CIS IBM DB2 9 Benchmark

v3.0.1 - 03-30-2017

Table of Contents

Overview

This document, Security Configuration Benchmark for IBM DB2, provides prescriptive guidance for establishing a secure configuration posture for DB2 versions 9.7 or 9.8 running on Linux and Windows. This guide was tested against DB2 version 9.7 and 9.8 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 of subject matter experts. Consensus participants provide perspective from a diverse set of backgrounds including consulting, software development, audit and compliance, security research, operations, government, and legal.

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

Typographical Conventions

The following typographical conventions are used throughout this guide:

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

Scoring Information

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

Scored

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

Not Scored

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

Profile Definitions

The following configuration profiles are defined by this Benchmark:

- **Level 1 - RDBMS**

  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:

Contributor
Chris Bielinski – *Trustwave*
Adam Montville – *Center for Internet Security*

Editor
Tim Harrison CISSP, ICP – *Center for Internet Security*
Karen Scarfone – *Scarfone Cybersecurity*

Recommendations

1 Installation and Patches

1.1 Install the latest fix packs (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

Periodically, IBM releases fix packs 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 fix pack will help protect the database from known vulnerabilities as well as reduce downtime that may otherwise result from functional defects.

Audit:

Perform the following DB2 commands to obtain the version:

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 fix pack as offered from IBM.

References:

1. http://www.ibm.com/products/finder/us/finders?Ne=5000000&finderN=1000188&pg=ddfinder&C1=5000002&C2=5000049

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 an attacker. For example, hostnames for DB2 instances often contain the DB2 version number, host type, or operating system type.

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

Linux:

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:

To reconfigure the connection string, launch the DB2 Configuration Assistant and change the connection's Hostname value from a text name to an IP address.

1.3 Leverage the 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 this service have the least privileges possible. Furthermore, it is advisable to have the DB2 service executed using the DB2 instance owner and monitor such accounts for unauthorized access to the sensitive data.

Rationale:

Leveraging a least privilege account for the DB2 service will reduce an attacker's ability to compromise the host operating system should the DB2 service process become compromised.

Audit:

Review all accounts that have access to the DB2 database service to ensure least privilege is applied.

Remediation:

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

1.4 Use non-default account names (Scored)

Profile Applicability:

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

Description:

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

Rationale:

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

Audit:

For Windows:

1. Right-click over the *`%DB2PATH%`* and select Properties from the menu.
2. Go to the Security tab and review all usernames that have access to this directory.

For Linux:

- Run *`ls -al $DB2PATH`* and review all usernames that have access to this directory.

Remediation:

For Windows:

1. Right-click over the `%DB2PATH%` and select Properties from the menu.
2. Go to the Security tab and re-assign all the user accounts with well-known default names to use non-default names.

For Linux, perform the following command:

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

Notes:

Review the impact of changing the usernames before performing this global change.

2 DB2 Directory and File Permissions

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

2.1 Secure the 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 needs to be secured so it grants only the necessary access to authorized users and administrators.

Rationale:

The DB2 runtime is comprised of files that are executed as part of the DB2 service. If these resources are not secured, an attacker may alter them to execute arbitrary code.

Audit:

Perform the following to obtain the value for this setting:

For Windows:

1. Connect to the DB2 host
2. Right-click on the `%DB2PATH%\sqllib` directory
3. Choose *Properties*
4. Select the *Security* tab
5. Review the permissions for all DB administrator accounts and all other accounts

For Linux:

1. Connect to the DB2 host
2. Change to the `$DB2PATH/sqllib` directory
3. Check the permission level of the directory

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

Remediation:

For Windows:

1. Connect to the DB2 host
2. Right-click on the `%DB2PATH%\sqllib` directory
3. Choose *Properties*
4. Select the *Security* tab
5. Select all DB administrator accounts and grant them the Full Control authority
6. Select all non-administrator accounts and revoke all privileges other than Read and Execute

For Linux:

1. Connect to the DB2 host
2. Change to the `$DB2PATH/sqllib` directory
3. Change the permission level of the directory to this recommended value:

```
OS => chmod -R 750
```

Default Value:

Linux: `$DB2PATH/sqllib` is owned by the DB2 administrator with read, write, and execute access.

MS Windows: `%DB2PATH%\sqllib` owned by the DB2 administrator with read, write, and execute access.

2.2 Secure the database container directory (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. Also, allowing excessive access to the containers may help an attacker to gain access to their contents. Therefore, secure the location(s) of the containers by restricting the access and ownership. Allow only the instance owner to have access to the tablespace containers.

Audit:

Review all users that have access to the directory of the containers to ensure only DB2 administrators have full access. All other users should have read-only access.

Remediation:

Set the privileges for the directory of the containers. The recommended values are that only DB2 administrators have full access, and all other users have read-only access.

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

Profile Applicability:

- Level 1 - Linux Host OS

Description:

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

Rationale:

The `umask` value should be set to `022` for the owner of the DB2 software at all times, including 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

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` in the above output.

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
```

Default Value:

The default value for `audit_buz_sz` is zero (`0`).

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=%20audit_buf_sz

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 the confidentiality of authentication credentials and user data as they traverse the network 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
```

Ensure that `AUTHENTICATION` is set to `DATA_ENCRYPT` in the output.

Remediation:

The suggested value is `DATA_ENCRYPT` so that authentication occurs at the server. To set this:

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
```

Default Value:

The default value for `AUTHENTICATION` is `SERVER`.

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=%2Fcom.ibm.db2.udb.admin.doc%2Fdoc%2Fr0000294.htm

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 `catalog_noauth` parameter be set to `NO`.

Rationale:

Cataloging a database is the process of registering a database from a remote client to allow remote call and access. Setting `catalog-noauth` to `YES` bypasses all permission checks and allows anyone to catalog and uncatalog databases.

Audit:

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

1. Attach to the DB2 instance:

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

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

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

3. Locate the value of `CATALOG_NOAUTH` in the output and ensure that it is set to NO:

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

Remediation:

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

1. Attach to the DB2 instance:

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

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

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

Default Value:

The default value for `CATALOG_NOAUTH` is `NO`.

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=catalog_noauth
2. http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=%2Fcom.ibm.db2.udb.admin.doc%2Fdoc%2Fr0000143.htm

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 because this will reduce the attack surface for 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 the value of `datalinks` value in the output:

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

Ensure that `DATALINKS` is set to `NO` in the 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
```

Default Value:

The default value for `datalinks` is `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 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 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
```

3.1.6 Secure permissions for default database file path (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- 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 permissions for this directory be set to full access for DB2 administrators and read and execute access only for all other accounts. It is also recommended that this directory be owned by the DB2 Administrator.

Rationale:

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

Audit:

For Windows and Linux:

1. Attach to the DB2 instance.

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

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

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

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

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

Additional steps for Windows:

1. Connect to the DB2 host
2. Right-click over the directory used for the default file path
3. Choose *Properties*
4. Select the *Security* tab
5. Review and verify the privileges for all accounts.
6. Review and verify that the DB2 Administrator is the owner of the directory.

Additional steps for Linux:

1. Connect to the DB2 host
2. Change to the directory used as the default file path
3. Review and verify the permissions for the directory for all users; also ensure that the DB2 Administrator is the owner.

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

Remediation:

For Windows and Linux:

1. Attach to the DB2 instance.
2. Run the following command from the DB2 command window to change the default file path, if necessary:

Additional steps for Windows:

1. Connect to the DB2 host
2. Right-click over the directory used as the default file path
3. Choose *Properties*
4. Select the *Security* tab
5. Assign ownership of the directory to the DB2 Administrator
6. Grant all DB administrator accounts the *Full Control* authority
7. Grant only read and execute privileges to all other users (revoke all other privileges)

Additional steps for Linux:

1. Connect to the DB2 host
2. Change to the directory used as the default file path
3. Assign the DB2 Administrator to be the owner of the directory using the `chown` command
4. Change the permissions for the directory

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

Default Value:

The default value for this directory is read and write access for non-administrator accounts.

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 `diaglevel` setting is `3`, but any value greater than `3` is also acceptable. A value of at least `3` will allow the DB2 instance to capture all errors and warnings that occur on the system.

Audit:

Perform the following 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 that `DIAGLEVEL` is set to at least `3` in the 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 diaglevel 3
```

Default Value:

The default value for `diaglevel` is `3`.

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=%2Fcom.ibm.db2.udb.admin.doc%2Fdoc%2Fr0000298.htm

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. The directory at this location should be secured so that users have read and execute privileges only (no write privileges). All DB2 administrators should have full access to the directory.

Rationale:

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

Audit:

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>
```

Additional steps for Windows:

1. Connect to the DB2 host
2. Right-click over the diagnostic log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Review the access for all accounts

Additional steps for Linux:

1. Connect to the DB2 host
2. Change to the diagnostic log directory
3. Review the permissions of the directory

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

Remediation:

For Windows and Linux, to change the directory for the diagnostic logs:

1. Attach to the DB2 instance.

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

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

```
db2 => update database manager configuration using diagpath
```

Additional steps for Windows:

1. Connect to the DB2 host
2. Right-click over the diagnostic log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Grant the *Full Control* authority to all DB2 administrator accounts
6. Grant only read and execute privileges to all other accounts (revoke any other privileges)

Additional steps for Linux:

1. Connect to the DB2 host
2. Change to the diagnostic log directory
3. Change the permissions of the directory

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

Default Value:

The default value for `diagpath` is `NULL`.

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof= diagpath

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 (`discover` parameter value of `known`).

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 `DIAGPATH` value in the output:

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

Ensure that `DISCOVER` is set to `KNOWN` in the 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:

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.

Default Value:

The default value for `discover` is `SEARCH`.

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=discover

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` value in the output:

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

Ensure that `DISCOVER_INST` is set to `DISABLE` in the 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
```

Default Value:

The default value for `discover_inst` is `ENABLE`.

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=discover_inst

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:

Setting `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
```

Verify that `FED_NOAUTH` is set to `NO` in the 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
```

Default Value:

The default value for `FED_NOAUTH` is `NO`.

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=fed_noauth

3.1.12 Enable instance health monitoring (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `health_mon` parameter allows you to specify whether you want to monitor the health of the instance, the databases, and the corresponding database objects. It is recommended that the `health_mon` parameter be set to `on`.

Rationale:

Enabling instance health monitoring will assist in ensuring instance data availability and integrity.

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 `HEALTH_MON` value in the output:

```
db2 => get database manager configuration
db2 => …
       Monitor health of instance and databases   (HEALTH_MON) = ON
```

Verify that `HEALTH_MON` is set to `ON` in the 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 health_mon on
```

Default Value:

The default value for `HEALTH_MON` is `ON`.

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=health_mon

3.1.13 Retain fenced model processes (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `keepfenced` parameter indicates whether or not external user-defined functions or stored procedures will reuse a DB2 process after each subsequent call. It is recommended that this parameter be set to `NO`.

Rationale:

All routines that were executed by the DB2 should be terminated when the instance is stopped.

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 `KEEPFENCED` value in the output:

```
db2 => get database manager configuration
db2 => …
       Keep fenced process                        (KEEPFENCED) = NO
```

Verify that `KEEPFENCED` is set to `NO` in the output.

Remediation:

Note: this procedure 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 keepfenced no
```

3. Restart the DB2 instance.

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

Default Value:

The default value for `KEEPFENCED` is `YES`.

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=keepfenced

3.1.14 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 maximum number of agents needed to perform 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 ensure that the lock limit is not reached, which would result in lock escalation issues.

Rationale:

By default, DB2 allows an unlimited number of users to access the DB2 instance. In addition to giving access to the DB2 instance to authorized users only, it is recommended to set a limit to the number of users allowed to access a DB2 instance. This helps prevent denial of service conditions should an authorized process malfunction and attempt a large number of simultaneous connections.

Audit:

Perform the following DB2 commands to obtain the values for the `max_connections` and `max_coordagents` parameters:

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 instance.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

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

```
db2 => get database manager 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 `AUTOMATIC`. Allowable range is 1 to 64,000, or -1 for unlimited. The recommended value is 100. The following command will set `max_coordagents` to 100, as well as set `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
```

3. If maxappls is NOT less than max_coordagents, 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_coordagents is AUTOMATIC. The default value for maxappls is AUTOMATIC.

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=max_connections
2. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=max_coordagents
3. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=maxappls

3.1.15 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` and `2`, will log all fatal errors, failing services, system integrity, as well as system health.

Rationale:

The system should be monitoring all Health Monitor alarms, warnings, and attentions. This may give an indication of any malicious usage on the DB2 instance.

Audit:

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

1. Attach to the DB2 instance.

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

2. Run the following command 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 for the `notifylevel` parameter is `3`.

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=notifylevel

3.1.16 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:

This parameter will take precedence over and override the authentication level. Authentication should be set on the server side.

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
```

Impact:

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.

Default Value:

The default value for `SRVCON_AUTH` is `NULL`.

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=srvcon_auth

3.1.17 Reserve the desired port number or name for incoming connection requests (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

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

Rationale:

When the database server is started, a port number or name is required to listen for incoming connection requests.

On Linux systems, the services file is found at: `/etc/services`

Audit:

Run the following command to determine if the `svcename` parameter value is correctly set and is not the default port (50000).

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

Remediation:

Run the following command to set the `svcename` parameter value.

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

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

Notes:

If `DB2COMM` contains both TCP/IP and SSL, the port specified by `svcename` must not be the same as `ssl_svcename`. Otherwise, the instance starts up without either TCP/IP or SSL protocol support.

3.1.18 Reserve the desired port number or name for incoming SSL connections (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

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

On Linux operating systems, the `ssl_svcename` file is located in: `/etc/services`

Rationale:

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

Audit:

Run the following command to determine if the current `ssl_svcename` parameter value is correctly set and is not a default port (50000).

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

Remediation:

Run the following command to set the `ssl_svcename` parameter value.

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

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

Notes:

If `DB2COMM` contains both TCP/IP and SSL, the port specified by `ssl_svcename` must not be the same as `svcename`. Otherwise, the instance starts up without either SSL or TCP/IP protocol support.

3.2 DB2 Database Configuration Parameters

This section provides guidance on how DB2 will control the data in the databases.

3.2.1 Set failed archive retry delay (Scored)

Profile Applicability:

- 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. You do not want the delay to be so short that the database ends up in a denial of service scenario, but you don't want the delay to be too long if an outside attack happens at the same time.

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE 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 $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

```
db2 => update database configuration using archretrydelay 20
```

Default Value:

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

3.2.2 Set the database instance to auto-restart after abnormal termination (Scored)

Profile Applicability:

- 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 $DB2INSTANCE 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 $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

Default Value:

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

3.2.3 Disable database discovery (Scored)

Profile Applicability:

- 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:

Disabling database discovery 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 $DB2INSTANCE 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 $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

Default Value:

The default value for `DISCOVER_DB` is `ENABLE`.

3.2.4 Secure permissions for the primary archive log location (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `logarchmeth1` parameter specifies the type of media and the location used as the primary destination of archived logs. It is recommended that the directory used for the archived logs be set to full access for DB2 administrator accounts and read and execute for all other accounts.

Rationale:

Restricting access to the contents of the primary archive log directory will help ensure that the confidentiality, integrity, and availability of archive logs are protected.

Although there are many ways to ensure that your primary logs will be archived, we recommend using the value of `DISK` as part of the `logarchmeth1` parameter. This will properly ensure that the primary logs are archived. A value of `OFF` is not acceptable.

Audit:

For Windows and Linux:

1. Attach to the DB2 database.

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

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

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

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

```
db2 => get database manager configuration

 db2 => ...

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

Additional steps for Windows:

1. Connect to the DB2 host
2. Right-click on the primary archive log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Review and verify the privileges for all accounts

Additional steps for Linux:

1. Connect to the DB2 host
2. Change to the primary archive log directory
3. Review and verify the permissions for the directory for all users.

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

Remediation:

For Windows and Linux:

1. Attach to the DB2 instance.
2. Run the following command from the DB2 command window to change the primary archive log directory, if necessary:

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

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

1. Connect to the DB2 host
2. Right-click on the primary archive log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Grant all DB2 administrator accounts the *Full Control* authority
6. Grant all other accounts read and execute privileges only (revoke all other privileges)

Additional steps for Linux (assuming that the `logarchmeth`1 parameter includes `DISK`):

1. Connect to the DB2 host
2. Change to the primary archive log directory
3. Change the permissions for the directory

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

Default Value:

The default value for `LOGARCHMETH1` is `OFF`.

The default permissions for the directory are read and write access.

3.2.5 Secure permissions for the secondary archive log location (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `logarchmeth2` parameter specifies the type of media and the location used as the secondary destination for archived logs. It is recommended that the directory used for the archived log be set to full access for DB2 administrator accounts and read and execute only for all other accounts.

Rationale:

Restricting access to the contents of the secondary archive log directory will help ensure that the confidentiality, integrity, and availability of archive logs are protected.

Although there are many ways to ensure that your logs will be archived, we recommend using the value of `DISK` as part of the `logarchmeth2` parameter. This will properly ensure that the logs are archived. A finding of `OFF` is not acceptable.

Audit:

To obtain the secondary archive log location:

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE 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.

Then perform the following additional steps:

For Windows:

1. Connect to the DB2 host
2. Right-click on the secondary archive log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Review and verify the privileges for all accounts

For Linux:

1. Connect to the DB2 host
2. Change to the secondary archive log directory
3. Review and verify the permissions of the directory for all users:

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

Remediation:

For Windows and Linux:

1. Attach to the DB2 instance.
2. Run the following command from the DB2 command window to change the secondary archive log directory, if necessary:

```
db2 => update database configuration using logarchmeth2
```

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

1. Connect to the DB2 host
2. Right-click on the secondary archive log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Grant all DB2 administrator accounts the *Full Control* authority
6. Grant all other accounts read and execute privileges only (revoke all other privileges)

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

1. Connect to the DB2 host
2. Change to the secondary archive log directory
3. Change the permissions for the directory

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

Default Value:

The default value for `LOGARCHMETH2` is `OFF`.

The default value for the directory is read and write access.

3.2.6 Secure permissions for the tertiary archive log location (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `failarchpath` parameter specifies the type of media and the location used as the tertiary destination for archived logs. It is recommended that the directory used for the archived logs be set to full access for DB2 administrator accounts and read and execute only for all other accounts.

Rationale:

Restricting access to the contents of the tertiary archive log directory will help ensure that the confidentiality, integrity, and availability of archive logs are protected.

Although there are many ways to ensure that your logs will be archived, we recommend using the value of `DISK` as part of the `failarchpath` parameter. This will properly ensure that the logs are archived. A finding of `OFF` is not acceptable.

Audit:

For Windows and Linux:

1. Attach to the DB2 database.

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

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

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

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

```
db2 => get database manager configuration

db2 => ...

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

Then perform the following commands:

For Windows:

1. Connect to the DB2 host
2. Right-click on the tertiary archive log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Review and verify the permissions for all accounts

For Linux:

1. Connect to the DB2 host
2. Change to the tertiary archive log directory
3. Review and verify the permissions for the directory for all users.

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

Remediation:

For Windows and Linux:

1. Attach to the DB2 instance.
2. Run the following command from the DB2 command window to change the tertiary archive log directory, if necessary:

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

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

1. Connect to the DB2 host
2. Right-click on the tertiary archive log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Grant all DB2 administrator accounts the *Full Control* authority
6. Grant all other accounts read and execute privileges only (revoke all other privileges)

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

1. Connect to the DB2 host
2. Change to the tertiary archive log directory
3. Change the permissions for the directory

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

Default Value:

The default value for `FAILARCHPATH` is `null`.

3.2.7 Secure permissions for the log mirror location (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `mirrorlogpath` parameter specifies the type of media and the location used to store the mirror copy of the logs. It is recommended that the directory used for the mirror copy of the logs be set to full access for DB2 administrator accounts and read and execute only for all other accounts.

Rationale:

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

Audit:

Perform the following DB2 commands to obtain the directory location:

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE 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.

Additional steps for Windows:

1. Connect to the DB2 host
2. Right-click on the mirror log directory
3. Choose *Properties*
4. Select the *Security* tab

5. Review and verify the privileges for all accounts

Additional steps for Linux:

1. Connect to the DB2 host
2. Change to the mirror log directory
3. Review and verify the permissions for the directory for all users.

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

Remediation:

For Windows and Linux:

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

2. Run the following command from the DB2 command window to change the mirror log directory, if necessary:

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

Additional steps for Windows:

1. Connect to the DB2 host
2. Right-click on the primary archive log directory
3. Choose *Properties*
4. Select the *Security* tab
5. Grant all DB2 administrator accounts the *Full Control* authority
6. Grant all other accounts read and execute privileges only (revoke all other privileges)

Additional steps for Linux:

1. Connect to the DB2 host
2. Change to the mirror log directory
3. Change the permissions for the directory

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

Default Value:

The default value for `mirrorlogpath` is `null`.

3.2.8 Establish retention set size for backups (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

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

Rationale:

Retain multiple copies of the database backup to ensure that the database can recover from an unexpected failure. This parameter should not be set to `0`. Multiple backups should be kept 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 $DB2INSTANCE 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 $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

Default Value:

The default value for `NUM_DB_BACKUPS` is `12`.

3.2.9 Set archive log failover retry limit (Scored)

Profile Applicability:

- 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`.

Audit:

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

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE 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 $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

Default Value:

The default value for `numarchretry` is `5`.

3.3 Database Administration Server Settings

This section provides guidance on configuring and securing the DB2 Database Administration Server (DAS).

3.3.1 Establish DAS administrative group (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `dasadm_group` parameter defines the group name with DAS Administration (DASADM) authority for the DAS. It is recommended that the `dasadm_group` group contains authorized users only.

Rationale:

The DAS is a special administrative tool that enables remote administration of DB2 servers. DASADM authority is the highest level of authority within the DAS.

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 admin configuration
```

3. Locate the value in the output:

```
db2 => get admin configuration
db2 => …
       DAS Administration Authority Group Name (DASADM_GROUP) = DASADM
```

Note: `DASADM_GROUP` is set to `DASADM` 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 configuration using dasadm_group <valid system
group>
```

Default Value:

The default value for `dasadm_group` is null.

3.3.2 Set a generic system name (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The `db2system` parameter specifies the DB2 system name that is used by users and database administrators to identify the DB2 server. It is recommended that this parameter be set to a value that does not represent sensitive aspects of the system.

Rationale:

Exposing OS or DB revision information may provide malicious users with enough information to identify vulnerabilities that may be present in the platforms.

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 admin configuration
```

3. Locate the value in the output:

```
db2 => get admin configuration
db2 => …
       Name of the DB2 Server System              (DB2SYSTEM) = QANODE1
```

Note: `DB2SYSTEM` is set to `QANODE1` in the above output.

Remediation:

1. Attach to the DB2 instance.

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

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

```
db2 => update database configuration using db2system <valid system
group>
```

Default Value:

The default value for `DB2SYSTEM` is the hostname.

3.3.3 Disable DAS discoverability (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The `discover` parameter specifies the discovery mode for the DB2 Administration Server. It is recommended that this parameter be set to `DISABLE`.

Rationale:

The DB2 Administration Server should not handle any type of discovery request. This will prevent a malicious user from discovering all DB2 servers on the network.

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 admin configuration
```

3. Locate the value in the output:

```
db2 => get admin configuration
db2 => …
       DAS Discovery Mode                          (DISCOVER) = DISABLE
```

Note: `DISCOVER` 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 admin configuration using discover disable
```

Default Value:

The default value for `DISCOVER` is `SEARCH`.

3.3.4 Prevent execution of expired tasks (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The `exec_exp_task` parameter controls whether the DB2 Scheduler will initialize past tasks that were scheduled but not yet executed. It is recommended that this parameter be set to `NO`.

Rationale:

This will help ensure sequestered jobs are not invoked by accident, which may have malicious scripts associated with the job. Ensure to review all expired jobs before restarting them.

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 admin configuration
```

3. Locate the value in the output:

```
db2 => get admin configuration
db2 => …
       Execute Expired Tasks                    (EXEC_EXP_TASK) = NO
```

Note: `EXEC_EXP_TASK` 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 exec_exp_task no
```

Default Value:

The default value for `exec_exp_task` is `NO`.

3.3.5 Secure the JDK 32-bit runtime library (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The `jdk_path` parameter specifies the 32-bit Software Developer's Kit (SDK) for Java directory for the DB2 Administration Server. It is recommended that the location pointed to by this parameter contain a current version of the JDK and be secured.

Rationale:

Maintaining JDK currency will ensure known exploitable conditions are mitigated. Ensuring that the location of the JDK is secure will help prevent attackers from compromising the integrity of Java runtime and therefore the administrative facilities of the DB 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 admin configuration
```

3. Locate the value in the output:

```
db2 => get admin configuration
db2 => …
       Java Development Kit Installation Path DAS   (JDK_PATH) =
C:\Program Files\Java
```

Note: `JDK_PATH` is set to `C:\Program Files\Java` 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 jdk_path <valid
path>
```

Default Value:

The default value for `jdk_path` is the default java install path.

3.3.6 Secure the JDK 64-bit runtime library (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The `jdk_64_path` parameter specifies the 64-bit Software Developer's Kit (SDK) for Java directory for the DB2 Administration Server. It is recommended that the location pointed to by this parameter contain a current version of the JDK and be secured.

Rationale:

Maintaining JDK currency will ensure known exploitable conditions are mitigated. Ensuring that the location of the JDK is secure will help prevent malicious entities from compromising the integrity of Java runtime and therefore the administrative facilities of the DB 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 admin configuration
```

3. Locate the value in the output:

```
db2 => get admin configuration
db2 => …
       Java Development Kit Installation Path DAS   (JDK_64_PATH) =
C:\Program Files\Java
```

Note: `JDK_64_PATH` is set to `C:\Program Files\Java` 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 jdk_64_path <valid
path>
```

Default Value:

The default value for `jdk_64_path` is the default install java path.

3.3.7 Disable unused task scheduler (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The `sched_enable` parameter specifies whether the DB2 Task Center utility is allowed to schedule and execute tasks at the administration server. It is recommended that this parameter be set to `OFF` when the Task Scheduler is not in use.

Rationale:

Enable this feature only when scheduling and executing tasks from the DB2 Task Center utility is necessary. This will ensure that malicious tasks are not executed unknowingly by the DB2 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 admin configuration
```

3. Locate the value in the output:

```
db2 => get admin configuration
db2 => …
       Scheduler Mode                              (SCHED_ENABLE)  = OFF
```

Note: `SCHED_ENABLE` is set to `OFF` 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 admin configuration using sched_enable off
```

Default Value:

The default value for `SCHED_ENABLE` is `OFF`.

4 Label-Based Access Controls (LBAC)

This section provides guidance on a new feature in DB2 V9.1 that can control the read and write access of a user at the table column and row level. This feature is a separately licensed component of DB2; therefore, apply these settings where appropriate.

4.1 Enforce label-based access controls implementation (Not Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

Ensure that the database has the label-based access controls (LBAC) component implemented to protect sensitive data. It is recommended that the policies and the components are properly enforced at the column and/or row level.

Rationale:

LBAC increases the control of your data by deciding exactly who has read and/or write access to individual rows and columns.

Audit:

Review all sensitive tables and views in your organization to determine who should have access to which columns and/or rows.

Remediation:

Impose LBAC capability on tables and rows with sensitive data.

4.2 Review security rule exemptions (Not Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

LBAC rule exemptions provide very powerful access. Do not grant them without careful consideration. It is recommended that all security rule exemptions are reviewed against users and their required access.

Rationale:

LBAC rule exemptions allow a particular rule within a particular security policy to not be enforced when trying to access data protected by that security policy.

Audit:

Review and justify all rule exemption grants.

Remediation:

Review all users that have LBAC rule exemptions for qualification according to needs of the business.

4.3 Review security label components (Not Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

A security label component represents any criteria that you use to decide if a user should have access to a given set of data. It is recommended that all security label components are reviewed.

Rationale:

A security label component should be implemented to provide different levels of access to different sensitive data.

Audit:

Review and justify all security label components.

Remediation:

Review all users and ensure those security label components are defined properly.

4.4 Review security label policies (Not Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

A security policy defines the criteria in an organization based on the label components, rules, and rule exemptions. It is recommended that all policies are reviewed.

Rationale:

A security policy defines all access to the table and the columns based on the user's login.

Audit:

Review and justify all security label policies.

Remediation:

Review all security label policies and ensure that they are set up properly.

4.5 Review security labels (Not Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

A security label defines the criteria of access to the protected data. It is recommended that all security labels are reviewed.

Rationale:

A security label must be properly set up on tables with sensitive data.

Audit:

Review and justify all security labels.

Remediation:

Review all security labels and ensure that they are set up properly.

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 should a DB2 service failure coincide with a corrupt backup.

Audit:

Review the replication of your backups based on organization policy.

Remediation:

Define a process to replicate your backups onto multiple locations.

5.2 Protect backups (Not Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

Backups of your database should be stored securely in a location with full access for administrators, read and execute access for group, and no access for users.

Rationale:

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

Audit:

Review the privileges applied to your backups.

Remediation:

Define a security policy for all backups that specifies the privileges they should be assigned.

5.3 Enable automatic 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 reduce database outages.

Audit:

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

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

3. Locate this value in the output:

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

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

Remediation:

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

Default Value:

The default value for `AUTO_MAINT` is `ON`.

5.4 Schedule Runstat and Reorg (Not Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

`runstat` and `reorg` are two DB2 utilities that maintain the database data. It is recommended that these utilities be executed when possible.

Rationale:

All statistics on tables and data should be monitored on a regular basis. A well-performing instance will require less system resources and provide better availability to users.

Audit:

Not applicable

Remediation:

Run the `runstat` and/or the `reorg` utility whenever a maintenance window permits such action.

6 Database Objects

Note: SYSCAT views have underlying SYSIBM tables that are also granted access by the PUBLIC group by default. Ensure that permissions 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 $DB2INSTANCE 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'
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

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 $DB2INSTANCE 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’
```

Remediation:

Revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

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 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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.5 Restrict Access to SYSCAT.EVENTS (Scored)

Profile Applicability:

- Level 1 - RDBMS
- 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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.6 Restrict Access to SYSCAT.EVENTTABLES (Scored)

Profile Applicability:

- Level 1 - RDBMS
- 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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.8 Restrict Access to SYSCAT.INDEXAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- 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 $DB2INSTANCE 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’
```

Remediation:

Revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.12 Restrict Access to SYSCAT.SECURITYLABELACCESS (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `SYSCAT.SECURITYLABELACCESS` view contains all accounts in the database that have a security label privilege. 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 $DB2INSTANCE 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 = 'SECURITYLABELACCESS' and grantee = 'PUBLIC'
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.13 Restrict Access to SYSCAT.SECURITYLABELCOMPONENTELEMENTS (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

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

Rationale:

`PUBLIC` should not be able to view all the elements of a security component and/or the database security policy.

Audit:

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

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE 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 = 'SECURITYLABELCOMPONENTELEMENTS' and grantee = ‘PUBLIC’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.14 Restrict Access to SYSCAT.SECURITYLABELCOMPONENTS (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `SYSCAT.SECURITYLABELCOMPONENTS` view contains the components of a security label. It is recommended that the `PUBLIC` role be restricted from accessing this view.

Rationale:

`PUBLIC` should not be able to view all the security components and/or the database security policy.

Audit:

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

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE 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 = 'SECURITYLABELCOMPONENTS' and grantee = ‘PUBLIC’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.15 Restrict Access to SYSCAT.SECURITYLABELS (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

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

Rationale:

`PUBLIC` should not be able to view all the security components and/or the database security policy.

Audit:

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

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE 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 = 'SECURITYLABELS' and grantee = ‘PUBLIC’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.16 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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.17 Restrict Access to SYSCAT.SECURITYPOLICYCOMPONENTRULES (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `SYSCAT.SECURITYPOLICYCOMPONENTRULES` view contains the access rights for a security label component. It is recommended that the `PUBLIC` role be restricted from accessing this view.

Rationale:

`PUBLIC` should not be able to view all the access rules of 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 $DB2INSTANCE 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 = 'SECURITYPOLICYCOMPONENTRULES' and grantee = ‘PUBLIC’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.18 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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.19 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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.20 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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.21 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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.22 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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.23 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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.24 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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.25 Restrict Access to SYSCAT.SEQUENCEAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `SYSCAT.SEQUENCEAUTH` contains users and/or groups that have access to one or more privileges 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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.26 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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.27 Restrict Access to SYSCAT.PROCEDURES (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `SYSCAT.PROCEDURES` contains all stored procedures in the database. 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 $DB2INSTANCE 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 = 'PROCEDURES' and grantee = ‘PUBLIC’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.28 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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.29 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 $DB2INSTANCE 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’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.30 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 $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

```
db2 => select grantee, tbspace from sysibm.systbspaceauth and grantee =
‘PUBLIC’
```

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

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

6.31 Restrict Access to SYSCAT.MODULEAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

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

Rationale:

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

Audit:

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 = 'MODULEAUTH' 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.moduleauth from public
```

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

6.32 Restrict Access to SYSCAT.VARIABLEAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

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

Rationale:

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

Audit:

Determine if `SYSCAT.VARIABLEAUTH` privileges for users, groups, and roles are correctly set.

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

1. Connect to the DB2 database.

```
db2 => connect to $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 = 'VARIABLEAUTH' and grantee = 'PUBLIC'
```

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
```

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

6.33 Restrict Access to SYSCAT.WORKLOADAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- 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 `GRANT` the `SELECT` privilege to `PUBLIC` for `SYSCAT` views. Therefore, it is strongly recommended to explicitly `REVOKE` the `SELECT` privilege on the `SYSCAT.WORKLOADAUTH` from `PUBLIC` to reduce risk to the organization's data.

Audit:

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

1. Connect to the DB2 database.

```
db2 => connect to $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 = 'WORKLOADAUTH' 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 => db2 => revoke select on syscat.workloadauth from public
```

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

6.34 Restrict Access to SYSCAT.XSROBJECTAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

```
db2 => connect to $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 = 'XSROBJECTAUTH' 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.xsrmoduleauth from public
```

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

6.35 Restrict Access to SYSIBMADM.OBJECTOWNERS (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `SYSIBMADM.OBJECTOWNERS` administrative view shows 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 `GRANT` the `SELECT` privilege to `PUBLIC` for views. Therefore, it is strongly recommended to explicitly `REVOKE` the `SELECT` privilege on the `SYSIBMADM.OBJECTOWNERS` view from `PUBLIC` to reduce risk to the organization's data.

Audit:

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

1. Connect to the DB2 database.

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

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

```
db2 => select grantee from sysibm.systabauth where tcreator =
'SYSIBMADM' and ttname = 'OBJECTOWNERS' 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 SYSIBMADM.OBJECTOWNERS from public
```

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

6.36 Restrict Access to SYSIBMADM.PRIVILEGES (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

The `SYSIBMADM.PRIVILEGES` administrative view displays all explicit privileges for all authorization IDs in the currently connected databases' system catalogs. `PRIVILEGES` schema is `SYSIBMADM`.

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

```
db2 => connect to $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 = 'PRIVILEGES' 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 SYSIBMADM.PRIVILEGES from public
```

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 Entitlements

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

7.1 Establish an administrator group (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The `sysadm_group` parameter defines the system administrator group with `SYSADM` authority for the DB2 instance. Accounts with this authority possess the highest level of authority within the database manager (i.e., stopping/starting services, backup/recovery, and maintenance) and control all database objects (i.e., data, system objects, and privileges). It is recommended that the `sysadm_group` group contains authorized users only.

Rationale:

If an account that possesses this authority is compromised or used in a malicious manner, the confidentiality, integrity, and availability of data in the DB2 instance will be at increased risk.

Audit:

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

1. Attach to the DB2 database.

```
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:

```
db2 => get database configuration
db2 => …
       SYSADM group name                        (SYSADM_GROUP) = DB2SYS
```

Note: `SYSADM_GROUP` is set to `DB2SYS` in the above output.

Remediation:

Define a valid group name for the `SYSADM` group.

1. Attach to the DB2 database.

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

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

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

Default Value:

The default value for `SYSADM_GROUP` is `NULL`.

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=sysadm_group

7.2 Establish a system control group (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 increased risk.

Audit:

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

1. Attach to the DB2 database.

```
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:

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

Note: `SYSCTRL_GROUP` is set to `DB2CTRL` in the above output.

Remediation:

Define a valid group name for the `SYSCTRL` group. **Note:** This parameter does not apply to Windows platforms.

1. Attach to the DB2 database.

```
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`.

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=sysctrl_group

7.3 Establish a system maintenance group (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `sysmaint_group` parameter defines the system administrator group that possesses the system maintenance (`SYSMAINT`) authority. It is recommended that the `sysmaint_group` group contains authorized users only.

Rationale:

If an account that possesses this authority is compromised or used in a malicious manner, the confidentiality, integrity, and availability of data in the DB2 instance will be at increased risk.

Audit:

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

1. Attach to the DB2 database.

```
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:

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

Note: `SYSMAINT_GROUP` is set to `DB2MAINT` in the above output.

Remediation:

Define a valid group name for the `SYSMAINT` group. **Note:** This parameter does not apply to Windows platforms.

1. Attach to the DB2 database.

```
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`.

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=sysmaint_group

7.4 Establish a system monitoring group (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 the `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 database.

```
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:

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

Note: `SYSMON_GROUP` is set to `DB2MON` in the above output.

Remediation:

Define a valid group name for the SYSMON group.

1. Attach to the DB2 database.

```
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.

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=sysmon_group

7.5 Secure the security administrator role (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 increased risk.

Audit:

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

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE 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 $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

```
db2 => REVOKE SECADM ON DATABASE FROM USER <username>
```

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=securityadm

7.6 Secure the database administration role (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 the dbadm role be granted to authorized users only.

Rationale:

If an account that possesses this authority is compromised or used in a malicious manner, the confidentiality, integrity, and availability of data in the database will be at increased risk.

Audit:

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

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE 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 $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

```
db2 db2 => REVOKE SECADM ON DATABASE FROM USER <username>
```

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=dbadm

7.7 Secure the create table role (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `CREATETAB` (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 $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
createtabauth = '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 $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

```
db2 => REVOKE CREATETAB ON DATABASE FROM USER <username>
```

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=createtab

7.8 Secure the bind application role (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 increased risk.

Audit:

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

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE 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 $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

```
db2 => REVOKE BINDADD ON DATABASE FROM USER <username>
```

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=bindadd

7.9 Secure the connect role (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `CONNECT` role grants the authority to a user to connect to a specific database. It is recommended that the CONNECT 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 $DB2INSTANCE 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 $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

```
db2 => REVOKE CONNECT ON DATABASE FROM USER <username>
```

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=connect

7.10 Secure the NOFENCE role (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `NOFENCE` role grants the authority to a user to create user-defined functions or procedures that are not fenced in the memory block of the database. It is recommended that the `NOFENCE` 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 $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
nofenceauth = '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 $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

```
db2 => REVOKE CREATE_NOT_FENCED_ROUTINE ON DATABASE FROM USER
<username>
```

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=nofence

7.11 Secure the implicit schema role (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `IMPLSCHEMA` (implicit schema) role grants the authority to a user to create objects without specifying a schema that already exists. It is recommended that the `IMPLSCHEMA` 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 $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
implschemaauth = '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 $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

```
db2 => REVOKE IMPLICIT_SCHEMA ON DATABASE FROM USER <username>
```

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=implschema

7.12 Secure the load role (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 $DB2INSTANCE 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 $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=load

7.13 Secure the external routine role (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `EXTERNALROUTINE` role grants the authority to a user to create user-defined functions and procedures in a specific database. It is recommended that the `EXTERNALROUTINE` 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 $DB2INSTANCE 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 $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

```
db2 => REVOKE CREATE_EXTERNAL_ROUTINE ON DATABASE FROM USER <username>
```

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=externalroutine

7.14 Secure the QUIESCECONNECT role (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. It is recommended that the `QUIESCECONNECT` 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 $DB2INSTANCE 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:

Revoke this permission from any unauthorized users.

1. Connect to the DB2 database.

```
db2 => connect to $DB2INSTANCE user $USERNAME using $PASSWORD
```

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

```
db2 => REVOKE QUIESCE_CONNECT ON DATABASE FROM USER <username>
```

References:

1. http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=quiesceconnect

7.15 Secure the SQLADM authority (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `SQLADM` authority is required to monitor, tune, and alter SQL statements.

Rationale:

The `SQLADM` authority can `CREATE`, `SET`, `FLUSH`, `DROP EVENT MONITORS` and perform `RUNSTATS` and `REORG INDEXES` and `TABLES`. `SQLADM` can be granted to users, groups, or roles or `PUBLIC`. `SQLADM` authority is a subset of the `DBADM` authority and can be granted by the `SECADM` authority.

Audit:

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

```
select distinct grantee, granteetype from syscat.dbauth where
sqladmauth = 'Y'
```

2. 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>
```

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

7.16 Secure the DATAACCESS authority (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `DATAACCESS` authority grants the authority to access data. It allows the grantee to leverage DML level commands, i.e., `SELECT`, `INSERT`, `UPDATE`, `DELETE`, `LOAD`, and `EXECUTE` any package or routine.

Rationale:

The `DATAACCESS` authority gives the grantee read access and also control over manipulating the data. `DATAACCESS` can be granted to users, groups, or roles, but not `PUBLIC`. `DATAACCESS` authority is a subset of the `DBADM` authority and can be granted by the `SECADM` authority.

Audit:

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

```
select distinct grantee, granteetype from syscat.dbauth where
dataaccessauth = 'Y'
```

2. 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

7.17 Secure the ACCESSCTRL authority (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `ACCESSCTRL` authority is the authority required to grant and revoke privileges on objects within a specific database. Some of these privileges include `BINDADD`, `CONNECT`, `CREATETAB`, `CREATE\_EXTERNAL\_ROUTINE`, `LOAD`, and `QUIESCE\_CONNECT`. It has no inherent privilege to access data stored in tables, except the catalog tables and views.

Rationale:

The `ACCESSCTRL` authority gives the grantee access control to a specified database. With this authority, the grantee can grant/revoke privileges to other users. `ACCESSCTRL` can be granted to users, groups, or roles, but not `PUBLIC`. `ACCESSCTRL` authority can only be granted by the `SECADM` authority.

Audit:

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

```
select distinct grantee, granteetype from syscat.dbauth where
accessctrlauth = 'Y'
```

2. 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

7.18 Secure the WLMADM authority (Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The `WLMADM` authority manages workload objects for a database. Holders of `DBADM` authority implicitly also hold `WLMADM` authority.

Rationale:

The `WLMADM` authority enables creating, altering, dropping, commenting, granting, and revoking access to workload objects for a database.

Audit:

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

```
select grantee, wlmadmauth from syscat.dbauth
```

2. Determine if the grantee(s) are correctly set.

Remediation:

Revoke any user who should NOT have `WLMADM` authority:

```
REVOKE WLMADM ON DATABASE FROM USER <username>
```

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

8 General Policy and Procedures

8.1 Restrict access to starting and stopping DB2 instances (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 allowing privileged users to start and stop the DB2 instance will help ensure that the DB2 instance is controlled by authorized administrators.

Audit:

On 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 Linux: Identify the members of the local DB2 admin group that have access to stop and start the DB2 instance.

Remediation:

1. Connect to the host
2. Review users and groups that have access to start and stop the DB2 instance
3. Revoke access from any unnecessary users.

8.2 Restrict access to starting and stopping the DB2 administration server (Not Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

The DB2 administration server responds to remote requests from administration tools and client utilities. It is recommended that only administrators are allowed to start and stop the DB2 administration server.

Rationale:

Allowing only privileged users to start and stop the DB2 administration server will help ensure that the DB2 administration server is controlled by authorized administrators.

Audit:

On Windows: go to *Start*, then to the *Run* option. Type in `services.msc` in the command prompt. Locate the `DB2DAS` service and identify the user/group that can start and stop the service.

On Linux: Identify the members of the local DB2 admin group that has access to stop and start the `db2admin` command.

Remediation:

1. Connect to the host
2. Review users and groups that have access to start and stop the DB2 instance
3. Revoke access from any unnecessary users.

8.3 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, so they should be removed.

Audit:

Perform the following commands to determine if there are any unused schemas.

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 and identify any unused schemas.

Remediation:

Perform the following commands to remove unused schemas:

1. Connect to the DB2 instance.

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

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

```
db2 => drop scheme  restrict
```

3. Remove the unused schemas.

8.4 Review system tablespaces for user data (Not 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 store system data only and not user data.

Rationale:

Do not install any user data in the following system tablespaces: `SYSCATSPACE` and `SYSTOOLSPACE`.

Audit:

1. Connect to the DB2 instance.

```
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 for any user data objects.

Remediation:

Drop, migrate, or otherwise remove all user data objects (tables, schemas, etc.) from within the system tablespaces. Also, revoke write access for the system tablespaces for all users. Note that these actions may cause loss of data and functionality for 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. Perform the appropriate actions to remediate the identified issues.

8.5 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 $DB2DATABASE
```

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            =
```

Note: Identify the default databases from the output above.

Remediation:

To drop sample databases:

1. Connect to the DB2 database.
2. Run the following command from the DB2 command window:

```
db2 => drop database sample
```

8.6 Enable SSL communication with LDAP server (Scored)

Profile Applicability:

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

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:

Update the parameter value:

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
```

Notes:

The file is located under `INSTANCE_HOME/sqllib/cfg/`, for Unix; and `%DB2PATH%\cfg\`, for MS Windows.

8.7 Secure the permissions 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 read-only (RO) to Everyone/Other/Users/Domain Users. This will ensure that the parameter file is protected.

Audit:

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

For 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 Linux:

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 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 Linux:

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

```
OS => chmod -R 740
```

Default Value:

The default value for this directory is read and write access for non-administrator accounts.

Notes:

The file is located under `INSTANCE_HOME/sqllib/cfg/`, for Unix; and `%DB2PATH%\cfg\`, for MS Windows.

8.8 Secure the permissions 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 read-only (RO) to Everyone/Other/Users/Domain Users. This will ensure that the parameter file is protected.

Audit:

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

For 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 Linux:

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 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 Linux:

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

```
OS => chmod -R 740
```

Default Value:

The default value for this directory is read-and-write access to non-administrator accounts.

Notes:

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.

9 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.

9.1 Review the 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:

1. Attach to a DB2 Instance:

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

2. Connect to DB2 database:

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

3. Run the following and review the results to determine if each role name still has a business requirement to access the data:

```
db2 => select rolename from syscat.roleauth where grantortype <> 'S'
group by rolename
```

Remediation:

To remove a role from the database:

1. Attach to a DB2 Instance:

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

2. Connect to DB2 database:

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

3. 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

9.2 Review the role members (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS

Description:

Having roles that have been granted specific privileges, then assigning users to the roles, is usually considered the best way to grant application access. Because granting privileges to individual users can be more difficult to track and maintain against unauthorized access, users should be assigned to organization-defined database roles according to the needs of the business. As users leave the organization or change responsibilities within the organization, the appropriate roles for them change as well, so role membership needs to be reviewed and verified periodically.

Rationale:

Users who have excessive privileges not needed to do their jobs pose an unnecessary risk to the organization as an insider threat.

Audit:

1. Attach to a DB2 Instance:

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

2. Connect to DB2 database:

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

3. Run the following to review and verify that the members are correct for each role:

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

Remediation:

To remove a member from a particular role:

1. Attach to a DB2 Instance:

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

2. Connect to DB2 database:

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

3. 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

9.3 Review nested roles (Scored)

Profile Applicability:

- Level 2 - RDBMS

Description:

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. By nesting roles the database administrator is saving time by only having to assign a group of users versus assigning them individually. Nesting roles properly can often ease the application of the security model if it's kept fairly shallow, and if the roles are logically named. If these are all true, then nesting of roles is a good idea.

Rationale:

As tracking multiple levels of permissions can result in unauthorized access to data resources, this capability should be restricted according to the needs of the business.

Audit:

1. Attach to DB2 Instance:

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

2. Connect to DB2 database:

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

3. Run the following to identify any nested roles:

```
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:

1. Attach to DB2 Instance:

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

2. Connect to DB2 database:

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

3. Run the following:

```
db2 => revoke role <role name> from role <role>
```

9.4 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:

As any role granted to `PUBLIC` can potentially allow the compromise of database availability, confidentiality, or integrity, these roles should be restricted according to the needs of the business.

Audit:

1. Attach to a DB2 Instance:

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

2. Connect to DB2 database:

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

3. Run the following:

```
db2 => select grantee, rolename from syscat.roleauth where grantee =
'PUBLIC'
```

Note: If the value returned is blank, that is considered a passable finding.

Remediation:

To remove `PUBLIC` from a particular role:

1. Attach to a DB2 Instance:

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

2. Connect to DB2 database:

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

3. 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

9.5 Review role grantees with the WITH ADMIN OPTION clause (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:

1. Attach to DB2 Instance:

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

2. Connect to DB2 database:

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

3. Perform the following query:

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

Note: If the value returned is blank, that is considered a passable finding.

Remediation:

1. Attach to DB2 Instance:

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

2. Connect to DB2 database:

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

3. Perform the following command:

```
db2=> revoke admin option for role <role name> from user <user name>
```

10 DB2 Utilities and Tools

10.1 Restrict access to the DB2 Control Center (Not Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The DB2 Control Center is a management tool that manages all registered DB2 instances and databases. It is recommended that access to the Control Center utility be granted to authorized users only.

Rationale:

Secure this application where applicable, since it has access to the DB2 instance name, the host it resides on, and the database name.

Audit:

Locate the `<DB2 install>\SQLLIB\BIN\db2cc` executable and identify the users/groups that have access to it.

Remediation:

To revoke access to the DB2 Control Center:

1. Connect to the host
2. Review users and groups that have access to start the DB2 Control Center
3. Revoke access from any unnecessary users.

10.2 Restrict access to the DB2 Configuration Assistant utility (Not Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The DB2 Configuration Assistant is a management tool that manages all connectivity setup to the DB2 instances and databases. It is recommended that access to the Configuration Assistant utility be granted to authorized users only.

Rationale:

Secure this application where applicable, since it has access to the DB2 instance name, the host it resides on, and the database name, and the port number.

Audit:

Locate the `<DB2 install>\SQLLIB\BIN\db2ca` executable and identify the users/groups that have access to it.

Remediation:

To revoke access to the DB2 Configuration Assistant from unnecessary users and groups:

1. Connect to the host
2. Review users and groups that have access to start the DB2 Configuration Assistant.
3. Revoke access from unnecessary users and groups.

10.3 Restrict access to the DB2 Health Monitor utility (Not Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The DB2 Health Monitor is a management tool that manages information about the database manager, database, tablespace and table space containers. It is recommended that access to the DB2 Health Monitor utility be granted to authorized users only.

Rationale:

Secure this application where applicable, since it has sensitive information about the health of the database.

Audit:

Locate the `<DB2 install>\SQLLIB\BIN\db2hc` executable and identify the users/groups that have access to it.

Remediation:

To revoke access to the DB2 Health Monitor from any unnecessary users and groups:

1. Connect to the host
2. Review users and groups that have access to start the DB2 Health Center
3. Revoke access from any unnecessary users and groups.

10.4 Restrict access to the DB2 Activity Monitor utility (Not Scored)

Profile Applicability:

- Level 1 - RDBMS

Description:

The DB2 Activity Monitor is a management tool that monitors all application performance and concurrency, resource consumption, and SQL statement usage of a database. It is recommended that access to the DB2 Activity Monitor utility be granted to authorized users only.

Rationale:

Secure this application where applicable, since it has vital statistics about the database.

Audit:

Locate the `<DB2 install>\SQLLIB\BIN\db2am` executable and identify the users/groups that have access to it.

Remediation:

To revoke access to the DB2 Activity Monitor from any unnecessary users and groups:

1. Connect to the host
2. Review users and groups that have access to start the DB2 Activity Monitor
3. Revoke access from all unnecessary users and groups.

Appendix: Summary Table

Control		Set Correctly	
		Yes	No
1	**Installation and Patches**		
1.1	Install the latest fix packs (Scored)	□	□
1.2	Use IP address rather than hostname (Scored)	□	□
1.3	Leverage the least privilege principle (Not Scored)	□	□
1.4	Use non-default account names (Scored)	□	□
2	**DB2 Directory and File Permissions**		
2.1	Secure the DB2 Runtime Library (Scored)	□	□
2.2	Secure the database container directory (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 permissions for default database file path (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	Enable instance health monitoring (Scored)	□	□
3.1.13	Retain fenced model processes (Scored)	□	□
3.1.14	Set maximum connection limits (Scored)	□	□
3.1.15	Set administrative notification level (Scored)	□	□
3.1.16	Enable server-based authentication (Scored)	□	□
3.1.17	Reserve the desired port number or name for incoming connection requests (Scored)	□	□
3.1.18	Reserve the desired port number or name for incoming SSL connections (Scored)	□	□
3.2	**DB2 Database Configuration Parameters**		
3.2.1	Set failed archive retry delay (Scored)	□	□
3.2.2	Set the database instance to auto-restart after abnormal termination (Scored)	□	□
3.2.3	Disable database discovery (Scored)	□	□

3.2.4	Secure permissions for the primary archive log location (Scored)	□	□
3.2.5	Secure permissions for the secondary archive log location (Scored)	□	□
3.2.6	Secure permissions for the tertiary archive log location (Scored)	□	□
3.2.7	Secure permissions for the log mirror location (Scored)	□	□
3.2.8	Establish retention set size for backups (Scored)	□	□
3.2.9	Set archive log failover retry limit (Scored)	□	□
3.3	**Database Administration Server Settings**		
3.3.1	Establish DAS administrative group (Scored)	□	□
3.3.2	Set a generic system name (Scored)	□	□
3.3.3	Disable DAS discoverability (Scored)	□	□
3.3.4	Prevent execution of expired tasks (Scored)	□	□
3.3.5	Secure the JDK 32-bit runtime library (Scored)	□	□
3.3.6	Secure the JDK 64-bit runtime library (Scored)	□	□
3.3.7	Disable unused task scheduler (Scored)	□	□
4	**Label-Based Access Controls (LBAC)**		
4.1	Enforce label-based access controls implementation (Not Scored)	□	□
4.2	Review security rule exemptions (Not Scored)	□	□
4.3	Review security label components (Not Scored)	□	□
4.4	Review security label policies (Not Scored)	□	□
4.5	Review security labels (Not Scored)	□	□
5	**Database Maintenance**		
5.1	Enable backup redundancy (Not Scored)	□	□
5.2	Protect backups (Not Scored)	□	□
5.3	Enable automatic database maintenance (Scored)	□	□
5.4	Schedule Runstat and Reorg (Not Scored)	□	□
6	**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.SECURITYLABELACCESS (Scored)	□	□
6.13	Restrict Access to	□	□

	SYSCAT.SECURITYLABELCOMPONENTELEMENTS (Scored)		
6.14	Restrict Access to SYSCAT.SECURITYLABELCOMPONENTS (Scored)	□	□
6.15	Restrict Access to SYSCAT.SECURITYLABELS (Scored)	□	□
6.16	Restrict Access to SYSCAT.SECURITYPOLICIES (Scored)	□	□
6.17	Restrict Access to SYSCAT.SECURITYPOLICYCOMPONENTRULES (Scored)	□	□
6.18	Restrict Access to SYSCAT.SECURITYPOLICYEXEMPTIONS (Scored)	□	□
6.19	Restrict Access to SYSCAT.SURROGATEAUTHIDS (Scored)	□	□
6.20	Restrict Access to SYSCAT.ROLEAUTH (Scored)	□	□
6.21	Restrict Access to SYSCAT.ROLES (Scored)	□	□
6.22	Restrict Access to SYSCAT.ROUTINEAUTH (Scored)	□	□
6.23	Restrict Access to SYSCAT.SCHEMAAUTH (Scored)	□	□
6.24	Restrict Access to SYSCAT.SCHEMATA (Scored)	□	□
6.25	Restrict Access to SYSCAT.SEQUENCEAUTH (Scored)	□	□
6.26	Restrict Access to SYSCAT.STATEMENTS (Scored)	□	□
6.27	Restrict Access to SYSCAT.PROCEDURES (Scored)	□	□
6.28	Restrict Access to SYSCAT.TABAUTH (Scored)	□	□
6.29	Restrict Access to SYSCAT.TBSPACEAUTH (Scored)	□	□
6.30	Restrict Access to Tablespaces (Scored)	□	□
6.31	Restrict Access to SYSCAT.MODULEAUTH (Scored)	□	□
6.32	Restrict Access to SYSCAT.VARIABLEAUTH (Scored)	□	□
6.33	Restrict Access to SYSCAT.WORKLOADAUTH (Scored)	□	□
6.34	Restrict Access to SYSCAT.XSROBJECTAUTH (Scored)	□	□
6.35	Restrict Access to SYSIBMADM.OBJECTOWNERS (Scored)	□	□
6.36	Restrict Access to SYSIBMADM.PRIVILEGES (Scored)	□	□
7	**Entitlements**		
7.1	Establish an administrator group (Scored)	□	□
7.2	Establish a system control group (Scored)	□	□
7.3	Establish a system maintenance group (Scored)	□	□
7.4	Establish a system monitoring group (Scored)	□	□
7.5	Secure the security administrator role (Scored)	□	□
7.6	Secure the database administration role (Scored)	□	□
7.7	Secure the create table role (Scored)	□	□
7.8	Secure the bind application role (Scored)	□	□
7.9	Secure the connect role (Scored)	□	□
7.10	Secure the NOFENCE role (Scored)	□	□
7.11	Secure the implicit schema role (Scored)	□	□
7.12	Secure the load role (Scored)	□	□
7.13	Secure the external routine role (Scored)	□	□
7.14	Secure the QUIESCECONNECT role (Scored)	□	□
7.15	Secure the SQLADM authority (Scored)	□	□

7.16	Secure the DATAACCESS authority (Scored)	□	□
7.17	Secure the ACCESSCTRL authority (Scored)	□	□
7.18	Secure the WLMADM authority (Scored)	□	□
8	**General Policy and Procedures**		
8.1	Restrict access to starting and stopping DB2 instances (Not Scored)	□	□
8.2	Restrict access to starting and stopping the DB2 administration server (Not Scored)	□	□
8.3	Remove unused schemas (Not Scored)	□	□
8.4	Review system tablespaces for user data (Not Scored)	□	□
8.5	Remove default databases (Scored)	□	□
8.6	Enable SSL communication with LDAP server (Scored)	□	□
8.7	Secure the permissions of the IBMLDAPSecurity.ini file (Scored)	□	□
8.8	Secure the permissions of the SSLconfig.ini file (Scored)	□	□
9	**DB2 Roles**		
9.1	Review the roles (Scored)	□	□
9.2	Review the role members (Scored)	□	□
9.3	Review nested roles (Scored)	□	□
9.4	Review roles granted to PUBLIC (Scored)	□	□
9.5	Review role grantees with the WITH ADMIN OPTION clause (Scored)	□	□
10	**DB2 Utilities and Tools**		
10.1	Restrict access to the DB2 Control Center (Not Scored)	□	□
10.2	Restrict access to the DB2 Configuration Assistant utility (Not Scored)	□	□
10.3	Restrict access to the DB2 Health Monitor utility (Not Scored)	□	□
10.4	Restrict access to the DB2 Activity Monitor utility (Not Scored)	□	□

Appendix: Change History

Date	Version	Changes for this version
11-05-2009	1.0.0	Initial Public Release
12-31-2009	1.1.0	- Section 1.0.2: Updated Rationale - Section 1.0.3: Updated Description - Section 1.0.4: Added a warning note before the Remediation step - Section 2.0.1: Changed remediation section, step #3 from 744 to 740 - Section 2.0.2: Updated Rationale - Section 3.
12-31-2011	1.2.0	Resolved technical and grammatical issues throughout document. Ticket details available here.
04-04-2016	2.0.0	Initial release.
12-15-2016	3.0.0	Merged sections/recommendations from v1.2.0 with those from v2.0.0.
12-30-2016	3.0.0	Planned Update
01-17-2017	3.0.1	Corrected typo in audit procedure for 6.15, changed 'SECURITYLABELS' to 'SECURITYLABELCOMPONENTS'
01-17-2017	3.0.1	Corrected type in 7.7, changed all instance of 'creatab' to 'createtab'