CIS Benchmarks

CIS IBM DB2 10 Benchmark

v1.1.0 - 08-31-2016

Terms of Use

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

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

Table of Contents

Overview

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

Intended Audience

This document is intended for system and application administrators, security specialists, auditors, help desk, and platform deployment personnel, who plan to develop, deploy, assess, or secure solutions that incorporate DB2 on Linux 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
Adam Montville
Timothy Harrison

Editor
Karen Scarfone
Chris Bielinski

Recommendations

1 Installation and Patches

1.1 Install the latest fix packs (Not Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 – RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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.

1.2 Use IP address rather than hostname (Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - 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:

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

Default Value:

IP address

1.3 Leverage the least privilege principle (Not Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 – RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - 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 account names may increase the DB2 service's susceptibility to unauthorized access by an attacker.

Audit:

For Windows:

1. Review the list of users for the system and confirm that none of the account names are `db2admin`, `db2inst1`, `dasusr1`, or `db2fenc1`.

For Linux:

1. Review `/etc/passwd` and confirm that none of the account names are `db2admin`, `db2inst1`, `dasusr1`, or `db2fenc1`.

Remediation:

For each account with a default name, either change the name to a name that is not well-known or delete the account if it is not needed.

1.5 Configure DB2 to use non-standard ports (Not Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

If enabled, the default DB2 instance will be assigned a default port of TCP:50000 for TCP/IP communication. TCP:50000 is a widely known DB2 port, so this port assignment should be changed. Though deprecated, if you still use the DAS, its default port uses TCP:523 and should be changed.

Rationale:

Using a non-default port helps reduce the number of attacks directed at the database through its port.

Audit:

Use the appropriate command below to identify the assigned port and confirm that it does not use the default value of 50000.

Linux:

```
cat etc/services | grep db2
```

Windows:

```
netstat -bao
```

Remediation:

Assign a non-default port (a value other than 50000) to the default DB2 instance.

Impact:

Changing the port will break connectivity for any servers, clients, etc. configured to access the DB instance at the original port. Any port number changes need to be coordinated to prevent inadvertent outages.

References:

1. http://www.ibm.com/support/knowledgecenter/en/SSEPGG_10.5.0/com.ibm.db2.luw.admin.config.doc/doc/c0060794.html

1.6 Creating the database with the RESTERICTIVE clause (Not Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

```
  db2=> select case when value = '-1' then 'automatic' when value = '' then 'NULL'
else value end as value from sysibmadm.dbcfg where name = 'restrict_access'
```

Remediation:

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

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

- CREATETAB
- BINDADD

- CONNECT
- IMPLICIT_SCHEMA

2. SYSCAT.TABAUTH – Ensure PUBLIC is **NOT** granted the following privileges:

- SELECT on all SYSCAT and SYSIBM tables
- SELECT and UPDATE on all SYSSTAT tables
- SELECT on the following views in schema SYSIBMADM:
    - ALL_*
    - USER_*
    - ROLE_*
    - SESSION_*
    - DICTIONARY
    - TAB

3. SYSCAT.ROUTINEAUTH – Ensure PUBLIC is **NOT** granted the following privileges:

- EXECUTE with GRANT on all procedures in schema SQLJ
- EXECUTE with GRANT on all functions and procedures in schema SYSFUN
- EXECUTE with GRANT on all functions and procedures in schema SYSPROC
- EXECUTE on all table functions in schema SYSIBM
- EXECUTE on all other procedures in schema SYSIBM

4. SYSCAT.MODULEAUTH – Ensure PUBLIC is **NOT** granted the following privileges:

- EXECUTE on the following modules in schema SYSIBMADM:
    - DBMS_DDL
    - DBMS_JOB
    - DBMS_LOB
    - DBMS_OUTPUT
    - DBMS_SQL
    - DBMS_STANDARD
    - DBMS_UTILITY

5. SYSCAT.PACKAGEAUTH – Ensure PUBLIC is **NOT** granted the following privileges:

- BIND on all packages created in the NULLID schema
- EXECUTE on all packages created in the NULLID schema

6. SYSCAT.SCHEMAAUTH – Ensure PUBLIC is **NOT** granted the following privileges:

- CREATEIN on schema SQLJ
- CREATEIN on schema NULLID

7. SYSCAT.TBSPACEAUTH – Ensure PUBLIC is **NOT** granted the USE privilege on table space USERSPACE1.

8. SYSCAT.WORKLOADAUTH – Ensure PUBLIC is **NOT** granted the USAGE privilege on SYSDEFAULTUSERWORKLOAD.

9. SYSCAT.VARIABLEAUTH – Ensure PUBLIC is **NOT** granted the READ privilege on schema global variables in the SYSIBM schema.

References:

1. https://www.ibm.com/support/knowledgecenter/en/SSEPGG_10.5.0/com.ibm.db2.luw.admin.cmd.doc/doc/r0001941.html
2. https://www.ibm.com/support/knowledgecenter/en/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0054269.html

2 DB2 Directory and File Permissions

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

2.1 Secure DB2 Runtime Library (Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - 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 NODE000x/sqldbdir directory
3. Choose *Properties*
4. Select the *Security* tab
5. Determine the permissions for DB administrator accounts and all other accounts

For Linux:

1. Connect to the DB2 host
2. Change to the `NODE000x/sqldbdir` directory
3. Determine the permissions for the directory

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

Remediation:

For Windows:

1. Connect to the DB2 host
2. Right-click on the \NODE000x\sqldbdir directory
3. Choose *Properties*
4. Select the *Security* tab
5. Select all DB administrator accounts and grant them the *Full Control* authority
6. Select all other accounts and revoke all privileges other than *Read* and *Execute*

For Linux:

1. Connect to the DB2 host
2. Change to the /NODE000x/sqldbdir directory
3. Change the permission level of the directory to this recommended value

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

2.2 Secure the database container directory (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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 so 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
- Level 2 - 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.

Audit:

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

Remediation:

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

2.4 Verify the groups within the DB2_GRP_LOOKUP environment variable are appropriate (Windows only) (Not Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 2 - Windows Host OS

Description:

The DB2_GRP_LOOKUP environment variable manages which groups are identified on a local machine/domain level.

Rationale:

Periodic review of these groups is required to ensure that non-essential groups do not have unnecessary authorization.

Audit:

Verify that the DB2_GRP_LOOKUP environment variable includes only the appropriate groups listed within the local machine/domain.

```
db2set -all
```

Remediation:

Alter the value of the DB2_GRP_LOOKUP environment variable so that it includes only the appropriate groups listed within the local machine/domain.

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/t0005914.html

2.5 Verify the domains within the DB2DOMAINLIST environment variable are appropriate (Windows only) (Not Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 2 - Windows Host OS

Description:

It is possible to have a user id be represented across multiple domains. Issues could arise when trying to authenticate such a user id. To prevent these issues, a listing of domains should be defined within the DB2DOMAINLIST environment variable. Note: the DB2DOMAINLIST is only effective if the authentication parameter is set to CLIENT.

Rationale:

Periodic review of the domain list assigned to the DB2DOMAINLIST environment variable helps ensure that non-essential domains do not have unnecessary authorizations.

Audit:

Verify that the DB2DOMAINLIST environment variable includes only the appropriate domains.

```
db2set -all
```

Remediation:

Alter the value of the DB2DOMAINLIST environment variable so that it includes only the appropriate domains.

References:

1. https://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/t0011962.html?lang=en

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
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

Increasing the audit buffer size to greater than 0 will allocate space for the audit records generated by the audit facility. At scheduled intervals, or when the audit buffer is full, the db2auditd audit daemon empties the audit buffer to disk, writing the audit records asynchronously.

Audit:

Perform the following to determine if the audit buffer is set as recommended:

1. Attach to the DB2 instance.

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

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

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

3. Locate `AUDIT_BUF_SZ` value in the output:

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

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

Remediation:

Perform the following to establish an audit buffer:

1. Attach to the DB2 instance

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

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

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

3.1.2 Encrypt user data across the network (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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
```

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

Remediation:

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

1. Attach to the DB2 instance

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

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

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

3.1.3 Require explicit authorization for cataloging (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

DB2 can be configured to allow users that do not possess the `SYSADM` authority to catalog and uncatalog databases and nodes. It is recommended that the `catalog_noauth` parameter be set to `NO`.

Rationale:

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

Audit:

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

1. Attach to the DB2 instance.

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

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

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

3. Locate the value of CATALOG_NOAUTH in the output:

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

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

Remediation:

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

1. Attach to the DB2 instance

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

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

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

References:

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

3.1.4 Disable datalinks support (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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 `datalinks` support be disabled.

Rationale:

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

Audit:

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

1. Attach to the DB2 instance.

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

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

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

3. Locate this value of `DATALINKS` in the output:

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

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

Remediation:

1. Attach to the DB2 instance

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

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

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

3.1.5 Secure permissions for default database file path (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - 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
```

3. 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.

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

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

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

Additional steps for Windows:

1. Connect to the DB2 host
2. Right-click over the directory used as the default file path
3. Choose *Properties*
4. Select the *Security* tab
5. Assign ownership of the directory to the DB2 Administrator
6. Grant all 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
```

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

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

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

1. Attach to the DB2 instance.

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

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

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

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

```
db2 => get database manager configuration
db2 => …
       Diagnostic error capture level (DIAGLEVEL) = 3
```

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

Remediation:

1. Attach to the DB2 instance

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

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

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

3.1.7 Secure permissions for all diagnostic logs (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

The `diagpath` parameter specifies the location of the diagnostic files for the DB2 instance. The directory at this location should be secured so that users have read and execute privileges only (no write privileges). All DB2 administrators should have full access to the directory.

Rationale:

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

Audit:

For both Windows and Linux, perform the following DB2 commands to obtain the location of the directory:

1. Attach to the DB2 instance.

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

2. Run the following command 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 <valid directory>
```

Additional steps for Windows:

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

Additional steps for Linux:

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

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

3.1.8 Require instance name for discovery requests (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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 `DISCOVER` value in the output:
```
db2 => get database manager configuration
db2 => …
       Discovery mode (DISCOVER) = KNOWN
```
Note: `DISCOVER` is set to `KNOWN` in the above output.

Remediation:

The recommended value is KNOWN. Note: this requires a DB2 restart.

1. Attach to the DB2 instance

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

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

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

3. Restart the DB2 instance.

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

Impact:

It is important to be aware that the implementation of this recommendation results in a brief downtime. It is advisable to ensure that the setting is implemented during an approved maintenance window.

3.1.9 Disable instance discoverability (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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
```

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

Remediation:

1. Attach to the DB2 instance

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

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

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

3.1.10 Authenticate federated users at the instance level (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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
```

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

Remediation:

1. Attach to the DB2 instance

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

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

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

3.1.11 Set maximum connection limits (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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 isn't 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 value(s) for these settings:

1. Attach to the DB2 instance.

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

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

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

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

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

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

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

1. Connect to the DB2 database.

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

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

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

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

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

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

Remediation:

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

1. Attach to the DB2 instance

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

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

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

If `maxappls` is NOT less than the value for `max_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`.

3.1.12 Set administrative notification level (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

The `notifylevel` parameter specifies the type of administration notification messages that are written to the administration notification log. It is recommended that this parameter be set greater than or equal to `3`. A setting of 3, which includes settings 1 & 2, will log all fatal errors, failing services, system integrity, as well as system health.

Rationale:

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

Audit:

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

1. Attach to the DB2 instance.

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

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

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

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

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

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

Remediation:

1. Attach to the DB2 instance

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

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

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

Default Value:

The default value of `notifylevel` is 3.

3.1.13 Enable server-based authentication (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

The `srvcon_auth` parameter specifies how and where authentication is to take place 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:

It is important to be aware that the implementation of this recommendation results in a brief downtime. It is advisable to ensure that the setting is implemented during an approved maintenance window.

3.1.14 Set failed archive retry delay (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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 $DB2DATABASE user $USERNAME using $PASSWORD
```

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

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

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

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

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

Remediation:

1. Connect to the DB2 database

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

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

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

Default Value:

The default value for `archretrydelay` is `20`

3.1.15 Auto-restart after abnormal termination (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

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

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

Remediation:

1. Connect to the DB2 database

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

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

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

3.1.16 Disable database discovery (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

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

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

Remediation:

1. Connect to the DB2 database

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

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

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

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

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

Restricting access to the contents of the primary archive log directory will help ensure that the confidentiality, integrity, and availability of archive logs are protected.
Although there are many ways to ensure that your primary logs will be archived, we recommend using the value of DISK as part of the `logarchmeth1` parameter. This will properly ensure that the primary logs are archived. A finding of OFF is not acceptable.

Audit:

For Windows and Linux:

1. Attach to the DB2 instance.

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

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

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

3. Locate this value in the output to find the 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
```

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

Profile Applicability:

- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

Restricting access to the contents of the secondary archive log directory will help ensure that the confidentiality, integrity, and availability of archive logs are protected.
Although there are many ways to ensure that your logs will be archived, we recommend using the value of DISK as part of the `logarchmeth2` parameter. This will properly ensure that the logs are archived. A finding of OFF is not acceptable.

Audit:

For Windows and Linux:

1. Attach to the DB2 instance.

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

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

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

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

```
db2 => get database manager configuration
db2 => ...
 Default database path (LOGARCHMETH2) = <valid directory>
```

Additional steps for Windows:

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

Additional steps for Linux:

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

```
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 <valid directory>
```

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

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

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

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

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

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

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

For Windows and Linux:

1. Attach to the DB2 instance.

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

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

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

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

```
db2 => get database manager configuration
db2 => ...
 Default database path (FAILARCHPATH) = <valid directory>
```

Additional steps for Windows:

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

Additional steps for Linux:

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

```
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)

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
```

3.1.20 Secure permissions for the log mirror location (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

For Windows and Linux, perform the following DB2 commands to obtain the directory location:

1. Connect to the DB2 database.

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

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

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

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

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

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

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 $DB2DATABASE 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 <valid path>
```

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
```

3.1.21 Establish retention set size for backups (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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 $DB2DATABASE user $USERNAME using $PASSWORD
```

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

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

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

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

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

Remediation:

1. Connect to the DB2 database

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

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

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

3.1.22 Set archive log failover retry limit (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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 $DB2DATABASE user $USERNAME using $PASSWORD
```

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

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

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

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

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

Remediation:

1. Connect to the DB2 database

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

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

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

3.2 Database Manager Configuration parameters

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

3.2.1 TCP/IP service name - svcename (Scored)

Profile Applicability:

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

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. The `svcename` parameter defines the port number or name 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).

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

Remediation:

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

```
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

3.2.2 SSL service name - ssl_svcename (Scored)

Profile Applicability:

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

Description:

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

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

Rationale:

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

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

Audit:

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

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

Remediation:

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

```
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

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

Profile Applicability:

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

Description:

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

Rationale:

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

Audit:

Run the following command to identify the current value of the `srvcon_auth` database configuration parameter:

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

Remediation:

Run the following command to update the current value of the `srvcon_auth` database configuration parameter to the correct value:

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

Default Value:

Not specified

References:

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

3.2.4 Database Manager Configuration parameter: trust_allclnts (Not Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

This parameter is used to determine where users are validated within the database environment (client-side authentication). If the parameter is set to 'YES', the server assumes that the client side is handling authentication to the database. If the parameter is set to 'NO', the client must provide authentication to the server on behalf of the user.

This parameter is only active when the `authentication` parameter is set to CLIENT.

Rationale:

This parameter is relied upon to determine whether each user (client) needs to be authenticated by the server or if the server should assume that each user (client) has already been sufficiently authenticated.

Audit:

Issue the following command to check the value of the parameter:

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

The value should be 'YES' for client-side authentication and 'NO' for server-side authentication.

Remediation:

To specify client-side authentication, issue the following command to set the parameter to 'YES':

```
db2=> update dbm cfg using trust_allclnts YES
```

To specify server-side authentication, issue the following command to set the parameter to 'NO':

```
db2=> update dbm cfg using trust_allclnts NO
```

References:

1. http://www.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.config.doc/doc/r0000380.html

3.2.5 Database Manager Configuration parameter: trust_clntauth (Not Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

This parameter specifies where a trusted client is authenticated (at the server or the client) if it provides a user ID and password.

If the parameter is set to 'CLIENT', the user ID and password are not needed, but if they are provided, authentication will occur at the client.

If the parameter is set to 'SERVER', the user ID and password are needed and will be authenticated at the server.

This parameter is only active if the `authentication` parameter is set to 'CLIENT'.

Rationale:

This parameter is relied upon to determine whether each trusted client needs to be authenticated by the server or the client after providing a user ID and password.

Audit:

Issue the following command to check the value of the parameter:

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

The value should be 'CLIENT' for client-side authentication and 'SERVER' for server-side authentication.

Remediation:

Issue the following command to set the parameter to 'CLIENT' or 'SERVER':

```
db2=> update dbm cfg using trust_clntauth <CLIENT/SERVER>
```

References:

1. http://www.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.config.doc/doc/r0000381.html

4 Row and Column Access Control (RCAC)

DB2 RCAC controls access to a table at the row and column level. Row and column access control is sometimes referred to as fine-grained access control or FGAC. Identify and gather the organization's security policies, management and staff roles, and user and group lists to compare against existing DB2 RCAC policies for compliance.

4.1 Review Organization's Policies against DB2 RCAC Policies (Not Scored)

Profile Applicability:

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

Description:

DB2 Row and Column Access Control (RCAC) Policies control access to DB2 tables. They should match the organization's security and database access policies, and they should be regularly reviewed for gaps.

Rationale:

Missing, incomplete, or incorrect DB2 RCAC policies will increase the risks to the organization's protected data and will prevent efforts to monitor, alert, and respond to these risks in the future.

Audit:

Schedule and complete a regular review of all organization security and data access database policies against the current DB2 policies to determine if gaps exist.

1. Identify each written organization policy.
2. Find the matching DB2 RCAC policy.
3. Determine if the RCAC policy applies and correctly supports the written policy.
4. If no matching DB2 RCAC policy is found, record a 'gap' for future remediation.

Remediation:

1. Create RCAC policies for each 'gap' listed from the Audit procedure.

2. Review the newly created DB2 RCAC policy against the organization's written policies.

Default Value:

Not installed

References:

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

4.2 Secure SECADM Authority (Not Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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:

It is important to consider reviewing the members of the `SECADM` authority when implementing this recommendation. Such consideration of this review is addressed in Section 7.5 of this document.

Remediation:

It is important to consider reviewing the members of the `SECADM` authority when implementing this recommendation. Such consideration of this review is addressed in Section 7.5 of this document.

References:

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

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

Profile Applicability:

- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

With row and column access control, individuals are permitted access to only the subset of data that is required to perform their job tasks. Periodic review of these individuals is crucial when trying to keep data secure. As business needs move forward, requirements behind accessing the data may change, leading to a revision in security policy. By inspecting the list of users, groups, and roles, you are identifying excessive privileges that could pose possible security threats within your infrastructure.

Rationale:

If a user (either by himself or part of a group or role) is no longer required access to the data that is protected by row and column access controls, allowing that individual to maintain access allows that individual to compromise the confidentiality, integrity, and/or availability of the data in the DB2 instance.

Audit:

1. Review the users within your database environment:

Linux:

```
cat /etc/passwd
```

Windows:

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

2. Review the groups within your database environment:

Linux:

```
cat /etc/group
```

Windows:

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

3. Review the roles and role members within your database environment:

a. Attach to DB2 Instance:

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

b. Connect to DB2 database:

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

c. Run the command:

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

Remediation:

1. To remove users from your database environment:

Linux:

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

Windows:

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

2. To remove groups from your database environment:

Linux:

```
groupdel <group name>
```

Windows:

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

3. To remove roles or role members from your database environment

a. Attach to DB2 Instance:

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

b. Connect to DB2 database:

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

c. To remove role members from roles:

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

d. To remove roles:

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

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

Profile Applicability:

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

Description:

The logic behind instituting row permissions is crucial for a successful security policy. Inspecting this logic and comparing it to the security policy will assure that all aspects of the data access controls are being adhered to.

Rationale:

Missing or incomplete DB2 RCAC Security Policies will increase the risks to the organization's protected data and will prevent efforts to monitor, alert, and respond to these risks in the future.

Audit:

1. Attach to the DB2 Instance:

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

2. Connect to database environment:

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

3. Run the following and review the results to confirm that the row permissions are correct and that they comply with the existing security policy:

```
db2 => select role.rolename, control.ruletext from syscat.roles role inner join
syscat.controls control on locate(role.rolename,control.ruletext) <> 0 where
enable = 'Y' and enforced = 'A' and valid = 'Y' and controltype = 'R'
```

Remediation:

1. Create RCAC Policies for each 'gap' listed from the Audit procedure.
2. Review the newly created DB2 RCAC policy against the organization's policy

References:

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

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

Profile Applicability:

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

Description:

The logic behind instituting column masks is crucial for a successful security policy. Inspecting this logic and comparing it to the security policy will assure that all aspects of the data access controls are being adhered to.

Rationale:

Missing or incomplete DB2 RCAC security policies will increase the risks to the organization's protected data and will prevent efforts to monitor, alert, and respond to these risks in the future.

Audit:

1. Attach to the DB2 Instance:

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

2. Connect to database environment:

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

3. Run the following and review the results to verify that the permissions are correct and that they comply with the organization's existing security policy:

```
db2 => select role.rolename, control.colname, control.ruletext from
syscat.roles role inner join syscat.controls control on
locate(role.rolename,control.ruletext) <> 0 where enable = 'Y' and enforced =
'A' and valid = 'Y' and controltype = 'C'
```

Remediation:

1. Create RCAC Policies for each 'gap' listed from the Audit procedure.

2. Review the newly created DB2 RCAC policy against the organization's written policy.

References:

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

5 Database Maintenance

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

5.1 Enable Backup Redundancy (Not Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

Backup redundancy ensures that multiple instances of database backups exist.

Rationale:

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

Audit:

Review the replication of your backups based on organization policy.

Remediation:

Define and implement a process to replicate your backups onto multiple locations.

5.2 Protecting Backups (Not Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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 $DB2DATABASE user $USERNAME using $PASSWORD
```

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

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

3. Locate this value in the output:

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

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

Remediation:

1. Connect to the DB2 database:

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

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

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

6 Securing 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
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.2 Restrict Access to SYSCAT.AUDITUSE (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

6.3 Restrict Access to SYSCAT.DBAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.4 Restrict Access to SYSCAT.COLAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

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

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

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

References:

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

6.5 Restrict Access to SYSCAT.EVENTS (Scored)

Profile Applicability:

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

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.6 Restrict Access to SYSCAT.EVENTTABLES (Scored)

Profile Applicability:

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

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.7 Restrict Access to SYSCAT.ROUTINES (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

6.8 Restrict Access to SYSCAT.INDEXAUTH (Scored)

Profile Applicability:

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

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.9 Restrict Access to SYSCAT.PACKAGEAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

The `SYSCAT.PACKAGEAUTH` view contains a list of users or groups that has `EXECUTE` privilege on a package. It is recommended that the `PUBLIC` role be restricted from accessing this view.

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

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

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

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

6.10 Restrict Access to SYSCAT.PACKAGES (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

6.11 Restrict Access to SYSCAT.PASSTHRUAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from PUBLIC.

1. Connect to the DB2 database.

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

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

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

References:

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

6.12 Restrict Access to SYSCAT.SECURITYPOLICIES (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.13 Restrict Access to SYSCAT.SECURITYPOLICYEXEMPTIONS (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.14 Restrict Access to SYSCAT.SURROGATEAUTHIDS (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

The `SYSCAT.SURROGATEAUTHIDS` contains the names of all accounts that have been granted `SETSESSIONUSER` privilege on a user or to `PUBLIC`. It is recommended that the `PUBLIC` role be restricted from accessing this view.

Rationale:

Public should not be able to view the names of all the surrogate accounts with `SETSESSIONUSER` privilege.

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.15 Restrict Access to SYSCAT.ROLEAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.16 Restrict Access to SYSCAT.ROLES (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.17 Restrict Access to SYSCAT.ROUTINEAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.18 Restrict Access to SYSCAT.SCHEMAAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

6.19 Restrict Access to SYSCAT.SCHEMATA (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.20 Restrict Access to SYSCAT.SEQUENCEAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.21 Restrict Access to SYSCAT.STATEMENTS (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

The `SYSCAT.STATEMENTS` view contains all SQL statements of a compiled package. It is recommended that the `PUBLIC` role be restricted from accessing this view.

Rationale:

`PUBLIC` should not have access to the SQL statements of a database package. This could lead to an exploit.

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.22 Restrict Access to SYSCAT.TABAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

The `SYSCAT.TABAUTH` view contains users or groups that have been granted one or more privileges on a table or view. It is recommended that the `PUBLIC` role be restricted from accessing this view.

Rationale:

`PUBLIC` should not have access to the grants of views and tables in a database. This could lead to an exploit.

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.23 Restrict Access to SYSCAT.TBSPACEAUTH (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

The `SYSCAT.TBSPACEAUTH` contains users or groups that have been granted the USE privilege on a particular tablespace in the database. It is recommended that the `PUBLIC` role be restricted from accessing this view.

Rationale:

`PUBLIC` should not have access to the grants of the tablespaces in a database. This could lead to an exploit.

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.24 Restrict Access to Tablespaces (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

A tablespace is where the data is physically stored. It is recommended that tablespace usage be restricted to authorized users.

Rationale:

Grant the `USE` of tablespace privilege to only authorized users. Restrict the privilege from `PUBLIC`, where applicable, as a malicious user can cause a denial of service at the tablespace level by overloading it with corrupted data.

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.25 Restrict Access to SYSCAT.MODULEAUTH (Scored)

Profile Applicability:

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

Description:

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

Rationale:

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

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.26 Restrict Access to SYSCAT.VARIABLEAUTH (Scored)

Profile Applicability:

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.27 Restrict Access to SYSCAT.WORKLOADAUTH (Scored)

Profile Applicability:

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

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&lang=en

6.28 Restrict Access to SYSCAT.XSROBJECTAUTH (Scored)

Profile Applicability:

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

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

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&lang=en

6.29 Restrict Access to SYSCAT.AUTHORIZATIONIDS (Scored)

Profile Applicability:

- Level 1 – RDBMS
- Level 2 – RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

SYSCAT.AUTHORIZATIONIDS is an administrative view for the currently connected server.

Rationale:

Databases created without the RESTRICT option automatically GRANT the SELECT privilege to PUBLIC for SYSCAT views. Therefore, it is strongly recommended to explicitly REVOKE the SELECT privilege on the SYSCAT.AUTHORIZATIONIDS 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 rtrim(grantee) as grantee, controlauth, alterauth, deleteauth,
indexauth, insertauth, selectauth, updateauth, refauth from sysibm.systabauth
where tcreator = 'SYSCAT' and ttname = 'AUTHORIZATIONIDS'
```

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

References:

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

6.30 Restrict Access to SYSIBMADM.OBJECTOWNERS (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 – RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

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&lang=en

6.31 Restrict Access to SYSIBMADM.PRIVILEGES (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 – RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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

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

Remediation:

Perform the following to revoke access from `PUBLIC`.

1. Connect to the DB2 database.

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

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

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

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&lang=en

7 DB2 Authorities

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

7.1 Secure SYSADM authority (Scored)

Profile Applicability:

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

Description:

The `sysadm_group` parameter defines the system administrator group (SYSADM) authority. It is recommended that the `sysadm_group` group contains authorized users only.

Rationale:

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

Audit:

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

1. Attach to the DB2 instance.

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

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

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

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

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

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

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

   Linux:

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

   Windows:

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

Remediation:

Define a valid group name for the SYSADM group.

1. Attach to the DB2 instance.

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

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

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

Default Value:

The default value for `sysadm_group` is NULL.

7.2 Secure SYSCTRL authority (Scored)

Profile Applicability:

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

Description:

The `sysctrl_group` parameter defines the system administrator group with system control (SYSCTRL) authority. It is recommended that the `sysctrl_group` group contains authorized users only.

Rationale:

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

Audit:

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

1. Attach to the DB2 instance.

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

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

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

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

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

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

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

   Linux:

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

   Windows:

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

Remediation:

Define a valid group name for the SYSCTRL group.

1. Attach to the DB2 instance.

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

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

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

Default Value:

The default value for `sysctrl_group` is NULL.

7.3 Secure SYSMAINT Authority (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 – RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

Rationale:

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

Audit:

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

1. Attach to the DB2 instance.

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

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

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

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

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

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

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

   Linux:

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

   Windows:

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

Remediation:

Define a valid group name for the SYSMAINT group.

1. Attach to the DB2 instance.

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

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

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

Default Value:

The default value for `sysmaint_group` is NULL.

7.4 Secure SYSMON Authority (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 – RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

The `sysmon_group` parameter defines the operating system groups with system monitor (SYSMON) authority. It is recommended that the `sysmon_group` group contain authorized users only.

Rationale:

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

Audit:

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

1. Attach to the DB2 instance.

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

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

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

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

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

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

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

   Linux:

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

   Windows:

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

Remediation:

Define a valid group name for the SYSMON group.

1. Attach to the DB2 instance.

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

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

   ```
   db2 => update database manager configuration using sysmon_group <sys monitor
   group name>
   ```

Default Value:

The default value for `sysmon_group` is NULL.

7.5 Secure SECADM Authority (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 – RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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 $DB2DATABASE user $USERNAME using $PASSWORD
```

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

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
securityadmauth = 'Y'
```

3. Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the DB2 database.

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

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

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

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0021054.html?lang=en

7.6 Secure DBADM Authority (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 – RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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 $DB2DATABASE user $USERNAME using $PASSWORD
```

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

```
db2 => select distinct grantee, granteetype from syscat.dbauth where dbadmauth
= 'Y'
```

3. Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the DB2 database.

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

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

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

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0005521.html?lang=en

7.7 Secure SQLADM Authority (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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

7.8 Secure DATAACCESS Authority (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

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

The DATAACCESS authority cannot be granted to PUBLIC.

Rationale:

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

Audit:

1. Run the following command 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

7.9 Secure ACCESSCTRL Authority (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

ACCESSCTRL authority is the authority required to grant and revoke privileges on objects within a specific database. Some of these privileges include BINDADD, CONNECT, CREATETAB, CREATE_EXTERNAL_ROUTINE, LOAD, and QUIESCE_CONNECT. It has no inherent privilege to access data stored in tables, except the catalog tables and views.

The ACCESSCTRL authority cannot be granted to PUBLIC.

Rationale:

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

Audit:

1. Run the following command 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

7.10 Secure WLMADM authority (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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

7.11 Secure CREATAB Authority (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

The `CREATAB` (create table) role grants the authority to a user to create tables within a specific database. It is recommended that the `CREATAB` role be granted to authorized users only.

Rationale:

Review all users that have access to this authority to avoid the addition of unnecessary and/or inappropriate users.

Audit:

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

1. Connect to the DB2 database.

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

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

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
creatabauth = 'Y'
```

3. Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the DB2 database.

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

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

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

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0054269.html?lang=en

7.12 Secure BINDADD Authority (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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 $DB2DATABASE user $USERNAME using $PASSWORD
```

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

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
bindaddauth = 'Y'
```

3. Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the DB2 database.

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

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

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

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0005524.html?lang=en

7.13 Secure CONNECT Authority (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

The `CONNECT` role grants the authority to a user to connect to mainframe and midrange databases from Windows, Unix, and Linux operating systems. It is recommended that the `CONNECT` role be granted to authorized users only.

Rationale:

All users that have access to this authority should be regularly reviewed.

Audit:

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

1. Connect to the DB2 database.

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

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

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
connectauth = 'Y'
```

3. Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the DB2 database.

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

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

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

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.qb.dbconn.doc/doc/r0059046.html?cp=SSEPGG_10.5.0%2F6&lang=en

7.14 Secure LOAD Authority (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

The `LOAD` role grants the authority to a user to load data into tables. It is recommended that the `LOAD` role be granted to authorized users only.

Rationale:

All users that have access to this authority should be regularly reviewed.

Audit:

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

1. Connect to the DB2 database.

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

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

```
db2 => select distinct grantee, granteetype from syscat.dbauth where loadauth =
'Y'
```

3. Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the DB2 database.

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

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

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

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0005522.html?lang=en

7.15 Secure EXTERNALROUTINE Authority (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

The `EXTERNALROUTINE` authority grants a user the privilege to create user-defined functions and procedures in a specific database.

Rationale:

All users with this authority should be regularly reviewed and approved.

Audit:

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

1. Connect to the DB2 database.

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

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

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
externalroutineauth = 'Y'
```

3. Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the DB2 database.

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

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

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

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.apdv.routines.doc/doc/c0009198.html?lang=en

7.16 Secure QUIESCECONNECT Authority (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

The `QUIESCECONNECT` role grants the authority to a user to access a database even in the quiesced state.

Rationale:

It is recommended that the `QUIESCECONNECT` role be granted to authorized users only.

Audit:

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

1. Connect to the DB2 database.

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

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

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
quiesceconnectauth = 'Y'
```

3. Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the DB2 database.

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

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

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

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.apdv.api.doc/doc/r0009331.html?lang=en

8 DB2 Roles

Roles simplify the administration and management of privileges by offering an equivalent capability as groups but without the same restrictions. A role is a database object that groups together one or more privileges and can be assigned to users, groups, PUBLIC, or other roles by using a GRANT statement. All the roles assigned to a user are enabled when that user establishes a connection, so all privileges and authorities granted to roles are taken into account when a user connects. Roles cannot be explicitly enabled or disabled.

8.1 Review Roles (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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

8.2 Review Role Members (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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 role members are correct for each role:

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

Remediation:

To remove a role member from a particular role:

1. 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

8.3 Nested Roles (Scored)

Profile Applicability:

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

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>
```

8.4 Review Roles granted to PUBLIC (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

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 a role 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 PUBLIC
```

References:

1. https://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0050531.html

8.5 Review Role Grantees with WITH ADMIN OPTION (Scored)

Profile Applicability:

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

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 query:

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

9 General Policy and Procedures

[This space intentionally left blank]

9.1 Start and Stop DB2 Instance (Not Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Windows Host OS
- Level 2 - Linux Host OS

Description:

The DB2 instance manages the database environment and sets the configuration parameters. It is recommended that only administrators are allowed to start and stop the DB2 instance.

Rationale:

Only privileged users should have access to start and stop the DB2 instance. This will ensure that the DB2 instance is controlled by authorized administrators.

Audit:

On Windows:

1. Go to Start, then to the Run option.
2. Type in `services.msc` in the command prompt.
3. Locate the DB2 service and identify the users/groups that can start and stop the service.

On Linux:

1. Identify the name of the local DB2 admin group.
2. Identify the members of that group.
3. Identify the members that have access to stop and start the DB2 instance.

Remediation:

Revoke access from any unnecessary users.

1. Connect to the host
2. Review users and groups that have access to start and stop the DB2 instance.
3. Remove start and stop privileges from all users and groups that should not have them.

9.2 Remove Unused Schemas (Not Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

A schema is a logical grouping of database objects. It is recommended that unused schemas be removed from the database.

Rationale:

Unused schemas can be left unmonitored and may be subjected to abuse and therefore should be removed.

Audit:

1. Connect to the DB2 database.

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

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

```
db2 => select schemaname from syscat.schemata
```

3. Review the list of schemas

Remediation:

Remote unnecessary 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 => drop scheme <scheme name> restrict
```

3. Review unused schemas and remove if necessary

9.3 Review System Tablespaces (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

System tablespaces store all system object data within that database. It is recommended that system tablespaces are used to stored system data only and not user data.

Rationale:

Users should not have privileges to create user data objects within the system tablespaces. User data objects created within the system tablespaces should be removed.

Audit:

1. Connect to the DB2 database.

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

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

```
db2 => select tabschema,tabname,tbspace from syscat.tables where tabschema not
in ('ADMINISTRATOR','SYSIBM','SYSTOOLS') and tbspace in
('SYSCATSPACE','SYSTOOLSPACE','SYSTOOLSTMPSPACE','TEMPSPACE')
```

3. Review the list of system tablespaces. If the output is BLANK, that is considered a successful finding.

Remediation:

1. Connect to the DB2 database.

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

2. Review the system tablespaces to identify any user data objects within them.
3. Drop, migrate, or otherwise remove all user data objects (tables, schemas, etc.) from within the system tablespaces.
4. Revoke write access for the system tablespaces from all users.

9.4 Remove Default Databases (Scored)

Profile Applicability:

- Level 1 - RDBMS
- Level 2 - RDBMS
- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

A DB2 instance may come installed with default databases. It is recommended that the `SAMPLE` database be removed.

Rationale:

Removing unused, well-known databases will reduce the attack surface of the system.

Audit:

Perform the following DB2 commands to obtain the list of databases:

1. Attach to the DB2 instance.

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

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

```
db2 => list database directory
```

3. Locate this value in the output:

```
db2 =>
Database 3 entry:
 Database alias                       = SAMPLE
 Database name                        = SAMPLE
 Local database directory             = C:
 Database release level               = c.00
 Comment = Directory entry type       = Indirect
 Catalog database partition number    = 0
 Alternate server hostname            =
```

4. Review the output above and identify the SAMPLE database. If there is no SAMPLE database, then it is considered a successful finding.

Remediation:

Drop unused sample databases:

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

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

9.5 Enable SSL communication with LDAP server (Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - 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 between the DB2 instance and the LDAP server.

Note: The file is located under `INSTANCE_HOME/sqllib/cfg/`, for Linux; and `%DB2PATH%\cfg\`, for Windows.

Audit:

Perform the following commands to obtain the parameter setting:

1. Connect to the DB2 host
2. Edit the `IBMLDAPSecurity.ini` file
3. Verify the existence of this parameter:

```
ENABLE_SSL = TRUE
```

Remediation:

Verify the parameter:

1. Connect to the DB2 host
2. Edit the `IBMLDAPSecurity.ini` file
3. Add or modify the file to include the following parameter:

```
ENABLE_SSL = TRUE
```

Default Value:

The default value is the omission of this parameter.

9.6 Secure the permission of the IBMLDAPSecurity.ini file (Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

The IBMLDAPSecurity.ini file contains the IBM LDAP security plug-in configurations.

Rationale:

Recommended value is read and write access to DB2 administrators only and read-only to Everyone/Other/Users/Domain Users. This will ensure that the parameter file is protected.
Note: the file is located under `INSTANCE_HOME/sqllib/cfg/`, for Linux; and `%DB2PATH%\cfg\`, for Windows.

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 for all accounts

For Linux:

1. Connect to the DB2 host
2. Change to the file directory
3. Check the permissions 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 administrator accounts and grant them *Read* and *Write* authority only (revoke all others).
6. Select all non-administrator accounts and grant them *Read* authority only (revoke all others).

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 664
```

9.7 Secure the permission of the SSLconfig.ini file (Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - 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 full access to DB2 administrators only, read and write access only to members of the SYSADM group, and no access to other users. This will ensure that the parameter file is protected.

Note: the file is located under `INSTANCE_HOME/sqllib/cfg/`, for Linux; and `%INSTHOME%\`, for Windows.

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 for all accounts

For Linux:

1. Connect to the DB2 host
2. Change to the file directory
3. Check the permissions 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 administrator accounts and grant them the *Full Control* authority
6. Select the SYSADM group and grant it *Read* and Write authority only (revoke all others)
7. Select all other accounts and revoke all privileges to the directory

For Unix:

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

```
OS => chmod -R 760
```

9.8 Ensure Trusted Contexts are enabled (Not Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

A Trusted Context object provides a means of enforcing encryption, assigning privileges based on roles, and ensuring that the actions performed on behalf of a user are performed in the context of the user's ID and privileges.

Rationale:

Creating Trusted Context objects to enforce encryption and assign roles will protect data in transit and limit access to information on a per user/role basis. Additionally, it ensures actions can be traced back to the user.

Audit:

Issue the following command to verify that a Trusted Context object is enabled:

```
select contextname, enabled from syscat.contexts where enabled = 'Y'
```

Remediation:

If there is no enabled Trusted Context object, create a Trusted Context object if needed and enable it.

References:

1. http://www.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0050514.html

9.9 Secure plug-in library locations (Not Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

Whether developing your own security plug-ins or migrating established plug-ins into your environment, it is important to ensure that the plug-in directories are secure.

Rationale:

If plug-in directories are not secure, the plug-ins could be misused, tampered with, or otherwise accessed in ways that could jeopardize the security of the server.

Audit:

Linux 32-bit:

Review the privileges assigned to the plug-in directories to ensure they are set to 755.

- For client-side plug-ins: $DB2PATH/security32/plugin/client
- For server-side plug-ins: $DB2PATH/security32/plugin/server
- For group plug-ins: $DB2PATH/security32/plugin/group

Linux 64-bit:

Review the privileges assigned to the plug-in directories to ensure they are set to 755.

- For client-side plug-ins: $DB2PATH/security64/plugin/client
- For server-side plug-ins: $DB2PATH/security64/plugin/server
- For group plug-ins: $DB2PATH/security64/plugin/group

Windows 32-bit and 64-bit:

Review the privileges assigned to the plug-in directories to ensure they are set to 755. Note: The sub-directories 'instance name' and 'client', 'server', and 'group' are not created automatically. The instance owner has to manually create them.

- For client-side plug-ins: $DB2PATH\security\plugin\instance name\client
- For server-side plug-ins: $DB2PATH\security\plugin\instance name\server
- For group plug-ins: $DB2PATH\security\plugin\instance name\group

Remediation:

Change the privileges for all plug-in directories so they are set to 755.

On a Linux system, perform the following for each directory needing its privileges changed:

```
[db2inst1@tgt-db2-101-abc123 IBM]$ chmod 755 <directory>
```

9.10 Ensure that security plug-in support for two-part user IDs is enabled (Not Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

The DB2 database manager on Windows supports the use of two-part user IDs and the mapping of two-part user IDs to two-part authorization IDs.

Rationale:

Having a two-part authorization scheme increases the security of user IDs by making them harder to compromise.

Audit:

Issue the following command and confirm that the `clnt_pw_plugin`, `srvcon_gssplugin_list`, and `srvcon_pw_plugin` parameters are all set to 'DISABLED':

```
db2=> select name, case when ((name = 'srvcon_pw_plugin' AND value in
('IBMOSauthserverTwoPart','IBMOSauthserverTwoPart64')) AND (name = 'clnt_pw_plugin'
and value in ('IBMOSauthclientTwoPart','IBMOSauthclientTwoPart64'))) OR ((name =
'srvcon_gssplugin_list' AND value in ('IBMOSkrb5TwoPart','IBMOSkrb5TwoPart64')) AND
(name = 'clnt_krb_plugin' and value in ('IBMkrb5TwoPart','IBMkrb5TwoPart64')))then
'ENABLED' else 'DISABLED' end as Status from sysibmadm.dbmcfg where (name =
'srvcon_pw_plugin' OR name = 'srvcon_gssplugin_list' OR name = 'clnt_pw_plugin')
```

Remediation:

To enable server authentication that maps two-part user IDs to two-part authorization IDs, you must set:

- `srvcon_pw_plugin` to IBMOSauthserverTwoPart
- `clnt_pw_plugin` to IBMOSauthclientTwoPart

To enable client authentication that maps two-part user IDs to two-part authorization IDs, you must set:

- `srvcon_pw_plugin` to IBMOSauthserverTwoPart
- `clnt_pw_plugin` to IBMOSauthclientTwoPart

To enable Kerberos authentication that maps two-part user IDs to two-part authorization IDs, you must set:

- `srvcon_gssplugin_list` to IBMOSkrb5TwoPart
- `clnt_krb_plugin` to IBMkrb5TwoPart

For example:

```
db2=> update dbm cfg using srvcon_pw_plugin IBMOSauthserverTwoPart
```

References:

1. http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0012039.html

9.11 Ensure permissions on communication exit library locations (Not Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

DB2 communication exit libraries must exist in specific directories. There should be proper permissions on these directories.

Rationale:

If the permissions on the DB2 communication exit library directories are not set properly, the contents of those directories could be misused, tampered with, or otherwise accessed to negatively impact the security of the server.

Audit:

Linux 64-bit:

Issue the following command to check the permissions for the communication exit library:

```
[db2inst1@tgt-db2-101-abcd plugin]$ ll /opt/ibm/db2/V10.5/security64/plugin
total 12
drwxr-x--- 2 db2iadm1 db2inst1 4096 Aug 17  2013 commexit
```

Remediation:

The database manager looks for communication exit libraries in the following directories:

- Linux 32-bit: $DB2PATH/security32/plugin/commexit
- Linux 64-bit: $DB2PATH/security64/plugin/commexit
- Windows 32-bit and 64-bit: $DB2PATH\security\plugin\commexit\instance_name

After locating the directory, update its permissions. The following is an example for a Linux 64-bit system:

```
[db2inst1@tgt-db2-101-abcd plugin]$ pwd
/opt/ibm/db2/V10.5/security64/plugin
[db2inst1@tgt-db2-101-abcd IBM]$ chmod -R 750 commexit
```

References:

1. http://www.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.admin.sec.doc/doc/c0060264.html

9.12 Ensure audit policies are enabled within the database (Not Scored)

Profile Applicability:

- Level 1 - Windows Host OS
- Level 2 - Windows Host OS
- Level 1 - Linux Host OS
- Level 2 - Linux Host OS

Description:

Creating and applying audit policies is crucial for securing and discovering issues within your databases. Audit policies can help trigger events for changes to data objects, table DML, and user access.

Rationale:

If audit policies are not enabled, issues may go undiscovered, and compromises and other incidents may occur without being quickly detected. It may also not be possible to provide evidence of compliance with security laws, regulations, and other requirements.

Audit:

Issue the following command to ensure that at least one audit policy returns an `auditstatus` not equal to 'N'. The assumption is that if there is an active policy, then information is being captured to audit.

```
db2=> select auditpolicyname, auditstatus from syscat.auditpolicies
```

Remediation:

Issue the following command to create an audit policy:

```
db2=> create audit policy AUDIT_TEST CATEGORIES ALL STATUS BOTH
```

References:

1. http://www.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.sql.ref.doc/doc/r0050607.html

Appendix: Summary Table

Control		Set Correctly	
		Yes	No
1	**Installation and Patches**		
1.1	Install the latest fix packs (Not 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)	☐	☐
1.5	Configure DB2 to use non-standard ports (Not Scored)	☐	☐
1.6	Creating the database with the RESTERICTIVE clause (Not Scored)	☐	☐
2	**DB2 Directory and File Permissions**		
2.1	Secure DB2 Runtime Library (Scored)	☐	☐
2.2	Secure the database container directory (Scored)	☐	☐
2.3	Set umask value for DB2 admin user .profile file (Scored)	☐	☐
2.4	Verify the groups within the DB2_GRP_LOOKUP environment variable are appropriate (Windows only) (Not Scored)	☐	☐
2.5	Verify the domains within the DB2DOMAINLIST environment variable are appropriate (Windows only) (Not 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 datalinks support (Scored)	☐	☐
3.1.5	Secure permissions for default database file path (Scored)	☐	☐
3.1.6	Set diagnostic logging to capture errors and warnings (Scored)	☐	☐
3.1.7	Secure permissions for all diagnostic logs (Scored)	☐	☐
3.1.8	Require instance name for discovery requests (Scored)	☐	☐
3.1.9	Disable instance discoverability (Scored)	☐	☐
3.1.10	Authenticate federated users at the instance level (Scored)	☐	☐
3.1.11	Set maximum connection limits (Scored)	☐	☐
3.1.12	Set administrative notification level (Scored)	☐	☐
3.1.13	Enable server-based authentication (Scored)	☐	☐
3.1.14	Set failed archive retry delay (Scored)	☐	☐
3.1.15	Auto-restart after abnormal termination (Scored)	☐	☐
3.1.16	Disable database discovery (Scored)	☐	☐
3.1.17	Secure permissions for the primary archive log location (Scored)	☐	☐

3.1.18	Secure permissions for the secondary archive log location (Scored)	☐	☐
3.1.19	Secure permissions for the tertiary archive log location (Scored)	☐	☐
3.1.20	Secure permissions for the log mirror location (Scored)	☐	☐
3.1.21	Establish retention set size for backups (Scored)	☐	☐
3.1.22	Set archive log failover retry limit (Scored)	☐	☐
3.2	**Database Manager Configuration parameters**		
3.2.1	TCP/IP service name - svcename (Scored)	☐	☐
3.2.2	SSL service name - ssl_svcename (Scored)	☐	☐
3.2.3	Authentication type for incoming connections at the server - srvcon_auth (Scored)	☐	☐
3.2.4	Database Manager Configuration parameter: trust_allclnts (Not Scored)	☐	☐
3.2.5	Database Manager Configuration parameter: trust_clntauth (Not Scored)	☐	☐
4	**Row and Column Access Control (RCAC)**		
4.1	Review Organization's Policies against DB2 RCAC Policies (Not Scored)	☐	☐
4.2	Secure SECADM Authority (Not Scored)	☐	☐
4.3	Review Users, Groups, and Roles (Not Scored)	☐	☐
4.4	Review Row Permission logic according to policy (Not Scored)	☐	☐
4.5	Review Column Mask logic according to policy (Not Scored)	☐	☐
5	**Database Maintenance**		
5.1	Enable Backup Redundancy (Not Scored)	☐	☐
5.2	Protecting Backups (Not Scored)	☐	☐
5.3	Enable Automatic Database Maintenance (Scored)	☐	☐
6	**Securing Database Objects**		
6.1	Restrict Access to SYSCAT.AUDITPOLICIES (Scored)	☐	☐
6.2	Restrict Access to SYSCAT.AUDITUSE (Scored)	☐	☐
6.3	Restrict Access to SYSCAT.DBAUTH (Scored)	☐	☐
6.4	Restrict Access to SYSCAT.COLAUTH (Scored)	☐	☐
6.5	Restrict Access to SYSCAT.EVENTS (Scored)	☐	☐
6.6	Restrict Access to SYSCAT.EVENTTABLES (Scored)	☐	☐
6.7	Restrict Access to SYSCAT.ROUTINES (Scored)	☐	☐
6.8	Restrict Access to SYSCAT.INDEXAUTH (Scored)	☐	☐
6.9	Restrict Access to SYSCAT.PACKAGEAUTH (Scored)	☐	☐
6.10	Restrict Access to SYSCAT.PACKAGES (Scored)	☐	☐
6.11	Restrict Access to SYSCAT.PASSTHRUAUTH (Scored)	☐	☐
6.12	Restrict Access to SYSCAT.SECURITYPOLICIES (Scored)	☐	☐
6.13	Restrict Access to SYSCAT.SECURITYPOLICYEXEMPTIONS (Scored)	☐	☐

6.14	Restrict Access to SYSCAT.SURROGATEAUTHIDS (Scored)	☐	☐
6.15	Restrict Access to SYSCAT.ROLEAUTH (Scored)	☐	☐
6.16	Restrict Access to SYSCAT.ROLES (Scored)	☐	☐
6.17	Restrict Access to SYSCAT.ROUTINEAUTH (Scored)	☐	☐
6.18	Restrict Access to SYSCAT.SCHEMAAUTH (Scored)	☐	☐
6.19	Restrict Access to SYSCAT.SCHEMATA (Scored)	☐	☐
6.20	Restrict Access to SYSCAT.SEQUENCEAUTH (Scored)	☐	☐
6.21	Restrict Access to SYSCAT.STATEMENTS (Scored)	☐	☐
6.22	Restrict Access to SYSCAT.TABAUTH (Scored)	☐	☐
6.23	Restrict Access to SYSCAT.TBSPACEAUTH (Scored)	☐	☐
6.24	Restrict Access to Tablespaces (Scored)	☐	☐
6.25	Restrict Access to SYSCAT.MODULEAUTH (Scored)	☐	☐
6.26	Restrict Access to SYSCAT.VARIABLEAUTH (Scored)	☐	☐
6.27	Restrict Access to SYSCAT.WORKLOADAUTH (Scored)	☐	☐
6.28	Restrict Access to SYSCAT.XSROBJECTAUTH (Scored)	☐	☐
6.29	Restrict Access to SYSCAT.AUTHORIZATIONIDS (Scored)	☐	☐
6.30	Restrict Access to SYSIBMADM.OBJECTOWNERS (Scored)	☐	☐
6.31	Restrict Access to SYSIBMADM.PRIVILEGES (Scored)	☐	☐
7	**DB2 Authorities**		
7.1	Secure SYSADM authority (Scored)	☐	☐
7.2	Secure SYSCTRL authority (Scored)	☐	☐
7.3	Secure SYSMAINT Authority (Scored)	☐	☐
7.4	Secure SYSMON Authority (Scored)	☐	☐
7.5	Secure SECADM Authority (Scored)	☐	☐
7.6	Secure DBADM Authority (Scored)	☐	☐
7.7	Secure SQLADM Authority (Scored)	☐	☐
7.8	Secure DATAACCESS Authority (Scored)	☐	☐
7.9	Secure ACCESSCTRL Authority (Scored)	☐	☐
7.10	Secure WLMADM authority (Scored)	☐	☐
7.11	Secure CREATAB Authority (Scored)	☐	☐
7.12	Secure BINDADD Authority (Scored)	☐	☐
7.13	Secure CONNECT Authority (Scored)	☐	☐
7.14	Secure LOAD Authority (Scored)	☐	☐
7.15	Secure EXTERNALROUTINE Authority (Scored)	☐	☐
7.16	Secure QUIESCECONNECT Authority (Scored)	☐	☐
8	**DB2 Roles**		
8.1	Review Roles (Scored)	☐	☐
8.2	Review Role Members (Scored)	☐	☐
8.3	Nested Roles (Scored)	☐	☐
8.4	Review Roles granted to PUBLIC (Scored)	☐	☐
8.5	Review Role Grantees with WITH ADMIN OPTION (Scored)	☐	☐
9	**General Policy and Procedures**		

9.1	Start and Stop DB2 Instance (Not Scored)	☐	☐
9.2	Remove Unused Schemas (Not Scored)	☐	☐
9.3	Review System Tablespaces (Scored)	☐	☐
9.4	Remove Default Databases (Scored)	☐	☐
9.5	Enable SSL communication with LDAP server (Scored)	☐	☐
9.6	Secure the permission of the IBMLDAPSecurity.ini file (Scored)	☐	☐
9.7	Secure the permission of the SSLconfig.ini file (Scored)	☐	☐
9.8	Ensure Trusted Contexts are enabled (Not Scored)	☐	☐
9.9	Secure plug-in library locations (Not Scored)	☐	☐
9.10	Ensure that security plug-in support for two-part user IDs is enabled (Not Scored)	☐	☐
9.11	Ensure permissions on communication exit library locations (Not Scored)	☐	☐
9.12	Ensure audit policies are enabled within the database (Not Scored)	☐	☐

Appendix: Change History

Date	Version	Changes for this version
12-19-2015	1.0.0	Initial Release
08-05-2016	1.1.0	Ticket # 144: Added a recommendation for DB2 to use non-standard ports.
08-10-2016	1.1.0	Ticket # 141: Added a recommendation for Trusted Contexts.
08-10-2016	1.1.0	Ticket # 148: Added a recommendation for the trust_allclnts parameter.
08-10-2016	1.1.0	Ticket # 152: Added a recommendation for secure plug-in library locations.
08-10-2016	1.1.0	Ticket # 155: Added a recommendation for security plug-in support of two-part user IDs.
08-10-2016	1.1.0	Ticket # 156: Added a recommendation for the communication exit library location.
08-10-2016	1.1.0	Ticket # 157: Added a recommendation for enabling audit policies.
08-10-2016	1.1.0	Ticket # 159: Added a recommendation for the DB2_GRP_LOOKUP environmental variable.
08-10-2016	1.1.0	Ticket # 160: Added a recommendation for the DB2DOMAINLIST environmental variable.
08-17-2016	1.1.0	Ticket # 146: Added a recommendation for the RESTRICTIVE parameter.