avoid a loss of confidentiality of the data which may be reside in these paths, they should be examined to ensure they match the values specified by the SOP.

Audit:

Perform the following Db2 commands to determine if a **EXTBL_LOCATION** is specified:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => get database configuration
```

3. Locate the **EXTBL_LOCATION** value in the output:

```
Allowed paths for external tables (EXTBL_LOCATION) = /home/db2inst1
```

If the value does not match what has been specified in the SOP, then follow the remediation steps.

Remediation:

To specify an external table location, perform the following commands:

1. Connect to the Db2 database

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => update database configuration using extbl_location <paths>
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=parameters-extbl-location-external-table-location>

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.			

4.2 Secure the Database Catalog Views

The database catalogs are tables and views that describe all the objects in the database as well as the authorities and privileges relevant to those objects. For example, a stored procedure object is described in the **SYSCAT.ROUTINES** view. The database catalogs can be broken into four areas:

1. Tables in the **SYSIBM** schema – As a general rule, these tables should not be referenced directly from within applications, as their table definitions are subject to change at any time. Applications coding SQL against these tables may break if the table definitions change. Instead, the **SYSCAT** views should be used.
2. Views in the **SYSCAT** schema – These are views on top of the underlying **SYSIBM** table and should be used to query the catalog tables as their table definition is stable within a release.
3. Udatable view in the **SYSSTAT** schema – These views contain statistical information that is used by the optimizer.

In general, the average user does not need to query the database catalogs. However, unless the database is created with the **RESTRICTIVE** option, then **PUBLIC** is granted **SELECT** on all of these tables. It is recommended that **SELECT** be revoked from **PUBLIC**. Administrators such as **DBADM** have implicit access to all of the catalog tables.

This section contains recommendations for security relevant catalog tables or those that contain sensitive values. There are other tables that may be considered base on individual business needs.

- <https://www.ibm.com/docs/en/db2/12.1?topic=sql-catalog-views>

4.2.1 Restrict Access to SYSCAT.AUDITPOLICIES (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.AUDITPOLICIES** view contains all audit policies for a database. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

This view contains sensitive information about the auditing security for this database. Access to the audit policies may aid attackers in avoiding detection.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tablename= 'AUDITPOLICIES'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.AUDITPOLICIES FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatauditpolicies>

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.			

4.2.2 Restrict Access to SYSCAT.AUDITUSE (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.AUDITUSE** view contains database audit policy for all non-database objects, such as authority, groups, roles, and users. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

This view contains sensitive information about the types of objects being audited. Access to the audit policy may aid attackers in avoiding detection.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'AUDITUSE'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.AUDITUSE FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscataudituse>

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.			

4.2.3 Restrict Access to SYSCAT.COLAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.COLAUTH** view contains the column privileges granted to the user or a group of users. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

This view contains the column privileges granted to the user, group, or role in the database and may be used as an attack vector.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'COLAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.COLAUTH FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatcolauth>

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.			

4.2.4 Restrict Access to SYSCAT.COLDIST (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.COLDIST** view contains the nth most frequent value of some column, or the nth quantile (cumulative distribution) value of the column. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.COLDIST** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname= 'COLDIST'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.COLDIST FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatcoldist>

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.			

4.2.5 Restrict Access to SYSCAT.COLGROUPDIST (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.COLGROUPDIST** view contains the *n*th most frequent value of the column group or the *n*th quantile value of the column group. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.COLGROUPDIST** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname= 'COLGROUPDIST'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.COLGROUPDIST FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatcolgroupdist>

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.			

4.2.6 Restrict Access to SYSCAT.COLUMNS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.COLUMNS** view contains all the columns in the database instance. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.COLUMNS** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname= 'COLUMNS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.COLUMNS FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatcolumns>

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.			

4.2.7 Restrict Access to SYSCAT.CONTEXTATTRIBUTES (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.CONTEXTATTRIBUTES** view contains all the trusted context attributes in the database instance. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.CONTEXTATTRIBUTES** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname= 'CONTEXTATTRIBUTES'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.CONTEXTATTRIBUTES FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatcontextattributes>

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.			

4.2.8 Restrict Access to SYSCAT.CONTEXTS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.CONTEXTS** view contains the trusted contexts in the database instance. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.CONTEXTS** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname= 'CONTEXTS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.CONTEXTS FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatcontexts>

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.			

4.2.9 Restrict Access to SYSCAT.CONTROLDEP (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.CONTROLDEP** view contains the dependency of a row permission or column mask on some other object. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.CONTROLDEP** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname= 'CONTROLDEP'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.CONTROLDEP FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatcontroldep>

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.			

4.2.10 Restrict Access to SYSCAT.CONTROLS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.CONTROLS** view contains row permissions and column masks. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.CONTROLS** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname= 'CONTROLS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.CONTROLS FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatcontrols>

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.			

4.2.11 Restrict Access to SYSCAT.DBAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.DBAUTH** view contains information on authorities granted to users or groups of users. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

This view contains all the grants in the database and may be used as an attack vector.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'DBAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.DBAUTH FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatdbauth>

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.			

4.2.12 Restrict Access to SYSCAT.EVENTS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.EVENTS** view contains all types of events that the database is currently monitoring. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

The types of events that the database is monitoring should not be made readily available to the public.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'EVENTS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.EVENTS FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatevents>

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.			

4.2.13 Restrict Access to SYSCAT.EVENTTABLES (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.EVENTTABLES** view contains the name of the destination table that will receive the monitoring events. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

PUBLIC should not have access to see the target name of the event monitoring table.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tablename = 'EVENTTABLES'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.EVENTTABLES FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscateventtables>

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.			

4.2.14 Restrict Access to SYSCAT.EXTERNALTABLEOPTIONS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.EXTERNALTABLEOPTIONS** view contains the external tables in the database instance. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.EXTERNALTABLEOPTIONS** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT'  
       and tabname= 'EXTERNALTABLEOPTIONS' and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.EXTERNALTABLEOPTIONS FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatexternaltableoptions>

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.			

4.2.15 Restrict Access to SYSCAT.INDEXAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.INDEXAUTH** view contains a list of users or groups that have CONTROL access on an index. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

The list of all users with access to an index should not be exposed to the public.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'INDEXAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.INDEXAUTH FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatindexauth>

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.			

4.2.16 Restrict Access to SYSCAT.MODULEAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.MODULEAUTH** view contains all granted privileges on a module for users, groups, or roles and is read only.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.MODULEAUTH** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'MODULEAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.MODULEAUTH FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatmoduleauth>

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.			

4.2.17 Restrict Access to SYSCAT.PACKAGEAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.PACKAGEAUTH** view contains the package privileges granted to the user or a group of users. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

The list of all users with access to a package should not be exposed to the public.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'PACKAGEAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.PACKAGEAUTH FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatpackageauth>

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.			

4.2.18 Restrict Access to SYSCAT.PACKAGES (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.PACKAGES** view contains the names of all packages created in the database instance. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

The names of packages created in the database can be used as an entry point if a vulnerable package exists.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth where tabschema = 'SYSCAT'
and tabname = 'PACKAGES' and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.PACKAGES FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatpackages>

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.			

4.2.19 Restrict Access to SYSCAT.PASSTHRUAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.PASSTHRUAUTH** view contains the names of user or group that have pass-through authorization to query the data source. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

The ability to see which accounts have the pass-through privilege could allow an attacker to exploit these accounts to access another data source.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'PASSTHRUAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.PASSTHRUAUTH FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatpassthruauth>

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.			

4.2.20 Restrict Access to SYSCAT.ROLEAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.ROLEAUTH** view contains information on all roles and their respective grantees. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

PUBLIC should not have access to see the grants of the roles because this could be used as a point of exploit.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'ROLEAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.ROLEAUTH FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatroleauth>

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.			

4.2.21 Restrict Access to SYSCAT.ROLES (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.ROLES** view contains all roles available in the database. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

PUBLIC should not have access to see all the roles because this could be used as a point of exploit.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'ROLES'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.ROLES FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatroles>

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.			

4.2.22 Restrict Access to SYSCAT.ROUTINEAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.ROUTINEAUTH** view contains a list of all users that have EXECUTE privilege on a routine (function, method, or procedure). It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

PUBLIC should not have access to see all the users because this could be used as a point of exploit.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'ROUTINEAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.ROUTINEAUTH FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatroutineauth>

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.			

4.2.23 Restrict Access to SYSCAT.ROUTINES (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.ROUTINES** view contains all user-defined routines, functions, and stored procedures in the database. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

User-defined functions and routines should not be exposed to the public for exploits.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'ROUTINES'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.ROUTINES FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatroutines>

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.			

4.2.24 Restrict Access to SYSCAT.SECURITYLABELACCESS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.SECURITYLABELACCESS** view contains security label that was granted access. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.SECURITYLABELACCESS** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT'  
       and tabname= 'SECURITYLABELACCESS' and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.SECURITYLABELACCESS FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatsecuritylabelaccess>

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.			

4.2.25 Restrict Access to SYSCAT.SECURITYLABELCOMPONENTELEMENTS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.SECURITYLABELCOMPONENTELEMENTS** view contains the element value for security label components. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.SECURITYLABELCOMPONENTELEMENTS** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT'  
       and tablename= 'SECURITYLABELCOMPONENTELEMENTS '  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.SECURITYLABELCOMPONENTELEMENTS  
FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatsecuritylabelcomponentelements>

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.			

4.2.26 Restrict Access to SYSCAT.SECURITYLABELCOMPONENTS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.SECURITYLABELCOMPONENTS** view contains the element value for security label components. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.SECURITYLABELCOMPONENTS** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth
       where tabschema = 'SYSCAT'
       and tabname= 'SECURITYLABELCOMPONENTS'
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.SECURITYLABELCOMPONENTS  
FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatsecuritylabelcomponents>

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.			

4.2.27 Restrict Access to SYSCAT.SECURITYLABELS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.SECURITYLABELS** view contains security labels. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.SECURITYLABELS** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname= 'SECURITYLABELS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.SECURITYLABELS FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatsecuritylabels>

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.			

4.2.28 Restrict Access to SYSCAT.SECURITYPOLICIES (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.SECURITYPOLICIES** view contains all database security policies. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

PUBLIC should not be able to view all the database security policies.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT'  
       and tabname = 'SECURITYPOLICIES' and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.SECURITYPOLICIES FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatsecuritypolicies>

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.			

4.2.29 Restrict Access to *SYSCAT.SECURITYPOLICYCOMPONENTRULES (Automated)*

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.SECURITYPOLICYCOMPONENTRULES** view contains the read and write access rules for security label components. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.SECURITYPOLICYCOMPONENTRULES** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT'  
       and tabname= 'SECURITYPOLICYCOMPONENTRULES'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.SECURITYPOLICYCOMPONENTRULES  
FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatsecuritypolicycomponentrules>

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.			

4.2.30 Restrict Access to SYSCAT.SECURITYPOLICYEXEMPTIONS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.SECURITYPOLICYEXEMPTIONS** view contains the exemption to a security policy that was granted to a database account. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

PUBLIC should not be able to view all the exemptions to the database security policies.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT'  
       and tablename = 'SECURITYPOLICYEXEMPTIONS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.SECURITYPOLICYEXEMPTIONS  
       FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatsecuritypolicyexemptions>

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.			

4.2.31 Restrict Access to SYSCAT.SERVEROPTIONS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.SERVEROPTIONS** view contains server-specific option values. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.SERVEROPTIONS** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname= 'SERVEROPTIONS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.SERVEROPTIONS FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatserveroptions>

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.			

4.2.32 Restrict Access to SYSCAT.SCHEMAAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.SCHEMAAUTH** view contains a list of all users that have one or more privileges or access to a particular schema. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

PUBLIC should not have access to see all the users because this could be used as a point of exploit.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'SCHEMAAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.SCHEMAAUTH FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatschemaauth>

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.			

4.2.33 Restrict Access to SYSCAT.SCHEMATA (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.SCHEMATA** view contains all schema names in the database. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

PUBLIC should not have access to see all the schema names in the database because this could be used as a point of exploit.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'SCHEMATA'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.SCHEMATA FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatschemata>

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.			

4.2.34 Restrict Access to SYSCAT.SEQUENCEAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.SEQUENCEAUTH** view contains users, groups, or roles granted privilege(s) on a sequence. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

PUBLIC should not have access to see all the granted access of a sequence in the database because this could be used as a point of exploit.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'SEQUENCEAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.SEQUENCEAUTH FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatsequenceauth>

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.			

4.2.35 Restrict Access to SYSCAT.STATEMENTS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.STATEMENTS** view contains all SQL statements of a compiled package. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

PUBLIC should not have access to the SQL statements of a database package. This could lead to an exploit.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'STATEMENTS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.STATEMENTS FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatstatements>

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.			

4.2.36 Restrict Access to SYSCAT.STATEMENTTEXTS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.STATEMENTTEXTS** view contains user-provided SQL statements for statement thresholds. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.STATEMENTTEXTS** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname= 'STATEMENTTEXTS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.STATEMENTTEXTS FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatstatementtexts>

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.			

4.2.37 Restrict Access to SYSCAT.SURROGATEAUTHIDS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.SURROGATEAUTHIDS** view contains the names of all accounts that have been granted SETSESSIONUSER privilege on a user or to **PUBLIC**. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

PUBLIC should not be able to view the names of all the surrogate accounts with SETSESSIONUSER privilege.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'SURROGATEAUTHIDS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.SURROGATEAUTHIDS FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatsurrogateauthids>

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.			

4.2.38 Restrict Access to SYSCAT.TABAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.TABAUTH** view contains users or groups that have been granted one or more privileges on a table or view. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

PUBLIC should not have access to the grants of views and tables in a database. This could lead to an exploit.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'TABAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.TABAUTH FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscattabauth>

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.			

4.2.39 Restrict Access to SYSCAT.TBSPACEAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.TBSPACEAUTH** view contains users or groups that have been granted the USE privilege on a particular tablespace in the database. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

PUBLIC should not have access to the grants of the tablespaces in a database. This could lead to an exploit.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'TBSPACEAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.TBSPACEAUTH FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscattbpaceauth>

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.			

4.2.40 Restrict Access to SYSCAT.USEROPTIONS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.USEROPTIONS** view contains server-specific user option values. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.USEROPTIONS** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname= 'USEROPTIONS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.USEROPTIONS FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatuseroptions>

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.			

4.2.41 Restrict Access to SYSCAT.VARIABLEAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.VARIABLEAUTH** view contains the granted privileges on a global variable for users, groups, or roles and is read only.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.VARIABLEAUTH** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Determine if **SYSCAT.VARIABLEAUTH** privileges for users, groups, and roles are correctly set. Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'VARIABLEAUTH'  
       and grantee = 'PUBLIC'
```

3. Review privileges for users, groups, and roles. If the output is BLANK, then it is considered a Pass.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.VARIABLEAUTH FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatvariableauth>

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.			

4.2.42 Restrict Access to SYSCAT.VARIABLES (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.VARIABLES** view contains global variables. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.VARIABLES** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname= 'VARIABLES'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.VARIABLES FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatvariables>

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.			

4.2.43 Restrict Access to SYSCAT.WORKLOADAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.WORKLOADAUTH** view represents the users, groups, or roles that have been granted the USAGE privilege on a workload.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.WORKLOADAUTH** from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'WORKLOADAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.WORKLOADAUTH FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatworkloadauth>

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.			

4.2.44 Restrict Access to SYSCAT.WRAPOPTIONS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.WRAPOPTIONS** view contains wrapper-specific options. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.WRAPOPTIONS** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname= 'WRAPOPTIONS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.WRAPOPTIONS FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatwrapoptions>

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.			

4.2.45 Restrict Access to SYSCAT.XSROBJECTAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.XSROBJECTAUTH** view contains granted USAGE privileges on a particular XSR object for users, groups, or roles and is read only.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSCAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSCAT.XSROBJECTAUTH** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'XSROBJECTAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.XSROBJECTAUTH FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscatxsrobjectauth>

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.			

4.2.46 Restrict Access to SYSSTAT.COLDIST (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSSTAT.COLDIST** view contains the nth most frequent value of some column, or the nth quantile (cumulative distribution) value of the column. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSSTAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSSTAT.COLDIST** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSSTAT' and tabname= 'COLDIST'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSSTAT.COLDIST FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-sysstatcoldist>

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.			

4.2.47 Restrict Access to SYSSTAT.COLGROUPEDIST (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSSTAT.COLGROUPEDIST** view contains the *n*th most frequent value of the column group or the *n*th quantile value of the column group. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSSTAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSSTAT.COLGROUPEDIST** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSSTAT' and tabname= 'COLGROUPEDIST'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSSTAT.COLGROUPEDIST FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-sysstatcolgroupdist>

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.			

4.2.48 Restrict Access to SYSSTAT.COLUMNS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSSTAT.COLUMNS** view contains all the columns in the database instance. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSSTAT** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSSTAT.COLUMNS** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSSTAT' and tabname= 'COLUMNS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSSTAT.COLUMNS FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-sysstatcolumns>

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.			

4.2.49 Restrict Access to SYSCAT.AUDITEXCEPTIONS (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.AUDITEXCEPTIONS** view contains audit exceptions that are associated with an object. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

This view contains sensitive information about what audit exceptions exist. Access to the audit exceptions may aid attackers in avoiding detection.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'AUDITEXCEPTIONS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.AUDITEXCEPTIONS FROM PUBLIC
```

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.			

4.2.50 Restrict Access to SYSCAT.BUFFERPOOLAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.BUFFERPOOLAUTH** view contains the bufferpool privileges granted to the user, group or role of a user. It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

This view contains the bufferpool privileges granted to a user, group or roles of a user, and may be used as an attack vector.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSCAT' and tabname = 'BUFFERPOOLAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSCAT.BUFFERPOOLAUTH FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1.x?topic=views-syscatbufferpoolauth>

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.			

4.3 Secure the Database Catalog Tables

The database catalogs are tables and views that describe all the objects in the database as well as the authorities and privileges relevant to those objects. For example, a stored procedure object is described in the **SYSCAT.ROUTINES** view. The database catalogs can be broken into four areas:

Tables in the **SYSIBM** schema – As a general rule, these tables should not be referenced directly from within applications, as their table definitions are subject to change at any time. Applications coding SQL against these tables may break if the table definitions change. Instead, the **SYSCAT** views should be used instead

Views in the **SYSCAT** schema – These are views on top of the underlying **SYSIBM** table and should be used to query the catalog tables as their table definition is stable within a release.

Updatable view in the **SYSSTAT** schema – These views contain statistical information that is used by the optimizer.

In general, the average user does not need to query the database catalogs. However, unless the database is created with the **RESTRICTIVE** option, then **PUBLIC** is granted **SELECT** on all of these tables. It is recommended that **SELECT** be revoked from **PUBLIC**. Administrators such as **DBADM** have implicit access to all of the catalog tables.

This section contains recommendations for security relevant catalog tables or those that contain sensitive values. There are other tables that may be considered base on individual business needs.

4.3.1 Restrict Access to SYSIBM.SYSAUDITPOLICIES (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSAUDITPOLICIES** table contains all audit policies for a database. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

This table contains sensitive information about the auditing security for this database. Access to the audit policies may aid attackers in avoiding detection.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname= 'SYSAUDITPOLICIES'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSAUDITPOLICIES FROM PUBLIC
```

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.			

4.3.2 Restrict Access to SYSIBM.SYSAUDITUSE (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSAUDITUSE** table contains database audit policy for all non-database objects, such as authority, groups, roles, and users. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

This table contains sensitive information about the types of objects being audited. Access to the audit policy may aid attackers in avoiding detection.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname = 'SYSAUDITUSE'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSAUDITUSE FROM PUBLIC
```


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.			

4.3.3 Restrict Access to SYSIBM.SYSCOLAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSCOLAUTH** table contains the column privileges granted to the user or a group of users. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

This table contains the column privileges granted to the user, group, or role in the database and may be used as an attack vector.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname = 'SYSCOLAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSCOLAUTH FROM PUBLIC
```

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.			

4.3.4 Restrict Access to SYSIBM.SYSCOLDIST (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSCOLDIST** table contains the nth most frequent value of some column, or the nth quantile (cumulative distribution) value of the column. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSCOLDIST** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname= 'SYSCOLDIST'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSCOLDIST FROM PUBLIC
```

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.			

4.3.5 Restrict Access to SYSIBM.SYSCOLGROUPDIST (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSCOLGROUPDIST** table contains the *n*th most frequent value of the column group or the *n*th quantile value of the column group. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSCOLGROUPDIST** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname= 'SYSCOLGROUPDIST'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSCOLGROUPDIST FROM PUBLIC
```

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.			

4.3.6 Restrict Access to SYSIBM.SYSCOLUMNS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSCOLUMNS** table contains all the columns in the database instance. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSCOLUMNS** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname= 'SYSCOLUMNS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSCOLUMNS FROM PUBLIC
```


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.			

4.3.7 Restrict Access to SYSIBM.SYSCONTEXTATTRIBUTES (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSCONTEXTATTRIBUTES** table contains all the trusted context attributes in the database instance. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSCONTEXTATTRIBUTES** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM'  
       and tabname= 'SYSCONTEXTATTRIBUTES' and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSCONTEXTATTRIBUTES FROM PUBLIC
```

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.			

4.3.8 Restrict Access to SYSIBM.SYSCONTEXTS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSCONTEXTS** table contains the trusted contexts in the database instance. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSCONTEXTS** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname= 'SYSCONTEXTS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSCONTEXTS FROM PUBLIC
```

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.			

4.3.9 Restrict Access to SYSIBM.SYSDEPENDENCIES (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSDEPENDENCIES** table contains the dependency of one object on another, such as a row permission or column mask on some other object. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSDEPENDENCIES** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM'  
       and tabname= 'SYSDEPENDENCIES'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSDEPENDENCIES FROM PUBLIC
```

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.			

4.3.10 Restrict Access to SYSIBM.SYSCONTROLS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSCONTROLS** table contains row permissions and column masks. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSCONTROLS** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname= 'SYSCONTROLS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSCONTROLS FROM PUBLIC
```


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.			

4.3.11 Restrict Access to SYSIBM.SYSDBAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSDBAUTH** table contains information on authorities granted to users or groups of users. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

This table contains all the grants in the database and may be used as an attack vector.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tablename = 'SYSDBAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSDBAUTH FROM PUBLIC
```

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.			

4.3.12 Restrict Access to SYSIBM.SYSEVENTS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSCAT.SYSEVENTS** table contains all types of events that the database is currently monitoring. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

The types of events that the database is monitoring should not be made readily available to the public.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname = 'SYSEVENTS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSEVENTS FROM PUBLIC
```

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.			

4.3.13 Restrict Access to SYSIBM.SYSEVENTTABLES (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSEVENTTABLES** table contains the name of the destination table that will receive the monitoring events. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

PUBLIC should not have access to see the target name of the event monitoring table.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname = 'SYSEVENTTABLES'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSEVENTTABLES FROM PUBLIC
```

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.			

4.3.14 Restrict Access to SYSIBM.SYSEXTTAB (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSEXTTAB** table contains the external tables in the database instance. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSEXTTAB** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname= 'SYSEXTTAB'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSEXTTAB FROM PUBLIC
```


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.			

4.3.15 Restrict Access to SYSIBM.SYSINDEXAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSINDEXAUTH** table contains a list of users or groups that have CONTROL access on an index. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

The list of all users with access to an index should not be exposed to the public.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tablename = 'SYSINDEXAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSINDEXAUTH FROM PUBLIC
```

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.			

4.3.16 Restrict Access to SYSIBM.SYSMODULEAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSMODULEAUTH** view contains all granted privileges on a module for users, groups, or roles and is read only.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSMODULEAUTH** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname = 'SYSMODULEAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSMODULEAUTH FROM PUBLIC
```

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.			

4.3.17 Restrict Access to SYSIBM.SYSPASSTHRUAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSPASSTHRUAUTH** table contains the names of user or group that have pass-through authorization to query the data source. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

The ability to see which accounts have the pass-through privilege could allow an attacker to exploit these accounts to access another data source.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname = 'SYSPASSTHRUAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSPASSTHRUAUTH FROM PUBLIC
```

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.			

4.3.18 Restrict Access to SYSIBM.SYSPLANAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSPLANAUTH** table contains the package privileges granted to the user or a group of users. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

The list of all users with access to a package should not be exposed to the public.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tablename = 'SYSPLANAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSPLANAUTH FROM PUBLIC
```


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.			

4.3.19 Restrict Access to SYSIBM.SYSPLAN (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSPLAN** table contains the names of all packages created in the database instance. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

The names of packages created in the database can be used as an entry point if a vulnerable package exists.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tablename = 'SYSPLAN'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSPLAN FROM PUBLIC
```

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.			

4.3.20 Restrict Access to SYSIBM.SYSROLEAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSROLEAUTH** table contains information on all roles and their respective grantees. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

PUBLIC should not have access to see the grants of the roles because this could be used as a point of exploit.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname = 'SYSROLEAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSROLEAUTH FROM PUBLIC
```

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.			

4.3.21 Restrict Access to SYSIBM.SYSROLES (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSROLES** table contains all roles available in the database. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

PUBLIC should not have access to see all the roles because this could be used as a point of exploit.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname = 'SYSROLES'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSROLES FROM PUBLIC
```

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.			

4.3.22 Restrict Access to SYSIBM.SYSROUTINEAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSROUTINEAUTH** table contains a list of all users that have EXECUTE privilege on a routine (function, method, or procedure). It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

PUBLIC should not have access to see all the users because this could be used as a point of exploit.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tablename = 'SYSROUTINEAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSROUTINEAUTH FROM PUBLIC
```


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.			

4.3.23 Restrict Access to SYSIBM.SYSROUTINES (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSROUTINES** table contains user-defined routines (scalar function, table function, sourced function, aggregate interface function, method, or procedure). It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSROUTINES** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname= 'SYSROUTINES'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSROUTINES FROM PUBLIC
```

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.			

4.3.24 Restrict Access to SYSIBM.ROUTINES_S (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The SYSIBM.ROUTINES_S view contains user-defined routines (scalar function, table function, sourced function, aggregate interface function, method, or procedure). It is recommended that the **PUBLIC** role be restricted from accessing this view.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the SYSIBM.ROUTINES_S view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname= 'ROUTINES_S'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.ROUTINES_S FROM PUBLIC
```

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.			

4.3.25 Restrict Access to SYSIBM.SYSSCHEMAAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSSCHEMAAUTH** table contains a list of all users that have one or more privileges or access to a particular schema. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

PUBLIC should not have access to see all the users because this could be used as a point of exploit.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname = 'SYSSCHEMAAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSSCHEMAAUTH FROM PUBLIC
```

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.			

4.3.26 Restrict Access to SYSIBM.SYSSCHEMATA (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSSCHEMATA** table contains all schema names in the database. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

PUBLIC should not have access to see all the schema names in the database because this could be used as a point of exploit.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname = 'SYSSCHEMATA'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSSCHEMATA FROM PUBLIC
```


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.			

4.3.27 Restrict Access to *SYSIBM.SYSSECURITYLABELACCESS (Automated)*

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSSECURITYLABELACCESS** table contains security label that was granted access. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSSECURITYLABELACCESS** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM'  
       and tabname= 'SYSSECURITYLABELACCESS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSSECURITYLABELACCESS FROM PUBLIC
```

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.			

4.3.28 Restrict Access to SYSIBM.SYSSECURITYLABELCOMPONENTELEMENTS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSSECURITYLABELCOMPONENTELEMENTS** table contains the element value for security label components. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSSECURITYLABELCOMPONENTELEMENTS** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM'  
       and tablename= 'SYSSECURITYLABELCOMPONENTELEMENTS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSSECURITYLABELCOMPONENTELEMENTS  
FROM PUBLIC
```

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.			

4.3.29 Restrict Access to *SYSIBM.SYSSECURITYLABELCOMPONENTS (Automated)*

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSSECURITYLABELCOMPONENTS** table contains security label components. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSSECURITYLABELCOMPONENTS** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM'  
       and tabname= 'SYSSECURITYLABELCOMPONENTS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSSECURITYLABELCOMPONENTS FROM PUBLIC
```

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.			

4.3.30 Restrict Access to SYSIBM.SYSSECURITYLABELS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSSECURITYLABELS** table contains security labels. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSSECURITYLABELS** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth where tabschema = 'SYSIBM'
and tabname= 'SYSSECURITYLABELS' and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSSECURITYLABELS FROM PUBLIC
```


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.			

4.3.31 Restrict Access to SYSIBM.SYSSECURITYPOLICIES (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSSECURITYPOLICIES** table contains all database security policies. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

PUBLIC should not be able to view all the database security policies.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM'  
       and tabname = 'SYSSECURITYPOLICIES' and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSSECURITYPOLICIES FROM PUBLIC
```

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.			

4.3.32 Restrict Access to SYSIBM.SYSSECURITYPOLICYCOMPONENTRULES (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSSECURITYPOLICYCOMPONENTRULES** table contains the read and write access rules for security label components. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSSECURITYPOLICYCOMPONENTRULES** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM'  
       and tablename= 'SYSSECURITYPOLICYCOMPONENTRULES'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSSECURITYPOLICYCOMPONENTRULES  
FROM PUBLIC
```

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.			

4.3.33 Restrict Access to *SYSIBM.SYSSECURITYPOLICYEXEMPTIONS (Automated)*

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSSECURITYPOLICYEXEMPTIONS** table contains the exemption to a security policy that was granted to a database account. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

PUBLIC should not be able to view all the exemptions to the database security policies.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM'  
       and tablename = 'SYSSECURITYPOLICYEXEMPTIONS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSSECURITYPOLICYEXEMPTIONS  
       FROM PUBLIC
```

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.			

4.3.34 Restrict Access to SYSIBM.SYSSERVEROPTIONS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSSERVEROPTIONS** table contains server-specific option values. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSSERVEROPTIONS** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM'  
       and tabname= 'SYSSERVEROPTIONS' and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSSERVEROPTIONS FROM PUBLIC
```


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.			

4.3.35 Restrict Access to SYSIBM.SYSSEQUENCEAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSSEQUENCEAUTH** table contains users, groups, or roles granted privilege(s) on a sequence. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

PUBLIC should not have access to see all the granted access of a sequence in the database because this could be used as a point of exploit.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM'  
       and tabname = 'SYSSEQUENCEAUTH' and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSSEQUENCEAUTH FROM PUBLIC
```

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.			

4.3.36 Restrict Access to SYSIBM.SYSSTATEMENTTEXTS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSSTATEMENTTEXTS** table contains user-provided SQL statements for statement thresholds. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSSTATEMENTTEXTS** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM'  
       and tabname= 'SYSSTATEMENTTEXTS' and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSSTATEMENTTEXTS FROM PUBLIC
```

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.			

4.3.37 Restrict Access to SYSIBM.SYSSTMT (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSSTMT** table contains all SQL statements of a compiled package. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

PUBLIC should not have access to the SQL statements of a database package. This could lead to an exploit.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname = 'SYSSTMT'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSSTMT FROM PUBLIC
```

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.			

4.3.38 Restrict Access to SYSIBM.SYSSURROGATEAUTHIDS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSSURROGATEAUTHIDS** table contains the names of all accounts that have been granted SETSESSIONUSER privilege on a user or to **PUBLIC**. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

PUBLIC should not be able to view the names of all the surrogate accounts with SETSESSIONUSER privilege.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM'  
       and tabname = 'SYSSURROGATEAUTHIDS' and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSSURROGATEAUTHIDS FROM PUBLIC
```


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.			

4.3.39 Restrict Access to SYSIBM.SYSTABAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSTABAUTH** table contains users or groups that have been granted one or more privileges on a table or view. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

PUBLIC should not have access to the grants of views and tables in a database. This could lead to an exploit.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname = 'SYSTABAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSTABAUTH FROM PUBLIC
```

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.			

4.3.40 Restrict Access to SYSIBM.SYSTBSPACEAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSTBSPACEAUTH** table contains users or groups that have been granted the USE privilege on a particular tablespace in the database. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

PUBLIC should not have access to the grants of the tablespaces in a database. This could lead to an exploit.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM'  
       and tabname = 'SYSTBSPACEAUTH' and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSTBSPACEAUTH FROM PUBLIC
```

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.			

4.3.41 Restrict Access to SYSIBM.SYSUSEROPTIONS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSUSEROPTIONS** table contains server-specific user option values. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSUSEROPTIONS** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM'  
       and tabname= 'SYSUSEROPTIONS' and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSUSEROPTIONS FROM PUBLIC
```

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.			

4.3.42 Restrict Access to SYSIBM.SYSVARIABLEAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSVARIABLEAUTH** view contains the granted privileges on a global variable for users, groups, or roles and is read only.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSVARIABLEAUTH** view from **PUBLIC** to reduce risk to the organization's data.

Audit:

Determine if **SYSIBM.SYSVARIABLEAUTH** privileges for users, groups, and roles are correctly set. Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname = 'SYSVARIABLEAUTH'  
       and grantee = 'PUBLIC'
```

3. Review privileges for users, groups, and roles. If the output is BLANK, then it is considered a Pass.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSVARIABLEAUTH FROM PUBLIC
```


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.			

4.3.43 Restrict Access to SYSIBM.SYSVARIABLES (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSVARIABLES** table contains global variables. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSVARIABLES** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname= 'SYSVARIABLES'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSVARIABLES FROM PUBLIC
```

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.			

4.3.44 Restrict Access to SYSIBM.SYSWORKLOADAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSWORKLOADAUTH** table represents the users, groups, or roles that have been granted the USAGE privilege on a workload.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSWORKLOADAUTH** from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname = 'SYSWORKLOADAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSWORKLOADAUTH FROM PUBLIC
```

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.			

4.3.45 Restrict Access to SYSIBM.SYSWRAPOPTIONS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSWRAPOPTIONS** table contains wrapper-specific options. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSWRAPOPTIONS** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname= 'SYSWRAPOPTIONS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSWRAPOPTIONS FROM PUBLIC
```

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.			

4.3.46 Restrict Access to SYSIBM.SYSXSROBJECTAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSXSROBJECTAUTH** table contains granted USAGE privileges on a particular XSR object for users, groups, or roles and is read only.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for **SYSIBM** tables. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBM.SYSXSROBJECTAUTH** table from **PUBLIC** to reduce risk to the organization's data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBMADM'  
       and tabname = 'AUTHORIZATIONIDS' and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBMADM.AUTHORIZATIONIDS FROM PUBLIC
```


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.			

4.3.47 Restrict Access to SYSIBM.SYSAUDITEXCEPTIONS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSAUDITEXCEPTIONS** table contains audit exceptions that are associated with an object. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

This table contains sensitive information about what audit exceptions exist. Access to the audit exceptions may aid attackers in avoiding detection.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname = 'SYSAUDITEXCEPTIONS'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSAUDITEXCEPTIONS FROM PUBLIC
```

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.			

4.3.48 Restrict Access to SYSIBM.SYSBUFFERPOOLAUTH (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBM.SYSBUFFERPOOLAUTH** table contains the bufferpool privileges granted to the user, group or role of a user. It is recommended that the **PUBLIC** role be restricted from accessing this table.

Rationale:

This table contains the bufferpool privileges granted to a user, group or roles of a user, and may be used as an attack vector.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBM' and tabname = 'SYSBUFFERPOOLAUTH'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBM.SYSBUFFERPOOLAUTH FROM PUBLIC
```

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.			

4.4 Secure the Database Administrative Views and Routines

Administrative views and routines provide additional information and functionality beyond that in the database catalogs. Many of these view and routines have **SELECT** and **EXECUTE** granted to **PUBLIC**. Security sensitive views and routines should be restricted to authorized users only.

4.4.1 Restrict Access to SYSIBMADM.AUTHORIZATIONIDS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

Administrative views and routines provide additional information and functionality beyond that in the database catalogs. Many of these view and routines have **SELECT** and **EXECUTE** granted to **PUBLIC**. Security sensitive views and routines should be restricted to authorized users only.

Rationale:

Databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for administrative views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBMADM.AUTHORIZATIONIDS** view from **PUBLIC** to reduce the risk a malicious user may use this view to conduct information gathering regarding the users that are known to the database server.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth
       where tabschema = 'SYSIBMADM'
       and tabname = 'AUTHORIZATIONIDS' and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SELECT ON SYSIBMADM.AUTHORIZATIONIDS FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-authorizationids-authorization-ids-types>

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.			

4.4.2 Restrict Access to SYSIBMADM.OBJECTOWNERS (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBMADM.OBJECTOWNERS** administrative view shows the complete object ownership information for each authorization ID of type **USER** owning a system catalog defined object from the connected database.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for administrative views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on the **SYSIBMADM.OBJECTOWNERS** view from **PUBLIC** to reduce risk of malicious user using this view to conduct information gathering regarding the users that own objects.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBMADM'  
       and tabname = 'OBJECTOWNERS' and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => revoke select on SYSIBMADM.OBJECTOWNERS from public
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-objectowners-object-ownership-information>

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.			

4.4.3 Restrict Access to SYSIBMADM.PRIVILEGES (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SYSIBMADM.PRIVILEGES** administrative view displays all explicit privileges for all authorization IDs in the currently connected databases' system catalogs. **PRIVILEGES** schema is **SYSIBMADM**.

Rationale:

Any databases created without the **RESTRICT** option automatically **GRANT** the **SELECT** privilege to **PUBLIC** for catalog views. Therefore, it is strongly recommended to explicitly **REVOKE** the **SELECT** privilege on **SYSIBMADM.PRIVILEGES** from **PUBLIC** to reduce risk a malicious user may use this view to conduct information gathering regarding the privileges that users have.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.tabauth  
       where tabschema = 'SYSIBMADM' and tabname = 'PRIVILEGES'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => revoke select on SYSIBMADM.PRIVILEGES from public
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-privileges-privilege-information>

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.			

4.4.4 Restrict Access to *SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID* (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

The table function **SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID** returns the instance and database authorities for the specified authorization ID. In a non-restrictive database this table function has **EXECUTE** granted to public. It is recommended that public should not be able to execute this routine.

Rationale:

A malicious user may use this function to conduct information gathering regarding users that have high level authorities.

Audit:

Perform the following Db2 commands to check if **PUBLIC** has **EXECUTE** on this routine:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.routineauth  
       where schema = 'SYSPROC'  
       and specificname = 'AUTH_LIST_AUTHORITIES_FOR_AUTHID'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **EXECUTE** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => revoke EXECUTE on function
        SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID
        from public RESTRICT
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-auth-list-authorities-authid-returns-list-agents>

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.			

4.4.5 Restrict Access to *SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID (Automated)*

Profile Applicability:

- Level 1 - RDBMS

Description:

The table function **SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID** returns the roles for the specified authorization ID. In a non-restrictive database this table function has **EXECUTE** granted to public. It is recommended that public should not be able to execute this routine.

Rationale:

A malicious user may use this function to conduct information gathering regarding the roles that a user belongs to.

Audit:

Perform the following Db2 commands to check if **PUBLIC** has **EXECUTE** on this routine:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.routineauth  
       where schema = 'SYSPROC'  
       and specificname = 'AUTH_LIST_ROLES_FOR_AUTHID'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **EXECUTE** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => revoke EXECUTE on function  
        SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID from public RESTRICT
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-auth-list-roles-authid-returns-roles-list>

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.			

4.4.6 Restrict Access to *SYSPROC.AUTH_LIST_GROUPS_FOR_AUTHID (Automated)*

Profile Applicability:

- Level 1 - RDBMS

Description:

The table function **SYSPROC.AUTH_LIST_GROUPS_FOR_AUTHID** returns the groups for the specified authorization ID. In a non-restrictive database this table function has **EXECUTE** granted to public. It is recommended that public should not be able to execute this routine.

Rationale:

A malicious user may use this function to conduct information gathering regarding the groups that users belong to.

Audit:

Perform the following Db2 commands to check if **PUBLIC** has **EXECUTE** on this routine:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee from syscat.routineauth  
       where schema = 'SYSPROC'  
       and specificname = 'AUTH_LIST_GROUPS_FOR_AUTHID'  
       and grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **EXECUTE** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => revoke EXECUTE on function  
        SYSPROC.AUTH_GROUPS_FOR_AUTHID from public RESTRICT
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=srv-auth-list-groups-authid-group-membership-authorization-ids>

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.			

4.5 General Database Considerations

4.5.1 Restrict Access to Tablespaces (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

A tablespace is where the data is physically stored. It is recommended that tablespace usage be restricted to authorized users.

Rationale:

Grant the **USE** of tablespace privilege to only authorized users. Restrict the privilege from **PUBLIC**, where applicable, as a malicious user can cause a denial of service at the tablespace level by overloading it with corrupted data.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select grantee, tbspace from sysibm.systbpaceauth  
       where grantee = 'PUBLIC'
```

3. If the output contains zero rows, then it is considered a Pass. Otherwise, **PUBLIC** has **SELECT** privilege and the remediation steps should be followed.

Remediation:

Perform the following to revoke access from **PUBLIC**.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE USE OF TABLESPACE [$tablespace_name\] FROM PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=views-syscattablespace>

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.			

4.5.2 Remove Unused Schemas (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

A schema is a logical grouping of database objects. It is recommended that unused schemas be removed from the database.

Rationale:

Unused schemas can be left unmonitored and may be subjected to abuse and therefore should be removed.

Audit:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select schemaname from syscat.schemata
```

3. Review the list of schemas

Remediation:

Remove unnecessary schemas.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => drop schema <schema name> restrict
```

4.5.3 Review System Tablespaces (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

System tablespaces store all system object data within that database. It is recommended that system tablespaces are used to store system data only and not user data. Only certain table spaces can be used to hold user tables.

Rationale:

Users should not have privileges to create user data objects within the system tablespaces. User data objects created within the system tablespaces should be removed.

Audit:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select tabschema,tablename,tbspace from syscat.tables
       where tabschema not in ('SYSIBM','SYSTOOLS')
       and tbspace in ('SYSTOOLSPACE','SYSTOOLSTMPSPACE')
```

3. Review the list of system tablespaces. If the output is **BLANK**, that is considered a Pass.

Remediation:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Review the system tablespaces to identify any user data objects within them.
3. Drop, migrate, or otherwise remove all user data objects (tables, schemas, etc.) from within the system tablespaces.
4. Revoke write access for the system tablespaces from all users.

5 Authentication Considerations

5.1 Specify a Secure Connection Authentication Type (SRVCON_AUTH) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SRVCON_AUTH** database manager configuration parameter specifies how and where authentication takes place for incoming connections to the database. This parameter overrides the **AUTHENTICATION** parameter.

This parameter can take on any of the following values:

- **NOT_SPECIFIED**
- **SERVER**
- **SERVER_ENCRYPT**
- **KERBEROS**
- **KRB_SERVER_ENCRYPT**
- **GSSPLUGIN**
- **GSS_SERVER_ENCRYPT**
- **SERVER_ENCRYPT_TOKEN**
- **KERBEROS_TOKEN**
- **GSSPLUGIN_TOKEN**
- **KRB_SVR_ENC_TOKEN**
- **GSS_SVR_ENC_TOKEN**

If this parameter is set to **NOT_SPECIFIED**, then the type of authentication for connections is determined by the **AUTHENTICATION** parameter.

Recommendations:

- Do not use **CLIENT** authentication type.

Rationale:

When using **CLIENT** authentication type, the server trusts the client to authenticate the connecting user. A malicious user can connect to the database as any user including a database administrator by simply creating that user on the client system. Note that **CLIENT** authentication is deprecated and can only be set if the **DB2AUTH** registry variable contains **ENABLE_CLIENT_AUTHN**.

When using **SERVER** authentication type without SSL enabled, the user ID and password that are sent from the client to the server during a connect or an attach are in plaintext format. Therefore, these credentials are exposed when sent across an unsecure network and can be intercepted by a malicious user.

The **SERVER_ENCRYPT** authentication type has been deprecated and may be removed in a future version of Db2. The Diffie-Hellman key exchange size does not meet NIST SP 80-131A recommendations. TLS is recommended to use to protect client-server communications.

Impact:

It is important to be aware that the implementation of this recommendation results in a brief downtime. It is advisable to ensure that the setting is implemented during an approved maintenance window.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the **SRVCON_AUTH** value in the output:

```
Server Connection Authentication (SRVCON_AUTH) = SERVER
```

Any value other than **CLIENT** is acceptable.

Remediation:

1. Attach to the Db2 instance

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => update database manager configuration parameter  
      using srvcon_auth <authentication type>
```

3. Restart the Db2 instance.

```
db2 => db2stop  
db2 => db2start
```


Refer to the 'encryption of data in motion' section for more information about using SSL for client-server communication.

References:

1. <https://www.ibm.com/docs/en/db2/12.1.x?topic=parameters-srvcon-auth-authentication-type-incoming-connections-server>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.7 Centralize Access Control Centralize access control for all enterprise assets through a directory service or SSO provider, where supported.			
v7	16.2 Configure Centralized Point of Authentication Configure access for all accounts through as few centralized points of authentication as possible, including network, security, and cloud systems.			

5.2 Specify a Secure Authentication Type (AUTHENTICATION) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **AUTHENTICATION** database manager configuration parameter specifies how and where authentication takes place for incoming connections to the database as well as local authorization of commands run outside a connection or attachment, such as tools run from the command line. This parameter can be overridden by the **SRVCON_AUTH** parameter for incoming connections.

This parameter can take on any of the following values:

- **SERVER**
- **SERVER_ENCRYPT**
- **KERBEROS**
- **KRB_SERVER_ENCRYPT**
- **GSSPLUGIN**
- **GSS_SERVER_ENCRYPT**

Recommendations:

- Do not use **CLIENT** authentication types.

Rationale:

When using **CLIENT** authentication type, the server trusts the client to authenticate the connecting user. A malicious user can connect to the database as any user including a database administrator by simply creating that user on the client system. Note that **CLIENT** authentication is deprecated and can only be set if the **DB2AUTH** registry variable contains **ENABLE_CLIENT_AUTHN**.

When using **SERVER** authentication type without SSL enabled, the user ID and password that are sent from the client to the server during a connect or an attach are in plaintext format. Therefore, these credentials are exposed when sent across an insecure network and can be intercepted by a malicious user.

The **SERVER_ENCRYPT** authentication type has been deprecated and may be removed in a future version of Db2. The Diffie-Hellman key exchange size does not meet NIST SP 80-131A recommendations. TLS is recommended to use to protect client-server communications.

Impact:

It is important to be aware that the implementation of this recommendation results in a brief downtime. It is therefore advisable to ensure that the setting is implemented during an approved maintenance window.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the **AUTHENTICATION** value in the output:

```
Database manager authentication (AUTHENTICATION) = SERVER
```

Remediation:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => update database manager configuration parameter  
using authentication <authentication type>
```

3. Restart the Db2 instance.

```
db2 => db2stop  
db2 => db2start
```

Refer to the 'encryption of data in motion' section for more information about using SSL for client-server communication.

References:

1. <https://www.ibm.com/docs/en/db2/12.1.x?topic=parameters-authentication-authentication-type>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.7 <u>Centralize Access Control</u> Centralize access control for all enterprise assets through a directory service or SSO provider, where supported.			
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.			

5.3 Database Manager Configuration Parameter: *ALTERNATE_AUTH_ENC* (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **ALTERNATE_AUTH_ENC** database manager configuration parameter specifies the encryption algorithm that is used to encrypt user ID and password that are sent from the client during a connect or attach. This parameter is in effect when the authentication method that is negotiated between the client and the server is **SERVER_ENCRYPT**.

It is recommended to set this parameter to **AES_ONLY**.

Rationale:

If this parameter is set to a value other than **AES_ONLY**, the server can accept the DES encryption algorithm to encrypt the user credentials and DES is cryptographically weak in comparison to AES.

Impact:

It is important to be aware that the implementation of this recommendation results in a brief downtime. It is therefore advisable to ensure that the setting is implemented during an approved maintenance window.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the **ALTERNATE_AUTH_ENC** value in the output:

```
Alternate authentication (ALTERNATE_AUTH_ENC) = AES_ONLY
```

Remediation:

1. Attach to the Db2 instance

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => update database manager configuration parameter  
using alternate_auth_enc aes_only
```

3. Restart the Db2 instance.

```
db2 => db2stop  
db2 => db2start
```

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.			

5.4 Database Manager Configuration Parameter: TRUST_ALLCLNTS (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

This database manager configuration parameter is only active when the authentication parameter is set to **CLIENT** which is not a recommended setting as discussed in the [authentication parameter section](#specify-a-secure-authentication-type-authentication). If the parameter is set to **YES**, the server assumes that the client side is handling authentication to the database. If the parameter is set to **NO**, the client must provide authentication to the server on behalf of the user.

The recommended value for this parameter is **NO**.

Rationale:

If the server trusts the client to authenticate the connecting user, a malicious user can connect to the database as any user including a database administrator by simply creating that user on the client system.

Impact:

It is important to be aware that the implementation of this recommendation results in a brief downtime. It is therefore advisable to ensure that the setting is implemented during an approved maintenance window.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the **AUTHENTICATION** and **SRVCON_AUTH** values in the output.

```
Server Connection Authentication          (SRVCON_AUTH) = NOT_SPECIFIED
Database manager authentication          (AUTHENTICATION) = SERVER
```

If **SRVCON_AUTH** does not equal **CLIENT**, or **SRVCON_AUTH** equals **NOT_SPECIFIED** and **AUTHENTICATION** does not equal **CLIENT**, then this is a PASS and you do not need to follow the remediation step. Otherwise one of **SRVCON_AUTH** or **AUTHENTICATION** equals **CLIENT**, and the following step must be followed:

Locate the **TRUST_ALLCLNTS** value in the output.

```
Trust all clients (TRUST_ALLCLNTS) = NO
```

Remediation:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => update database manager configuration parameter  
using trust_allclnts no
```

3. Restart the Db2 instance.

```
db2 => db2stop  
db2 => db2start
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.7 Centralize Access Control Centralize access control for all enterprise assets through a directory service or SSO provider, where supported.			

5.5 Database Manager Configuration Parameter: *TRUST_CLNTAUTH* (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

This database manager configuration parameter is only active when the authentication parameter is set to **CLIENT** which is not a recommended setting as discussed in the [authentication parameter section](#specify-a-secure-authentication-type-authentication). If the parameter is set to **CLIENT**, the user ID and password are not needed, but if they are provided, authentication will occur at the client. If the parameter is set to **SERVER**, the user ID and password are needed and will be authenticated at the server.

The recommended value for this parameter is **SERVER**.

Rationale:

If the server trusts the client to authenticate the connecting user, a malicious user can connect to the database as any user including a database administrator by simply creating that user on the client system.

Impact:

It is important to be aware that the implementation of this recommendation results in a brief downtime. It is therefore advisable to ensure that the setting is implemented during an approved maintenance window.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the **AUTHENTICATION** and **SRVCON_AUTH** values in the output.

```
Server Connection Authentication      (SRVCON_AUTH) = NOT_SPECIFIED
Database manager authentication      (AUTHENTICATION) = SERVER
```

If **SRVCON_AUTH** does not equal **CLIENT**, or **SRVCON_AUTH** equals **NOT_SPECIFIED** and **AUTHENTICATION** does not equal **CLIENT**, then this is a PASS and you do not need to follow the remediation step. Otherwise one of **SRVCON_AUTH** or **AUTHENTICATION** equaled **CLIENT**, and the following step must be followed:

Locate the **TRUST_CLNTAUTH** value in the output:

```
Trusted client authentication (TRUST_CLNTAUTH) = SERVER
```

Remediation:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => update database manager configuration parameter  
using trust_clntauth SERVER
```

3. Restart the Db2 instance.

```
db2 => db2stop  
db2 => db2start
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.7 Centralize Access Control Centralize access control for all enterprise assets through a directory service or SSO provider, where supported.			

5.6 Secure Permissions for All Authentication Plugins (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The following database manager configuration parameters allow the use of custom plugins for authentication and group lookup purposes.

- `CLNT_PW_PLUGIN`
- `CLNT_KRB_PLUGIN`
- `GROUP_PLUGIN`
- `LOCAL_GSSPLUGIN`
- `SRVCON_GSSPLUGIN_LIST`
- `SRVCON_PW_PLUGIN`

The permissions on the plugins specified by any of the above parameters should be secured so that users other than the instance owner don't have write privileges.

Rationale:

If a malicious user has write access to a plugin, they can overwrite it with their own thereby manipulating the authentication and authorization behavior for connecting users.

Audit:

Steps for Linux:

- 32-bit and 64-bit client side user authentication plugins are found in `$DB2PATH/security32/plugin/client` and `$DB2PATH/security64/plugin/client` directories respectively
- 32-bit and 64-bit server side user authentication plugins are found in `$DB2PATH/security32/plugin/server` and `$DB2PATH/security64/plugin/server` directories respectively

Review the permissions of the plugins that are in use:

```
ls -al
```

Steps for Windows:

- 32-bit and 64-bit client side user authentication plugins are found in `$DB2PATH\security\plugin\instance name\client`
- 32-bit and 64-bit server side user authentication plugins are found in `$DB2PATH\security\plugin\instance name\server`

Review the permissions of the plugins that are in use:

1. Right-click over the plugin file
2. Choose properties
3. Select the Security tab
4. Review the access for all accounts

Remediation:

To change permissions of a file on Linux:

```
chmod 755 <file>
```

To change permissions of a file on Windows:

1. Right-click on the file
2. Choose properties
3. Select the Security tab
4. Grant the Full Control authority to all Db2 administrator accounts
5. Grant only read and execute privileges to all other accounts (revoke any other privileges)

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.			

5.7 DB2_GRP_LOOKUP Registry Variable (Windows only) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DB2_GRP_LOOKUP** registry variable specifies which Windows security mechanism is used to enumerate the groups that a user belongs to. Periodic review of this variable is required to ensure that the correct location is being used for group definitions during authentication.

Rationale:

Incorrectly configured **DB2_GRP_LOOKUP** registry variable could result in unexpected authorization behavior where a low privileged user could potentially get access to sensitive data.

Audit:

Verify that the **DB2_GRP_LOOKUP** registry variable is set to the correct location by running the following command:

```
db2set -all
```

Remediation:

Run the following command to set the **DB2_GRP_LOOKUP** registry variable to the appropriate location for group lookup:

```
db2set DB2_GRP_LOOKUP=<location for group lookup>
```

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.			

5.8 DB2DOMAINLIST Registry Variable (Windows only) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

This registry variable is only active if the authentication parameter is set to CLIENT which is not a recommended setting as discussed in section 6.2. It is possible to have a user id be represented across multiple domains. Issues could arise when trying to authenticate such a user id. To prevent these issues, a listing of domains should be defined within the **DB2DOMAINLIST** registry variable.

Periodic review of the domain list assigned to the **DB2DOMAINLIST** registry variable helps ensure that non-essential domains do not have unnecessary authorizations.

Rationale:

Incorrectly configured **DB2DOMAINLIST** registry variable could result in unexpected authorization behavior where a low privileged user could potentially get access to sensitive data.

Audit:

Verify that the **DB2DOMAINLIST** registry variable includes only the appropriate domains by running the following command:

```
db2set -all
```

Remediation:

Run the following command to set the **DB2DOMAINLIST** registry variable to the appropriate domains:

```
db2set DB2DOMAINLIST=<ordered list of domains separated by comma>
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.1 <u>Establish and Maintain an Inventory of Accounts</u> Establish and maintain an inventory of all accounts managed in the enterprise. The inventory must include both user and administrator accounts. The inventory, at a minimum, should contain the person's name, username, start/stop dates, and department. 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.			

5.9 DB2AUTH Registry Variable (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DB2AUTH** registry variable is used to control various authentication related behaviors in Db2.

The following settings are recommended for this registry variable:

1. Use **DISABLE_CHGPASS** which disables the ability to change a user's password from the client.
2. Use **JCC_ENFORCE_SECMEC** which enforces that the Db2 server does not accept a clear text password security mechanism when using **SERVER_ENCRYPT** authentication type.
3. If **CLIENT** authentication is being used which is not recommended as discussed in section 6.2, it is also recommended to set this registry variable to **TRUSTEDCLIENT_SRVENC** and not **TRUSTEDCLIENT_DATAENC**. **TRUSTEDCLIENT_SRVENC** forces untrusted clients to use **SERVER_ENCRYPT** authentication type while **TRUSTEDCLIENT_DATAENC** forces them to use **DATA_ENCRYPT**.

If **DB2AUTH** is not set to **DISABLE_CHGPASS**, refer to the **DB2CHGPWD_EEE** registry variable section which specifies whether users are able to change passwords through Db2 in a partitioned database environment.

The following settings are recommended to not be set for this registry variable:

1. **ENABLE_CLIENT_AUTHN**

Rationale:

1. Allowing a client to change a user's password through Db2 commands is not recommended since Db2 may not enforce the expected password change rules. In addition, the password change requests going through Db2 may not be audited as expected.
2. Plain text passwords sent across an unsecure network are exposed and can be intercepted by a malicious user.
3. Enforcing **SERVER_ENCRYPT** ensures that the user ID and password are encrypted when sent from the client to the server.
4. Setting **ENABLE_CLIENT_AUTHN** allows the use of **CLIENT** authentication for the **AUTHENTICATION** or **SRVCON_AUTH** database manager configuration parameter, which is not recommended.

Audit:

Verify that the **DB2AUTH** registry variable is set to the appropriate values by running the following command:

```
db2set -all
```

Remediation:

Run the following command to set the **DB2AUTH** registry variable to the appropriate values:

```
db2set DB2AUTH=<comma separated values>
```

References:

1. https://www.ibm.com/docs/en/db2/12.1.x?topic=variables-miscellaneous#r0005669_M_DB2AUTH

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.			

5.10 DB2CHGPWD_EEE Registry Variable (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DB2CHGPWD_EEE** registry variable specifies whether users are able to change passwords through Db2 in a partitioned database environment.

It is recommended to set this variable to **NO**.

Rationale:

If password management in the partitioned database environment is not centralized, then it could result in a situation where one partition has the updated password for a given user while the rest have the old password which is a security risk.

Audit:

Verify that the **DB2CHGPWD_EEE** registry variable is set to the correct location by running the following command:

```
db2set -all
```

Remediation:

Run the following command to set the **DB2CHGPWD_EEE** registry variable to NO:

```
db2set DB2CHGPWD_EEE=NO
```

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.			

6 Authorization Considerations

6.1 Secure Database Authorities

6.1.1 Secure SYSADM Authority (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `sysadm_group` parameter defines the system administrator group (**SYSADM**) authority. It is recommended that the `sysadm_group` group contains authorized users only.

Rationale:

If an account that possesses this authority is compromised or used in a malicious manner, the confidentiality, integrity, and availability of data in the Db2 instance will be at increased risk.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the `sysadm_group` value in the output and ensure the value is not NULL:

```
(SYSADM_GROUP) = DB2ADM
```

4. Review the members of the `sysadm_group` on the operating system.

Linux:

```
cat /etc/group | grep <sysadm group name>
```

Windows:

1. Run `compmgmt.msc`
2. Click 'Local Users and Groups'

3. Double click 'Groups'
4. Double click <sysadm group name>
5. Review group members

Remediation:

Define a valid group name for the **SYSADM** group.

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => update database manager configuration
       using sysadm_group <sys adm group name>
```

Default Value:

The default value for **sysadm_group** is **NULL**.

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=authorities-system-administration-authority-sysadm>

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.			

6.1.2 Secure SYSCTRL Authority (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **sysctrl_group** parameter defines the system administrator group with system control (**SYSCTRL**) authority. It is recommended that the **sysctrl_group** group contains authorized users only.

Rationale:

If an account that possesses this authority is compromised or used in a malicious manner, the confidentiality, integrity, and availability of data in the Db2 instance will be at increased risk.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the **sysctrl_group** value in the output and ensure the value is not **NULL**:

```
SYSCTRL group name (SYSCTRL_GROUP) = DB2CTRL
```

4. Review the members of the **sysctrl_group** on the operating system.

Linux:

```
cat /etc/group | grep <sysctrl group name>
```

Windows:

1. Run **compmgmt.msc**
2. Click 'Local Users and Groups'
3. Double click 'Groups'
4. Double click <sysadm group name>
5. Review group members

Remediation:

Define a valid group name for the **SYSCTRL** group.

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => update dbm cfg using sysctrl_group <sys control group name>
```

Default Value:

The default value for **sysctrl_group** is **NULL**.

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=authorities-system-control-authority-sysctrl>

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.			

6.1.3 Secure SYSMAINT Authority (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **sysmaint_group** parameter defines the system administrator group that possesses the system maintenance (**SYSMAINT**) authority. It is recommended that the **sysmaint_group** group contains authorized users only.

Rationale:

If an account that possesses this authority is compromised or used in a malicious manner, the confidentiality, integrity, and availability of data in the Db2 instance will be at increased risk.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the **sysmaint_group** value in the output and ensure the value

is not **NULL**:

```
SYSMAINT group name (SYSMAINT_GROUP) = DB2MAINT
```

4. Review the members of the **sysmaint_group** on the operating system.

Linux:

```
cat /etc/group | grep <sysmaint group name>
```

Windows:

1. Run **compmgmt.msc**
2. Click 'Local Users and Groups'
3. Double click 'Groups'
4. Double click <sysadm group name>
5. Review group members

Remediation:

Define a valid group name for the **SYSMAINT** group.

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => update database manager configuration  
       using sysmaint_group <sys maintenance group name>
```

Default Value:

The default value for **sysmaint_group** is **NULL**.

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=authorities-system-maintenance-authority-sysmaint>

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.			

6.1.4 Secure SYSMON Authority (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **sysmon_group** parameter defines the operating system groups with system monitor (**SYSMON**) authority. It is recommended that the **sysmon_group** group contain authorized users only.

Rationale:

If an account that possesses this authority is compromised or used in a malicious manner, the confidentiality, integrity, and availability of data in the Db2 instance will be at increased risk.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the **sysmon_group** value in the output and ensure the value is not **NULL**:

```
SYSMON group name (SYSMON_GROUP) = DB2MON
```

4. Review the members of the **sysmon_group** on the operating system.

Linux:

```
cat /etc/group | grep <sysmon group name>
```

Windows:

1. Run **compmgmt.msc**
2. Click 'Local Users and Groups'
3. Double click 'Groups'
4. Double click <sysadm group name>
5. Review group members

Remediation:

Define a valid group name for the SYSMON group.

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => update database manager configuration  
       using sysmon_group <sys monitor group name>
```

Default Value:

The default value for `sysmon_group` is `NULL`.

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=authorities-system-monitor-authority-sysmon>

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.			

6.1.5 Secure SECADM Authority (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SECADM** (security administrator) role grants the authority to create, alter (where applicable), and drop roles, trusted contexts, audit policies, security label components, security policies and security labels. It is also the authority required to grant and revoke roles, security labels and exemptions, and the **SETSESSIONUSER** privilege. **SECADM** authority has no inherent privilege to access data stored in tables. It is recommended that the **SECADM** role be granted to authorized users only.

Rationale:

If an account that possesses this authority is compromised or used in a malicious manner, the confidentiality, integrity, and availability of data in the Db2 instance will be at increased risk.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select distinct grantee, granteetype from syscat.dbauth  
       where securityadmauth = 'Y'
```

3. Review the list of users, groups, and roles in the above output to ensure only approved users, groups, and roles are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SECADM ON DATABASE FROM USER <username>
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=authorities-security-administration-authority-secadm>

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.			

6.1.6 Secure DBADM Authority (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DBADM** (database administration) role grants the authority to a user to perform administrative tasks on a specific database. It is recommended that the **DBADM** role be granted to authorized users only.

Rationale:

If an account that possesses this authority is compromised or used in a malicious manner, the confidentiality, integrity, and availability of data in the database will be at increased risk.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select distinct grantee, granteetype from syscat.dbauth  
       where dbadmauth = 'Y'
```

3. Review the list of users, groups, and roles in the above output to ensure only approved users, groups, and roles are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE DBADM ON DATABASE FROM USER <username>
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=authorities-database-administration-authority-dbadm>

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.			

6.1.7 Secure SQLADM Authority (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SQLADM** authority is required to monitor, tune, and alter SQL statements.

Rationale:

The **SQLADM** authority can **CREATE**, **SET**, **FLUSH**, **DROP EVENT MONITORS** and perform **RUNSTATS** and **REORG INDEXES** and **TABLES**. **SQLADM** can be granted to users, groups, or roles or **PUBLIC**. **SQLADM** authority is a subset of the **DBADM** authority and can be granted by the **SECADM** authority.

Audit:

1. Run the following command:

```
db2 => select distinct grantee, granteetype from syscat.dbauth  
       where sqladmauth = 'Y'
```

2. Review the list of users, groups, and roles in the above output to ensure only approved users, groups, and roles are assigned.

Remediation:

Revoke **SQLADM** authority from any unauthorized users.

```
db2 => REVOKE SQLADM ON DATABASE FROM USER <username>
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=authorities-sql-administration-authority-sqladm>

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 Secure DATAACCESS Authority (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

Grants the authority to access data. The DATAACCESS authority allows the grantee to leverage DML level commands i.e. **SELECT**, **INSERT**, **UPDATE**, **DELETE**, **LOAD**, and **EXECUTE** any package or routine.

The DATAACCESS authority cannot be granted to **PUBLIC**.

Rationale:

The DATAACCESS authority gives the grantee read access and also control over manipulating the data. DATAACCESS can be granted to users, groups, or roles, but not **PUBLIC**. DATAACCESS authority is a subset of the **DBADM** authority and can be granted by the **SECADM** authority.

Audit:

1. Run the following command:

```
db2 => select distinct grantee, granteetype from syscat.dbauth
       where dataaccessauth = 'Y'
```

2. Review the list of users, groups, and roles in the above output to ensure only approved users, groups, and roles are assigned.

Remediation:

Revoke DATAACCESS authority from any unauthorized users.

```
db2 => REVOKE DATAACCESS ON DATABASE FROM USER <username>
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=ownership-database-authorities>

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.			

6.1.9 Secure ACCESSCTRL Authority (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

ACCESSCTRL authority is the authority required to grant and revoke privileges on objects within a specific database. Some of these privileges include **BINDADD**, **CONNECT**, **CREATETAB**, **CREATE_EXTERNAL_ROUTINE**, **LOAD**, and **QUIESCE_CONNECT**. It has no inherent privilege to access data stored in tables, except the catalog tables and views.

The **ACCESSCTRL** authority cannot be granted to **PUBLIC**.

Rationale:

The **ACCESSCTRL** authority gives the grantee access control to a specified database. With this authority, the grantee can grant/revoke privileges to other users. **ACCESSCTRL** can be granted to users, groups, or roles, but not **PUBLIC**. **ACCESSCTRL** authority can only be granted by the **SECADM** authority.

Audit:

1. Run the following command:

```
db2 => select distinct grantee, granteetype from syscat.dbauth  
       where accessctrlauth = 'Y'
```

2. Review the list of users, groups, and roles in the above output to ensure only approved users, groups, and roles are assigned.

Remediation:

Revoke **ACCESSCTRL** authority from any unauthorized users.

```
db2 => REVOKE ACCESSCTRL ON DATABASE FROM USER <username>
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=authorities-access-control-authority-accessctrl>

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 Secure WLMADM Authority (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **WLMADM** authority manages workload objects for a database. Holders of DBADM authority implicitly also hold **WLMADM** authority.

Rationale:

The **WLMADM** authority enables creating, altering, dropping, commenting, granting, and revoking access to workload objects for a database.

Audit:

1. Run the following command:

```
db2 => select distinct grantee, granteetype from syscat.dbauth
       where wlmadmauth = 'Y'
```

2. Determine if the grantee(s) are correctly set.

Remediation:

Revoke any user who should NOT have **WLMADM** authority:

```
db2 => REVOKE WLMADM ON DATABASE FROM USER <username>
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=authorities-workload-administration-authority-wlmadm>

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.			

Controls Version	Control	IG 1	IG 2	IG 3
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 Secure CREATAB Authority (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **CREATAB** (create table) role grants the authority to a user to create tables within a specific database. It is recommended that the **CREATAB** role be granted to authorized users only.

Rationale:

Review all users that have access to this authority to avoid the addition of unnecessary and/or inappropriate users.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select distinct grantee, granteetype from syscat.dbauth  
       where creatabauth = 'Y'
```

3. Review the list of users, groups, and roles in the above output to ensure only approved users, groups, and roles are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE CREATAB ON DATABASE FROM USER <username>
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=ownership-default-privileges-granted-creating-database>

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.			

6.1.12 Secure BINDADD Authority (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **BINDADD** (bind application) role grants the authority to a user to create packages on a specific database. It is recommended that the **BINDADD** role be granted to authorized users only.

Rationale:

If an account that possesses this authority is compromised or used in a malicious manner, the confidentiality, integrity, and availability of data in the database will be at increased risk.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select distinct grantee, granteetype from syscat.dbauth  
       where bindaddauth = 'Y'
```

3. Review the list of users, groups, and roles in the above output to ensure only approved users, groups, and roles are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE BINDADD ON DATABASE FROM USER <username>
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=ownership-database-authorities>

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.			

6.1.13 Secure CONNECT Authority (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **CONNECT** role grants the authority to a user to connect to mainframe and midrange databases from Windows, Unix, and Linux operating systems. It is recommended that the **CONNECT** role be granted to authorized users only.

Rationale:

All users that have access to this authority should be regularly reviewed.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select distinct grantee, granteetype from syscat.dbauth  
       where connectauth = 'Y'
```

3. Review the list of users, groups, and roles in the above output to ensure only approved users, groups, and roles are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE CONNECT ON DATABASE FROM USER <username>
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=statements-connect-type-1>

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.			

6.1.14 Secure LOAD Authority (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **LOAD** role grants the authority to a user to load data into tables. It is recommended that the **LOAD** role be granted to authorized users only.

Rationale:

All users that have access to this authority should be regularly reviewed.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select distinct grantee, granteetype from syscat.dbauth  
       where loadauth = 'Y'
```

3. Review the list of users, groups, and roles in the above output to ensure only approved users, groups, and roles are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE LOAD ON DATABASE FROM USER <username>
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=authorities-load-authority>

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.			

6.1.15 Secure *EXTERNALROUTINE* Authority (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **EXTERNALROUTINE** authority grants a user the privilege to create user-defined functions and procedures in a specific database.

Rationale:

All users with this authority should be regularly reviewed and approved.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select distinct grantee, granteetype from syscat.dbauth  
       where externalroutineauth = 'Y'
```

3. Review the list of users, groups, and roles in the above output to ensure only approved users, groups, and roles are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE CREATE_EXTERNAL_ROUTINE ON DATABASE  
       FROM USER <username>
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=routines-security>

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.			

6.1.16 Secure QUIESCECONNECT Authority (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **QUIESCECONNECT** role grants the authority to a user to access a database even in the quiesced state.

Rationale:

It is recommended that the **QUIESCECONNECT** role be granted to authorized users only.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select distinct grantee, granteetype from syscat.dbauth  
       where quiesceconnectauth = 'Y'
```

3. Review the list of users, groups, and roles in the above output to ensure only approved users, groups, and roles are assigned.

Remediation:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE QUIESCE_CONNECT ON DATABASE FROM USER <username>
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=apis-db2databasequiesce-quiesce-database>

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.			

6.1.17 Secure SETSESSIONUSER Privilege (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SETSESSIONUSER** privilege allows one user to switch the session authorization ID of the connection to another user. This allows one user to run SQL statements as another user. It is recommended that Trusted Context be used as a stronger assertion of user identity where such a feature is required.

Rationale:

A user can switch their identity to another user for whom they have been authorized and run SQL statements as that user.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select distinct trustedid, trustedidtype, surrogateauthid,  
surrogateauthidtype from syscat.surrogateauthids  
where trustedidtype != 'C'
```

3. Review the list of users, groups, and roles in the above output to ensure only approved users, groups, and roles are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SETSESSIONUSER ON <authid> FROM <authid>
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=statements-set-session-authorization>
2. <https://www.ibm.com/docs/en/db2/12.1?topic=statements-grant-setsessionuser-privilege>

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.			

6.1.18 Secure SCHEMAADM Authority (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SCHEMAADM** (schema administration) role grants the authority to a user to perform administrative tasks on a specific schema. It is recommended that the **SCHEMAADM** authority be granted to authorized users only.

Rationale:

If an account that possesses this authority is compromised or used in a malicious manner, the confidentiality, integrity, and availability of data in the database will be at increased risk.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select distinct schemaname, grantee, granteetype  
       from syscat.schemaauth where schemaadmauth = 'Y'
```

3. Review the list of users, groups, and roles in the above output to ensure only approved users, groups, and roles are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE SCHEMAADM ON SCHEMA <schema> FROM USER <username>
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=authorities-schema-administration-authority-schemaadm>

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.			

6.1.19 Secure Schema ACCESSCTRL Authority (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The Schema **ACCESSCTRL** authority allows a user grant and revoke privileges within a specific schema. It is recommended that the Schema **ACCESSCTRL** authority be granted to authorized users only.

Rationale:

If an account that possesses this authority is compromised or used in a malicious manner, the confidentiality, integrity, and availability of data in the database will be at increased risk.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select distinct schemaname, grantee, granteetype  
       from syscat.schemaauth where ACCESSCTRLAUTH = 'Y'
```

3. Review the list of users, groups, and roles in the above output to ensure only approved users, groups, and roles are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE ACCESSCTRL ON SCHEMA <schema> FROM USER <username>
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=authorities-schema-access-control-authority-accessctrl>

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.			

6.1.20 Secure Schema DATAACCESS Authority (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The Schema **DATAACCESS** authority allows the user to leverage DML level commands i.e. **SELECT**, **INSERT**, **UPDATE**, **DELETE**, **LOAD**, and **EXECUTE** any package or routine within a schema. It is recommended that the Schema **DATAACCESS** authority be granted to authorized users only.

Rationale:

If an account that possesses this authority is compromised or used in a malicious manner, the confidentiality, integrity, and availability of data in the database will be at increased risk.

Audit:

Perform the following Db2 commands to obtain the value for this setting:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => select distinct schemaname, grantee, granteetype  
       from syscat.schemaauth where dataaccessauth = 'Y'
```

3. Review the list of users, groups, and roles in the above output to ensure only approved users, groups, and roles are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Run the following command:

```
db2 => REVOKE DATAACCESSAUTH ON SCHEMA <schema> FROM USER <username>
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=authorities-schema-data-access-authority-dataaccess>

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 General Authorization

6.2.1 Review Users, Groups, and Roles (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

With row and column access control, individuals are permitted access to only the subset of data that is required to perform their job tasks. Periodic review of these individuals is crucial when trying to keep data secure. As business needs move forward, requirements behind accessing the data may change, leading to a revision in security policy. By inspecting the list of users, groups, and roles, you are identifying excessive privileges that could pose possible security threats within your infrastructure.

Rationale:

If a user (either by himself or part of a group or role) is no longer required access to the data that is protected by row and column access controls, allowing that individual to maintain access allows that individual to compromise the confidentiality, integrity, and/or availability of the data in the Db2 instance.

Audit:

1. Review the users within your database environment:
 - Linux:

```
cat /etc/passwd
```

- Windows:
 1. Run **compmgmt.msc**
 2. Click 'Local Users and Groups'
 3. Double click 'Users'
 4. Review users
2. Review the groups within your database environment:
 - Linux:

```
cat /etc/group
```

- Windows:
 1. Run **compmgmt.msc**
 2. Click 'Local Users and Groups'
 3. Double click 'Groups'
 4. Review groups
3. Review the roles and role members within your database environment:

1. Connect to Db2 database:

```
db2 => connect to <dbname>
```

2. Run the command:

```
db2 => select rolename, grantee from syscat.roleauth  
       where grantortype <> 'S'
```

Remediation:

1. To remove users from your database environment:

- o Linux:

```
userdel -r <user name>
```

- o Windows:

1. Run **compmgmt.msc**
2. Click 'Local Users and Groups'
3. Double click 'Users'
4. Right-click on
5. Select 'Delete'

2. To remove groups from your database environment:

- o Linux:

```
groupdel -r <group name>
```

- o Windows:

1. Run **compmgmt.msc**
2. Click 'Local Users and Groups'
3. Double click 'Groups'
4. Right-click on
5. Select 'Delete'

3. To remove roles or role members from your database environment:

1. Connect to Db2 database:

```
db2 => connect to <dbname>
```

2. To remove role members from roles:

```
db2 => revoke role <role name> from <usergroup/role member>
```

3. To remove roles:

```
db2 => drop role <role 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.			

6.2.2 Review Roles (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

Roles provide several advantages that make it easier to manage privileges in a database system. Security administrators can control access to their databases in a way that mirrors the structure of their organizations (they can create roles in the database that map directly to the job functions in their organizations). The assignment of privileges is simplified. Instead of granting the same set of privileges to each individual user in a particular job function, the administrator can grant this set of privileges to a role representing that job function and then grant that role to each user in that job function.

Rationale:

Reviewing the roles within a database helps minimize the possibility of unwanted access.

Audit:

1. Connect to Db2 database:

```
db2 => connect to <dbname>
```

2. Run the following and review the results to determine if each role name still has a business requirement to access the data:

```
db2 => select rolename from syscat.roleauth  
       where grantortype <> 'S' group by rolename
```

Remediation:

To remove a role from the database:

1. Connect to Db2 database:

```
db2 => connect to <dbname>
```

2. Run the following:

```
db2 => drop role <role name>
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=security-roles>

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 Review Role Members (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

Having roles that have been granted specific privileges, then assigning users to the roles, is usually considered the best way to grant application access. Because granting privileges to individual users can be more difficult to track and maintain against unauthorized access, users should be assigned to organization-defined database roles according to the needs of the business. As users leave the organization or change responsibilities within the organization, the appropriate roles for them change as well, so role membership needs to be reviewed and verified periodically.

Rationale:

Users who have excessive privileges not needed to do their jobs pose an unnecessary risk to the organization as an insider threat.

Audit:

1. Connect to Db2 database:

```
db2 => connect to <dbname>
```

2. Run the following to review and verify that the role members are correct for each role:

```
db2 => select rolename, grantee from syscat.roleauth  
       where grantortype <> 'S' group by rolename, grantee
```

Remediation:

To remove a role member from a particular role:

1. Connect to Db2 database:

```
db2 => connect to <dbname>
```

2. Run the following:

```
db2 => revoke role <role name> from <role member>
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=security-roles>

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 Nested Roles (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The user-defined roles in Db2 can be nested in the same fashion as Windows security groups, i.e. a nested group has both its directly assigned permissions as well as the assigned group permissions. By nesting roles, the database administrator is saving time by only having to assign a group of users versus assigning them individually. Nesting roles properly can often ease the application of the security model if it's kept fairly shallow, and if the roles are logically named. If these are all true, then nesting of roles is a good idea.

Rationale:

As tracking multiple levels of permissions can result in unauthorized access to data resources, this capability should be restricted according to the needs of the business.

Audit:

1. Connect to Db2 database:

```
db2 => connect to <dbname>
```

2. Run the following to identify any nested roles:

```
db2 => select grantee, rolename from syscat.roleauth  
       where grantee in (select rolename from syscat.roles)
```

If value is blank, this would be considered passing.

Remediation:

To remove a nested role, perform the following:

1. Connect to Db2 database:

```
db2 => connect to <dbname>
```

2. Run the following:

```
db2 => revoke role <role name> from role <role>
```

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 Review Roles Granted to PUBLIC (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

Granting to **PUBLIC** increases the risk of unauthorized entry into the database. Because **PUBLIC** is accessible by any database user, it is important to understand the exposure it has on all database objects. It would make sense to grant role membership to **PUBLIC** if all users required all the privileges granted through that role.

Rationale:

As any role granted to **PUBLIC** can potentially allow the compromise of database availability, confidentiality, or integrity, these roles should be restricted according to the needs of the business.

Audit:

1. Connect to Db2 database:

```
db2 => connect to <dbname>
```

2. Run the following:

```
db2 => select grantee, rolename from syscat.roleauth  
       where grantee = 'PUBLIC'
```

If the value returned is blank, that is considered a Pass.

Remediation:

To remove a role member from a particular role:

1. Connect to Db2 database:

```
db2 => connect to <dbname>
```

2. Run the following:

```
db2 => revoke role <role name> from PUBLIC
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=security-roles>

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 Review Role Grantees with WITH ADMIN OPTION (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

Using the **WITH ADMIN OPTION** clause of the **GRANT** (Role) SQL statement, the security administrator can delegate the management and control of membership in a role to someone else.

Rationale:

The **WITH ADMIN OPTION** clause gives another user the authority to grant membership in the role to other users, to revoke membership in the role from other members of the role, and to comment on a role, but not to drop the role.

Audit:

1. Connect to Db2 database:

```
db2 => connect to <dbname>
```

2. Perform the following query:

```
db2 => select rolename, grantee, admin from syscat.roleauth  
       where grantortype <> 'S' and admin = 'Y'
```

If the value returned is blank, that is considered a Pass.

Remediation:

1. Connect to Db2 database:

```
db2 => connect to <dbname>
```

2. Perform the following query:

```
db2 => revoke admin option for role <role name> from user <user 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.			

6.3 Row and Column Access Control

6.3.1 Review Organization's Policies Against Db2 RCAC Policies (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

DB2 Row and Column Access Control (RCAC) Policies control access to Db2 tables. They should match the organization's security and database access policies, and they should be regularly reviewed for gaps.

Rationale:

Missing, incomplete, or incorrect Db2 RCAC policies will increase the risks to the organization's protected data and will prevent efforts to monitor, alert, and respond to these risks in the future.

Audit:

Schedule and complete a regular review of all organization security and data access database policies against the current Db2 policies to determine if gaps exist.

1. Identify each written organization policy.
2. Find the matching Db2 RCAC policy.
3. Determine if the RCAC policy applies and correctly supports the written policy.
4. If no matching Db2 RCAC policy is found, record a 'gap' for future remediation.

Remediation:

1. Create RCAC policies for each 'gap' listed from the Audit procedure.
2. Review the newly created Db2 RCAC policy against the organization's written policies.

Default Value:

Not installed

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=security-row-column-access-control-rcac>

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.			

6.3.2 Review Row Permission Logic According to Policy (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The logic behind instituting row permissions is crucial for a successful security policy. Inspecting this logic and comparing it to the security policy will assure that all aspects of the data access controls are being adhered to.

Rationale:

Missing or incomplete Db2 RCAC Security Policies will increase the risks to the organization's protected data and will prevent efforts to monitor, alert, and respond to these risks in the future.

Audit:

1. Connect to database environment:

```
db2 => connect to <dbname>
```

2. Run the following and review the results to confirm that the row permissions are correct and that they comply with the existing security policy:

```
db2 => select role.rolename, control.ruletext
       from syscat.roles role
       inner join syscat.controls control
           on locate(role.rolename,control.ruletext) <> 0
       where enable = 'Y' and enforced = 'A' and valid = 'Y'
       and controltype = 'R'
```

Remediation:

1. Create RCAC Policies for each 'gap' listed from the Audit procedure.
2. Review the newly created Db2 RCAC policy against the organization's policy.

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=security-row-column-access-control-rcac>

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.			

6.3.3 Review Column Mask Logic According to Policy (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The logic behind instituting column masks is crucial for a successful security policy. Inspecting this logic and comparing it to the security policy will assure that all aspects of the data access controls are being adhered to.

Rationale:

Missing or incomplete Db2 RCAC security policies will increase the risks to the organization's protected data and will prevent efforts to monitor, alert, and respond to these risks in the future.

Audit:

1. Connect to database environment:

```
db2 => connect to <dbname>
```

2. Run the following and review the results to verify that the permissions are correct and that they comply with the organization's existing security policy:

```
db2 => select role.rolename, control.colname, control.ruletext
       from syscat.roles role
       inner join syscat.controls control
           on locate(role.rolename,control.ruletext) <> 0
       where enable = 'Y' and enforced = 'A' and valid = 'Y'
       and controltype = 'C'
```

Remediation:

1. Create RCAC Policies for each 'gap' listed from the Audit procedure.
2. Review the newly created Db2 RCAC policy against the organization's written policy.

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=security-row-column-access-control-rcac>

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 Trusted context Considerations

6.4.1 Ensure Trusted Contexts are Enabled (Manual)

Profile Applicability:

- Level 2 - RDBMS (extends Level 1 - RDBMS)

Description:

A Trusted Context object provides a means of enforcing encryption, assigning privileges based on roles, and ensuring that the actions performed on behalf of a user are performed in the context of the user's ID and privileges.

Rationale:

Creating Trusted Context objects to enforce encryption and assign roles will protect data in transit and limit access to information on a per user/role basis. Additionally, it ensures actions can be traced back to the user.

Audit:

Issue the following command to verify that a Trusted Context object is enabled:

```
db2 => select contextname, enabled from syscat.contexts where enabled = 'Y'
```

Remediation:

If there is no enabled Trusted Context object, create a Trusted Context object if needed and enable it.

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=connections-trusted-contexts-trusted>

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.			

6.4.2 Do Not Allow Trusted Context to Switch Users Without Authentication (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

A Trusted Context can provide a middle tier with the option of performing end user authentication, and then switching to that user within the database without further authentication. The middle tier is asserting the identity of the end user they have already authenticated. The organizations Standard Operating Procedures (SOP) will determine whether such trust has been placed with the middle tier establishing the trusted connection.

Rationale:

Allowing an untrusted middle tier to establish a trusted context and switch users without authentication is a security risk. Only middle tier applications that have been validated and included in the SOP should have this functionality enabled. All other trusted contexts should not use with **WITHOUT AUTHENTICATION** clause during the **CREATE TRUSTED CONTEXT** statement.

Audit:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Issue the following command to verify that no trusted context

```
db2 => select * from syscat.surrogateauthids where trustedidtype='C'
and authenticate='N' and trustedID!='SYSATSCONTEXT'
```

3. Review any rows returned to ensure they are allowed by the SOP for performing a switch user without authentication.

Remediation:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Disable any trusted context identified above until it can be determined why such trusted context was created with the following command

```
db2 => alter trusted context <contextname> alter disable
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=security-using-trusted-contexts-trusted-connections>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.8 <u>Define and Maintain Role-Based Access Control</u> Define and maintain role-based access control, through determining and documenting the access rights necessary for each role within the enterprise to successfully carry out its assigned duties. Perform access control reviews of enterprise assets to validate that all privileges are authorized, on a recurring schedule at a minimum annually, or more frequently.			

7 Audit Considerations

7.1 General Audit Considerations

7.1.1 Disable the Audit Buffer (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

Db2 can be configured to use an audit buffer where individual audit events are gathered into a large buffer to improve performance by reducing the number of writes to disk. It is recommended that the audit buffer be disabled by setting the size to 0.

Rationale:

Increasing the audit buffer size to greater than 0 will allocate space for the audit records generated by the audit facility. At scheduled intervals, or when the audit buffer is full, the `db2auditd` audit daemon empties the audit buffer to disk, writing the audit records asynchronously. As the events are held in memory for some time before being written to disk, if the database server happened to crash those event would be lost. Setting the buffer size to 0 ensure events are written directly to disk.

Impact:

Disabling the audit buffer may have noticeable impact on overall performance of the database server.

Audit:

Perform the following to determine if the audit buffer is set as recommended:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate `AUDIT_BUF_SZ` value in the output:

```
Audit buffer size (4KB) (AUDIT_BUF_SZ) = 0
```

Ensure `AUDIT_BUF_SZ` is equal to 0.

Remediation:

Perform the following to disable the audit buffer:

1. Attach to the Db2 instance

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => update database manager configuration using audit_buf_sz 0
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=parameters-audit-buf-sz-audit-buffer-size>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.3 <u>Ensure Adequate Audit Log Storage</u> Ensure that logging destinations maintain adequate storage to comply with the enterprise's audit log management process.			
v7	6.4 <u>Ensure adequate storage for logs</u> Ensure that all systems that store logs have adequate storage space for the logs generated.			

7.1.2 Disable Limited Audit of Applications (DB2_LIMIT_AUDIT_APPS) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DB2_LIMIT_AUDIT_APPS** registry variable contains a list of application names that should not be audited. It is recommended that this variable should not be set and all applications should be audited.

The **DB2_LIMIT_AUDIT_APPS** registry variable is not documented.

Rationale:

The application name not to be audited is determined by the client and not validated by the server. It is possible for a malicious user to change their application name to avoid being audited.

Audit:

Perform the following to determine if the **DB2_LIMIT_AUDIT_APPS** registry variable is set:

```
db2set -all | grep DB2_LIMIT_AUDIT_APPS
```

The above command should not yield a value.

Remediation:

Perform the following command to remove any applications from the list:

```
db2set DB2_LIMIT_AUDIT_APPS=
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.1 <u>Establish and Maintain an Audit Log Management Process</u> Establish and maintain an audit log management process that defines the enterprise's logging requirements. At a minimum, address the collection, review, and retention of audit logs for enterprise assets. Review and update documentation annually, or when significant enterprise changes occur that could impact this Safeguard.			

7.1.3 Ensure Audit Policies are Enabled Within the Database (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

Creating and applying audit policies is crucial for securing and discovering issues within your databases.

Audit policies can help trigger events for changes to data objects, table DML, and user access.

Rationale:

If audit policies are not enabled, issues may go undiscovered, and compromises and other incidents may occur without being quickly detected. It may also not be possible to provide evidence of compliance with security laws, regulations, and other requirements.

Impact:

Auditing all categories within the database can have an impact on the performance of the database server depending on the workload and number of transactions. If enabling audit as part of the remediation, analysis should be performed on which categories are required to meet business needs.

Audit:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Issue the following command to ensure that at least one audit policy exists and has been associated with an object. The SYSCAT.AUDITUSE catalog view shows all of the audit policies that have been associated with auditing an object within the database.

```
db2=> select * from syscat.audituse
```

3. If there is at least 1 row then it is considered a Pass. If there are zero rows returned then nothing is being audited within the database and the remediation steps should be followed.

Remediation:

1. Connect to the Db2 database.

```
db2 => connect to <dbname>
```

2. Issue the following command to create an audit policy. This policy audits all categories. An analysis should be performed to determine which categories are required to meet business needs.

```
db2 => create audit policy AUDITDB CATEGORIES ALL STATUS BOTH ERROR  
TYPE AUDIT
```

3. Audit the database using the policy just created with the following command:

```
db2 => audit database using policy AUDITDB
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=statements-create-audit-policy>
2. <https://www.ibm.com/docs/en/db2/12.1?topic=statements-audit>

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.			
v7	6.2 <u>Activate audit logging</u> Ensure that local logging has been enabled on all systems and networking devices.			

7.1.4 Ensure Audit is Enabled Within the Instance (Automated)

Profile Applicability:

- Level 1 - RDBMS

Description:

Auditing is crucial for securing and discovering issues within your databases.

Auditing can help trigger events for changes to data objects, table DML, and user access.

Rationale:

If instance auditing is not enabled, issues may go undiscovered, and compromises and other incidents may occur without being quickly detected. It may also not be possible to provide evidence of compliance with security laws, regulations, and other requirements.

Audit:

1. Issue the following command to ensure that auditing is active at the instance level.

```
$ db2audit describe

DB2 AUDIT SETTINGS:

Audit active: "TRUE "
Log audit events: "FAILURE"
Log checking events: "FAILURE"
Log object maintenance events: "FAILURE"
Log security maintenance events: "FAILURE"
Log system administrator events: "FAILURE"
Log validate events: "FAILURE"
Log context events: "NONE"
Return SQLCA on audit error: "FALSE "
Audit Data Path: ""
Audit Archive Path: ""

AUD0000I  Operation succeeded.
```

2. If the output contains **Audit active: "TRUE "**, then it is considered a Pass. Otherwise, instance level auditing is not enabled and the remediation steps should be followed.

Remediation:

Issue the following command to activate instance level auditing:

```
$ db2audit start
```

References:

1. <https://www.ibm.com/docs/en/db2/12.1?topic=commands-db2audit-audit-facility-administrator-tool>

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.			
v7	6.2 <u>Activate audit logging</u> Ensure that local logging has been enabled on all systems and networking devices.			

8 Encryption Considerations

8.1 Encryption of Data in Motion

8.1.1 Configure a Server-side Key Store for TLS (SSL_SVR_KEYDB) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

To enable TLS support in a Db2 server, it is necessary to configure a key store that will contain certificates to be used for secure TLS communication between a Db2 client and Db2 server.

Rationale:

On the server side, Db2 requires a key store to be configured. Otherwise, TLS support cannot be enabled.

Audit:

Perform the following to determine if **SSL_SVR_KEYDB** is set.

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the value of CATALOG_NOAUTH in the output:

```
SSL server keydb file (SSL_SVR_KEYDB) =
```

Ensure **SSL_SVR_KEYDB** is not blank.

Remediation:

Perform the following to set **SSL_SVR_KEYDB**:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```


2. Run the following command, where *<path>* is the fully qualified path to the keystore file:

```
db2 => update dbm cfg using SSL_SVR_KEYDB <path>
```

3. (Optional) To use the Microsoft certificate store on Windows, set **SSL_SVR_KEYDB** to **GSK_MS_CERTIFICATE_STORE**:

```
db2 => update dbm cfg using SSL_SVR_KEYDB GSK_MS_CERTIFICATE_STORE
```

If a keystore file is being used, ensure only the instance owner and administrators have access to the file. Do not grant world readable or world writable permissions on the keystore file.

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.			

8.1.2 Configure a Server-side Stash File for TLS (SSL_SVR_STASH) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

If a key store file is being used to configure TLS support in a Db2 instance, a stash file must also be configured to allow the Db2 server to be able to read certificates from the keystore. If the Microsoft certificate store is configured on a Windows platform, a stash file is not required.

Rationale:

Db2 does not have a method for an operator to enter a password for a server-side SSL key store, so a stash file must be used to provide the password to Db2. The Microsoft certificate store does not require a password, so a stash file is not required.

Audit:

Perform the following to determine if **SSL_SVR_STASH** is required, and if it is set.

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the value of **SSL_SVR_KEYDB** in the output:

```
SSL server keydb file (SSL_SVR_KEYDB) =
```

4. A stash file is not required on Windows if using **GSK_MS_CERTIFICATE_STORE**. If the value of **SSL_SVR_KEYDB** is not **GSK_MS_CERTIFICATE_STORE**, locate the value of **SSL_SVR_STASH** and ensure **SSL_SVR_STASH** is not blank.

```
SSL server stash file (SSL_SVR_STASH) =
```

Remediation:

Perform the following to set **SSL_SVR_STASH**:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command, where *<path>* is the fully qualified path to the keystore file:

```
db2 => update dbm cfg using SSL_SVR_STASH <path>
```

If a stash file is being used, ensure only the instance owner and administrators have access to the file. Do not grant world readable or world writable permissions on the stash file.

If the Microsoft certificate store is being used on Windows, it is not necessary to set **SSL_SVR_STASH**.

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.			

8.1.3 Configure an Endpoint Certificate (SSL_SVR_LABEL) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SSL_SVR_LABEL** database manager configuration parameter controls which certificate Db2 will serve to clients. This certificate must have its associated certificate chain present in the server-side key store and must be associated with a private key.

Rationale:

It is highly recommended to set **SSL_SVR_LABEL**. Leaving this parameter blank and allowing Db2 to utilize a default certificate will only work with CMS(.KDB) format key stores, and the feature is deprecated.

Audit:

Perform the following to determine if **SSL_SVR_LABEL** is set.

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the value of **SSL_SVR_LABEL** in the output:

```
SSL server certificate label (SSL_SVR_LABEL) =
```

Remediation:

Perform the following to set **SSL_SVR_LABEL**:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command, where **<label>** is the name of a certificate present in the server-side key store.

```
db2 => update dbm cfg using SSL_SVR_LABEL <label>
```

Updating the value of **SSL_SVR_LABEL** while attached to the instance will cause the certificate served by Db2 to change while instance is running, with no effect on existing connections.

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).			
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	16.4 <u>Encrypt or Hash all Authentication Credentials</u> Encrypt or hash with a salt all authentication credentials when stored.			
v7	16.5 <u>Encrypt Transmittal of Username and Authentication Credentials</u> Ensure that all account usernames and authentication credentials are transmitted across networks using encrypted channels.			

8.1.4 Configure the Service Name for TLS (SSL_SVCENAME) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SSL_SVCENAME** database manager configuration parameter controls which port Db2 will listen on for TLS encrypted connections. **SSL_SVCENAME** can consist of one of the following:

- A port number
- Service name defined in **/etc/services** (UNIX/Linux) or **%WINDIR%\system32\drivers\etc\services** (windows)

Rationale:

Db2 must have a port number or service name defined to enable TLS communication. Db2 does not choose a default port number if **SSL_SVCENAME** is unset.

Audit:

Perform the following to determine if **SSL_SVCENAME** is set.

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the value of **SSL_SVCENAME** in the output:

```
SSL service name (SSL_SVCENAME) =
```

Remediation:

Perform the following to set **SSL_SVCENAME**:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command, where *<service>* is a port number or named service.

```
db2 => update dbm cfg using SSL_SVCENAME <service>
```

Db2 must be recycled (**db2stop/db2start**) for changes to **SSL_SVCENAME** to take effect.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>16.10 Apply Secure Design Principles in Application Architectures</u> Apply secure design principles in application architectures. Secure design principles include the concept of least privilege and enforcing mediation to validate every operation that the user makes, promoting the concept of "never trust user input." Examples include ensuring that explicit error checking is performed and documented for all input, including for size, data type, and acceptable ranges or formats. Secure design also means minimizing the application infrastructure attack surface, such as turning off unprotected ports and services, removing unnecessary programs and files, and renaming or removing default accounts.			
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.			

8.1.5 Configure a Secure TLS Version (SSL_VERSIONS) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SSL_VERSIONS** database manager configuration parameter controls which versions of the TLS protocol Db2 enables.

Rationale:

The versions of TLS considered secure changes over time. This recommendation will be updated to reflect those changes. Db2 12.1 supports TLS 1.2 and 1.3, both of which are considered secure. Previous versions of Db2 supported TLS 1.0 and 1.1 which are insecure and should not be used.

Audit:

Perform the following to determine if **SSL_VERSIONS** is set.

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the value of **SSL_VERSIONS** in the output:

```
SSL versions (SSL_VERSIONS) =
```

At the current time, either having **SSL_VERSIONS** not set, or set to either or both of **TLSV12** and **TLSV13** is a pass. This may change in future updates if there are changes to the accepted security of TLS versions.

Remediation:

Perform the following to set **SSL_VERSIONS**:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command to enable TLS 1.3 within the Db2 server.


```
db2 => update dbm cfg using SSL_VERSIONS TLSV13
```

Default Value:

At the DB2 server, TLS 1.2 and TLS 1.3 are enabled by default if SSL_VERSIONS is not set.

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.			

8.1.6 Configure Secure TLS Cipher Suites (SSL_CIPHERSPECS) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SSL_CIPHERSPECS** database manager configuration parameter controls which cipher suites are enabled by Db2. If it is unset, Db2 will enable a default list of ciphers.

Rationale:

Some cipher suites have known issues or do not enable perfect forward secrecy. Specifically the **TLS_RSA_WITH_*** cipher have known weaknesses and do not support perfect forward secrecy and should not be enabled.

All of the TLS 1.3 ciphers suites are considered secure, it is only some of those for TLS 1.2 that are of concern.

Audit:

There are 4 conditions that are required for the **TLS_RSA_WITH_*** cipher suites to be enabled.

1. **TLS** has been enabled
2. **TLSV12** has been enabled
3. **NOFIPS** mode has been enabled
4. **SSL_CIPHERSPECS** is either blank (default) or lists one or more of **TLS_RSA_WITH_*** cipher suite.

Perform the following to see if **TLS** is in use:

1. Check if TLS has been turned on. As the instance owner, run the following command.

```
db2set -all DB2COMM
```

2. The value of **DB2COMM** will be returned in the output. If it does not contain SSL, then TLS is not in use and this condition is a pass.

```
[i] SSL
```

Perform the following to see if **TLSV12** is in use:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the value of **SSL_VERSIONS** in the output:

```
SSL versions (SSL_VERSIONS) =
```

If the output is blank, or **TLSV12** is listed, then **TLSV12** is enabled, then you must continue to see if **NOFIPS** mode is enabled. If only **TLSV13** is enabled the this is a pass.

Perform the following to see if **NOFIPS** has been set:

1. As the instance owner, run the following command.

```
db2set -all DB2AUTH
```

2. The value of **DB2AUTH** will be returned in the output.

```
[i] STRICT_FIPS
```

If the output does not contain **NOFIPS** then this is a pass. If **NOFIPS** is listed then the **SSL_CIPHERSPECS** must be examined.

Perform the following to determine if **SSL_CIPHERSPECS** is set.

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the value of **SSL_CIPHERSPECS** in the output:

```
SSL cipher specs (SSL_CIPHERSPECS) =
```

It is considered a pass if the value is set to any list of values that does not contain **TLS_RSA_WITH_*** cipher suites. If it is blank or contains a **TLS_RSA_WITH_*** cipher suite then this is a fail and the remediation should be followed.

Remediation:

Perform the following to set **SSL_CIPHERSPECS**:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command to enable a specific set of ciphers within the Db2 server.

```
db2 => update dbm cfg using SSL_CIPHERSPECS <LIST>
```

Replace **<LIST>** with one or multiple of the following cipher suites. If multiple items are specified, separate them with a single comma and no spaces.

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

The choice of which values to use will depend on the type of certificate that the server is using. Even if TLS 1.3 is in use, it is acceptable not to specify any TLS 1.3 cipher suites in **SSL_CIPHERSPECS**, as the default list for TLS 1.3, all of which are secure, will be used.

Default Value:

At the Db2 server, the default cipher suites contain only recommended values when running in FIPS or **STRICT_FIPS** mode. Specifically the **TLS_RSA_WITH_*** cipher suites are not enabled by default in **FIPS** or **STRICT_FIPS** mode, nor can they even be set. In **NOFIPS** mode, the **TLS_RSA_WITH_*** cipher suites are enabled by default for **TLSV12**. Consult the link in the references for the full list of default ciphersuites depending on which version of TLS has been enabled.

References:

1. <https://www.ibm.com/docs/en/db2/12.1.x?topic=parameters-ssl-cipherspecs-supported-cipher-specifications-server>

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.			

8.1.7 Unset the Service Name for Plaintext Communication (SVCENAME) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **SSVCENAME** database manager configuration parameter controls which port Db2 will listen on for unencrypted connections.

Rationale:

To prevent unencrypted remote connections to the DB2 instance, it is good practice to unset the value of the **SVCENAME DBM CFG** parameter to prevent Db2 from starting the TCP listener, even if **DB2COMM** is set to **TCPIP**.

Audit:

Perform the following to determine if **SSL_SVCENAME** is set.

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the value of **SVCENAME** in the output:

```
TCP/IP Service name (SVCENAME) =
```

Remediation:

Perform the following to unset **SVCENAME**:

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command, where **<service>** is a port number or named service.

```
db2 => update dbm cfg using SVCENAME NULL
```

Db2 must be recycled (**db2stop/db2start**) for changes to **SSL_SVCENAME** to take effect.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>16.10 Apply Secure Design Principles in Application Architectures</u> Apply secure design principles in application architectures. Secure design principles include the concept of least privilege and enforcing mediation to validate every operation that the user makes, promoting the concept of "never trust user input." Examples include ensuring that explicit error checking is performed and documented for all input, including for size, data type, and acceptable ranges or formats. Secure design also means minimizing the application infrastructure attack surface, such as turning off unprotected ports and services, removing unnecessary programs and files, and renaming or removing default accounts.			
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.			

8.1.8 Configure a Client-side Key Store for TLS (SSL_CLNT_KEYDB) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

To enable TLS support in a Db2 client, it is possible to configure a key store in the database manager configuration that will contain root certificates to be used for secure TLS communication between a Db2 client and Db2 server.

Rationale:

On the client side, Db2 requires the root certificates for the server to be available. This can be achieved by configuring a client-side keystore. This parameter is optional and is not needed if clients use the `SSLClientKeystoredb` or `SSLServerCertificate` parameters in the `db2cli.ini` or `db2dsdriver.cfg` files, or in the connection string.

Audit:

Perform the following to determine if `SSL_CLNT_KEYDB` is set.

1. Run the following command:

```
db2 => get database manager configuration
```

2. Locate the value of `SSL_CLNT_KEYDB` in the output:

```
SSL client keydb file (SSL_CLNT_KEYDB) =
```

Remediation:

Perform the following to set `SSL_CLNT_KEYDB`:

1. Run the following command, where `<path>` is the fully qualified path to the keystore file:

```
db2 => update dbm cfg using SSL_CLNT_KEYDB <path>
```

2. (Optional) To use the Microsoft certificate store on Windows, set `SSL_CLNT_KEYDB` to `GSK_MS_CERTIFICATE_STORE`:

```
db2 => update dbm cfg using SSL_CLNT_KEYDB GSK_MS_CERTIFICATE_STORE
```

3. Restart the client application. If the CLP is being used, run the following command to terminate the background process


```
db2 => terminate
```

If a client-side keystore file is being used, ensure that any users running client applications, and administrators have access to the file. Do not grant world readable or world writable permissions on the stash file.

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.			

8.1.9 Configure a Client-side Stash File for TLS (SSL_CLNT_STASH) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

If a key store file is being used to configure client-side TLS support in a Db2 instance, a stash file should also be configured to allow the Db2 client to be able to read certificates from the keystore. If the Microsoft certificate store is configured on a Windows platform, a stash file is not required.

Rationale:

The database manager configuration does not provide a method for specifying a password for the client-side SSL key store. The Microsoft certificate store does not require a password, so a stash file is not required.

Impact:

Perform the following to determine if **SSL_CLNT_STASH** is required, and if it is set.

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the value of SSL_CLNT_KEYDB in the output:

```
SSL client keydb file (SSL_CLNT_KEYDB) =
```

4. If the value of SSL_CLNT_KEYDB is not **GSK_MS_CERTIFICATE_STORE**, locate the value of **SSL_CLNT_STASH** and ensure **SSL_CLNT_STASH** is not blank.

```
SSL client stash file (SSL_CLNT_STASH)=
```

Audit:

Perform the following to set **SSL_CLNT_STASH**:

1. Run the following command, where **<path>** is the fully qualified path to the keystore file:

```
db2 => update dbm cfg using SSL_CLNT_KEYDB <path>
```

2. Restart the client application. If the CLP is being used, run the following command to terminate the background process

```
db2 => terminate
```

If a stash file is being used, ensure that any users running client applications, and administrators have access to the file. Do not grant world readable or world writable permissions on the stash file.

If the Microsoft certificate store is being used on Windows, it is not necessary to set **SSL_CLNT_STASH**.

Remediation:

Perform the following to set **SSL_CLNT_STASH**:

1. Run the following command, where is the fully qualified path to the keystore file:

```
db2 => update dbm cfg using SSL_CLNT_KEYDB <path>
```

2. Restart the client application.

If the CLP is being used, run the following command to terminate the background process

```
db2 => terminate
```

If a stash file is being used, ensure that any users running client applications, and administrators have access to the file. Do not grant world readable or world writable permissions on the stash file.

If the Microsoft certificate store is being used on Windows, it is not necessary to set **SSL_CLNT_STASH**.

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.			

8.1.10 Enable TLS Communication Between HADR Primary and Standby Instances (HADR_SSL_LABEL) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **HADR_SSL_LABEL** database configuration parameter controls whether connections between HADR peers are encrypted, and which certificate is served to an HADR peer when establishing an HADR connection.

Rationale:

To protect database data and log records when they are sent from a primary database to a standby database, the **HADR_SSL_LABEL** database configuration should be set.

Audit:

Perform the following to determine if the **HADR_SSL_LABEL** is set. These steps should be performed on the primary database and any standby databases.

1. As the instance owner, run the following command.

```
db2 => get db cfg for <database>
```

2. Locate the value of **HADR_SSL_LABEL** in the output:

```
HADR SSL certificate label (HADR_SSL_LABEL) =
```

Remediation:

Perform the following steps on both the primary and any standby databases to enable TLS encrypted HADR. A server side keystore and stash file (**SSL_SVR_KEYDB/SSL_SVR_STASH**) must be configured to enable TLS encrypted HADR communication:

- Run the following command as the instance owner.

```
db2 => update db cfg for <database> using HADR_SSL_LABEL <label>
```

Replace **<label>** with the name of a certificate present in the server-side keystore (**SSL_SVR_KEYDB**).

If it is active, HADR must be recycled for changes to the **HADR_SSL_LABEL** registry variable to take effect.

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.			

8.1.11 Enable Remote TLS Connections to Db2 (DB2COMM) (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The **DB2COMM** registry variable controls what types of remote connections Db2 will accept. It can be configured to enable plaintext communication, encrypted communication, or both.

Rationale:

For security, **DB2COMM** should be set to enable only TLS encrypted communications.

Audit:

Perform the following to determine if **DB2COMM** is set.

1. As the instance owner, run the following command.

```
db2set -all DB2COMM
```

2. The value of **DB2COMM** will be returned in the output. If the **DB2COMM** registry variable is not set, the db2set command will not return any data.

```
[i] TCPIP
```

Remediation:

Perform the following to set **DB2COMM** and enable TLS:

1. Run the following command as the instance owner.

```
db2set DB2COMM=SSL
```

Db2 must be recycled (**db2stop/db2start**) for changes to the **DB2COMM** registry variable to take effect.

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.			

8.2 Encryption of Data at Rest

Db2 Native Encryption is transparent data encryption for data at rest. Db2 native encryption uses a two-tier approach to data encryption. Data is encrypted with a Data Encryption Key (DEK), which is in turn encrypted with a Master Key (MK). The encrypted DEK is stored with the data while the MK is stored in a keystore external to Db2.

When you encrypt a database, Db2 native encryption protects all files that contain your data, such as:

- All table spaces (both system-defined and user-defined)
- All types of data in a table space (including **LOB** and **XML** data types)
- All transaction logs, including archived log files
- **LOAD COPY** data
- **LOAD** staging files

Db2 native encryption can also be used to encrypt database backups, even if the source database is not encrypted. This section provides guidance on using native encryption within Db2.

8.2.1 Encrypt the Database (Manual)

Profile Applicability:

- Level 2 - RDBMS (extends Level 1 - RDBMS)

Description:

Encrypting the database will protect your data and may be required for compliance with certain government regulations (e.g. NIST).

Rationale:

A malicious user can steal physical media like the hard drive, and restore the database and browse the data. To prevent access to the data at rest, encrypt sensitive data in a database and use an encryption key that encrypt the data. This prevents anyone without the keys from using the data.

Audit:

Check that the database is encrypted via examining configuration settings.

Remediation:

To determine if a database is encrypted you should check the **Encrypted database** database configuration parameter:

```
db2 => get db cfg | grep -i encrypt
Encryption Library for Backup (ENCRLIB) = libdb2encr.so
Encryption Options for Backup (ENCROPTS) = CIPHER=AES:MODE=CBC:KEY LENGTH=256
Encrypted database = YES
```

You should see that the encrypted database is set to **YES**.

You should check the type of key manger used by checking dbm configuration:

```
db2 => get dbm cfg | grep -i keystore
Keystore type (KEYSTORE_TYPE) = KMIP
Keystore location (KEYSTORE_LOCATION) = /path/ekeystore.cfg
```

You can also check the current database encryption settings:

```
db2 => SELECT * FROM TABLE(SYSPROC.ADMIN_GET_ENCRYPTION_INFO())
```

You must be connected to the database to run this command. The following information will be shown for the connected database:

- **OBJECT_NAME**
- **OBJECT_TYPE**
- **ALGORITHM**
- **ALGORITHM_MODE**
- **KEY_LENGTH**
- **MASTER_KEY_LABEL KEYSTORE_NAME KEYSTORE_TYPE**

- KEYSTORE_HOST KEYSTORE_IP
- KEYSTORE_IP_TYPE
- PREVIOUS_MASTER_KEY_LABEL AUTH_ID
- APPL_ID
- ROTATION_TIME

This information should be filled in for an encrypted database.

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.			●

8.2.2 Do Not Use Encryption Algorithms that are Not Secure (Manual)

Profile Applicability:

- Level 2 - RDBMS (extends Level 1 - RDBMS)

Description:

Encryption can be performed with various algorithms, some of which are outdated and should no longer be used.

Rationale:

Using an outdated algorithm can leave the data vulnerable. As new methods of attack are created and hardware processing speeds increase, the encryption algorithms can become vulnerable. Thus, only current encryption algorithms should be used. 3DES is an outdated encryption algorithm and should not be used.

Audit:

Check that you are not using an outdated algorithm (like 3DES) by looking at the configuration settings.

Remediation:

To determine what encryption options used to encrypt the database you can run the following command:

```
db2 => get db cfg | grep -i encrypt
Encryption Library for Backup (ENCRLIB) = libdb2encr.so
Encryption Options for Backup (ENCROPTS) = CIPHER=AES:MODE=CBC:KEY LENGTH=256
Encrypted database = YES
```

ENCROPTS should not contain **CIPHER=3DES** algorithm, because **3DES** is not secure. **CIPHER=AES** is secure.

You can also check the current database encryption settings:

```
db2 => SELECT * FROM TABLE (SYSPROC.ADMIN_GET_ENCRYPTION_INFO())
```

You must be connected to the database to run this command. From the information retrieved, the **ALGORITHM** should not be **3DES**.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>16.11 Leverage Vetted Modules or Services for Application Security Components</u> Leverage vetted modules or services for application security components, such as identity management, encryption, and auditing and logging. Using platform features in critical security functions will reduce developers' workload and minimize the likelihood of design or implementation errors. Modern operating systems provide effective mechanisms for identification, authentication, and authorization and make those mechanisms available to applications. Use only standardized, currently accepted, and extensively reviewed encryption algorithms. Operating systems also provide mechanisms to create and maintain secure audit logs.			
v7	<u>18.5 Use Only Standardized and Extensively Reviewed Encryption Algorithms</u> Use only standardized and extensively reviewed encryption algorithms.			

8.2.3 Secure the Configuration File (Automated)

Profile Applicability:

- Level 2 - RDBMS (extends Level 1 - RDBMS)

Description:

A configuration file, **ekeystore.cfg**, is created by the user in order to configure external keymanager functionality. This file should be secured against tampering via OS permissions.

Rationale:

Set this file to be readable and writeable by only the Db2 instance owner. If this file is not secured, an attacker may delete it, causing potential interruption of operations.

Audit:

A configuration file is created when configuring external keymanager access. You can verify the permissions of the stash file via regular OS operations, such as **ls -la** on a Unix-type system.

Check the OS permissions for the stash file. On a Unix-type system they should be:

```
$ ls -la
-rw----- 1 db2inst1 db2inst1 129 May 19 11:44 ekeystore.cfg
```

Remediation:

Change the permissions for the file:

```
$ chmod 600 ekeystore.cfg
```

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.			

8.2.4 Secure the Stash File (Manual)

Profile Applicability:

- Level 2 - RDBMS (extends Level 1 - RDBMS)

Description:

A stash file is an obfuscated file that contains the credentials that are needed to access the keystore. If a keystore password was not provided during **db2start**, the password will be retrieved from the stash file.

A stash file is created when **-stash** command is used during the creation of the keystore.

Rationale:

Set this file to be readable by only the Db2 instance owner. If this file is not secured, an attacker may delete it, causing potential interruption of operations.

Audit:

A stash file is used when a file with the same name as the PKCS12 keystore file exists but with a **.sth** extension. For example, if the keystore is called **keystore.p12**, the stash file, if it exists, will be in the same directory with the name **keystore.sth**.

You can determine the location of the stash file by examining the **KEYSTORE_LOCATION** database manager configuration parameter.

1. Attach to the Db2 instance.

```
db2 => attach to <db2instance>
```

2. Run the following command:

```
db2 => get database manager configuration
```

3. Locate the **KEYSTORE_LOCATION** value in the output:

```
Keystore location (KEYSTORE_LOCATION) = /path/keystore.p12
```

You can verify the permissions of the stash file via regular OS operations, such as **ls -la** on a Unix-type system.

4. Check the OS permissions for the stash file.

```
$ ls -la
-rw----- 1 db2inst1 db2inst1 129 May 19 11:44 keystore.sth
```

Remediation:

Change the permissions for the file:

```
$ chmod 600 keystore.sth
```

References:

1. <https://www.ibm.com/support/pages/internal-format-gskit-stash-files-has-changed>

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.			

8.2.5 Backup the Stash File (Manual)

Profile Applicability:

- Level 2 - RDBMS (extends Level 1 - RDBMS)

Description:

A stash file is an obfuscated file that contains the credentials that are needed to access the keystore. If a keystore password was not provided during **db2start**, the password will be retrieved from the stash file.

A stash file is created when **-stash** command is used during the creation of the keystore.

Rationale:

Backup the stash file. If access to the stash file is lost, and it can not be re-created because knowledge of the password has been lost, then you will not have access to the keystore file. This may result in the inability to decrypt the database or backup files.

Audit:

Check that the stash file has been backed up according to company policy.

Remediation:

Backup the stash file to a safe location.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	11.2 <u>Perform Automated Backups</u> Perform automated backups of in-scope enterprise assets. Run backups weekly, or more frequently, based on the sensitivity of the data.			
v7	10.2 <u>Perform Complete System Backups</u> Ensure that each of the organization's key systems are backed up as a complete system, through processes such as imaging, to enable the quick recovery of an entire system.			

8.2.6 Create a Strong Password (Manual)

Profile Applicability:

- Level 2 - RDBMS (extends Level 1 - RDBMS)

Description:

When creating or changing passwords for local keystone files, ensure that the passwords are strong, by using the **-strong** parameter of the **gsk8capicmd_64** command.

Rationale:

A stronger password prevents unauthorized access to the database.

Audit:

Password is set during the keystore create command. You cannot check what password you used via a command; you have to manage the the password yourself.

Remediation:

Use the **-strong** parameter on the **gsk8capicmd_64** command:

```
$ gsk8capicmd_64 -keydb -create -dbmykeystore.p12  
-pw <yourpasswordhere> -strong -stash
```

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.			

8.2.7 Backup Your Keystore (Manual)

Profile Applicability:

- Level 2 - RDBMS (extends Level 1 - RDBMS)

Description:

The keystore stores encryption keys used to encrypt your database. Losing the key will make the data inaccessible. If the keystore with encryption keys is lost, there is no way to decrypt the data.

Rationale:

The contents of your keystore are critical and it is important that you back up the keystore at regular intervals. Backups should be done whenever the contents of the keystore changes, such as when a key or certificate is added, a master key (MK) is rotated, or the password is changed.

For local keystore files, the configuration file is not included as part of a Db2 database backup and must be backed up manually.

For a centralized keystore, consult the documentation for your keystore product to understand their recommendations for keystore backups.

Audit:

Check that you are backing up your keystore regularly. Keep in mind that even if the encryption key has been rotated, logs may still need access to the old key. Do not delete keys.

Remediation:

Regularly backup your keystore and stash files, using mechanisms outside of Db2.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	11.2 Perform Automated Backups Perform automated backups of in-scope enterprise assets. Run backups weekly, or more frequently, based on the sensitivity of the data.			
v7	10.2 Perform Complete System Backups Ensure that each of the organization's key systems are backed up as a complete system, through processes such as imaging, to enable the quick recovery of an entire system.			

8.2.8 Backup Your Password In Case Stash File is Inaccessible or Corrupted (Manual)

Profile Applicability:

- Level 2 - RDBMS (extends Level 1 - RDBMS)

Description:

If the keystore is inaccessible, there is no way to decrypt the data in the database. Therefore, protecting and backing up the password to the keystore is important to avoid data loss.

Rationale:

Ensure that you back up your passwords, in addition to using a stash file. This applies particularly to the password used for a local keystore file. Should your stash file ever become corrupted, you will need to manually supply the password. If you forget the password, and do not created a backup, access to your keys and data is lost.

Audit:

Store the password in a safe place, like other similar passwords.

Remediation:

You may use a password manager to store the password in a secure manner.

8.2.9 Rotate the Master Key (Manual)

Profile Applicability:

- Level 2 - RDBMS (extends Level 1 - RDBMS)

Description:

Key rotation refers to the process of changing encryption keys and is often required for compliance purposes. Similar to a password change, key rotation is done to reduce the risk that can come from exposure of the key, while it exists. Since the DEK used by Db2 for encryption is never outside of the encrypted database, backup, or transaction log, there is little risk of exposure. The same is not true for the MK, which lives outside of the database.

The rotation of the MK does not affect the encryption of the DEK within existing backups or archived transaction logs.

The master key (MK) should be rotated based on the frequency needed for compliance. Rotating MK requires decrypting any DEK encrypted with the old MK and then re-encrypting it with the new MK. The data is encrypted with a DEK and does not get re-encrypted.

Rationale:

Consider rotating the MK as being similar to changing passwords every X number of days. You may also have external requirements to rotate the MK after a certain period of time.

Audit:

Check the current database is encryption settings:

```
db2 => SELECT * FROM TABLE(SYSPROC.ADMIN_GET_ENCRYPTION_INFO())
```

You must be connected to the database to run this command. The following information will be shown for the connected database:

- OBJECT_NAME
- OBJECT_TYPE
- ALGORITHM
- ALGORITHM_MODE
- KEY_LENGTH
- MASTER_KEY_LABEL KEYSTORE_NAME KEYSTORE_TYPE
- KEYSTORE_HOST KEYSTORE_IP
- KEYSTORE_IP_TYPE
- PREVIOUS_MASTER_KEY_LABEL AUTH_ID
- APPL_ID
- ROTATION_TIME

Remediation:

The **SYSPROC.ADMIN_ROTATE_MASTER_KEY** procedure can be used to change the database key to comply with key rotation requirement. You must be connected to the database to run this command.

```
db2 => CALL SYSPROC.ADMIN_ROTATE_MASTER_KEY('newMasterKeyLabel')
```

Db2 will automatically generate the new MK and label unless you provide a MK label for an existing MK. Key rotation is logged in the **db2diag.log** file:

```
$ grep -A 3 "Key Rotation" ~/sqlllib/db2dump/db2diag.log
Key Rotation successful using label:
DATA #2 : String, 46 bytes
DB2_SYSGEN_db2inst1_SECRET_2021-04-29-11.22.01
```

The DEK is not externalized and does not need to be rotated. However, if you wish to rotate the DEK you can take an offline backup and restore to a new encrypted database, thus generating a new DEK.

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.			

8.2.10 Turn Off

ALLOW_KEY_INSERT_WITHOUT_KEYSTORE_BACKUP (Manual)

Profile Applicability:

- Level 2 - RDBMS (extends Level 1 - RDBMS)

Description:

The **ALLOW_KEY_INSERT_WITHOUT_KEYSTORE_BACKUP** setting in **ekeystore.cfg** file for external key managers (KMIP or PKCS11) allows writing to the keystore. The setting is **false** by default. Check that the setting has not been turned on.

Rationale:

Keeping the setting as false does not allow Db2 to create master keys. If you choose to turn the setting on, you acknowledge that Db2 does not backup changes to the keystore. Always backup your keystore before making changes.

Audit:

Check that **ekeystore.cfg** file that's being used has the setting turned OFF.

```
ALLOW_KEY_INSERT_WITHOUT_KEYSTORE_BACKUP=false
```

Remediation:

```
$ cat ekeystore.cfg

VERSION=1
PRODUCT_NAME=Luna
ALLOW_KEY_INSERT_WITHOUT_KEYSTORE_BACKUP=false
LIBRARY=/usr/safenet/lunaclient/luna6.1/lib/libCryptoki2_64.so
SLOT_LABEL=DB2Partition
NEW_OBJECT_TYPE=PRIVATE
KEYSTORE_STASH=/home/db2inst1/sqlllib/security/pkcs11.sth
```

For maximum security, turn the setting off and create the keys outside of Db2. This way you will also be able to manage the labeling scheme across the Db2 instances and prevent name collision between different databases.

8.2.11 Keep Master Key Labels Unique (Manual)

Profile Applicability:

- Level 2 - RDBMS (extends Level 1 - RDBMS)

Description:

Db2 uses the MK label to uniquely identify each MK, and stores the label value in each encrypted object, be it a database, transaction log, or backup file. This stored MK label value identifies the MK key material that is used to decrypt the data in the object. Creating MK labels is part of an organizations standard operating procedures (SOP).

Rationale:

It is critical to use unique MK labels across the organization to avoid duplication. If unique labels are not used, access to encrypted data can be lost through human error. Access to encrypted data is lost when the key that is retrieved from the keystore for a label is different from the key that was used to encrypt the database using the same label.

It may not be possible to migrate databases using a local keystore file to a centralized keystore file if multiple databases are using the same master key label, but different master keys associated with that label.

Audit:

Check the organization's SOP for creating master key labels to ensure it includes specific guidance on creating unique master key labels.

Remediation:

Update the SOP to provide guidance found missing during the audit.

8.2.12 Retain All Master Keys (Manual)

Profile Applicability:

- Level 2 - RDBMS (extends Level 1 - RDBMS)

Description:

Master keys are needed to access the DEKs that are stored in encrypted databases, transaction logs, and backup images. Since multiple MKs can exist over the life time of these objects, it is necessary to retain them all while the encrypted data is retained. If a MK is lost, the encrypted data in the object cannot be retrieved. An organizations standard operating procedures (SOP) should have a methodology for tracking what data is encrypted with a specific master key, and be consulted before any master keys are archived.

Rationale:

Any failure or human error in tracking what data encrypted with a specific master key could result in loss of access to that data should the master key be deleted. Master keys should not be deleted. Instead they should be archived to a secure location for long term storage in case they are required at some point in the future.

Audit:

Check the organization's SOP for specific instructions on tracking what data is encrypted with a particular master key, and that there are details on how to archive a master key for long term storage. Ensure the SOP clearly states master keys should never be deleted.

Remediation:

Update the SOP to provide guidance found missing during the audit.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>11.2 Perform Automated Backups</u> Perform automated backups of in-scope enterprise assets. Run backups weekly, or more frequently, based on the sensitivity of the data.			
v7	<u>10.2 Perform Complete System Backups</u> Ensure that each of the organization's key systems are backed up as a complete system, through processes such as imaging, to enable the quick recovery of an entire system.			

8.2.13 Set CFG Values in a Single Command (Manual)

Profile Applicability:

- Level 2 - RDBMS (extends Level 1 - RDBMS)

Description:

Db2 uses **KEYSTORE_TYPE** and **KEYSTORE_LOCATION** to access the keystore.

Rationale:

Although the Db2 database manager configuration parameters **KEYSTORE_TYPE** and **KEYSTORE_LOCATION** are configurable online, you should set them in the a single **DB2 UPDATE DBM CFG** command. Otherwise, Db2 might attempt to access the keystore between the updates and report an access error.

Audit:

You can check what values you have set for these parameters as follows:

```
$ db2 get dbm cfg | grep KEYSTORE
Keystore type (KEYSTORE_TYPE) = PKCS12
Keystore location (KEYSTORE_LOCATION) = /path/to/file.p12
```

Remediation:

You can execute the following command to change the values in one line:

```
db2 => update dbm cfg using keystore_type pkcs12
      keystore_location /path/to/file.p12;

DB20000I The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.
```

8.2.14 Key Rotation in HADR Environment (Manual)

Profile Applicability:

- Level 2 - RDBMS (extends Level 1 - RDBMS)

Description:

Some additional considerations are needed when working in an HADR environment.

A key rotation on the primary database in an HADR environment drives a key rotation automatically on the standby. The change, however, does not occur until other log records are sent to the standby database. If you want to force the rotated key to the standby, the archive log command can be used to generate the log records that are needed to replay the rotation on the standby.

Rationale:

When the MK is rotated, the database begins to use the new key immediately, but access to the old MK value is still needed in the following scenarios:

- Transaction log files that have not been reused since the key rotation
- Archived encrypted transaction log files that used the previous MK value
- Encrypted backup images that used the previous MK value

Do not delete an MK from the keystore unless you are certain it is no longer referenced by any encrypted object on any of the hosts of the HADR environment.

When using a local keystore file, you need to provide an identical copy of the keystore at each Db2 member that is associated with the database. If you choose to use a shared file system, ensure that network access is maintained for that file system while Db2 is actively working with the encrypted database. In an HADR setup, you should copy the keystore containing the encryption keys between the hosts.

Audit:

You can check what values you have set for these parameters as follows on each HADR member:

```
db2 => get dbm cfg | grep KEYSTORE
Keystore type (KEYSTORE_TYPE) = PKCS12
Keystore location (KEYSTORE_LOCATION) = /path/to/file.p12
```

We recommend you use a centralized key manager for HADR environments.

Remediation:

Retain all Master Keys until you are certain they are not use on any of the hosts in an HADR environment.

The same DEK and master key is used on all members in a DPF or pureScale deployment and each member requires access to the keystore to access master keys, which can be a shared file.

9 Additional Considerations

This section provides additional considerations on protecting and maintaining the database instance.

9.1 Leverage the Least Privilege Principle (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

The Db2 database instance will execute under the context of a given security principle. It is recommended that this service have the least privileges possible. Furthermore, it is advisable to have the Db2 service executed using the Db2 instance owner and monitor such accounts for unauthorized access to the sensitive data.

Rationale:

Leveraging a least privilege account for the Db2 service will reduce an attacker's ability to compromise the host operating system should the Db2 service process become compromised.

Audit:

Review all accounts that have access to the Db2 database service to ensure least privilege is applied.

Remediation:

Ensure that all accounts have the absolute minimal privilege granted to perform their tasks.

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.			

9.2 Enable Backup Redundancy (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

Backup redundancy ensures that multiple instances of database backups exist.

Rationale:

Maintaining redundant copies of database backups will increase business continuity capabilities should a Db2 service failure coincide with a corrupt backup.

Audit:

Review the replication of your backups based on organization policy.

Remediation:

Define and implement a process to replicate your backups onto multiple locations.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	11.4 <u>Establish and Maintain an Isolated Instance of Recovery Data</u> Establish and maintain an isolated instance of recovery data. Example implementations include, version controlling backup destinations through offline, cloud, or off-site systems or services.			

9.3 Protecting Backups (Manual)

Profile Applicability:

- Level 1 - RDBMS

Description:

Backups of your database should be stored securely in a location with full access for administrators, read and execute access for group, and no access for users.

Rationale:

Backups may contain sensitive data that attackers can use to retrieve valuable information about the organization.

Audit:

Review the privileges applied to your backups.

Remediation:

Define a security policy for all backups that specifies the privileges they should be assigned.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	11.3 <u>Protect Recovery Data</u> Protect recovery data with equivalent controls to the original data. Reference encryption or data separation, based on requirements.			
v7	10.4 <u>Ensure Protection of Backups</u> Ensure that backups are properly protected via physical security or encryption when they are stored, as well as when they are moved across the network. This includes remote backups and cloud services.			

Appendix: Summary Table

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
1	Staying Current		
1.1	General Considerations		
1.1.1	Install Available Updates (Manual)	☐	☐
2	Securing the Server Environment		
2.1	Prevent Database Users from Logging into the Operating System (Manual)	☐	☐
3	Securing the Server Instance		
3.1	Database Manager Configuration Parameters		
3.1.1	Require Explicit Authorization for Cataloging (CATALOG_NOAUTH) (Automated)	☐	☐
3.1.2	Secure Permissions for Default Database File Path (DFTDBPATH) (Automated)	☐	☐
3.1.3	Set Diagnostic Logging to Capture Errors and Warnings (DIAGLEVEL) (Automated)	☐	☐
3.1.4	Secure Permissions for All Diagnostic Logs (DIAGPATH) (Automated)	☐	☐
3.1.5	Secure Permissions for Alternate Diagnostic Log Path (ALT_DIAGPATH) (Automated)	☐	☐
3.1.6	Set Maximum Connection Limits (MAX_CONNECTIONS and MAX_COORDAGENTS) (Automated)	☐	☐
3.1.7	Set Administrative Notification Level (NOTIFYLEVEL) (Automated)	☐	☐
3.1.8	Secure the Java Development Kit Installation Path (JDK_PATH) (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
3.1.9	Secure the Python Runtime Path (PYTHON_PATH) (Automated)	☐	☐
3.1.10	Secure the R Runtime Path (R_PATH) (Automated)	☐	☐
3.1.11	Secure the Communication Buffer Exit Library (COMM_EXIT_LIST) (Automated)	☐	☐
3.2	Db2 Registry Values		
3.2.1	Specify Secure Remote Shell Command (DB2RSHCMD) (Automated)	☐	☐
3.2.2	Turn Off Remote Command Legacy Mode (DB2RCMD_LEGACY_MODE) (Automated)	☐	☐
3.2.3	Disable Grants During Restore (DB2_RESTORE_GRANT_ADMIN_AUTHORITIES) (Automated)	☐	☐
3.2.4	Enable Extended Security (DB2_EXTSECURITY) (Automated)	☐	☐
3.2.5	Limit OS Privileges of Fenced Mode Process (DB2_LIMIT_FENCED_GROUP) (Automated)	☐	☐
3.3	General Considerations		
3.3.1	Secure Db2 Runtime Library (Manual)	☐	☐
3.3.2	Secure the Database Container Directory (Manual)	☐	☐
3.3.3	Set umask Value in the Db2 Instance Owner's .profile (Automated)	☐	☐
4	Securing the Database		
4.1	Database Configuration Parameters		
4.1.1	Creating the Database Without PUBLIC Grants (RESTRICTIVE) (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
4.1.2	Set Failed Archive Retry Delay (ARCHRETRYDELAY) (Automated)	☐	☐
4.1.3	Auto-restart After Abnormal Termination (AUTORESTART) (Automated)	☐	☐
4.1.4	Secure Permissions for the Primary Archive Log Location (LOGARCHMETH1) (Automated)	☐	☐
4.1.5	Secure Permissions for the Secondary Archive Log Location (LOGARCHMETH2) (Automated)	☐	☐
4.1.6	Secure Permissions for the Tertiary Archive Log Location (FAILARCHPATH) (Automated)	☐	☐
4.1.7	Secure Permissions for the Log Mirror Location (MIRRORLOGPATH) (Automated)	☐	☐
4.1.8	Secure Permissions for the Log Overflow Location (OVERFLOWLOGPATH) (Manual)	☐	☐
4.1.9	Establish Retention Set Size for Backups (NUM_DB_BACKUPS) (Manual)	☐	☐
4.1.10	Set Archive Log Failover Retry Limit (NUMARCHRETRY) (Automated)	☐	☐
4.1.11	Set Maximum Number of Applications (MAXAPPLS) (Automated)	☐	☐
4.1.12	Ensure a Secure Connect Procedure is Used (CONNECT_PROC) (Manual)	☐	☐
4.1.13	Specify a Secure Location for External Tables (EXTBL_LOCATION) (Manual)	☐	☐
4.2	Secure the Database Catalog Views		
4.2.1	Restrict Access to SYSCAT.AUDITPOLICIES (Automated)	☐	☐
4.2.2	Restrict Access to SYSCAT.AUDITUSE (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
4.2.3	Restrict Access to SYSCAT.COLAUTH (Automated)	☐	☐
4.2.4	Restrict Access to SYSCAT.COLDIST (Automated)	☐	☐
4.2.5	Restrict Access to SYSCAT.COLGROUPDIST (Automated)	☐	☐
4.2.6	Restrict Access to SYSCAT.COLUMNS (Automated)	☐	☐
4.2.7	Restrict Access to SYSCAT.CONTEXTATTRIBUTES (Automated)	☐	☐
4.2.8	Restrict Access to SYSCAT.CONTEXTS (Automated)	☐	☐
4.2.9	Restrict Access to SYSCAT.CONTROLDEP (Automated)	☐	☐
4.2.10	Restrict Access to SYSCAT.CONTROLS (Automated)	☐	☐
4.2.11	Restrict Access to SYSCAT.DBAUTH (Automated)	☐	☐
4.2.12	Restrict Access to SYSCAT.EVENTS (Automated)	☐	☐
4.2.13	Restrict Access to SYSCAT.EVENTTABLES (Automated)	☐	☐
4.2.14	Restrict Access to SYSCAT.EXTERNALTABLEOPTIONS (Automated)	☐	☐
4.2.15	Restrict Access to SYSCAT.INDEXAUTH (Automated)	☐	☐
4.2.16	Restrict Access to SYSCAT.MODULEAUTH (Automated)	☐	☐
4.2.17	Restrict Access to SYSCAT.PACKAGEAUTH (Automated)	☐	☐
4.2.18	Restrict Access to SYSCAT.PACKAGES (Automated)	☐	☐
4.2.19	Restrict Access to SYSCAT.PASSTHRUAUTH (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
4.2.20	Restrict Access to SYSCAT.ROLEAUTH (Automated)	☐	☐
4.2.21	Restrict Access to SYSCAT.ROLES (Automated)	☐	☐
4.2.22	Restrict Access to SYSCAT.ROUTINEAUTH (Automated)	☐	☐
4.2.23	Restrict Access to SYSCAT.ROUTINES (Automated)	☐	☐
4.2.24	Restrict Access to SYSCAT.SECURITYLABELACCESS (Automated)	☐	☐
4.2.25	Restrict Access to SYSCAT.SECURITYLABELCOMPONENTELEMENTS (Automated)	☐	☐
4.2.26	Restrict Access to SYSCAT.SECURITYLABELCOMPONENTS (Automated)	☐	☐
4.2.27	Restrict Access to SYSCAT.SECURITYLABELS (Automated)	☐	☐
4.2.28	Restrict Access to SYSCAT.SECURITYPOLICIES (Automated)	☐	☐
4.2.29	Restrict Access to SYSCAT.SECURITYPOLICYCOMPONENTRULES (Automated)	☐	☐
4.2.30	Restrict Access to SYSCAT.SECURITYPOLICYEXEMPTIONS (Automated)	☐	☐
4.2.31	Restrict Access to SYSCAT.SERVEROPTIONS (Automated)	☐	☐
4.2.32	Restrict Access to SYSCAT.SCHEMAAUTH (Automated)	☐	☐
4.2.33	Restrict Access to SYSCAT.SCHEMATA (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
4.2.34	Restrict Access to SYSCAT.SEQUENCEAUTH (Automated)	☐	☐
4.2.35	Restrict Access to SYSCAT.STATEMENTS (Automated)	☐	☐
4.2.36	Restrict Access to SYSCAT.STATEMENTTEXTS (Automated)	☐	☐
4.2.37	Restrict Access to SYSCAT.SURROGATEAUTHIDS (Automated)	☐	☐
4.2.38	Restrict Access to SYSCAT.TABAUTH (Automated)	☐	☐
4.2.39	Restrict Access to SYSCAT.TBSPACEAUTH (Automated)	☐	☐
4.2.40	Restrict Access to SYSCAT.USEROPTIONS (Automated)	☐	☐
4.2.41	Restrict Access to SYSCAT.VARIABLEAUTH (Automated)	☐	☐
4.2.42	Restrict Access to SYSCAT.VARIABLES (Automated)	☐	☐
4.2.43	Restrict Access to SYSCAT.WORKLOADAUTH (Automated)	☐	☐
4.2.44	Restrict Access to SYSCAT.WRAPOPTIONS (Automated)	☐	☐
4.2.45	Restrict Access to SYSCAT.XSROBJECTAUTH (Automated)	☐	☐
4.2.46	Restrict Access to SYSSTAT.COLDIST (Automated)	☐	☐
4.2.47	Restrict Access to SYSSTAT.COLGROUPDIST (Automated)	☐	☐
4.2.48	Restrict Access to SYSSTAT.COLUMNS (Automated)	☐	☐
4.2.49	Restrict Access to SYSCAT.AUDITEXCEPTIONS (Manual)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
4.2.50	Restrict Access to SYSCAT.BUFFERPOOLAUTH (Automated)	☐	☐
4.3	Secure the Database Catalog Tables		
4.3.1	Restrict Access to SYSIBM.SYSAUDITPOLICIES (Automated)	☐	☐
4.3.2	Restrict Access to SYSIBM.SYSAUDITUSE (Automated)	☐	☐
4.3.3	Restrict Access to SYSIBM.SYSCOLAUTH (Automated)	☐	☐
4.3.4	Restrict Access to SYSIBM.SYSCOLDIST (Automated)	☐	☐
4.3.5	Restrict Access to SYSIBM.SYSCOLGROUPDIST (Automated)	☐	☐
4.3.6	Restrict Access to SYSIBM.SYSCOLUMNS (Automated)	☐	☐
4.3.7	Restrict Access to SYSIBM.SYSCONTEXTATTRIBUTES (Automated)	☐	☐
4.3.8	Restrict Access to SYSIBM.SYSCONTEXTS (Automated)	☐	☐
4.3.9	Restrict Access to SYSIBM.SYSDEPENDENCIES (Automated)	☐	☐
4.3.10	Restrict Access to SYSIBM.SYSCONTROLS (Automated)	☐	☐
4.3.11	Restrict Access to SYSIBM.SYSDBAUTH (Automated)	☐	☐
4.3.12	Restrict Access to SYSIBM.SYSEVENTS (Automated)	☐	☐
4.3.13	Restrict Access to SYSIBM.SYSEVENTTABLES (Automated)	☐	☐
4.3.14	Restrict Access to SYSIBM.SYSEXTTAB (Automated)	☐	☐
4.3.15	Restrict Access to SYSIBM.SYSINDEXAUTH (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
4.3.16	Restrict Access to SYSIBM.SYSMODULEAUTH (Automated)	☐	☐
4.3.17	Restrict Access to SYSIBM.SYSPASSTHRUAUTH (Automated)	☐	☐
4.3.18	Restrict Access to SYSIBM.SYSPLANAUTH (Automated)	☐	☐
4.3.19	Restrict Access to SYSIBM.SYSPLAN (Automated)	☐	☐
4.3.20	Restrict Access to SYSIBM.SYSROLEAUTH (Automated)	☐	☐
4.3.21	Restrict Access to SYSIBM.SYSROLES (Automated)	☐	☐
4.3.22	Restrict Access to SYSIBM.SYSROUTINEAUTH (Automated)	☐	☐
4.3.23	Restrict Access to SYSIBM.SYSROUTINES (Automated)	☐	☐
4.3.24	Restrict Access to SYSIBM.ROUTINES_S (Automated)	☐	☐
4.3.25	Restrict Access to SYSIBM.SYSSCHEMAAUTH (Automated)	☐	☐
4.3.26	Restrict Access to SYSIBM.SYSSCHEMATA (Automated)	☐	☐
4.3.27	Restrict Access to SYSIBM.SYSSECURITYLABELACCESS (Automated)	☐	☐
4.3.28	Restrict Access to SYSIBM.SYSSECURITYLABELCOMPONENTELEMEN TS (Automated)	☐	☐
4.3.29	Restrict Access to SYSIBM.SYSSECURITYLABELCOMPONENTS (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
4.3.30	Restrict Access to SYSIBM.SYSSECURITYLABELS (Automated)	☐	☐
4.3.31	Restrict Access to SYSIBM.SYSSECURITYPOLICIES (Automated)	☐	☐
4.3.32	Restrict Access to SYSIBM.SYSSECURITYPOLICYCOMPONENTRULES (Automated)	☐	☐
4.3.33	Restrict Access to SYSIBM.SYSSECURITYPOLICYEXEMPTIONS (Automated)	☐	☐
4.3.34	Restrict Access to SYSIBM.SYSSERVEROPTIONS (Automated)	☐	☐
4.3.35	Restrict Access to SYSIBM.SYSSEQUENCEAUTH (Automated)	☐	☐
4.3.36	Restrict Access to SYSIBM.SYSSTATEMENTTEXTS (Automated)	☐	☐
4.3.37	Restrict Access to SYSIBM.SYSSTMT (Automated)	☐	☐
4.3.38	Restrict Access to SYSIBM.SYSSURROGATEAUTHIDS (Automated)	☐	☐
4.3.39	Restrict Access to SYSIBM.SYSTABAUTH (Automated)	☐	☐
4.3.40	Restrict Access to SYSIBM.SYSTBSPACEAUTH (Automated)	☐	☐
4.3.41	Restrict Access to SYSIBM.SYSUSEROPTIONS (Automated)	☐	☐
4.3.42	Restrict Access to SYSIBM.SYSVARIABLEAUTH (Automated)	☐	☐
4.3.43	Restrict Access to SYSIBM.SYSVARIABLES (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
4.3.44	Restrict Access to SYSIBM.SYSWORKLOADAUTH (Automated)	☐	☐
4.3.45	Restrict Access to SYSIBM.SYSWRAPOPTIONS (Automated)	☐	☐
4.3.46	Restrict Access to SYSIBM.SYSXSROBJECTAUTH (Automated)	☐	☐
4.3.47	Restrict Access to SYSIBM.SYSAUDITEXCEPTIONS (Automated)	☐	☐
4.3.48	Restrict Access to SYSIBM.SYSBUFFERPOOLAUTH (Automated)	☐	☐
4.4	Secure the Database Administrative Views and Routines		
4.4.1	Restrict Access to SYSIBMADM.AUTHORIZATIONIDS (Automated)	☐	☐
4.4.2	Restrict Access to SYSIBMADM.OBJECTOWNERS (Automated)	☐	☐
4.4.3	Restrict Access to SYSIBMADM.PRIVILEGES (Automated)	☐	☐
4.4.4	Restrict Access to SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID (Automated)	☐	☐
4.4.5	Restrict Access to SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID (Automated)	☐	☐
4.4.6	Restrict Access to SYSPROC.AUTH_LIST_GROUPS_FOR_AUTHID (Automated)	☐	☐
4.5	General Database Considerations		
4.5.1	Restrict Access to Tablespaces (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
4.5.2	Remove Unused Schemas (Automated)	☐	☐
4.5.3	Review System Tablespaces (Automated)	☐	☐
5	Authentication Considerations		
5.1	Specify a Secure Connection Authentication Type (SRVCON_AUTH) (Manual)	☐	☐
5.2	Specify a Secure Authentication Type (AUTHENTICATION) (Manual)	☐	☐
5.3	Database Manager Configuration Parameter: ALTERNATE_AUTH_ENC (Manual)	☐	☐
5.4	Database Manager Configuration Parameter: TRUST_ALLCLNTS (Manual)	☐	☐
5.5	Database Manager Configuration Parameter: TRUST_CLNTAUTH (Automated)	☐	☐
5.6	Secure Permissions for All Authentication Plugins (Manual)	☐	☐
5.7	DB2_GRP_LOOKUP Registry Variable (Windows only) (Manual)	☐	☐
5.8	DB2DOMAINLIST Registry Variable (Windows only) (Manual)	☐	☐
5.9	DB2AUTH Registry Variable (Manual)	☐	☐
5.10	DB2CHGPWD_EEE Registry Variable (Manual)	☐	☐
6	Authorization Considerations		
6.1	Secure Database Authorities		
6.1.1	Secure SYSADM Authority (Manual)	☐	☐
6.1.2	Secure SYSCTRL Authority (Manual)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
6.1.3	Secure SYSMAINT Authority (Manual)	☐	☐
6.1.4	Secure SYSMON Authority (Manual)	☐	☐
6.1.5	Secure SECADM Authority (Manual)	☐	☐
6.1.6	Secure DBADM Authority (Manual)	☐	☐
6.1.7	Secure SQLADM Authority (Manual)	☐	☐
6.1.8	Secure DATAACCESS Authority (Manual)	☐	☐
6.1.9	Secure ACCESSCTRL Authority (Manual)	☐	☐
6.1.10	Secure WLMADM Authority (Manual)	☐	☐
6.1.11	Secure CREATAB Authority (Manual)	☐	☐
6.1.12	Secure BINDADD Authority (Manual)	☐	☐
6.1.13	Secure CONNECT Authority (Manual)	☐	☐
6.1.14	Secure LOAD Authority (Manual)	☐	☐
6.1.15	Secure EXTERNALROUTINE Authority (Manual)	☐	☐
6.1.16	Secure QUIESCECONNECT Authority (Manual)	☐	☐
6.1.17	Secure SETSESSIONUSER Privilege (Manual)	☐	☐
6.1.18	Secure SCHEMAADM Authority (Manual)	☐	☐
6.1.19	Secure Schema ACCESSCTRL Authority (Manual)	☐	☐
6.1.20	Secure Schema DATAACCESS Authority (Manual)	☐	☐
6.2	General Authorization		
6.2.1	Review Users, Groups, and Roles (Manual)	☐	☐
6.2.2	Review Roles (Manual)	☐	☐
6.2.3	Review Role Members (Manual)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
6.2.4	Nested Roles (Manual)	☐	☐
6.2.5	Review Roles Granted to PUBLIC (Manual)	☐	☐
6.2.6	Review Role Grantees with WITH ADMIN OPTION (Manual)	☐	☐
6.3	Row and Column Access Control		
6.3.1	Review Organization's Policies Against Db2 RCAC Policies (Manual)	☐	☐
6.3.2	Review Row Permission Logic According to Policy (Manual)	☐	☐
6.3.3	Review Column Mask Logic According to Policy (Manual)	☐	☐
6.4	Trusted context Considerations		
6.4.1	Ensure Trusted Contexts are Enabled (Manual)	☐	☐
6.4.2	Do Not Allow Trusted Context to Switch Users Without Authentication (Manual)	☐	☐
7	Audit Considerations		
7.1	General Audit Considerations		
7.1.1	Disable the Audit Buffer (Automated)	☐	☐
7.1.2	Disable Limited Audit of Applications (DB2_LIMIT_AUDIT_APPS) (Manual)	☐	☐
7.1.3	Ensure Audit Policies are Enabled Within the Database (Manual)	☐	☐
7.1.4	Ensure Audit is Enabled Within the Instance (Automated)	☐	☐
8	Encryption Considerations		
8.1	Encryption of Data in Motion		

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
8.1.1	Configure a Server-side Key Store for TLS (SSL_SVR_KEYDB) (Manual)	☐	☐
8.1.2	Configure a Server-side Stash File for TLS (SSL_SVR_STASH) (Manual)	☐	☐
8.1.3	Configure an Endpoint Certificate (SSL_SVR_LABEL) (Manual)	☐	☐
8.1.4	Configure the Service Name for TLS (SSL_SVCENAME) (Manual)	☐	☐
8.1.5	Configure a Secure TLS Version (SSL_VERSIONS) (Manual)	☐	☐
8.1.6	Configure Secure TLS Cipher Suites (SSL_CIPHERSPECS) (Manual)	☐	☐
8.1.7	Unset the Service Name for Plaintext Communication (SVCENAME) (Manual)	☐	☐
8.1.8	Configure a Client-side Key Store for TLS (SSL_CLNT_KEYDB) (Manual)	☐	☐
8.1.9	Configure a Client-side Stash File for TLS (SSL_CLNT_STASH) (Manual)	☐	☐
8.1.10	Enable TLS Communication Between HADR Primary and Standby Instances (HADR_SSL_LABEL) (Manual)	☐	☐
8.1.11	Enable Remote TLS Connections to Db2 (DB2COMM) (Manual)	☐	☐
8.2	Encryption of Data at Rest		
8.2.1	Encrypt the Database (Manual)	☐	☐
8.2.2	Do Not Use Encryption Algorithms that are Not Secure (Manual)	☐	☐
8.2.3	Secure the Configuration File (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
8.2.4	Secure the Stash File (Manual)	☐	☐
8.2.5	Backup the Stash File (Manual)	☐	☐
8.2.6	Create a Strong Password (Manual)	☐	☐
8.2.7	Backup Your Keystore (Manual)	☐	☐
8.2.8	Backup Your Password In Case Stash File is Inaccessible or Corrupted (Manual)	☐	☐
8.2.9	Rotate the Master Key (Manual)	☐	☐
8.2.10	Turn Off ALLOW_KEY_INSERT_WITHOUT_KEYSTORE_BACK UP (Manual)	☐	☐
8.2.11	Keep Master Key Labels Unique (Manual)	☐	☐
8.2.12	Retain All Master Keys (Manual)	☐	☐
8.2.13	Set CFG Values in a Single Command (Manual)	☐	☐
8.2.14	Key Rotation in HADR Environment (Manual)	☐	☐
9	Additional Considerations		
9.1	Leverage the Least Privilege Principle (Manual)	☐	☐
9.2	Enable Backup Redundancy (Manual)	☐	☐
9.3	Protecting Backups (Manual)	☐	☐

Appendix: Change History

Date	Version	Changes for this version
27 Mar 2026	1.0.0	Initial Release