CENTER FOR INTERNET SECURITY

CIS IBM DB2 10 Benchmark

v1.0.0 - 11-20-2015

The CIS Security Benchmarks division provides consensus-oriented information security products, services, tools, metrics, suggestions, and recommendations (the "SB Products") as a public service to Internet users worldwide. Downloading or using SB Products in any way signifies and confirms your acceptance of and your binding agreement to these CIS Security Benchmarks Terms of Use.

CIS SECURITY BENCHMARKS TERMS OF USE

BOTH CIS SECURITY BENCHMARKS DIVISION MEMBERS AND NON-MEMBERS MAY:

- Download, install, and use each of the SB Products on a single computer, and/or
- Print one or more copies of any SB Product that is in a .txt, .pdf, .doc, .mcw, or .rtf format, but only if each such copy is printed in its entirety and is kept intact, including without limitation the text of these CIS Security Benchmarks Terms of Use.

UNDER THE FOLLOWING TERMS AND CONDITIONS:

- **SB Products Provided As Is**. CIS is providing the SB Products "as is" and "as available" without: (1) any representations, warranties, or covenants of any kind whatsoever (including the absence of any warranty regarding: (a) the effect or lack of effect of any SB Product on the operation or the security of any network, system, software, hardware, or any component of any of them, and (b) the accuracy, utility, reliability, timeliness, or completeness of any SB Product); or (2) the responsibility to make or notify you of any corrections, updates, upgrades, or fixes.
- **Intellectual Property and Rights Reserved**. You are not acquiring any title or ownership rights in or to any SB Product, and full title and all ownership rights to the SB Products remain the exclusive property of CIS. All rights to the SB Products not expressly granted in these Terms of Use are hereby reserved.
- **Restrictions**. You acknowledge and agree that you may not: (1) decompile, dis-assemble, alter, reverse engineer, or otherwise attempt to derive the source code for any software SB Product that is not already in the form of source code; (2) distribute, redistribute, sell, rent, lease, sublicense or otherwise transfer or exploit any rights to any SB Product in any way or for any purpose; (3) post any SB Product on any website, bulletin board, ftp server, newsgroup, or other similar mechanism or device; (4) remove from or alter these CIS Security Benchmarks Terms of Use on any SB Product; (5) remove or alter any proprietary notices on any SB Product; (6) use any SB Product or any component of an SB Product with any derivative works based directly on an SB Product or any component of an SB Product; (7) use any SB Product or any component of an SB Product with other products or applications that are directly and specifically dependent on such SB Product or any component for any part of their functionality; (8) represent or claim a particular level of compliance or consistency with any SB Product; or (9) facilitate or otherwise aid other individuals or entities in violating these CIS Security Benchmarks Terms of Use.
- **Your Responsibility to Evaluate Risks**. You acknowledge and agree that: (1) no network, system, device, hardware, software, or component can be made fully secure; (2) you have the sole responsibility to evaluate the risks and benefits of the SB Products to your particular circumstances and requirements; and (3) CIS is not assuming any of the liabilities associated with your use of any or all of the SB Products.
- **CIS Liability**. You acknowledge and agree that neither CIS nor any of its employees, officers, directors, agents or other service providers has or will have any liability to you whatsoever (whether based in contract, tort, strict liability or otherwise) for any direct, indirect, incidental, consequential, or special damages that arise out of or are connected in any way with your use of any SB Product.
- **Indemnification**. You agree to indemnify, defend, and hold CIS and all of CIS's employees, officers, directors, agents and other service providers harmless from and against any liabilities, costs and expenses incurred by any of them in connection with your violation of these CIS Security Benchmarks Terms of Use.
- **Jurisdiction**. You acknowledge and agree that: (1) these CIS Security Benchmarks Terms of Use will be governed by and construed in accordance with the laws of the State of Maryland; (2) any action at law or in equity arising out of or relating to these CIS Security Benchmarks Terms of Use shall be filed only in the courts located in the State of Maryland; and (3) you hereby consent and submit to the personal jurisdiction of such courts for the purposes of litigating any such action.
- **U.S. Export Control and Sanctions laws**. Regarding your use of the SB Products with any non-U.S. entity or country, you acknowledge that it is your responsibility to understand and abide by all U.S. sanctions and export control laws as set from time to time by the U.S. Bureau of Industry and Security (BIS) and the U.S. Office of Foreign Assets Control (OFAC).

SPECIAL RULES FOR CIS MEMBER ORGANIZATIONS: CIS reserves the right to create special rules for: (1) CIS Members; and (2) Non-Member organizations and individuals with which CIS has a written contractual relationship. CIS hereby grants to each CIS Member Organization in good standing the right to distribute the SB Products within such Member's own organization, whether by manual or electronic means. Each such Member Organization acknowledges and agrees that the foregoing grants in this paragraph are subject to the terms of such Member's membership arrangement with CIS and may, therefore, be modified or terminated by CIS at any time.

Table of Contents

Overview

This document, Security Configuration Benchmark for DB2, provides prescriptive guidance for establishing a secure configuration posture for DB2 versions 10.x running on Linux, UNIX, and Windows. This guide was tested against DB2 version 10.5 installed on Windows Server 2008 R2 and CentOS 6. To obtain the latest version of this guide, please visit http://cisecurity.org. If you have questions, comments, or have identified ways to improve this guide, please write us at feedback@cisecurity.org.

Intended Audience

This 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 Linux, UNIX, and Windows platforms.

Consensus Guidance

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

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

Typographical Conventions

The following typographical conventions are used throughout this guide:

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

Scoring Information

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

Scored

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

Not Scored

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

Profile Definitions

The following configuration profiles are defined by this Benchmark:

- **Level 1 - RDBMS**

  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**

  This profile extends the "Level 1" 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

- **Level 1 - Windows Host OS**

  Items in this profile apply to the Windows Host OS 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 - Windows Host OS**

  This profile extends "Level 1 - Windows Host OS". 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

- **Level 1 - Linux Host OS**

  Items in this profile apply to the Linux Host OS 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 - Linux Host OS**

  This profile extends "Level 1 - Linux Host OS". 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:

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:

Authors

Contributors and Reviews

Recommendations

1 Installation and Patches

[This space intentionally left blank]

1.1 Install the latest Fixpaks (Not Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

Periodically, IBM releases "Fixpaks" to enhance features and resolve defects, including security defects. It is recommended that the DB2 instance remain current with all fix packs.

Rationale:

Installing the latest DB2 fixpak will help protect the database from known vulnerabilities as well as reducing downtime that may otherwise result from functional defects.

Audit:

Perform the following DB2 commands to obtain the version:

1. Open the DB2 Command Window and type in `db2level`:

```
$ db2level
DB21085I Instance "DB2" uses "32" bits and DB2 code release "SQL09050" with
level identifier "03010107".
Informational tokens are "DB2 v9.5.0.808", "s071001", "NT3295", and Fix Pack
"3".
```

Remediation:

Apply the latest fixpak as offered from IBM.

1.2 Use IP address rather than hostname (Scored)

Profile Applicability:

- Level 1 - Windows Host OS

- Level 1 - Linux Host OS

Description:

Use an IP address rather than a hostname to connect to the host of the DB2 instance.

Rationale:

Using a hostname to connect to a DB2 instance can display useful information about the host to a attacker. For example, do not include version number, type of host, or the type of operating system in the hostname.

Audit:

Windows:

1. Run DB2 Command Prompt - Administrator
2. type 'db2 list node directory show detail'
3. Verify that the 'HOSTNAME' values for all nodes listed are in IP address form and not hostnames

Unix:

1. Log into DB2 as DB2 Instance owner
2. type 'db2 list node directory show detail'
3. Verify that the 'HOSTNAME' values for all nodes listed are in IP address form and not hostnames

Sample:

```
Node Directory

 Number of entries in the directory = 2

Node 1 entry:

 Node name = SAMPLE
 Comment =
 Directory entry type = LDAP
 Protocol = TCPIP
 Hostname = 192.168.145.10
 Service name = 50000
```

Remediation:

1. Drop all existing nodes
2. Recreate node directory using IP addresses and not hostnames

1.3 Leverage a least privilege principle (Not Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The DB2 database instance will execute under the context of a given security principle. It is recommended that the DB2 service execute under a least privilege security principle. Furthermore, it is advisable to have the DB2 service executed under using the or Administrator account and monitor such accounts from 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 segregation of duties and least privilege is applied.

Remediation:

Ensure that all accounts have the absolute minimal privilege granted to perform their tasks.

1.4 Use non-standard account names (Not Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 1 - Linux Host OS

Description:

The DB2 service is installed with default, well-known accounts such as `db2admin`, `db2inst1`, `dasusr1`, or `db2fenc1`. It is recommended that the use of these accounts be avoided.

Rationale:

The use of default accounts may increase the DB2 service's susceptibility to unauthorized access as an attacker.

Audit:

1. For MS Windows: Right-click over the `%DB2PATH%` and select *Properties* from the menu. Go to the *Security* tab and review all group and user names that have access to this directory.

   For Unix: Run `ls -al $DB2PATH` and review all group and user names that have access to this directory.

Remediation:

1. For MS Windows: Right-click over the `%DB2PATH%` and select *Properties* from the menu. Go to the *Security* tab and re-assign all the groups or user names with a not well-known account.
2. For Unix:

```
chown -R <new user name>:<new group name> $DB2PATH
```


2 DB2 Directory and File Permissions

This section provides guidance on securing all operating system specific objects for DB2.

2.1 Secure DB2 Runtime Library (Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 1 - Linux Host OS

Description:

A DB2 software installation will place all executables under the default `<DB2PATH>\sqllib` directory. This directory should grant access to DB2 administrator only. All other users should only have `read` privilege.

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 DB2 commands to obtain the value for this setting:
For MS Windows:

1. Connect to the DB2 host
2. 2. Right-click on the NODE000x/sqldbdir directory
3. 3. Choose *Properties*
4. 4. Select the *Security* tab
5. 5. Review access from all non-administrator accounts

For Unix:

1. Connect to the DB2 host
2. Change to the `NODE000x/sqldbdir` directory
3. Determine the permission level of the directory

```
OS => ls -al
```

Remediation:

For MS Windows:

1. Connect to the DB2 host
2. Right-click on the \`NODE000x\sqldbdir` directory
3. Choose *Properties*
4. Select the *Security* tab
5. Select all non-administrator accounts and revoke the *Full Control* authority

For Unix:

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

```
u=rwx and go=rx
```

Impact:

Organizations should plan and implement policies for restricting access to Databases. Restricting access to the DB2 SQLLIB directory to only authorized groups & persons will reduce the risk of unauthorized access.

2.2 Secure all database containers (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

A DB2 database container is the physical storage of the data.

Rationale:

The containers are needed in order for the database to operate properly. The loss of the containers can cause down time and possibly allow attackers to gain access to sensitive data stored within the containers. 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 to ensure only DB2 administrators have access.

Remediation:

Secure the directory of the containers. The recommended value is "read-only" to all non-DB2 administrator accounts.

2.3 Set umask value for DB2 admin user .profile file (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The DB2 Admin .profile file in UNIX sets the environment variables and the settings for the user.

Rationale:

Ensure the `umask` value is `022` for the owner of the DB2 software before installing DB2. Regardless of where the `umask` is set, `umask` must be set to `022` before installing DB2.

Audit:

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

Remediation:

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

3 DB2 Configurations

[This space intentionally left blank]

3.1 DB2 Instance Parameter Settings

This section provides guidance on how DB2 will control the data in the databases and the system resources that are allocated to the instance.

3.1.1 Enable audit buffer (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

DB2 can be configured to use an audit buffer. It is recommended that the audit buffer size be set to at least 1000.

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.

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 from the DB2 command window:

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

3. Locate `AUDIT_BUF_SZ` value in the output:

```
db2 => get database manager configuration
db2 => …
         Audit buffer size (4KB) (AUDIT_BUF_SZ) = 1000
```

Ensure `AUDIT_BUF_SZ` is greater than or equal to `1000`.

Remediation:

Perform the following to establish an audit buffer:

1. Attach to the DB2 instance

```
db2 => attach to $DB2INSTANCE
```

2. Run the following command from the DB2 command window:

```
db2 => update database manager configuration using audit_buf_sz 1000
```

3.1.2 Encrypt user data across the network (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

DB2 supports a number of authentication mechanisms. It is recommended that the `DATA_ENCRYPT` authentication mechanism be used.

Rationale:

The `DATA_ENCRYPT` authentication mechanism employs cryptographic algorithms to protect both the authentication credentials and user data as it traverses the network. Given this, the confidentiality of authentication credentials and user data is ensured while in transit between the DB2 client and server.

Audit:

Perform the following to determine if the authentication mechanism is set as recommended:

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 the AUTHENTICATION value in the output:

```
db2 => get database manager configuration db2 => … Database manager
authentication (AUTHENTICATION) = DATA_ENCRYPT
```

Note: AUTHENTICATION is set to DATA_ENCRYPT in the above output.

Remediation:

Suggested value is DATA_ENCRYPT so that authentication occurs at the server.

1. Attach to the DB2 instance

```
db2 => attach to $DB2INSTANCE
```

2. Run the following command from the DB2 command window:

```
db2 => update database manager configuration using authentication data_encrypt
```


3.1.3 Require explicit authorization for cataloging (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - 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 SYSADM authority be required 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. This procedure should be restricted to users with a valid DB2 account with the SYSADM or SYSCTRL authority. Setting catalog-noauth to YES by-passes all permissions checks and allows anyone to catalog and uncatalog databases.

Audit:

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

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 the value of CATALOG_NOAUTH in the output:

```
db2 => get database manager configuration
db2 => …
       Cataloging allowed without authority (CATALOG_NOAUTH) = NO
```

Note: `CATALOG_NOAUTH` is set to `NO` in the above output.

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 from the DB2 command window:

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

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_9.1.0/com.ibm.db2.udb.admin.doc/doc/r0000143.htm?cp=SSEPGG_9.1.0%2F11-0-0-4-3

3.1.4 Disable data links support (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

`Datalinks` enables the database to support the Data Links Manager to manage unstructured data, such as images, large files and other unstructured files on the host. It is recommended that data links support be disabled.

Rationale:

Disable `datalinks` if there is no use for them as this will reduce the attack surface of the DB2 service.

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 from the DB2 command window:

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

3. Locate this value of datalinks in the output:

```
db2 => get database manager configuration
db2 => …
       Data Links support (DATALINKS) = NO
```

Note: DATALINKS is set to NO in the above output.

Remediation:

1. Attach to the DB2 instance

```
db2 => attach to $DB2INSTANCE
```

2. Run the following command from the DB2 command window:

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

3.1.5 Secure default database location (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `dftdbpath` parameter contains the default file path used to create DB2 databases. It is recommended that this parameter is set to a directory that is owned by the DB2 Administrator.

Rationale:

Securing the default database path will ensure that the confidentiality, integrity, and availability of data contained in the DB2 service is preserved.

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 from the DB2 command window:

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

3. Locate this value in the output:

```
db2 => get database manager configuration
db2 => …
       Default database path (DFTDBPATH) = <valid directory>
```


Remediation:

1. Attach to the DB2 instance

```
db2 => attach to $DB2INSTANCE
```

2. Run the following command from the DB2 command window:

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

3.1.6 Secure permission of default database location (Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 1 - Linux Host OS

Description:

The `dftdbpath` parameter contains the default file path used to create DB2 databases. It is recommended that the database files permissions be set to `read-only` for non-administrator accounts.

Rationale:

Recommended value is ready-only (RO) to Everyone/Other/Users/Domain Users. This will ensure that the archive logs are protected.

Audit:

Perform the following DB2 commands to obtain the value for this setting:
For MS Windows:

1. Connect to the DB2 host
2. Right-click over the file directory
3. Choose *Properties*
4. Select the *Security* tab
5. Review access from all non-administrator accounts

For Unix:

1. Connect to the DB2 host
2. Change to the file directory
3. Review the permission level of the directory

```
OS => ls -al
```

Remediation:

For MS Windows:

1. Connect to the DB2 host
2. Right-click over the file directory
3. Choose *Properties*
4. Select the *Security* tab
5. Select all non-administrator accounts and revoke the *Full Control* authority

For Unix:

1. Connect to the DB2 host
2. Change to the file directory
3. Change the permission level of the directory

```
OS => chmod -R 755
```

3.1.7 Set diagnostic logging to capture errors and warnings (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - 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 diagnostic level setting is 3. This 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 from the DB2 command window:

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

3. Locate the DIAGLEVEL value in the output:

```
db2 => get database manager configuration
db2 => …
       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. 2. Run the following command from the DB2 command window:

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

3.1.8 Secure all diagnostic logs (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `diagpath` parameter specifies the location of the diagnostic files for the DB2 instance. It is recommended that this parameter be set to a secure location.

Rationale:

Specify a path that is secure and grant permission to appropriate users only.

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 from the DB2 command window:

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

3. Locate the DIAGPATH value in the output:

```
db2 => get database manager configuration
db2 => …
       Diagnostic data directory path (DIAGPATH) = <valid directory>
```

Remediation:

1. Attach to the DB2 instance

```
db2 => attach to $DB2INSTANCE
```

2. 2. Run the following command from the DB2 command window:

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

3.1.9 Require instance name for discovery requests (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `discover` parameter determines what kind of discovery requests, if any, the DB2 server will fulfill. It is recommended that the DB2 server only fulfill requests from clients that know the given instance name.

Rationale:

Discovery capabilities may be used by a malicious entity to derive the names of and target DB2 instances. In this configuration, the client has to specify a known instance name to be able to detect the 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 from the DB2 command window:

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

3. Locate the DISCOVER value in the output:

```
db2 => get database manager configuration
db2 => …
       Discovery mode (DISCOVER) = KNOWN
```

Note: `DISCOVER` is set to `KNOWN` in the above output.

Remediation:

The recommended value is KNOWN. Note: this requires a db2 restart.

1. Attach to the DB2 instance

```
db2 => attach to $DB2INSTANCE
```

2. Run the following command from the DB2 command window:

```
db2 => update database manager configuration using discover known
```

3. Restart the DB2 instance.

```
db2 => db2stop
db2 => db2start
```

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.

3.1.10 Disable instance discoverability (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `discover_inst` parameter specifies whether the instance can be discovered in the network. It is recommended that instances not be discoverable.

Rationale:

Discovery capabilities may be used by a malicious entity to derive the names of and target DB2 instances.

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 from the DB2 command window:

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

3. Locate the DISCOVER_INST is value in the output:

```
db2 => get database manager configuration
db2 => …
       Discover server instance (DISCOVER_INST) = DISABLE
```

Note: DISCOVER_INST is set to DISABLE in the above output.

Remediation:

1. Attach to the DB2 instance

```
db2 => attach to $DB2INSTANCE
```

2. Run the following command from the DB2 command window:

```
db2 => update database manager configuration using discover_inst disable
```

3.1.11 Authenticate federated users at the instance level (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `fed_noauth` parameter determines whether federated authentication will be bypassed at the instance. It is recommended that this parameter be set to `no`.

Rationale:

Set `fed_noauth` to `no` will ensure that authentication is checked at the instance level. This will prevent any federated authentication from bypassing the client and the server.

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 from the DB2 command window:

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

3. Locate the FED_NOAUTH value in the output:

```
db2 => get database manager configuration
db2 => …
       Bypass federated authentication (FED_NOAUTH) = NO
```

Note: `FED_NOAUTH` is set to `NO` in the above output.

Remediation:

1. Attach to the DB2 instance

```
db2 => attach to $DB2INSTANCE
```

2. Run the following command from the DB2 command window:

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

3.1.12 Set maximum connection limits (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `max_connections` parameter indicates the maximum number of client connections allowed per database partition. It is recommended that this parameter be set equal to the `max_coordagents` parameter. The `max_coordagents` parameter equals the number of maximum agents needed to performs connections to the database or attachments to the instance.

NOTE: Ensure that dependent parameters, such as `maxappls`, are set less than the `max_coordagents` parameter. This would be to ensure that the lock limit isn't reached resulting in lock escalation issues.

Rationale:

DB2 allows an unlimited number of users to access the DB2 instance. Set a limit to the number of users allowed to access a DB2 instance to reduce the chances of open connections to attackers. Also, give access to the DB2 instance to only authorized users.

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 from the DB2 command window:

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

3. Locate the `MAX_CONNECTIONS` and `MAX_COORDAGENTS` values in the output:

```
db2 => get database manager configuration
db2 => …
       Max number of client connections (MAX_CONNECTIONS) = 150
       Max number of existing agents (MAX_COORDAGENTS) = 150
```

Note: `MAX_CONNECTIONS` is set to `150` and the `MAX_COORDAGENTS` is set to `150` in the above output.

Perform the following DB2 commands to obtain the value of the MAXAPPLS parameter:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

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

```
db2 => get database configuration
db2 => …
       Max Number of Active Applications (MAXAPPLS) = [99]
```

Note: `MAXAPPLS` is set to `99` in the above output.

Remediation:

The default value for `max_coordagents` is set to `AUTOMATIC` (realistically 200). Allowable range is 1 to 64,000. Or -1 for unlimited. The recommended value is 100. The following command will set the `max_cooradagents` to 100, as well as, set the `max_connections` to `AUTOMATIC` which is also recommended.

1. Attach to the DB2 instance

```
db2 => attach to $DB2INSTANCE
```

2. Run the following command from the DB2 command window:

```
db2 => update database manager configuration using max_coordagents 100
AUTOMATIC
```

If `maxappls` is NOT less than the value for `max_cooradagents` then adjust the value of `maxappls` accordingly:

```
 db2 => update database configuration using maxappls <a number less then
max_coordagents>
```

Default Value:

The default value for `max_connections` is `AUTOMATIC`.

The default value for `max_cooradagents` is -1.

The default value for maxappls is `AUTOMATIC`

3.1.13 Set administrative notification level (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - 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, Health Monitor warnings, and Health Monitor attentions. This may give an indication of any malicious usage on the DB2 instance.

Audit:

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 from the DB2 command window:

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

3. Locate the `NOTIFYLEVEL` value in the output:

```
db2 => get database manager configuration
db2 => …
       Notify Level (NOTIFYLEVEL) = 3
```

Note: `NOTIFYLEVEL` is set to `3` in the above output.

Remediation:

1. Attach to the DB2 instance

```
db2 => attach to $DB2INSTANCE
```

2. Run the following command from the DB2 command window:

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

Default Value:

The default value of `notifylevel` is 3.

3.1.14 Enable server-based authentication (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The srvcon_auth parameter specifies how and where authentication is to take for incoming connections to the server. It is recommended that this parameter is not set to CLIENT.

Rationale:

Ensure that this parameter is not set to CLIENT, since this parameter will take precedence and override the authentication level. Authentication should be set at the server level or use a security plug-in.

Note: If the authentication setting at the database configuration level is set to DATA_ENCRYPT (in benchmark 3.1.2), then leave this setting to NULL.

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 from the DB2 command window:

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

3. Locate the SRVCON_AUTH value in the output:

```
db2 => get database manager configuration
db2 => …
       Server Connection Authentication (SRVCON_AUTH) = SERVER
```

Note: SRVCON_AUTH is set to SERVER in the above output.

Remediation:

The recommended value is SERVER. Note: this will require a DB2 restart.

1. Attach to the DB2 instance

```
db2 => attach to $DB2INSTANCE
```

2. Run the following command from the DB2 command window:

```
db2 => update database manager configuration using srvcon_auth server
```

3. Restart the DB2 instance.

```
db2 => db2stop
db2 => db2start
```

3.1.15 Set failed archive retry delay (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - 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. The range is given as to how flexible your hardware environment is. 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 for an outside attack to happen at the same time.

Rationale:

Ensure that the value is non-zero else archive logging will not retry after the first failure. A denial of service attack can render the database without an archivelog if this setting is not set. An archivelog 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

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

```
db2 => get database configuration
db2 => …
       Log archive retry Delay (secs) (ARCHRETRYDELAY) = 20
```

Note: `ARCHRETRYDELAY` is set to `20` in the above output.

Remediation:

1. Connect to the DB2 database

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. To successfully set the `archretrydelay` within the 10-30 range, run the following command from the DB2 command window:

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

Default Value:

The default value for `archretrydelay` is 20

3.1.16 Auto-restart after abnormal termination (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `autorestart` parameter specifies if the database instance should restart after an abnormal termination. It is recommended that this parameter be set to `ON`.

Rationale:

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

Audit:

Perform the following DB2 commands to obtain the value for this setting:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

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

```
db2 => get database configuration db2 => … Auto restart enabled (AUTORESTART) =
ON
```

Note: `AUTORESTART` is set to `ON` in the above output.

Remediation:

1. Connect to the DB2 database

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

3.1.17 Disable database discovery (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `discover_db` parameter specifies if the database will respond to a discovery request from a client. It is recommended that this parameter be set to `DISABLE`.

Rationale:

Setting the database discovery to disabled can hide a database with sensitive data.

Audit:

Perform the following DB2 commands to obtain the value for this setting:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

3. Locate the `DISCOVER_DB` value in the output:

```
db2 => get database configuration
db2 => …
       Discovery support for this database (DISCOVER_DB) = DISABLE
```

Note: `DISCOVER_DB` is set to `DISABLE` in the above output.

Remediation:

1. Connect to the DB2 database

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => update database configuration using discover_db disable
```

3.1.18 Establish secure archive log location (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `logarchmeth1` parameter specifies the type of media used for the primary destination of archived logs. The value of this parameter will be dependent on your system environment and logging requirements.

Rationale:

Though there are many ways to ensure that your primary logs will be archived, we recommend using the value of `DISK`. This will properly ensure that the primary logs are archived.

NOTE: We are looking to see that the logs are being archived, so a finding of `OFF` is not acceptable.

Audit:

Perform the following DB2 commands to obtain the value for this setting:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

3. Locate the `LOGARCHMETH1` value in the output:

```
4. db2 => get database configuration
   db2 => …
          First log archive method (LOGARCHMETH1) = DISK:C:\DB2LOGS
```

Note: `LOGARCHMETH1` is set to `DISK:C:\DB2LOGS` in the above output.

Remediation:

1. Connect to the DB2 database

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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


3.1.19 Secure permission of the primary archive log location (Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 1 - Linux Host OS

Description:

The `logarchmeth1` parameter specifies the type of media used for the primary destination of archived logs. It is recommended that the archive log permission setting be set to read-only for non-administrator accounts.

Rationale:

Recommended value is ready-only (RO) to Everyone/Other/Users/Domain Users. This will ensure that the archive logs are protected.

Audit:

Perform the following DB2 commands to obtain the value for this setting:
For MS Windows:

1. Follow the Audit steps in section Establish secure archive log location to obtain the primary archive log directory.
2. Connect to the DB2 host
3. Right-click on the directory obtained in step #1
4. Choose *Properties*
5. Select the *Security* tab
6. Review access from all non-administrator accounts

For Unix:

1. Right-click on the directory obtained in step #1.
2. Connect to the DB2 host
3. Change to the directory obtained in step #1
4. Review the permission level of the directory

```
OS => ls -al
```


Remediation:

For MS Windows:

1. Follow the Audit steps in section Establish secure archive log location to obtain the primary archive log directory.
2. Connect to the DB2 host
3. Right-click on the directory obtained in step #1
4. Choose *Properties*
5. Select the *Security* tab
6. Select all non-administrator accounts and revoke the *Full Control* authority

For Unix:

1. Follow the Audit steps in section Establish secure archive log location to obtain the primary archive log directory.
2. Connect to the DB2 host
3. Change to the directory obtained in step #1
4. Change the permission level of the directory

```
OS => chmod -R 755
```

3.1.20 Establish secure secondary archive location (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `logarchmeth2` parameter specifies the type of media used as the secondary destination for archived logs. It is recommended that this parameter be set to a secure location.

Rationale:

Recommended value is `DISK:`*<valid directory>*. This will ensure that the secondary logs are written to disk.

Audit:

Perform the following DB2 commands to obtain the value for this setting:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```


2. Run the following command from the DB2 command window:

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

3. Locate the `LOGARCHMETH2` value in the output:

```
db2 => get database configuration
db2 => …
       Second log archive method (LOGARCHMETH2) = DISK:C:\DB2LOGS2
```

Note: `LOGARCHMETH2` is set to `DISK:C:\DB2LOGS2` in the above output.

Remediation:

1. Connect to the DB2 database

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

3.1.21 Secure permission of the secondary archive location (Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 1 - Linux Host OS

Description:

The `logarchmeth2` parameter specifies where the type of media used as the secondary destination for archived logs. It is recommended that the archive log permissions be set to read-only for non-administrator accounts.

Rationale:

Recommended value is ready-only (RO) to Everyone/Other/Users/Domain Users. This will ensure that the archive logs are protected.

Audit:

Perform the following DB2 commands to obtain the value for this setting:
For MS Windows:

1. Follow the Audit steps in section Establish secure secondary archive location to obtain the primary archive log directory.
2. Connect to the DB2 host
3. Right-click on the directory obtained in step #1
4. Choose *Properties*
5. Select the *Security* tab
6. Review access from all non-administrator accounts

For Unix:

1. Follow the Audit steps in section Establish secure secondary archive location to obtain the primary archive log directory.
2. Connect to the DB2 host
3. Change to the directory obtained in step #1
4. Review the permission level of the directory

```
OS => ls -al
```

Remediation:

MS Windows:

1. Follow the Audit steps in section Establish secure secondary archive location to obtain the primary archive log directory.
2. Connect to the DB2 host

3. Right-click on the directory obtained in step #1
4. Choose *Properties*
5. Select the *Security* tab
6. Select all non-administrator accounts and revoke the *Full Control* authority

For Unix:

1. Follow the Audit steps in section Establish secure secondary archive location to obtain the primary archive log directory.
2. Connect to the DB2 host
3. Change to the directory obtained in step #1
4. Change the permission level of the directory

```
OS => chmod -R 755
```

3.1.22 Establish secure tertiary archive log location (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `failarchpath` parameter specifies the location for the archive logs if the primary or secondary archive destination is not available. It is recommended that this parameter be set to point to a secure location.

Rationale:

Ensure that a valid path is specified for this setting so that archive logs can have an alternate failover destination due to media problems. Access to the destination location should only be granted to the DB2 system administrator; and give read-only privilege to non-privileged users.

Audit:

Perform the following DB2 commands to obtain the value for this setting:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

3. Locate the `FAILARCHPATH` value in the output:

```
db2 => get database configuration
db2 => …
       Failover log archive path (FAILARCHPATH) = <valid path>
```

Note: `FAILARCHPATH` is set to a valid path in the above output.

Remediation:

1. Connect to the DB2 database

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => update database configuration using failarchpath <valid path>
```

3.1.23 Secure permission of the tertiary archive location (Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 1 - Linux Host OS

Description:

The `failarchpath` parameter specifies the location of the tertiary destination for archived logs. It is recommended that the archive log permission be set to read-only for non-administrator accounts.

Rationale:

Recommended value is ready-only (RO) to Everyone/Other/Users/Domain Users. This will ensure that the archive logs are protected.

Audit:

Perform the following DB2 commands to obtain the value for this setting:
For MS Windows:

1. Connect to the DB2 host
2. Right-click over the file directory
3. Choose *Properties*
4. Select the *Security* tab
5. Review access from all non-administrator accounts

For Unix:

1. Connect to the DB2 host
2. Change to the file directory
3. Review the permission level of the directory

```
OS => ls -al
```

Remediation:

For MS Windows:

1. Connect to the DB2 host
2. Right-click over the file directory
3. Choose *Properties*
4. Select the *Security* tab
5. Select all non-administrator accounts and revoke the *Full Control* authority

For Unix:

1. Connect to the DB2 host
2. Change to the file directory
3. Change the permission level of the directory

```
OS => chmod -R 755
```

3.1.24 Establish secure log mirror location (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `mirrorlogpath` parameter specifies a location to store the mirror copy of the logs. It is recommended that this parameter be set to a secure location.

Rationale:

A mirror log path should not be empty and it should be a valid path that is secure. The mirror log path stores a second copy of the active log files.

Audit:

Perform the following DB2 commands to obtain the value for this setting:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

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

```
db2 => get database configuration
db2 => …
       Mirror log path (MIRRORLOGPATH) = C:\DB2MIRRORLOGS
```

Note: `MIRRORLOGPATH` is set to `C:\DB2MIRRORLOGS` in the above output.

Remediation:

1. Connect to the DB2 database

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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


3.1.25 Establish retention set size for backups (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `num_db_backups` parameter specifies the number of backups to retain for a database before the old backups is marked 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 to ensure 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

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

```
db2 => get database configuration
db2 => …
       Number of database backups to retain (NUM_DB_BACKUPS) = 12
```

Note: `NUM_DB_BACKUPS` is set to `12` in the above output.

Remediation:

1. Connect to the DB2 database

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

3.1.26 Set archive log failover retry limit (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - 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. The recommended value is 5.

Audit:

Perform the following DB2 commands to obtain the value for this setting:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

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

```
db2 => get database configuration
db2 => …
       Number of log archive retries on error (NUMARCHRETRY) = 5
```

Note: `NUMARCHRETRY` is set to `5` in the above output.

Remediation:

1. Connect to the DB2 database

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

3.2 Database Manager Configuration parameters

Database Configuration Parameters set several resource limits [values] to be allocated to a database. Many database configuration parameters can be modified to optimize performance and capacity.

3.2.1 Priority of agents - agentpri (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

Using the operating system scheduler, the agentpri parameter controls the priority of CPU time given to all database manager agents, threads, and processes.

Rationale:

Performance issues can result as users, applications, and database manager configuration parameters require CPU time. Balancing the access to CPU time by setting the 'agentpri' database configuration parameter can improve performance for all processes.

Audit:

1. Run the following command to determine the current 'agentpri' value

```
db2 => select name, value from sysibmadm.dbmcfg where name = 'agentpri'
```


Remediation:

1. Determine the correct value to set 'agentpri' and enter the following command:

```
db2 => update dbm cfg using agentpri <enter proper value> (immediate or deferred)
```

Impact:

User and applications can be severely impacted by changing the 'agentpri' database manager configuration parameter. Careful database peformance planning should be done prior to making changes.

Default Value:

Windows

-1 (system) [0-6]

Unix

-1 (system) [41-128]

Linux

-1 (system) [1-99]

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.config.doc/doc/r0000261.html%23r0000261?cp=SSEPGG_10.1.0%2F2-2-4-7-1

3.2.2 Application support layer heap - aslheapsz (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The 'aslheapsz' database configuration parameter is allocated as shared memory and forms a buffer for communication between the agent and the local application.

Rationale:

As different application have changing and dynamic memory requirements for communicating between their agents and local applications, setting aside a memory buffer can reduce application performance degradation.

Audit:

```
select value from sysibmadm.dbmcfg where name = 'aslheapsz'
```

Determine if the aslheapsz value is set correctly.

Remediation:

```
update dbm cfg using aslheapsz <value> immediate
```

Impact:

Changing the aslheapsz parameter can adversely affect local and remote application performance. If memory is limited, the aslheapsz parameter can be reduced. Conversely, if memory is available, increasing the aslheapsz, especially when queries are large, is recommended.

Default Value:

15 [1- 524 288]

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.perf.doc/doc/c0005387.html?lang=en

3.2.3 Fast communication manager - fcm_parallelism (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The fcm_parallelism parameter sets the number of conduit pairs for sending and receiving. One pair: a sender and a receiver carry all communication back and forth to other members in the instance.

Rationale:

The fcm_parallelism parameter sets the range of conduit pairs used for parallel communication between DB2 instance members. When 'fcm_parallelism' parameter is set to '1', no parallel communication is enabled.

Audit:

1. Determine if the current value is correct:

```
db2 => select name, value from sysibmadm.dbmcfg where name = 'fcm_parallelism'
```

Remediation:

1. Shut down and restart the system for the update to commit.

```
db2 => update dbm cfg using fcm_parallelism <value> immediate or deferred
```

Default Value:

1 [1-8]

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.config.doc/doc/r0060554.html%23r0060554?cp=SSEPGG_10.1.0(=en

3.2.4 Intrapartition parallelism - intra_parallel (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The 'intra_parallel' database configuration parameter enables or disables intra-partition query. The 'intra_parallel' database configuration parameter applies to database servers with local clients, partitioned database servers with local and remote clients, and database servers with local clients.

Rationale:

Query performance can be degraded across multiple partitions. Enabling the 'intra_parallel' parameter can improve query performance across partitioned databases.

Audit:

```
select name, value from sysibmadm.dbmcfg where name = 'intra_parallel'
```

Determine if the current value is correctly set.

Remediation:

```
update dbm cfg using intra_parallel <value> immediate or deferred
```

Impact:

A value of SYSTEM causes the parameter value to be set to YES or NO based on the hardware on which the database manager is running. If the number of logical CPUs on the system is > 1, when the value is set to SYSTEM, intrapartition query parallelism is enabled.

Default Value:

NO (0) [SYSTEM (-1), NO (0), YES (1)]

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.config.doc/doc/r0000146.html%23r0000146?cp=SSEPGG_10.1.0(=en

3.2.5 Maximum query degree of parallelism - max_querydegree (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The max_querydegree database manager parameter sets the level of intrapartition parallelism used for executing SQL statements.

Rationale:

When the 'max_querydegree' database parameter is executed any SQL statement will limit the number of parallel operations within a partition according to the max_querydegree parameters' value.

Audit:

```
select name, value from sysibmadm.dbmcfg where name = 'max_querydegree'
```

Determine if the current value of 'max_querydegree' is correctly set.

Remediation:

```
update dbm cfg using max_querydegree <insert value> immediate or deferred
```

Impact:

Database partitions can inhibit the performance of database queries. Enabling max_querydegree can improve SQL statement query peformance.

Default Value:

-1 (Any) [Any, 1 - 32 767]

Any means system determined.

This value means that the system uses the degree of parallelism determined by the optimizer; otherwise, the user-specified value is used.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.config.doc/doc/r0000140.html%23r0000140?cp=SSEPGG_10.1.0

3.2.6 Agent pool size - num_poolagents (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The 'num_poolagents' parameter sets the maximum size of the idle agent pool. The parameter applies to Partitioned databases with local and remote clients, database servers with local and remote clients.

Rationale:

When set to AUTOMATIC, the database manager dynamically manages the number of idle agents to pool. Once an agent completes its work, although not terminated, it becomes idle for a period of time and depending on the workload and agent type, it may terminate at some point.

Audit:

```
select name, value from sysibmadm.dbmcfg where name = 'num_poolagents'
```

1. Determine if the 'num_poolagents' value is correctly set.

Remediation:

```
update dbm cfg using num_poolagents <value> immediate or deferred or automatic
```

Impact:

When using AUTOMATIC, you can still specify a value for the num_poolagents configuration parameter. Additional idle agents will always be pooled when the current number of pooled idle agents is less than or equal to the value that you specified.

Default Value:

100, AUTOMATIC [-1, 0 - 64000]

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.config.doc/doc/r0000145.html%23r0000145?cp=SSEPGG_10.1.0

3.2.7 Sort heap threshold - sheapthres (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The sort heap threshold (sheapthres) parameter provides a continuous soft limit on the total amount of memory used by private sorts across the instance.

Rationale:

As private sort memory consumption increases, the 'sheapthres' parameter provides a limit, once reached, the additional memory for additional incoming private sort requests is significantly reduced.

Audit:

```
select name, value from sysibmadm.dbmcfg where name = 'sheapthres'
```

Determine if the current value for 'sheapthres' is correctly set.

Remediation:

```
update dbm cfg using sheapthres <value> immediate or deferred
```

Impact:

Implementing the 'sheapthres' parameter defines a fixed memory pool for incoming private sort request and once reached will degrade additional incoming private sort requests and could adversely impact the Organizations' operations.

Default Value:

UNIX 32-bit platforms

0 [0, 250 - 2097152]

Windows 32-bit platforms

0 [0, 250 - 2097152]

64-bit platforms

0 [0, 250 - 2147483647]

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.config.doc/doc/r0000260.html%23r0000260?cp=SSEPGG_10.1.0

3.2.8 Instance impact policy configuration - util_impact_lim (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The util_impact_lim parameter limits the performance degradation of a throttled utility on a workload.

Rationale:

Setting a limit or throttle on utilities impacting production workloads reduces the risk of utilities degrading production work.

Audit:

```
select name, value from sysibmadm.dbmcfg where name = 'util_impact_lim'
```

Remediation:

```
update dbm cfg using util_impact_lim <value> immediate or deferred
```

Impact:

The util_impact_lim parameter will throttle utility invocations. If not throttled (util_impact_lim 100) the utilities could impact the production workload.

Default Value:

10 [1 - 100]

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.config.doc/doc/r0010968.html%23r0010968?cp=SSEPGG_10.1.0(=en

3.2.9 Client I/O block size configuration parameter - rqrioblk (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The rqrioblk parameter sets the block size at the Data Server Runtime Client for use when a blocking cursor is opened, which is taken from the applications' private address space and measured in bytes.

Rationale:

When planning to transfer a large number of rows or a large size of rows (over 4096 bytes); choosing larger row blocks might improve performance. Using the OPTIMIZE FOR clause on the SELECT statement can control the number of fetch requests.

Audit:

```
select name, value from sysibmadm.dbmcfg where name = 'rqrioblk'
```

Determine if the 'rqrioblk' parameters' value is correctly set.

Remediation:

```
update dbm cfg using rqrioblk <value> immediate or deferred
```

Impact:

Choosing larger row blocks can result in two costs: larger working set memory for each connection and causing more fetch requests than the application usually requires.

Default Value:

32 767 [4 096 - 65 535] bytes

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.config.doc/doc/r0000270.html%23r0000270?cp=SSEPGG_10.1.0(=en

3.2.10 TCP/IP service name - svcename (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The `svcename` parameter reserves the port number for listening to incoming communications from a Data Server Runtime Client. Both the database server port number and the TCP/IP service name must be defined on the database client.

Rationale:

When the database server is started a port number is required to listen for incoming connection requests. The `svcename` parameter defines the port number for incoming connection requests. On UNIX and Linux systems, the services file is found at: /etc/services

Audit:

```
select name, value from sysibmadm.dbmcfg where name = 'svcename'
```

Determine if the `svcename` parameter value is correctly set and is not the default port (50000)

Remediation:

```
update dbm cfg using svcename <value> immediate or deferred
```

Impact:

Unless the `svcename` parameter defines the port number, the database cannot accept incoming TCP/IP connection requests.

References:

1. https://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.config.doc/doc/r0000273.html?lang=en

3.2.11 SSL service name - ssl_svcename (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The `ssl_svcename` configuration parameter defines the name of the port the database server listens for communications from remote client nodes using SSL protocol. The `ssl_svcename` and the svcename port numbers cannot be the same.

UNIX operating systems the ssl_svcename file is located in: /etc/services

Rationale:

The database requires a defined port to listen for incoming remote clients using the SSL protocol. The `ssl_svcename` configuration parameter defines the port for communicating with remote clients.

Consider using a non-default port to help protect the database from attacks directed to a default port.

Audit:

```
select name, value from sysibmadm.dbmcfg where name = 'ssl_svcename'
```

Determine if the current `ssl_svcename` parameter value is correctly set and is not a default port (50000)

Remediation:

```
update dbm cfg using ssl_svcename <value> immediate or deferred
```

Impact:

Unless the `ssl_svcename` parameter defines the port number, the database cannot accept incoming SSL connection requests.

Default Value:

Null

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.config.doc/doc/r0053615.html

3.2.12 Maximum Java interpreter heap size - java_heap_sz (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The java_heap_sz parameter defines the maximum size of the heap for Java interpreter to service UDFs and DB2 stored procedures.

Rationale:

When UDF or a Java stored procedure begins these require memory to operate and the Java_heap_sz parameter defines the memory heap used by the Java interpreter.

Audit:

```
select name, value from sysibmadm.dbmcfg where name = 'java_heap_sz'
```

Determine if the 'java_heap_sz' parameters' value is correctly set

Remediation:

```
update dbm cfg using java_heap_sz <value> immediate or deferred
```

Impact:

Java stored procedures and UDFs require memory to operate and both are serviced by the java_heap_sz configuration parameter. As db2fmp stored procedures and multithreaded db2fmp processes using threadsafe fenced routines are both serviced by a single heap, the java_heap_sz parameter value set correctly is critical to their operations.

Default Value:

2048 [0-524288]

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.config.doc/doc/r0000137.html%23r0000137?cp=SSEPGG_10.1.0

3.2.13 Authentication type for incoming connections at the server - srvcon_auth (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The **srvcon_auth** parameter defines where and how user authentication is done for incoming connections at the server. If no value is used, DB2 uses the database manager configuration parameter **authentication.**

Rationale:

Incoming connections to the DB2 server must follow an authentication protocol. The **srvcon_aut**h server configuration parameter defines how and where user authentication is done.

Audit:

Identify the current value of the **srvcon_auth** database configuration parameter:

```
select name, value from sysibmadm.dbmcfg where name = 'srvcon_auth'
```

Remediation:

Update the current value of the **srvcon_auth** database configuration parameter to the correct value:

```
 db2 => update dbm cfg using srvcon_auth <any supported authentication>
```

Impact:

Defining a srvcon_auth server configuration parameter, which restricts how and where incoming connections to the server are executed, could adversely impact database operations if not carefully planned and implemented.

Default Value:

Not specified

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.config.doc/doc/r0011454.html?lang=en

4 Row and Column Access Control (RCAC)

DB2 RCAC controls access to a table at the row and column level. Row and column access control is sometimes referred to as fine-grained access control or FGAC. Identify and Gather the Organizations' Security Policies, Management and Staff Roles, and User and Group Lists to compare against existing DB2 RCAC policies for compliance.

4.1 Review Organizations' Policies against DB2 RCAC Policies (Not Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

DB2 Row and Column Access Control (RCAC) Database Policies control access to DB2 Tables, should match the Organizations' Security and Database Access Policies, and be regularly reviewed for gaps.

Rationale:

Missing or incomplete DB2 RCAC Security Policies will increase the risks to the Organizations' 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 Organizations Security and Data Access Database Policies against the current DB2 Policies to determine if gaps exist.

```
1. Identify each written Organization Security Policy.
2. Find the matching DB2 Row and Column Access Control 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 newly created DB2 RCAC policy against the Organizations' policy
```

Impact:

Implementing DB2 RCAC Policies will apply restricted access to 'rows and columns' for specified DB2 tables and can negatively impact database operations unless carefully planned and implemented.

Default Value:

Not installed

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0057423.html?lang=en

4.2 Secure SECADM Authority (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

Security administration, as well as the security administrator play an important part in implementing RCAC rules. As a result, constant review of the SECADM authority is warranted to ensure the members still have a business need belonging to this group.

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 increase risk.

Audit:

It is important to consider reviewing the members of the `SECADM` authority when implementing this recommendation. Such consideration of this review is addressed in Section 7.1 of this document.

Remediation:

It is important to consider reviewing the members of the `SECADM` authority when implementing this recommendation. Such consideration of this review is addressed in Section 7.1 of this document.

Impact:

Only the security administrator has the ability to grant other users, groups, or roles the ACCESSCTRL, DATAACCESS, DBADM, and SECADM authorities.

References:

1. https://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.sec.doc/doc/c0021054.html?lang=en

4.3 Review Users, Groups, and Roles (Not Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 1 - Linux Host OS

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 safe guarding possible security threats within your infrastructure.

Rationale:

If an 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, still has access to said data then that individual could compromise or use the information in a malicious manner. Doing so would damage the confidentiality, integrity, and availability of the data in the DB2 instance.

Audit:

Review the users within your database environment:

UNIX:

```
cat /etc/passwd
```

Windows:

1. Run compmgmt.msc
2. Click 'Local Users and Groups'
3. Double click 'Users'
4. Review users

Review the groups within your database environment:

UNIX:

```
cat /etc/group
```

Windows:

1. Run compmgmt.msc
2. Click 'Local Users and Groups'
3. Double click 'Groups'
4. Review groups

Review the Roles and role members within your database environment:

Attach to DB2 Instance:

```
db2 => attach to $DB2INSTANCE
```

Connect to DB2 database:

```
db2 => connect to $DBNAME
```

Run the command:

```
db2 => select rolename, grantee from syscat.roleauth where grantortype <> 'S'
```

Remediation:

To remove users from your database environment:

UNIX:

```
userdel -r <user name>
```

Windows:

1.
    1. Run compmgmt.msc
    2. Click 'Local Users and Groups'
    3. Double click 'Users'
    4. Right-click on <*user name*>
    5. Select 'Delete'

To remove groups from your database environment:

UNIX:

```
groupdel <group name>
```

Windows:

1. Run compmgmt.msc
2. Click 'Local Users and Groups'
3. Double click 'Groups'
4. Right-click on <*group name*>
5. Select 'Delete'

To remove Roles or Role Members from your database environment

Attach to DB2 Instance:

```
db2 => attach to $DB2INSTANCE
```

Connect to DB2 database:

```
db2 => connect to $DBNAME
```

To remove Role members from Roles:

```
db2 => revoke role <role name> from <user/group/role member>
```

To remove Roles:

```
db2 => drop role <role name>
```

4.4 Review Row Permission logic according to policy (Not Scored)

Profile Applicability:

- Level 2 - 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 Organizations' protected data and will prevent efforts to monitor, alert, and respond to these risks in the future.

Audit:

Attach to the DB2 Instance:

```
db2 => attach to $DB2INSTANCE
```

Connect to database environment:

```
db2 => connect to $DBNAME
```

Run the following:

```
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 newly created DB2 RCAC policy against the Organizations' policy
```

Impact:

Implementing DB2 RCAC Policies will apply restricted access to 'rows and columns' for specified DB2 tables and can negatively impact database operations unless carefully planned and implemented.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0057423.html?lang=en

4.5 Review Column Mask logic according to policy (Not Scored)

Profile Applicability:

- Level 2 - 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 Organizations' protected data and will prevent efforts to monitor, alert, and respond to these risks in the future.

Audit:

Attach to the DB2 Instance:

```
db2 => attach to $DB2INSTANCE
```

Connect to database environment:

```
db2 => connect to $DBNAME
```

Run the following:

```
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 newly created DB2 RCAC policy against the Organizations' policy
```

Impact:

Implementing DB2 RCAC Policies will apply restricted access to 'rows and columns' for specified DB2 tables and can negatively impact database operations unless carefully planned and implemented.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0057423.html?lang=en

5 Database Maintenance

This section provides guidance on protecting and maintaining the database instance.

5.1 Enable Backup Redundancy (Not Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

Backup redundancy ensures that multiple instances of database backups exist.

Rationale:

Maintaining redundant copies of database backups will increase business continuity capabilities in should a DB2 service failure coincides with a corrupt backup.

Audit:

Review the replication of your backups based on company policy.

Remediation:

Define a process to replicate your backups onto multiple locations.

5.2 Protecting Backups (Not Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

Backups of your database should be stored in a secure location. It is recommended that backups be created to ensure that the instance can be recovered.

Rationale:

Backups may contain sensitive data that attackers can use to retrieve valuable information about the organization.

Audit:

Review the access of your backups based on company policy.

Remediation:

Define a security policy for all backups stored.

5.3 Enable Database Maintenance (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

Enable automatic database maintenance on your DB2 instance. It is recommended that the DB2 Automatic Maintenance tool be used to ensure that the instance is performing optimally.

Rationale:

A well maintained DB2 instance will provide access to the data and reduces database outages.

Audit:

Perform the following DB2 commands to obtain the value for this setting:

1. Connect to the DB2 database:

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => update database configuration
```

3. Locate this value in the output:

```
db2 => get database configuration
db2 => …
       Automatic maintenance (AUTO_MAINT) = ON
```

Note: `AUTO_MAIN` is set to `ON` in the above output.

Remediation:

A well maintained DB2 instance will provide access to the data and reduces database outages.
Remediation:

1. Connect to the DB2 database:

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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


6 Securing Database Objects

Note: `SYSCAT` views have underlying `SYSIBM` tables that are also granted to `PUBLIC` group by default. Ensure that permission applied to these tables revoke access from unnecessary users. If the database was created using the `RESTRICTIVE` option, then grants to `PUBLIC` are voided.

6.1 Restrict Access to SYSCAT.AUDITPOLICIES (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - 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 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'AUDITPOLICIES' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

Impact:

As SYSCAT.AUDITPOLICIES contains all the Audit policies, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0050610.html?cp=SSEPGG_10.5.0%2F2-12-8-2(=en

6.2 Restrict Access to SYSCAT.AUDITUSE (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - 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 on the types of objects are being audited. Access to the audit usage may aid 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'AUDITUSE'Restrict Access to SYSCAT.DBAUTH and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

Impact:

As SYSCAT.AUDITUSE contains an audit policy for an object, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

6.3 Restrict Access to SYSCAT.DBAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'DBAUTH' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

Impact:

As SYSCAT.DBAUTH contains one or more database -level authorities granted to a user, group, or role, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0001041.html?cp=SSEPGG_10.5.0%2F2-12-8-30(=en

6.4 Restrict Access to SYSCAT.COLAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The SYSCAT.COLAUTH view contains the column privileges granted to the user, group, or role in the database

Rationale:

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

Audit:

Perform the following DB2 commands to obtain the value for this setting:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'COLAUTH' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

Impact:

This view contains the column privileges granted to the user, group, or role in the database and may be used as an attack vector. Therefore, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.dbobj.doc/doc/t0005379.html?lang=en

6.5 Restrict Access to SYSCAT.EVENTS (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The `SYSCAT.EVENTS` view contains all 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'EVENTS' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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


Impact:

SYSCAT.EVENTS contains an event being monitored, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0001043.html?cp=SSEPGG_10.5.0%2F2-12-8-34(=en

6.6 Restrict Access to SYSCAT.EVENTTABLES (Scored)

Profile Applicability:

- Level 2 - 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'EVENTTABLES' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

Impact:

As SYSCAT.EVENTTABLES contains the target SQL table of an event monitor, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0007483.html?cp=SSEPGG_10.5.0%2F2-12-8-35(=en

6.7 Restrict Access to SYSCAT.ROUTINES (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'ROUTINES' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

Impact:

As `SYSCAT.ROUTINES` view contains all user-defined routines, functions, and stored procedures in the database, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

6.8 Restrict Access to SYSCAT.INDEXAUTH (Scored)

Profile Applicability:

- Level 2 - 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'INDEXAUTH' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

Impact:

As SYSCAT.INDEXAUTH represents a user, group, or role that has been granted CONTROL privilege on an index, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0001046.html?cp=SSEPGG_10.5.0%2F2-12-8-44(=en

6.9 Restrict Access to SYSCAT.PACKAGEAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `SYSCAT.PACKAGEAUTH` view contains a list of users or groups that has `EXECUTE` privilege on a package. 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'PACKAGEAUTH' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

Impact:

The SYSCAT.PACKAGEAUTH view contains a list of users or groups that has EXECUTE privilege on a package, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

6.10 Restrict Access to SYSCAT.PACKAGES (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `SYSCAT.PACKAGES` view contains 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'PACKAGES' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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


Impact:

As SYSCAT.PACKAGES lists all the packages created by binding an application program, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

6.11 Restrict Access to SYSCAT.PASSTHRUAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'PASSTHRUAUTH' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

Impact:

As SYSCAT.PASSTHRUAUTH lists users, groups, or roles granted pass-through authorization to query a data source, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0002184.html?cp=SSEPGG_10.5.0%2F2-12-8-70(=en

6.12 Restrict Access to SYSCAT.SECURITYPOLICIES (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'SECURITYPOLICIES' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

Impact:

As SYSCAT.SECURITYPOLICIES view contains all database security policies, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0020048.html?cp=SSEPGG_10.5.0%2F2-12-8-91(=en

6.13 Restrict Access to SYSCAT.SECURITYPOLICYEXEMPTIONS (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `SYSCAT.SECURITYPOLICYEXEMPTIONS` contains the exemption on 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 exemption rules 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'SECURITYPOLICYEXEMPTIONS' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

Impact:

As SYSCAT.SECURITYPOLICYEXEMPTIONS view contains all database security policies, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0020042.html?cp=SSEPGG_10.5.0%2F2-12-8-93(=en

6.14 Restrict Access to SYSCAT.SURROGATEAUTHIDS (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The SYSCAT.SURROGATEAUTHIDS contains 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 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'SURROGATEAUTHIDS' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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


Impact:

The SYSCAT.SURROGATEAUTHIDS contains all accounts that have been granted SETSESSIONUSER privilege on a user or to PUBLIC, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0020044.html?cp=SSEPGG_10.5.0%2F2-12-8-102(=en

6.15 Restrict Access to SYSCAT.ROLEAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The SYSCAT.ROLEAUTH 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'ROLEAUTH' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

Impact:

The SYSCAT.ROLEAUTH contains information on all roles and their respective grantees, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0050619.html?cp=SSEPGG_10.5.0%2F2-12-8-74(=en

6.16 Restrict Access to SYSCAT.ROLES (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `SYSCAT.ROLES` 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'ROLES' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

Impact:

The SYSCAT.ROLES contains all roles available in the database, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0050612.html?cp=SSEPGG_10.5.0%2F2-12-8-75(=en

6.17 Restrict Access to SYSCAT.ROUTINEAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `SYSCAT.ROUTINEAUTH` 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 grants of routines to users or groups 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'ROUTINEAUTH' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

Impact:

The SYSCAT.ROUTINEAUTH contains a list of all users that have EXECUTE privilege on a routine (function, method, or procedure), REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0007491.html?cp=SSEPGG_10.5.0%2F2-12-8-76(=en

6.18 Restrict Access to SYSCAT.SCHEMAAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `SYSCAT.SCHEMAAUTH` 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 grants of schemas to users or groups 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'SCHEMAAUTH' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

Impact:

The SYSCAT.SCHEMAAUTH contains a list of all users that have one or more privileges or access to a particular schema, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

6.19 Restrict Access to SYSCAT.SCHEMATA (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The SYSCAT.SCHEMATA 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 created schemas 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'SCHEMATA' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

Impact:

The SYSCAT.SCHEMATA contains all schema names in the database, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0001059.html?cp=SSEPGG_10.5.0%2F2-12-8-85(=en

6.20 Restrict Access to SYSCAT.SEQUENCEAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `SYSCAT.SEQUENCEAUTH` 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'SEQUENCEAUTH' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

Impact:

The SYSCAT.SEQUENCEAUTH contains users, groups, or roles granted privilege(s) on a sequence, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0008181.html?cp=SSEPGG_10.5.0%2F2-12-8-94(=en

6.21 Restrict Access to SYSCAT.STATEMENTS (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `SYSCAT.STATEMENTS` 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 source code or 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'STATEMENTS' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

Impact:

The SYSCAT.STATEMENTS contains all SQL statements of a compiled package, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0001060.html?cp=SSEPGG_10.5.0%2F2-12-8-99(=en

6.22 Restrict Access to SYSCAT.TABAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `SYSCAT.TABAUTH` 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'TABAUTH' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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


Impact:

The SYSCAT.TABAUTH contains users or groups that have been granted one or more privileges on a table or view, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0001061.html?cp=SSEPGG_10.5.0%2F2-12-8-103&lang=en

6.23 Restrict Access to SYSCAT.TBSPACEAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The SYSCAT.TBSPACEAUTH contains users or groups that has been granted the USE privilege on a particular table space 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee from sysibm.systabauth where tcreator = 'SYSCAT' and
ttname = 'TBSPACEAUTH' and grantee = 'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

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

Impact:

The SYSCAT.TBSPACEAUTH contains users or groups that has been granted the USE privilege on a particular table space in the database, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0002201.html?cp=SSEPGG_10.5.0%2F2-12-8-110(=en

6.24 Restrict Access to Tablespaces (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select grantee, tbspace from sysibm.systbspaceauth where grantee =
'PUBLIC'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => REVOKE USE OF TABLESPACE [$tablespace_name] FROM PUBLIC
```

Impact:

A tablespace is where the data is physically stored, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0001064.html?cp=SSEPGG_10.5.0%2F2-12-8-108(=en

6.25 Restrict Access to SYSCAT.MODULEAUTH (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

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

Rationale:

Any databases created without the RESTRICT option, automatically GRANTS THE SELECT privilege to PUBLIC for SYSCAT views. Therefore, it is strongly recommended to explicitly REVOKE the SELECT privilege on the SYSCAT.MODULEAUTH from PUBLIC to reduce risk to the Organizations' data.

Audit:

Perform the following DB2 commands to obtain the value for this setting:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select rtrim(grantee) as grantee, controlauth, alterauth, deleteauth,
indexauth, insertauth, selectauth, updateauth, refauth  from sysibm.systabauth
where tcreator = 'SYSCAT' and ttname = 'MODULEAUTH'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => revoke select on syscat.moduleauth from public
```


Impact:

As SYSCAT.MODULEAUTH contains all granted privileges on a module for users, groups, and roles, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0054748.html?lang=en

6.26 Restrict Access to SYSCAT.VARIABLEAUTH (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The SYSCAT.VARIABLEAUTH contains the granted privileges on a global variable for users, groups, or roles and are read only.

Rationale:

Any databases created without the RESTRICT option, automatically GRANTS THE SELECT privilege to PUBLIC for SYSCAT views. Therefore, it is strongly recommended to explicitly REVOKE the SELECT privilege on the SYSCAT.VARIABLEAUTH from PUBLIC to reduce risk to the Organizations' 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select rtrim(grantee) as grantee, controlauth, alterauth, deleteauth,
indexauth, insertauth, selectauth, updateauth, refauth  from sysibm.systabauth
where tcreator = 'SYSCAT' and ttname = 'VARIABLEAUTH'
```

3. Review privileges for users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => revoke select on syscat.variableauth from public
```

Impact:

As SYSCAT.VARIABLEAUTH contains all granted privileges on a module for users, groups, and roles, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0050504.html?lang=en

6.27 Restrict Access to SYSCAT.WORKLOADAUTH (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The SYSCAT.WORKLOADAUTH catalog 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 GRANTS 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 Organizations' data.

Audit:

Perform the following DB2 commands to obtain the value for this setting:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select rtrim(grantee) as grantee, controlauth, alterauth, deleteauth,
indexauth, insertauth, selectauth, updateauth, refauth  from sysibm.systabauth
where tcreator = 'SYSCAT' and ttname = 'WORKLOADAUTH'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => db2 => revoke select on syscat.workloadauth from public
```

Impact:

As SYSCAT.WORKLOADAUTH contains all granted privileges on a module for users, groups, and roles, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0050558.html?cp=SSEPGG_10.5.0%2F2-12-8-127(=en

6.28 Restrict Access to SYSCAT.XSROBJECTAUTH (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The SYSCAT.XSROBJECTAUTH contains granted USAGE privileges on a particular XSR object for users, groups, or roles and are read only.

Rationale:

Any databases created without the RESTRICT option, automatically GRANTS THE SELECT privilege to PUBLIC for SYSCAT views. Therefore, it is strongly recommended to explicitly REVOKE the SELECT privilege on the SYSCAT.XSROBJECTAUTH from PUBLIC to reduce risk to the Organizations' data.

Audit:

Perform the following DB2 commands to obtain the value for this setting:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select rtrim(grantee) as grantee, controlauth, alterauth, deleteauth,
indexauth, insertauth, selectauth, updateauth, refauth  from sysibm.systabauth
where tcreator = 'SYSCAT' and ttname = 'XSROBJECTAUTH'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => revoke select on syscat.xsrmoduleauth from public
```


Impact:

As SYSCAT.XSROBJECTAUTH contains USAGE privileges on a particularXSR object for users, groups, and roles, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0021693.html?cp=SSEPGG_10.5.0%2F2-12-8-135(=en

6.29 Restrict Access to SYSCAT.AUTHORIZATIONIDS (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

AUTHORIZATIONIDS is an administrative view, using GRANT statements, retrieve users, roles, and groups of the currently connected server

Rationale:

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.AUTHORIZATIONIDS from PUBLIC to reduce risk to the Organizations' data.

Audit:

Perform the following DB2 commands to obtain the value for this setting:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select rtrim(grantee) as grantee, controlauth, alterauth, deleteauth,
indexauth, insertauth, selectauth, updateauth, refauth from sysibm.systabauth
where tcreator = 'SYSCAT' and ttname = 'AUTHORIZATIONIDS'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => revoke select on syscat.AUTHORIZATIONIDS from public
```

Impact:

As SYSCAT.AUTHORIZATIONIDS contains all granted privileges on a module for users, groups, and roles, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.rtn.doc/doc/r0021977.html?lang=en

6.30 Restrict Access to SYSIBMADM.OBJECTOWNERS (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

OBJECTOWNERS administrative view show the complete object ownership information for each authorization ID for USER owning a system catalog defined object from the connected database.

Rationale:

Any databases created without the RESTRICT option, automatically GRANTS the SELECT privilege to PUBLIC for views. Therefore, it is strongly recommended to explicitly REVOKE the SELECT privilege on the OBJECTOWNERS from PUBLIC to reduce risk to the Organizations' data.

Audit:

Perform the following DB2 commands to obtain the value for this setting:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select rtrim(grantee) as grantee, controlauth, alterauth, deleteauth,
indexauth, insertauth, selectauth, updateauth, refauth from sysibm.systabauth
where tcreator = 'SYSIBMADM' and ttname = 'OBJECTOWNERS'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => revoke select on SYSIBMADM.OBJECTOWNERS from public
```

Impact:

As OBJECTOWNERS administrative view show complete object ownership information for each authorization ID for USER owning a system catalog defined object from the connected database. Although REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access, it could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.rtn.doc/doc/r0021979.html?cp=SSEPGG_10.5.0%2F3-6-1-3-12-6(=en

6.31 Restrict Access to SYSIBMADM.PRIVILEGES (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

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 GRANTS 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 to the Organizations' data.

Audit:

Perform the following DB2 commands to obtain the value for this setting:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select rtrim(grantee) as grantee, controlauth, alterauth, deleteauth,
indexauth, insertauth, selectauth, updateauth, refauth from sysibm.systabauth
where tcreator = 'SYSIBMADM' and ttname = 'PRIVILEGES'
```

3. Review privileges granted to users, groups, and roles. If the output is BLANK, then it is considered a successful finding.

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => revoke select on SYSIBMADM.PRIVILEGES from public
```


Impact:

As SYSIBMADM.PRIVILEGES reveals all explicit privileges for all defined authorization IDs in the system catalog for the connected database, REVOKING the SELECT privilege from PUBLIC reduces the risk of unauthorized access but could have adverse affects on database operations.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.rtn.doc/doc/r0021978.html?cp=SSEPGG_10.5.0%2F3-6-1-3-12-7(=en

7 DB2 Authorities

This section provides guidance on securing the authorities that exist in the DB2 instance and database.

7.1 Secure SYSADM authority (Scored)

Profile Applicability:

- Level 2 - RDBMS
- Level 2 - Windows Host OS
- Level 2 - Linux Host OS

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 increase 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 from the DB2 command window:

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

3. Locate the sysadm_group value in the output and ensure the value is not NULL:

```
db2 => get database manager configuration db2 => … SYSADM group name (SYSADM_GROUP) = DB2ADM
```

Note: *sysadm_group is set to DB2ADM in the above output.*

4. Review the members of the sysadm_group on the operating system.

Unix:

```
cat /etc/group | grep <sysadm group name>
```

Windows:

1.
    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 to the SYSADM group.

1. Attach to the DB2 instance

```
db2 => attach to $DB2INSTANCE
```

2. Run the following command from the DB2 command window:

```
db2 => update database manager configuration using sysadm_group <sys adm group name>
```

Default Value:

The default value for sysadm_group is NULL.

7.2 Secure SYSCTRL authority (Scored)

Profile Applicability:

- Level 2 - 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 increase 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 from the DB2 command window:

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

3. Locate the sysctrl_group value in the output and ensure the value is not NULL:

```
db2 => get database manager configuration db2 => … SYSCTRL group name (SYSCTRL_GROUP)
= DB2CTRL
```

Note: *sysctrl_group is set to DB2CTRL in the above output.*

4. Review the members of the sysctrl_group on the operating system.

Unix:

```
cat /etc/group | grep <sysctrl group name>
```

Windows:

1.
    1. Run compmgmt.msc
    2. Click 'Local Users and Groups'
    3. Double click 'Groups'
    4. Double click <sysctrl group name>
    5. Review group members

Remediation:

Define a valid group name to the SYSCTRL group.

1. Attach to the DB2 instance

```
db2 => attach to $DB2INSTANCE
```

2. Run the following command from the DB2 command window:

```
db2 => update database manager configuration using sysctrl_group <sys control group
name>
```

Default Value:

The default value for sysctrl_group is NULL.

7.3 Secure SYSMAINT Authority (Scored)

Profile Applicability:

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

Description:

The sysmaint_group parameter defines the system administrator group that possess the system maintenance (SYSMAINT) authority. It is recommended that 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 increase 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 from the DB2 command window:

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

3. Locate the sysmaint_group value in the output and ensure the value is not NULL:

```
db2 => get database manager configuration db2 => … SYSMAINT group name
(SYSMAINT_GROUP) = DB2MAINT
```

Note: *sysmaint_group is set to DB2MAINT in the above output.*

4. Review the members of the sysmaint_group on the operating system.

Unix:

```
cat /etc/group | grep <sysmaint group name>
```

Windows:

1.
    1. Run compmgmt.msc
    2. Click 'Local Users and Groups'
    3. Double click 'Groups'
    4. Double click <sysmaint group name>
    5. Review group members

Remediation:

Define a valid group name to the SYSMAINT group.

1. Attach to the DB2 instance

```
db2 => attach to $DB2INSTANCE
```

2. Run the following command from the DB2 command window:

```
db2 => update database manager configuration using sysmaint_group <sys maintenance
group name>
```


Default Value:

The default value for sysmaint_group is NULL.

7.4 Secure SYSMON Authority (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The sysmon_group parameter defines the operating system groups with system monitor (SYSMON) authority. It is recommended that sysmon_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 increase 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 from the DB2 command window:

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

3. Locate the sysmon_group value in the output and ensure the value is not NULL:

```
db2 => get database manager configuration db2 => … SYSMON group name (SYSMON_GROUP) = DB2MON
```

Note: *sysmon_group is set to DB2MON in the above output.*

4. Review the members of the sysmon_group on the operating system.

Unix:

```
cat /etc/group | grep <sysmon group name>
```


Windows:

1.
    1. Run compmgmt.msc
    2. Click 'Local Users and Groups'
    3. Double click 'Groups'
    4. Double click <sysmon group name>
    5. Review group members

Remediation:

Define a valid group name to the SYSMON group.

1. Attach to the DB2 instance.

```
db2 => attach to $DB2INSTANCE
```

2. Run the following command from the DB2 command window:

```
db2 => update database manager configuration using sysmon_group <sys monintor group name>
```

Default Value:

The default value for sysmon_group is NULL.

7.5 Secure SECADM Authority (Scored)

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 increase risk.

Audit:

Perform the following DB2 commands to obtain the value for this setting:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
securityadmauth = 'Y'
```

3. Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => REVOKE SECADM ON DATABASE FROM USER <username>
```

Impact:

Only the security administrator has the ability to grant other users, groups, or roles the ACCESSCTRL, DATAACCESS, DBADM, and SECADM authorities.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0021054.html?lang=en

7.6 Secure DBADM Authority (Scored)

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 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 increase risk.

Audit:

Perform the following DB2 commands to obtain the value for this setting:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select distinct grantee, granteetype from syscat.dbauth where dbadmauth
= 'Y'
```

3. Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => REVOKE DBADM ON DATABASE FROM USER <username>
```

Impact:

The DBADM is an inherent DB2 authority.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0005521.html?lang=en

7.7 Secure SQLADM Authority (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - 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 subject to DBADM authority and can be granted by the SECADM authority.

Audit:

Run the following command from the DB2 command window:

```
select distinct grantee, granteetype from syscat.dbauth where sqladmauth = 'Y'
```

Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

Revoke SQLADM authority from any unauthorized users.

```
REVOKE SQLADM ON DATABASE FROM USER <username>
```

Impact:

SQLADM is an inherent DB2 authority.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0053931.html?lang=en

7.8 Secure DATAACCESS Authority (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - 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 subject to DBADM authority and can be granted by the SECADM authority.

Audit:

Run the following command from the DB2 command window:

```
select distinct grantee, granteetype from syscat.dbauth where dataaccessauth = 'Y'
```

Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

Revoke DATAACCESS authority from any unauthorized users.

```
 REVOKE DATAACCESS ON DATABASE FROM USER <username>
```

References:

1. https://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0005524.html?lang=en

7.9 Secure ACCESSCTRL Authority (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

ACCESSCTRL authority is the authority required to grant and revoke privileges on objects within a specific database. It has no inherent privilege to access data stored in tables, except the catalog tables and views. Some of these privileges include BINDADD, CONNECT, CREATETAB, CREATE_EXTERNAL_ROUTINE,LOAD, and QUIESCE_CONNECT

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. Some of these privileges include BINDADD, CONNECT, CREATETAB, and LOAD. ACCESSCTRL can be granted to users, groups, or roles, but not PUBLIC. ACCESSCTRL authority can only be granted by the SECADM authority.

Audit:

Run the following command from the DB2 command window:

```
select distinct grantee, granteetype from syscat.dbauth where accessctrlauth= 'Y'
```

Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

Revoke ACCESSCTRL authority from any unauthorized users.

```
REVOKE ACCESSCTRL ON DATABASE FROM USER <username>
```


References:

1. https://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0053933.html?lang=en

7.10 Secure WLMADM authority (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - 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 creation, alteration, dropping, commenting, granting, and revoking access to workload objects for a database.

Audit:

1. Run the following command from the DB2 command window:

```
select grantee, wlmadmauth from syscat.dbauth
```

2. Determine if the grantee(s) and granteetype(s) are correctly set

Remediation:

1. Revoke any user who should NOT have WLMADM authority:

```
REVOKE WLMADM ON DATABASE FROM USER <username>
```


Impact:

The SECADM authority can grant the WLMADM authority or a user who possesses ACCESSCTRL authority. The WLMADM authority is an inherent DB2 authority.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0053932.html?lang=en

7.11 Secure CREATAB Authority (Scored)

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 createtab 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
creatabauth = 'Y'
```

3. Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => REVOKE CREATAB ON DATABASE FROM USER <username>
```

Impact:

The CREATAB (create table) role grants the authority to a user to create tables within a specific database and should be properly restricted to reduce the risk of unauthorized use.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0054269.html?lang=en

7.12 Secure BINDADD Authority (Scored)

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 increase risk.

Audit:

Perform the following DB2 commands to obtain the value for this setting:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
bindaddauth = 'Y'
```

3. Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => REVOKE BINDADD ON DATABASE FROM USER <username>
```

Impact:

The BINDADD (bind application) role grants the authority to a user to create packages on a specific database, the BINDADD authority should be restricted to prevent unauthorized use.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0005524.html?lang=en

7.13 Secure CONNECT Authority (Scored)

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:

The CONNECT role grants the authority to a user to connect to mainframe and mid-range databases from Windows, Unix, and Linux operating systems and 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
connectauth = 'Y'
```

3. Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => REVOKE CONNECT ON DATABASE FROM USER <username>
```

Impact:

DB2 CONNECT enables connectivity to other databases and should be appropriately restricted to reduce the risk of unauthorized access.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.qb.dbconn.doc/doc/r0059046.html?cp=SSEPGG_10.5.0%2F6(=en

7.14 Secure LOAD Authority (Scored)

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:

Review all users that have access to this authority.

Audit:

Perform the following DB2 commands to obtain the value for this setting:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select distinct grantee, granteetype from syscat.dbauth where loadauth =
'Y'
```

3. Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => REVOKE LOAD ON DATABASE FROM USER <username>
```

Impact:

As the LOAD authority grants Users' the privilege to load data into tables on the database. These users should be regularly reviewed and approved to prevent unauthorized access or changes to the Organizations' data.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0005522.html?lang=en

7.15 Secure EXTERNALROUTINE Authority (Scored)

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:

Because the EXTERNALROUTINE authority grants a user the privilege to create user-defined functions and procedures in a database, 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
externalroutineauth = 'Y'
```

3. Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => REVOKE CREATE_EXTERNAL_ROUTINE ON DATABASE FROM USER <username>
```


Impact:

As the EXTERNALROUTINE authority grants a user the privilege to create user-defined functions and procedures in a specific database, which is a significant risk to the Organizations' data, the externalroutine authority should be granted to authorized users only.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.apdv.routines.doc/doc/c0009198.html?lang=en

7.16 Secure QUIESCECONNECT Authority (Scored)

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:

The QUIESCECONNECT role grants the authority to a user to access a database even in the quiesced state. 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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
quiesceconnectauth = 'Y'
```

3. Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

this permission from any unauthorized users.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => REVOKE QUIESCE_CONNECT ON DATABASE FROM USER <username>
```

Impact:

Granting the QUIESCECONNECT authority gives a user the privilege to access a database while in the quiesced state, presenting a significant risk to the Organizations' data.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.apdv.api.doc/doc/r0009331.html?lang=en

8 DB2 Roles

Roles simplify the administration and management of privileges by offering an equivalent capability as groups but without the same restrictions. A role is a database object that groups together one or more privileges and can be assigned to users, groups, PUBLIC, or other roles by using a GRANT statement. All the roles assigned to a user are enabled when that user establishes a connection, so all privileges and authorities granted to roles are taken into account when a user connects. Roles cannot be explicitly enabled or disabled.

8.1 Review Roles (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - 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:

Attach to a DB2 Instance:

```
db2 => attach to $DB2INSTANCE
```

Connect to Db2 database:

```
db2 => connect to $DBNAME
```

Run the following:

```
db2 => select rolename from syscat.roleauth where grantortype <> 'S' group by rolename
```

Remediation:

To remove a role from the database:

Attach to a DB2 Instance:

```
db2 => attach to $DB2INSTANCE
```

Connect to Db2 database:

```
db2 => connect to $DBNAME
```

Run the following:

```
db2 => drop role <role name>
```

References:

1. https://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0050531.html

8.2 Review Role Members (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

Rationale:

Having roles that have been granted specific privileges, then assigning users to the roles, are usually considered the best way to grant application access. As having privileges granted to individual users can be more difficult to track and maintain against unauthorized access, users should be assigned to database roles according to the needs of the business.

Audit:

Attach to a DB2 Instance:

```
db2 => attach to $DB2INSTANCE
```

Connect to Db2 database:

```
db2 => connect to $DBNAME
```

Run the following:

```
db2 => select rolename,grantee from syscat.roleauth where grantortype <> 'S' group by
rolename, grantee
```

Remediation:

To remove a role member from a particular role:

Attach to a DB2 Instance:

```
db2 => attach to $DB2INSTANCE
```

Connect to Db2 database:

```
db2 => connect to $DBNAME
```

Run the following:

```
db2 => revoke role <role name> from <role member>
```


References:

1. https://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0050531.html

8.3 Nested Roles (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

Nested groups are convenient when assigning permissions on certain objects. 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:

The user-defined roles in DB2 can be nested in the same fashion as Windows security groups--a nested group has both its directly assigned permissions as well as the assigned group permissions. As tracking multiple levels of permissions can result in unauthorized access to data resources, this capability should restricted according to the needs of the business.

Audit:

Attach to DB2 Instance:

```
db2 => attach to $DB2INSTANCE
```

Connect to database:

```
db2 => connect to $DBNAME
```

Run the following:

```
db2 => select grantee, rolename from syscat.roleauth where grantee in (select rolename from syscat.roles)
```

NOTE: If value is blank, this would be considered passing.

Remediation:

To remove a nested role, perform the following:

Attach to DB2 Instance:

```
db2 => attach to $DB2INSTANCE
```

Connect to database:

```
db2 => connect to $DBNAME
```

Run the following:

```
db2 => revoke role <role name> from role <role>
```

8.4 Review Roles granted to PUBLIC (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - 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:

The roles granted directly to PUBLIC are those available to all users. 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:

Attach to a DB2 Instance:

```
db2 => attach to $DB2INSTANCE
```

Connect to Db2 database:

```
db2 => connect to $DBNAME
```

Run the following:

```
db2 => select grantee, rolename from syscat.roleauth where grantee = 'PUBLIC'
```

NOTE: If value returned in blank, that is considered a passable finding.

Remediation:

To remove a role member from a particular role:

Attach to a DB2 Instance:

```
db2 => attach to $DB2INSTANCE
```

Connect to Db2 database:

```
db2 => connect to $DBNAME
```

Run the following:

```
db2 => revoke role <role name> from PUBLIC
```

References:

1. https://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0050531.html

8.5 Review Role Grantees with WITH ADMIN OPTION (Scored)

Profile Applicability:

- Level 2 - 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:

Attach to DB2 Instance:

```
db2 => attach to $DB2INSTANCE
```

Connect to database:

```
db2 => connect to $DBNAME
```

Perform the following query:

```
db2 => select rolename, grantee, admin from syscat.roleauth where grantortype <> 'S'
and admin = 'Y'
```

NOTE: If value returned in blank, that is considered a passable finding.

Remediation:

Attach to DB2 Instance:


```
db2 => attach to $DB2INSTANCE
```

Connect to database:

```
db2 => connect to $DBNAME
```

Perform the following query:

```
db2=> revoke admin option for role <role name> from user <user name>
```

9 General Policy and Procedures

[This space intentionally left blank]

9.1 Start and Stop DB2 Instance (Not Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 1 - Linux Host OS

Description:

The DB2 instance manages the database environment and sets the configuration parameters. It is recommended that only administrators are allowed to start and stop the DB2 instance.

Rationale:

Only privileged users should have access to start and stop the DB2 instance. This will ensure that the DB2 instance is controlled by authorized administrators.

Audit:

On MS Windows: Go to Start, then to the Run option. Type in `services.msc` in the command prompt. Locate the DB2 service and identify the users/groups that can start and stop the service.

On Unix: Identify the members of the local DB2 admin group that have access to stop and start the DB2 instance.

Remediation:

Revoke access from any unnecessary users.

1. Connect to the host
2. Review users and groups that have access to start and stop the DB2 instance

9.2 Remove Unused Schemas (Not Scored)

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 $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select schemaname from syscat.schemata
```

3. Review the list of schemas

Remediation:

Revoke access from any unnecessary users.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => drop scheme <scheme name> restrict
```

3. Review unused schemas and remove if necessary

9.3 Review System Tablespaces (Scored)

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 stored system data.

Rationale:

Do not install any user data in the following system tablespaces: SYSCATSPACE and SYSTOOLSPACE.

Audit:

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window:

```
db2 => select tabschema,tabname,tbspace from syscat.tables where tabschema not
in ('ADMINISTRATOR','SYSIBM','SYSTOOLS') and tbspace in
('SYSCATSPACE','SYSTOOLSPACE','SYSTOOLSTMPSPACE','TEMPSPACE')
```

3. Review the list of system tablespaces. If the output is BLANK, that is considered a successful finding.

Remediation:

Revoke access from any unnecessary users.

1. Connect to the DB2 database.

```
db2 => connect to $DB2DATABASE user $USERNAME using $PASSWORD
```

2. Review unused users and user objects that are stored in the system tablespaces

9.4 Remove Default Databases (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

A DB2 Instance may come installed with default databases. It is recommended that the SAMPLE database be removed.

Rationale:

Removing unused, well-known, databases will reduce the attack surface of the system.

Audit:

Perform the following DB2 commands to obtain the list of databases:

1. Attach to the DB2 instance

```
db2 => attach to $DB2INSTANCE
```

2. Run the following command from the DB2 command window:

```
db2 => list database directory
```

3. Locate this value in the output:

```
db2 =>
Database 3 entry:

 Database alias                       = SAMPLE
 Database name                        = SAMPLE
 Local database directory             = C:
 Database release level               = c.00
 Comment = Directory entry type       = Indirect
 Catalog database partition number    = 0
 Alternate server hostname            =
```

4. Review the output above and identify the SAMPLE database. If there is no SAMPLE database, then it is considered a successful finding.

Remediation:

Drop unused sample databases

1. Connect to the DB2 instance
2. Run the following command from the DB2 command window:

```
db2 => drop database sample
```

9.5 Enable SSL communication with LDAP server (Scored)

Profile Applicability:

- Level 1 - Windows Host OS

Description:

The communication layer between a DB2 instance and the LDAP server should be encrypted. It is recommended that the `ENABLE_SSL` parameter in the `IBMLDAPSecurity.ini` file be set to `TRUE`.

Rationale:

Enabling SSL will help ensure the confidentiality of authentication credentials and other information that is sent to and from the DB2 instance and the LDAP server.

Note: The file is located under `INSTANCE_HOME/sqllib/cfg/`, for Unix; and `%DB2PATH%\cfg\`, for MS Windows.

Audit:

Perform the following commands to obtain the parameter setting:

1. Connect to the DB2 host
2. Edit the `IBMLDAPSecurity.ini` file
3. Verify the existence of this parameter:

```
ENABLE_SSL = TRUE
```

Note: The default setting is the omission of this parameter.

Remediation:

Verify the parameter

1. Connect to the DB2 host
2. Edit the `IBMLDAPSecurity.ini` file
3. Add or modify the file to include the following parameter:

```
ENABLE_SSL = TRUE
```

9.6 Secure the permission of the IBMLDAPSecurity.ini file (Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 1 - Linux Host OS

Description:

The IBMLDAPSecurity.ini file contains the IBM LDAP security plug-in configurations.

Rationale:

Recommended value is ready-only (RO) to Everyone/Other/Users/Domain Users. This will ensure that the parameter file is protected.

Note: the file is located under `INSTANCE_HOME/sqllib/cfg/`, for Unix; and `%DB2PATH%\cfg\`, for MS Windows.

Audit:

Perform the following DB2 commands to obtain the value for this setting:
For MS Windows:

1. Connect to the DB2 host
2. Right-click over the file directory
3. Choose *Properties*
4. Select the *Security* tab
5. Review access from all non-administrator accounts

For Unix:

1. Connect to the DB2 host
2. Change to the file directory
3. Change the permission level of the directory

```
OS => ls -al
```


Remediation:

For MS Windows:

1. Connect to the DB2 host
2. Right-click over the file directory
3. Choose *Properties*
4. Select the *Security* tab
5. Select all non-administrator accounts and revoke the *Full Control* authority

For Unix:

1. Connect to the DB2 host
2. Change to the file directory
3. Change the permission level of the directory

```
OS => chmod -R 740
```

9.7 Secure the permission of the SSLconfig.ini file (Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 1 - Linux Host OS

Description:

The SSLconfig.ini file contains the SSL configuration parameters for the DB2 instance, including the password for KeyStore.

Rationale:

Recommended value is ready-only (RO) to Everyone/Other/Users/Domain Users. This will ensure that the parameter file is protected.

Note: the file is located under `INSTANCE_HOME/cfg/`, for Unix; and `%INSTHOME%\`, for MS Windows. Only the instance owner should have access to this file.

Audit:

Perform the following DB2 commands to obtain the value for this setting:
For MS Windows:

1. Connect to the DB2 host
2. Right-click over the file directory
3. Choose *Properties*
4. Select the *Security* tab
5. Review access from all non-administrator accounts

For Unix:

1. Connect to the DB2 host
2. Change to the file directory
3. Change the permission level of the directory

```
OS => ls -al
```

Remediation:

For MS Windows:

1. Connect to the DB2 host
2. Right-click over the file directory
3. Choose *Properties*
4. Select the *Security* tab

5. Select all non-administrator accounts and revoke the *Full Control* authority

For Unix:

1. Connect to the DB2 host
2. Change to the file directory
3. Change the permission level of the directory

```
OS => chmod -R 740
```

Control		Set Correctly	
		Yes	No
1	**Installation and Patches**		
1.1	Install the latest Fixpaks (Not Scored)	☐	☐
1.2	Use IP address rather than hostname (Scored)	☐	☐
1.3	Leverage a least privilege principle (Not Scored)	☐	☐
1.4	Use non-standard account names (Not Scored)	☐	☐
2	**DB2 Directory and File Permissions**		
2.1	Secure DB2 Runtime Library (Scored)	☐	☐
2.2	Secure all database containers (Scored)	☐	☐
2.3	Set umask value for DB2 admin user .profile file (Scored)	☐	☐
3	**DB2 Configurations**		
3.1	**DB2 Instance Parameter Settings**		
3.1.1	Enable audit buffer (Scored)	☐	☐
3.1.2	Encrypt user data across the network (Scored)	☐	☐
3.1.3	Require explicit authorization for cataloging (Scored)	☐	☐
3.1.4	Disable data links support (Scored)	☐	☐
3.1.5	Secure default database location (Scored)	☐	☐
3.1.6	Secure permission of default database location (Scored)	☐	☐
3.1.7	Set diagnostic logging to capture errors and warnings (Scored)	☐	☐
3.1.8	Secure all diagnostic logs (Scored)	☐	☐
3.1.9	Require instance name for discovery requests (Scored)	☐	☐
3.1.10	Disable instance discoverability (Scored)	☐	☐
3.1.11	Authenticate federated users at the instance level (Scored)	☐	☐
3.1.12	Set maximum connection limits (Scored)	☐	☐
3.1.13	Set administrative notification level (Scored)	☐	☐
3.1.14	Enable server-based authentication (Scored)	☐	☐
3.1.15	Set failed archive retry delay (Scored)	☐	☐
3.1.16	Auto-restart after abnormal termination (Scored)	☐	☐
3.1.17	Disable database discovery (Scored)	☐	☐
3.1.18	Establish secure archive log location (Scored)	☐	☐

3.1.19	Secure permission of the primary archive log location (Scored)	☐	☐
3.1.20	Establish secure secondary archive location (Scored)	☐	☐
3.1.21	Secure permission of the secondary archive location (Scored)	☐	☐
3.1.22	Establish secure tertiary archive log location (Scored)	☐	☐
3.1.23	Secure permission of the tertiary archive location (Scored)	☐	☐
3.1.24	Establish secure log mirror location (Scored)	☐	☐
3.1.25	Establish retention set size for backups (Scored)	☐	☐
3.1.26	Set archive log failover retry limit (Scored)	☐	☐
3.2	**Database Manager Configuration parameters**		
3.2.1	Priority of agents - agentpri (Scored)	☐	☐
3.2.2	Application support layer heap - aslheapsz (Scored)	☐	☐
3.2.3	Fast communication manager - fcm_parallelism (Scored)	☐	☐
3.2.4	Intrapartition parallelism - intra_parallel (Scored)	☐	☐
3.2.5	Maximum query degree of parallelism - max_querydegree (Scored)	☐	☐
3.2.6	Agent pool size - num_poolagents (Scored)	☐	☐
3.2.7	Sort heap threshold - sheapthres (Scored)	☐	☐
3.2.8	Instance impact policy configuration - util_impact_lim (Scored)	☐	☐
3.2.9	Client I/O block size configuration parameter - rqrioblk (Scored)	☐	☐
3.2.10	TCP/IP service name - svcename (Scored)	☐	☐
3.2.11	SSL service name - ssl_svcename (Scored)	☐	☐
3.2.12	Maximum Java interpreter heap size - java_heap_sz (Scored)	☐	☐
3.2.13	Authentication type for incoming connections at the server - srvcon_auth (Scored)	☐	☐
4	**Row and Column Access Control (RCAC)**		
4.1	Review Organizations' Policies against DB2 RCAC Policies (Not Scored)	☐	☐
4.2	Secure SECADM Authority (Scored)	☐	☐
4.3	Review Users, Groups, and Roles (Not Scored)	☐	☐
4.4	Review Row Permission logic according to policy (Not Scored)	☐	☐
4.5	Review Column Mask logic according to policy (Not Scored)	☐	☐
5	**Database Maintenance**		
5.1	Enable Backup Redundancy (Not Scored)	☐	☐
5.2	Protecting Backups (Not Scored)	☐	☐
5.3	Enable Database Maintenance (Scored)	☐	☐
6	**Securing Database Objects**		
6.1	Restrict Access to SYSCAT.AUDITPOLICIES (Scored)	☐	☐
6.2	Restrict Access to SYSCAT.AUDITUSE (Scored)	☐	☐
6.3	Restrict Access to SYSCAT.DBAUTH (Scored)	☐	☐
6.4	Restrict Access to SYSCAT.COLAUTH (Scored)	☐	☐
6.5	Restrict Access to SYSCAT.EVENTS (Scored)	☐	☐
6.6	Restrict Access to SYSCAT.EVENTTABLES (Scored)	☐	☐
6.7	Restrict Access to SYSCAT.ROUTINES (Scored)	☐	☐

6.8	Restrict Access to SYSCAT.INDEXAUTH (Scored)	☐	☐
6.9	Restrict Access to SYSCAT.PACKAGEAUTH (Scored)	☐	☐
6.10	Restrict Access to SYSCAT.PACKAGES (Scored)	☐	☐
6.11	Restrict Access to SYSCAT.PASSTHRUAUTH (Scored)	☐	☐
6.12	Restrict Access to SYSCAT.SECURITYPOLICIES (Scored)	☐	☐
6.13	Restrict Access to SYSCAT.SECURITYPOLICYEXEMPTIONS (Scored)	☐	☐
6.14	Restrict Access to SYSCAT.SURROGATEAUTHIDS (Scored)	☐	☐
6.15	Restrict Access to SYSCAT.ROLEAUTH (Scored)	☐	☐
6.16	Restrict Access to SYSCAT.ROLES (Scored)	☐	☐
6.17	Restrict Access to SYSCAT.ROUTINEAUTH (Scored)	☐	☐
6.18	Restrict Access to SYSCAT.SCHEMAAUTH (Scored)	☐	☐
6.19	Restrict Access to SYSCAT.SCHEMATA (Scored)	☐	☐
6.20	Restrict Access to SYSCAT.SEQUENCEAUTH (Scored)	☐	☐
6.21	Restrict Access to SYSCAT.STATEMENTS (Scored)	☐	☐
6.22	Restrict Access to SYSCAT.TABAUTH (Scored)	☐	☐
6.23	Restrict Access to SYSCAT.TBSPACEAUTH (Scored)	☐	☐
6.24	Restrict Access to Tablespaces (Scored)	☐	☐
6.25	Restrict Access to SYSCAT.MODULEAUTH (Scored)	☐	☐
6.26	Restrict Access to SYSCAT.VARIABLEAUTH (Scored)	☐	☐
6.27	Restrict Access to SYSCAT.WORKLOADAUTH (Scored)	☐	☐
6.28	Restrict Access to SYSCAT.XSROBJECTAUTH (Scored)	☐	☐
6.29	Restrict Access to SYSCAT.AUTHORIZATIONIDS (Scored)	☐	☐
6.30	Restrict Access to SYSIBMADM.OBJECTOWNERS (Scored)	☐	☐
6.31	Restrict Access to SYSIBMADM.PRIVILEGES (Scored)	☐	☐
7	**DB2 Authorities**		
7.1	Secure SYSADM authority (Scored)	☐	☐
7.2	Secure SYSCTRL authority (Scored)	☐	☐
7.3	Secure SYSMAINT Authority (Scored)	☐	☐
7.4	Secure SYSMON Authority (Scored)	☐	☐
7.5	Secure SECADM Authority (Scored)	☐	☐
7.6	Secure DBADM Authority (Scored)	☐	☐
7.7	Secure SQLADM Authority (Scored)	☐	☐
7.8	Secure DATAACCESS Authority (Scored)	☐	☐
7.9	Secure ACCESSCTRL Authority (Scored)	☐	☐
7.10	Secure WLMADM authority (Scored)	☐	☐
7.11	Secure CREATAB Authority (Scored)	☐	☐
7.12	Secure BINDADD Authority (Scored)	☐	☐
7.13	Secure CONNECT Authority (Scored)	☐	☐
7.14	Secure LOAD Authority (Scored)	☐	☐
7.15	Secure EXTERNALROUTINE Authority (Scored)	☐	☐
7.16	Secure QUIESCECONNECT Authority (Scored)	☐	☐
8	**DB2 Roles**		

8.1	Review Roles (Scored)	☐	☐
8.2	Review Role Members (Scored)	☐	☐
8.3	Nested Roles (Scored)	☐	☐
8.4	Review Roles granted to PUBLIC (Scored)	☐	☐
8.5	Review Role Grantees with WITH ADMIN OPTION (Scored)	☐	☐
9	**General Policy and Procedures**		
9.1	Start and Stop DB2 Instance (Not Scored)	☐	☐
9.2	Remove Unused Schemas (Not Scored)	☐	☐
9.3	Review System Tablespaces (Scored)	☐	☐
9.4	Remove Default Databases (Scored)	☐	☐
9.5	Enable SSL communication with LDAP server (Scored)	☐	☐
9.6	Secure the permission of the IBMLDAPSecurity.ini file (Scored)	☐	☐
9.7	Secure the permission of the SSLconfig.ini file (Scored)	☐	☐

Appendix: Change History

Date	Version	Changes for this version