CIS IBM DB2 10 Benchmark

1.1 INSTALL THE LATEST FIX PACKS

AUDIT:

Perform the following DB2 commands to obtain the version:

Open the DB2 Command Window and type in db2level:

COMMAND: $ 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

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.

REMEDIATION:

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

Default value:

IP address

1.3 Use non-default account names

AUDIT:

Windows:

Review the list of users for the system and confirm that none of the account names are db2admin, db2instl, dasusrl, or db2fenc1.

Linux:

Review /etc/passwd and confirm that none of the account names are db2admin, db2instl, dasusrl, or db2fencl.

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.

1.4 Configure DB2 to use non-standard ports

AUDIT:

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

Windows:

```
netstat -bao
```

Linux:

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

REMEDIATION:

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

1.5 Creating the database with the RESTRICTIVE clause

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.

2.1 Secure DB2 Runtime Library

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

```
COMMAND: 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

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

2.2 Secure the database container directory

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

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

Audit:

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

```
COMMAND: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.

2.5 Verify the domains within the DB2DOMAINLIST environment variable are appropriate

Audit:

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

```
COMMAND:db2set -all
```

Remediation:

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

3.1.1 Enable audit buffer

Audit:

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

1. Attach to the DB2 instance.
   ```
   COMMAND: db2 => attach to $DB2INSTANCE
   ```
2. Run the following command from the DB2 command window:
   ```
   COMMAND: db2 => get database manager configuration
   ```
3. Locate `AUDIT_BUF_SZ` value in the output:
   ```
   COMMAND: 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

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

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

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

3.1.2 Encrypt user data across the network

Audit:

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

1. Attach to the DB2 instance.

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

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

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

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

```
COMMAND: 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

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

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

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

3.1.3 Require explicit authorization for cataloging

Audit:

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

1. Attach to the DB2 instance.

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

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

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

3. Locate the value of CATALOG_NOAUTH in the output:

```
COMMAND: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

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

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

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

3.1.4 Disable datalinks support

Audit:

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

1. Attach to the DB2 instance.

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

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

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

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

```
COMMAND: 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

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

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

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

3.1.5 Secure permissions for default database file path

Audit:

For Windows and Linux:

1. Attach to the DB2 instance.

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

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

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

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

```
COMMAND: 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.

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

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

```
COMMAND: 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

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

3.1.6 Set diagnostic logging to capture errors and warnings

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

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

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

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

3.1.7 Secure permissions for all diagnostic logs

Audit:

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

1. Attach to the DB2 instance.

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

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

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

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

```
COMMAND: 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

```
COMMAND: OS => ls -al
```

Remediation:

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

1. Attach to the DB2 instance

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

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

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

Additional steps for Windows:

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

Additional steps for Linux:

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

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

3.1.8 *Require instance name for discovery requests*

Audit:

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

1. Attach to the DB2 instance.

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

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

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

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

```
COMMAND: 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

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

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

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

3. Restart the DB2 instance.

```
COMMAND: 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

Audit:

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

1. Attach to the DB2 instance.

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

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

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

3. Locate the DISCOVER_INST value in the output:

```
COMMAND: 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

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

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

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

3.1.10 Authenticate federated users at the instance level

Audit:

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

1. Attach to the DB2 instance.

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

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

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

3. Locate the FED_NOAUTH value in the output:

```
COMMAND: 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

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

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

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

3.1.11 Set maximum connection limits

Audit:

Perform the following DB2 commands to obtain the value(s) for these settings:
1. Attach to the DB2 instance.

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

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

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

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

```
COMMAND: 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.

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

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

```
COMMAND: db2 => get database configuration
```

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

```
COMMAND: 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

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

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

```
COMMAND: 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:

```
COMMAND: 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

Audit:

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

1. Attach to the DB2 instance.

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

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

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

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

```
COMMAND: 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

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

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

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

Default Value:

The default value of `notifylevel` is 3.

3.1.13 Enable server-based authentication

Audit:

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

1. Attach to the DB2 instance.

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

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

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

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

```
COMMAND: 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

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

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

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

3. Restart the DB2 instance.

```
COMMAND: 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

Audit:

Perform the following DB2 commands to obtain the value for this setting:
1. Connect to the DB2 database.

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

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

```
COMMAND: db2 => get database configuration
```

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

```
COMMAND: 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

```
COMMAND: 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:

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

Default Value:

The default value for `archretrydelay` is 20

3.1.15 Auto-restart after abnormal termination

Audit:

Perform the following DB2 commands to obtain the value for this setting:
1. Connect to the DB2 database.

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

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

```
COMMAND: db2 => get database configuration
```

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

```
COMMAND: 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

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

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

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

3.1.16 Disable database discovery

Audit:

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

1. Connect to the DB2 database.

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

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

```
COMMAND: db2 => get database configuration
```

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

```
COMMAND: 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

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

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

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

3.1.17 Secure permissions for the primary archive log location

Audit:

For Windows and Linux:

1. Attach to the DB2 instance.

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

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

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

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

```
COMMAND: 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.

```
COMMAND: 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:

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

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

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

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

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

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

3.1.18 Secure permissions for the secondary archive log location

Audit:

For Windows and Linux:

1. Attach to the DB2 instance.

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

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

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

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

```
COMMAND: 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

```
COMMAND: 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:

```
COMMAND: 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

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

3.1.19 Secure permissions for the tertiary archive log location

Audit:

For Windows and Linux:

1. Attach to the DB2 instance.

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

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

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

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

```
COMMAND: 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.

```
COMMAND: 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:

```
COMMAND: 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

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

3.1.20 Secure permissions for the log mirror location

Audit:

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

1. Connect to the DB2 database.

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

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

```
COMMAND: db2 => get database configuration
```

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

```
COMMAND: 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.

```
COMMAND: OS => ls -al
```

Remediation:

For Windows and Linux:

1. Connect to the DB2 database

```
COMMAND: 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:

```
COMMAND: 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

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

3.1.21 Establish retention set size for backups

Audit:

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

1. Connect to the DB2 database.

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

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

```
COMMAND: 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

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

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

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

3.1.22 Set archive log failover retry limit

Audit:

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

1. Connect to the DB2 database.

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

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

```
COMMAND: db2 => get database configuration
```

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

```
COMMAND: 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

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

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

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

3.2.2 SSL service name - ssl_svcename

Audit:

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

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

Remediation:

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

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

Default Value:

Null

3.2.3 Authentication type for incoming connections at the server -srvcon_auth

Audit:

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

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

Remediation:

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

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

Default Value:

Not specified

3.2.4 Database Manager Configuration parameter: trust_allclnts

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
```

3.2.5 Database Manager Configuration parameter: trust_clntauth

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>
```

4.1 Review Organization's Policies against DB2 RCAC Policies

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

4.2 Secure SECADM Authority

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

4.3 Review Users, Groups, and Roles

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

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

4.5 Review Column Mask logic according to policy

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.

5.1 Enable Backup Redundancy

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

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

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
```

6.1 Restrict Access to SYSCAT.AUDITPOLICIES

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
```

6.2 Restrict Access to SYSCAT.AUDITUSE

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

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
```

6.4 Restrict Access to SYSCAT.COLAUTH

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
```

6.5 Restrict Access to SYSCAT.EVENTS

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
```

6.6 Restrict Access to SYSCAT.EVENTTABLES

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
```

6.7 Restrict Access to SYSCAT.ROUTINES

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

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
```

6.9 Restrict Access to SYSCAT.PACKAGEAUTH

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

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

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
```

6.12 Restrict Access to SYSCAT.SECURITYPOLICIES

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
```

6.13 Restrict Access to SYSCAT.SECURITYPOLICYEXEMPTIONS

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
```

6.14 Restrict Access to SYSCAT.SURROGATEAUTHIDS

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
```

6.15 Restrict Access to SYSCAT.ROLEAUTH

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
```

6.16 Restrict Access to SYSCAT.ROLES

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
```

6.17 Restrict Access to SYSCAT.ROUTINEAUTH

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
```

6.18 Restrict Access to SYSCAT.SCHEMAAUTH

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

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
```

6.20 Restrict Access to SYSCAT.SEQUENCEAUTH

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
```

6.21 Restrict Access to SYSCAT.STATEMENTS

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
```

6.22 Restrict Access to SYSCAT.TABAUTH

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
```

6.23 Restrict Access to SYSCAT.TBSPACEAUTH

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
```

6.24 Restrict Access to Tablespaces

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
```

6.25 Restrict Access to SYSCAT.MODULEAUTH

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
```

6.26 Restrict Access to SYSCAT.VARIABLEAUTH

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'
```

4. 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
```

6.27 Restrict Access to SYSCAT.WORKLOADAUTH

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
```

6.28 Restrict Access to SYSCAT.XSROBJECTAUTH

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
```

6.29 Restrict Access to SYSCAT.AUTHORIZATIONIDS

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
```

6.30 Restrict Access to SYSIBMADM.OBJECTOWNERS

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
```

6.31 Restrict Access to SYSIBMADM.PRIVILEGES

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
```

7.1 Secure SYSADM authority

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

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

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

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

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>
```

7.6 Secure DBADM Authority

Audit:

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

1. Connect to the DB2 database.

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

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

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
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>
```

7.7 Secure SQLADM 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:

1. Revoke SQLADM authority from any unauthorized users.

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

7.8 Secure DATAACCESS 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:

1. Revoke DATAACCESS authority from any unauthorized users.

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

7.9 Secure ACCESSCTRL 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:

1. Revoke ACCESSCTRL authority from any unauthorized users.

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

7.10 Secure WLMADM authority

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:

1. Revoke any user who should NOT have WLMADM authority:

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

7.11 Secure CREATAB Authority

Audit:

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

1. Connect to the DB2 database.

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

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

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
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>
```

7.12 Secure BINDADD Authority

Audit:

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

1. Connect to the DB2 database.

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

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

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
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>
```

7.13 Secure CONNECT Authority

Audit:

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

1. Connect to the DB2 database.

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

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

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
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>
```

7.14 Secure LOAD Authority

Audit:

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

1. Connect to the DB2 database.

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

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

```
db2 => select distinct grantee, granteetype from syscat.dbauth where loadauth
= 'Y'
```

3. Review the list of users in the above output to ensure only approved users are assigned.

Remediation:

Revoke this permission from any unauthorized users.

1. Connect to the DB2 database.

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

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

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

7.15 Secure EXTERNALROUTINE Authority

Audit:

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

1. Connect to the DB2 database.

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

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

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
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>
```

7.16 Secure QUIESCECONNECT Authority

Audit:

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

1. Connect to the DB2 database.

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

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

```
db2 => select distinct grantee, granteetype from syscat.dbauth where
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>
```

8.1 Review Roles

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>
```

8.2 Review Role Members

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>
```

8.3 Nested Roles

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

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
```

8.5 Review Role Grantees with WITH ADMIN OPTION

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>
```

9.1 Start and Stop DB2 Instance

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

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

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

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

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

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

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

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.

9.9 Secure plug-in library locations

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

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
```

9.11 Ensure permissions on communication exit library locations

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
```

9.12 Ensure audit policies are enabled within the database

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
```