the CENTER for
INTERNET SECURITY

Security Configuration Benchmark For

DB2 8, 9 & 9.5 for Linux, UNIX, and Windows

Version 1.1.0
December 31st, 2009

http://cisecurity.org
feedback@cisecurity.org

Terms of Use Agreement

Background.

CIS provides benchmarks, scoring tools, software, data, information, suggestions, ideas, and other services and materials from the CIS website or elsewhere ("**Products**") as a public service to Internet users worldwide. Recommendations contained in the Products ("**Recommendations**") result from a consensus-building process that involves many security experts and are generally generic in nature. The Recommendations are intended to provide helpful information to organizations attempting to evaluate or improve the security of their networks, systems and devices. Proper use of the Recommendations requires careful analysis and adaptation to specific user requirements. The Recommendations are not in any way intended to be a "quick fix" for anyone's information security needs.

No representations, warranties and covenants.

CIS makes no representations, warranties or covenants whatsoever as to (i) the positive or negative effect of the Products or the Recommendations on the operation or the security of any particular network, computer system, network device, software, hardware, or any component of any of the foregoing or (ii) the accuracy, reliability, timeliness or completeness of any Product or Recommendation. CIS is providing the Products and the Recommendations "as is" and "as available" without representations, warranties or covenants of any kind.

User agreements.

By using the Products and/or the Recommendations, I and/or my organization ("**we**") agree and acknowledge that:

No network, system, device, hardware, software or component can be made fully secure;
We are using the Products and the Recommendations solely at our own risk;

We are not compensating CIS to assume any liabilities associated with our use of the Products or the Recommendations, even risks that result from CIS's negligence or failure to perform;

We have the sole responsibility to evaluate the risks and benefits of the Products and Recommendations to us and to adapt the Products and the Recommendations to our particular circumstances and requirements;

Neither CIS, nor any CIS Party (defined below) has any responsibility to make any corrections, updates, upgrades or bug fixes or to notify us if it chooses at it sole option to do so; and

Neither CIS nor any CIS Party has or will have any liability to us whatsoever (whether based in contract, tort, strict liability or otherwise) for any direct, indirect, incidental, consequential, or special damages (including without limitation loss of profits, loss of sales, loss of or damage to reputation, loss of customers, loss of software, data, information or emails, loss of privacy, loss of use of any computer or other equipment, business interruption, wasted management or other staff resources or claims of any kind against us from third parties) arising out of or in any way connected with our use of or our inability to use any of the Products or Recommendations (even if CIS has been advised of the possibility of such damages), including without limitation any liability associated with infringement of intellectual property, defects, bugs, errors, omissions, viruses, worms, backdoors, Trojan horses or other harmful items.

Grant of limited rights.

CIS hereby grants each user the following rights, but only so long as the user complies with all of the terms of these Agreed Terms of Use:

Except to the extent that we may have received additional authorization pursuant to a written agreement with CIS, each user may download, install and use each of the Products on a single computer;

Each user may print one or more copies of any Product or any component of a Product that is in a .txt, .pdf, .doc, .mcw, or .rtf format, provided that all such copies are printed in full and are kept intact, including without limitation the text of this Agreed Terms of Use in its entirety.

Retention of intellectual property rights; limitations on distribution.

The Products are protected by copyright and other intellectual property laws and by international treaties. We acknowledge and agree that we are not acquiring title to any intellectual property rights in the Products and that full title and all ownership rights to the Products will remain the exclusive property of CIS or CIS Parties. CIS reserves all rights not expressly granted to users in the preceding section entitled "Grant of limited rights." Subject to the paragraph entitled "Special Rules" (which includes a waiver, granted to some classes of CIS Members, of certain limitations in this paragraph), and except as we may have otherwise agreed in a written agreement with CIS, we agree that we will not (i) decompile, disassemble, reverse engineer, or otherwise attempt to derive the source code for any software Product that is not already in the form of source code; (ii) distribute, redistribute, encumber, sell, rent, lease, lend, sublicense, or otherwise transfer or exploit rights to any Product or any component of a Product; (iii) post any Product or any component of a Product on any website, bulletin board, ftp server, newsgroup, or other similar mechanism or device, without regard to whether such mechanism or device is internal or external, (iv) remove or alter trademark, logo, copyright or other proprietary notices, legends, symbols or labels in any Product or any component of a Product; (v) remove these Agreed Terms of Use from, or alter these Agreed Terms of Use as they appear in, any Product or any component of a Product; (vi) use any Product or any component of a Product with any derivative works based directly on a Product or any component of a Product; (vii) use any Product or any component of a Product with other products or applications that are directly and specifically dependent on such Product or any component for any part of their functionality, or (viii) represent or claim a particular level of compliance with a CIS Benchmark, scoring tool or other Product. We will not facilitate or otherwise aid other individuals or entities in any of the activities listed in this paragraph.

We hereby agree to indemnify, defend and hold CIS and all of its officers, directors, members, contributors, employees, authors, developers, agents, affiliates, licensors, information and service providers, software suppliers, hardware suppliers, and all other persons who aided CIS in the creation, development or maintenance of the Products or Recommendations ("**CIS Parties**") harmless from and against any and all liability, losses, costs and expenses (including attorneys' fees and court costs) incurred by CIS or any CIS Party in connection with any claim arising out of any violation by us of the preceding paragraph, including without limitation CIS's right, at our expense, to assume the exclusive defense and control of any matter subject to this indemnification, and in such case, we agree to cooperate with CIS in its defense of such claim. We further agree that all CIS Parties are third-party beneficiaries of our undertakings in these Agreed Terms of Use.

Special rules.

CIS has created and will from time to time create special rules for its members and for other persons and organizations with which CIS has a written contractual relationship. Those special rules will override and supersede these Agreed Terms of Use with respect to the users who are covered by the special rules. CIS hereby grants each CIS Security Consulting or Software Vendor Member and each CIS Organizational User Member, but only so long as such Member remains in good standing with CIS and complies with all of the terms of these Agreed Terms of Use, the right to distribute the Products and Recommendations within such Member's own organization, whether by manual or electronic means. Each such Member acknowledges and agrees that the foregoing grant is subject to the terms of such Member's membership arrangement with CIS and may, therefore, be modified or terminated by CIS at any time.

Choice of law; jurisdiction; venue.

We acknowledge and agree that these Agreed Terms of Use will be governed by and construed in accordance with the laws of the State of Maryland, that any action at law or in equity arising out of or relating to these Agreed Terms of Use shall be filed only in the courts located in the State of Maryland, that we hereby consent and submit to the personal jurisdiction of such courts for the purposes of litigating any such action. If any of these Agreed Terms of Use shall be determined to be unlawful, void, or for any reason unenforceable, then such terms shall be deemed severable and shall not affect the validity and enforceability of any remaining provisions. We acknowledge and agree that we have read these Agreed Terms of Use in their entirety, understand them and agree to be bound by them in all respects.

Table of Contents

Overview

This document, *Security Configuration Benchmark for DB2*, provides prescriptive guidance for establishing a secure configuration posture for *DB2* versions 8, 9 & 9.5 running on Linux, UNIX, and Windows. This guide was tested against *DB2 versions 9 and 9.5,* as installed by Fixpak 3a. To obtain the latest version of this guide, please visit http://cisecurity.org. If you have questions, comments, or have identified ways to improve this guide, please write us at feedback@cisecurity.org.

Consensus Guidance

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

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

Intended Audience

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

Acknowledgements

This benchmark exemplifies the great things a community of users, vendors, and subject matter experts can accomplish through consensus collaboration. The CIS community thanks the entire consensus team with special recognition to the following individuals who contributed greatly to the creation of this guide:

Authors
Nam Wu, *Qualys, Inc.*

Contributors and Reviews
Paul Griffiths, *Goldman Sachs*
David Futter, *JPMorgan Chase*
Blake Frantz, *Center for Internet Security*
Walid Rjaibi, *IBM*

Typographic Conventions

The following typographical conventions are used throughout this guide:

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

Configuration Levels

This section defines the configuration levels that are associated with each benchmark recommendation. Configuration levels represent increasing levels of security assurance.

Level-I Benchmark settings/actions

Level-I Benchmark recommendations are intended to:

- be practical and prudent;
- provide a clear security benefit; and
- do not negatively inhibit the utility of the technology beyond acceptable means

Level-II Benchmark settings/actions

Level-II Benchmark recommendations exhibit one or more of the following characteristics:

- may negatively inhibit the utility or performance of the technology
- acts as defense in depth measure

Scoring Status

This section defines the scoring statuses used within this document. The scoring status indicates whether compliance with the given recommendation is discernable in an automated manner.

Scorable

The platform’s compliance with the given recommendation can be determined via automated means.

Not Scorable

The platform’s compliance with the given recommendation cannot be determined via automated means.

Database Version Affected

This section defines the database version(s) that is affected by this benchmark.

- *DB2 UDB v8*
- *DB2 UDB v9*
- *DB2 UDB v9.5*

1. Installation and Patches

1.0.1 Install the latest Fixpak (Level 2, Scorable, 8, 9, 9.5)

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

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

Remediation:
Apply the latest fixpak as offered from IBM.

Audit:
Perform the following DB2 command to obtain the version:

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

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

References:

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

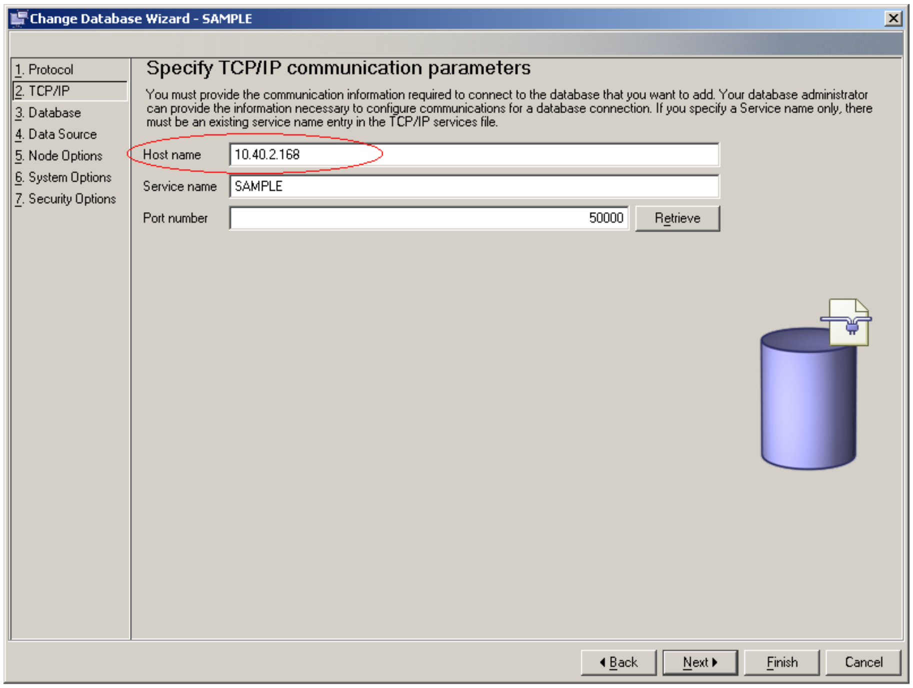
1.0.2 Use IP address rather than hostname (Level 1, Scorable, 8, 9, 9.5)

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

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

Remediation:
Reconfigure the connection string using the DB2 Configuration Assistant.

1. Launch the DB2 Configuration Assistant:



Specify TCP/IP communication parameters

You must provide the communication information required to connect to the database that you want to add. Your database administrator can provide the information necessary to configure communications for a database connection. If you specify a Service name only, there must be an existing service name entry in the TCP/IP services file.

Host name: 10.40.2.168

Service name: SAMPLE

Port number: 50000

Default Value:
The default value in the hostname field is an IP address.

1.0.3 Leverage a least privilege principle (Level 1, Not Scorable, 8, 9, 9.5)

Description:
The DB2 database instance will execute under the context of a given security principle. It is recommended that the DB2 service execute under a least privilege security principle. Furthermore, it is advisable to have the DB2 be executed under root or Administrator account and monitor such accounts from unauthorized access to the sensitive data.

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

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

Audit:
Review all accounts that have access to the DB2 database service.

1.0.4 Use non-standard account names (Level 1, Scorable, 8, 9, 9.5)

Description:
Do not install the DB2 software using well-known DB2 accounts. It is recommended not to install create well-known accounts.

Rationale:
The DB2 installation shall refrain from using well-known defined accounts such as 'db2admin', 'db2inst1', 'dasusr1', or 'db2fenc1'.

Note: review the impact of changing the group names and/or user names before performing this global change.

Remediation:

1. For MS Windows: right-click over the `($DB2 software directory)` and select *Properties* from the menu. Go to the *Security* tab and re-assign all the groups or user names with a not well-known account.

2. For Unix:

```
chown <new user name>:<new group name> -R <$DB2 software directory>
```

Audit:

1. For MS Windows: right-click over the `($DB2 software directory)` and select *Properties* from the menu. Go to the *Security* tab and review all groups or user names that access to this directory.

   For Unix: `ls -al {$DB2 software directory}` and review all groups or user names that access to this directory.

2. DB2 Directory and File Permissions

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

2.0.1 Secure DB2 Runtime Library (Level 1, Scorable, 8, 9, 9.5)

Description:

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

Rationale:
The files contain in this directory and in the sub-directories are executables and have direct impact to the DB2 instance.

Remediation:
For MS Windows:

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

For Unix:

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

For MS Windows:

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

For Unix:

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

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

Default Value:
Unix: `<$DB2 Directory>/sqllib` owned by the DB2 administrator with read, write, and execute access.

MS Windows: `<Drive:>\Program Files\IBM\SQLLIB` owned by the DB2 administrator with read, write, and execute access.

2.0.2 Secure all database containers (Level 1, Scorable, 8, 9, 9.5)

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

Rationale:

The containers are needed in order for the database to operate properly. The loss of the containers can cause down time and possibly allow hackers to read sensitive data stored in the containers. Therefore, secure the location(s) of the containers by restricting the access and ownership. Allow only the instance owner to have access to the tablespace containers.

Remediation:
Secure the directory of the containers. The recommended value is “read-only” to all non-DB2 administrator accounts.

Audit:
Review all users that have access to the directory of the containers.

2.0.3 Set umask value for DB2 admin user .profile file (Level 1, Scorable, 8, 9, 9.5)

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

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

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

Audit:
Ensure that the umask 022 setting exists in the .profile.

3. DB2 Configurations

3.1 DB2 Instance Parameter Settings

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

3.1.1 Enable audit buffer (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:
Increasing the audit buffer size to greater than 0 will allocate space for the audit records generated by the audit facility; and will cause the audit records to write asynchronously, thus ensuring no loss of audit records.

Remediation:
Perform the following to establish an audit buffer:

1. Attach to the DB2 instance

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

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

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

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

1. Attach to the DB2 instance.

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

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

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

3. Locate `AUDIT_BUF_SZ` value in the output:

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

Note: `AUDIT_BUF_SZ` is set to `1000` in the above output.

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

References:

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

3.1.2 Encrypt user data across the network (Level 2, Scorable, 8, 9, 9.5)

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

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

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

1. Attach to the DB2 instance

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

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

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

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

1. Attach to the DB2 instance.

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

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

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

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

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

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

Default Value:
The default value for `AUTHENTICATION` is `SERVER`.

References:

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

3.1.3 Require explicit authorization for cataloging (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:
Cataloging a database is the process of registering a database from a remote client to allow remote call and access. This procedure should only be restricted to user with a valid DB2 account and must have the SYSADM or SYSCTRL authority. Setting the `catalog-noauth` to `NO` by-passes all permission checks and allow anyone to catalog and uncatalog databases.

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

1. Attach to the DB2 instance

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

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

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

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

1. Attach to the DB2 instance.

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

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

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

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

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

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

Default Value:
The default value for `CATALOG_NOAUTH` is `NO`.

References:

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

3.1.4 Disable data links support (Level 2, Scorable, 8)

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

Rationale:

Disable `datalinks` if there is no use for them. `Datalinks` can be a point of attack from hackers using corrupted or infected files.

Remediation:

1. Attach to the DB2 instance

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

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Attach to the DB2 instance.

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

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

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

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

```

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

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

Default Value:
The default value for `datalinks` is `NO`.

3.1.5 Secure default database location (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:
Specify a path that is secure and has proper permission granted to authorize user.

Remediation:

1. Attach to the DB2 instance

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

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Attach to the DB2 instance.

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

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

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

3. Locate this value in the output:

```

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

3.1.6 Secure permission of default database location (Level 1, Scorable, 8, 9, 9.5)

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

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

Remediation:
For MS Windows:

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

For Unix:

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

```
OS => chmod -R 744
```

Audit:
Perform the following DB2 command to obtain the value for this setting:

For MS Windows:

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

For Unix:

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

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

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

3.1.7 Set diagnostic logging to capture errors and warnings (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:
The recommended diagnostic level setting is 3. This will allow the DB2 instance to capture all errors and warnings that occur on the system.

Remediation:

1. Attach to the DB2 instance

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

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

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

Audit:
Perform the following DB2 command 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
```

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

Default Value:
The default value for `diaglevel` is `3`.

References:

1. [http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof= diaglevel](http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.admin.doc/doc/r0000103.htm?resultof=diaglevel)

3.1.8 Secure all diagnostic logs (Level 1, Scorable, 8, 9, 9.5)

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

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

Remediation:

1. Attach to the DB2 instance

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

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Attach to the DB2 instance.

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

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

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

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

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

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

Default Value:
The default value for `diagpath` is NULL.

References:

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

3.1.9 Require instance name for discovery requests (Level 2, Scorable, 8, 9, 9.5)

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

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

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

1. Attach to the DB2 instance

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

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

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

3. Restart the DB2 instance.

```
db2 => db2stop
```

```
db2 => db2start
```

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Attach to the DB2 instance.

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

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

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

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

```

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

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

Default Value:
The default value for `discover` is `SEARCH`.

References:

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

3.1.10 Disable instance discoverability (Level 2, Scorable, 8, 9, 9.5)

Description:
The `discover_inst` parameter specifies whether the instance can be discovered in the network. It is recommended that instances be undiscoverable.

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

Remediation:

1. Attach to the DB2 instance

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

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Attach to the DB2 instance.

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

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

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

3. Locate the `DISCOVER_INST` is value in the output:

```

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

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

Default Value:
The default value for `discover_inst` is `ENABLE`.

References:

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

3.1.11 Authenticate federated users at the instance level (Level 2, Scorable, 8, 9, 9.5)

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

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

Remediation:

1. Attach to the DB2 instance

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

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

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

Audit:

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

1. Attach to the DB2 instance.

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

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

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

3. Locate the FED_NOAUTH value in the output:

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

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

Default Value:
The default value for `FED_NOAUTH` is `NO`.

References:

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

3.1.12 Enable instance health monitoring (Level 2, Scorable, 8, 9, 9.5)

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

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

Remediation:

1. Attach to the DB2 instance

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

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

```
db2 => update database manager configuration using health_mon on
```

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Attach to the DB2 instance.

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

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

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

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

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

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

Default Value:
The default value for `HEALTH_MON` is `ON`.

References:

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

3.1.13 Retain fenced model processes (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:
All routines that were executed by the DB2 shall be terminated when the instance is stopped.

Remediation:

1. Attach to the DB2 instance

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

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

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

3. Restart the DB2 instance.

```
db2 => db2stop
```

```
db2 => db2start
```

Note: this will require a db2 restart.

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Attach to the DB2 instance.

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

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

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

3. Locate the KEEPFENCED value in the output:

```

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

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

Default Value:
The default value for KEEPFENCED is YES.

References:

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

3.1.14 Set maximum connection limits (Level 2, Scorable, 8, 9, 9.5)

Description:
The **max_connections** parameter indicates the maximum number of client connections allowed per database partition. It is recommended that this parameter be set equal to the max_coordagents parameter; the max_coordagents parameter should be set to 100. Ensure that dependent parameter, such as maxappls, be set less than the max_coordagents parameter as well.

Rationale:
DB2 allows unlimited number of users to access the db2 instance. Set a limit to the number of users allowed to access a DB2 instance to reduce the chances of open connections to attackers. Also, give access to the DB2 instance to only authorized users.

Remediation:
The default value is AUTOMATIC, where the system will determine the limit. Allowable range is 1 to 64,000. Or -1 for unlimited.

1. Attach to the DB2 instance

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

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

```
db2 => update database manager configuration using max_connections
[integer]
```

Audit:
Perform the following DB2 command 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 `MAX_CONNECTIONS` and `MAXAGENTS` values in the output:

```

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

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

1. Connect to the DB2 database.

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

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

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

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

```

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

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

Default Value:

The default value for `MAX_CONNECTIONS` is `AUTOMATIC`.
The default value for `MAX_COORDAGENTS` is `-1`.
The default value for `MAXAPPLS` is `AUTOMATIC`.

References:

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

3.1.15 Set administrative notification level (Level 2, Scorable, 8, 9, 9.5)

Description:
The `notifylevel` parameter specifies the type of administration notification messages that are written to the administration notification log. It is recommended that this parameter be set to `3`. A setting of 3 will log all fatal errors, failing services, system integrity, as well as system health.

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

Remediation:

1. Attach to the DB2 instance

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

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Attach to the DB2 instance.

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

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

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

3. Locate the NOTIFYLEVEL value in the output:

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

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

Default Value:
The default value for NOTIFYLEVEL is 3.

References:

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

3.1.16 Enable server-based authentication (Level 2, Scorable, 8, 9, 9.5)

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

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

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

Remediation:
The recommended value is SERVER. Note: this will require a db2 restart.

1. Attach to the DB2 instance

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

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

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

3. Restart the DB2 instance.

```
db2 => db2stop

db2 => db2start
```

Audit:

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

1. Attach to the DB2 instance.

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

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

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

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

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

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

Default Value:
The default value for `SRVCON_AUTH` is `NULL`.

References:

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

3.2.1 Set failed archive retry delay (Level 2, Scorable, 8, 9, 9.5)

Description:
The `archretrydelay` parameter specifies the wait time (in seconds) to retry to archive the log file after a failure. It is recommended that this parameter be set to `20`.

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

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 archretrydelay 25
```

Audit:

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

1. Connect to the DB2 database.

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

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

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

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

```

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

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

Default Value:
The default value for `ARCHRETRYDELAY` is `20`.

3.2.2 Auto-restart after abnormal termination (Level 2, Scorable, 8, 9, 9.5)

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

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

Remediation:

1. Connect to the DB2 database

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

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Connect to the DB2 database.

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

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

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

3. Locate the AUTORESTART value in the output:

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

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

Default Value:
The default value for autorestart is ON.

3.2.3 Disable database discovery (Level 2, Scorable, 8, 9, 9.5)

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

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

Remediation:

1. Connect to the DB2 database

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

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Connect to the DB2 database.

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

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

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

3. Locate the DISCOVER_DB value in the output:

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

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

Default Value:
The default value for `DISCOVER_DB is ENABLE`.

3.2.4 Establish secure archive log location (Level 1, Scorable, 8, 9, 9.5)

Description:
The `logarchmeth1` parameter specifies the type of media used for the primary destination for archived logs. It is recommended that this parameter be set to a secure location.

Rationale:
Recommended value is `DISK:<`*`valid directory`*`>`. This will ensure that the primary logs are archived.

Remediation:

1. Connect to the DB2 database

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

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Connect to the DB2 database.

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

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

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

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

```

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

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

Default Value:
The default value for `LOGARCHMETH1` is `OFF`.

3.2.5 Secure permission of the primary archive log location (Level 1, Scorable, 8, 9, 9.5)

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

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

Remediation:
For MS Windows:

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

For Unix:

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

```
OS => chmod -R 744
```

Audit:
Perform the following DB2 command to obtain the value for this setting:

For MS Windows:

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

For Unix:

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

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

Default Value:
The default value for a directory is read-and-write access.

3.2.6 Establish secure secondary archive location (Level 1, Scorable, 8, 9, 9.5)

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

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

Remediation:

1. Connect to the DB2 database

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

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Connect to the DB2 database.

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

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

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

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

```

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

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

Default Value:
The default value for `LOGARCHMETH2` is `OFF`.

3.2.7 Secure permission of the secondary archive location (Level 1, Scorable, 8, 9, 9.5)

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

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

Remediation:
For MS Windows:

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

For Unix:

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

```
OS => chmod -R 744
```

Audit:
Perform the following DB2 command to obtain the value for this setting:

For MS Windows:

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

For Unix:

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

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

Default Value:
The default value for a directory is read-and-write access.

3.2.8 Establish secure tertiary archive log location (Level 1, Scorable, 8, 9, 9.5)

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

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

Remediation:

1. Connect to the DB2 database

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

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Connect to the DB2 database.

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

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

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

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

```

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

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

Default Value:
The default value for `FAILARCHPATH` is null.

3.2.9 Secure permission of the tertiary archive location (Level 1, Scorable, 8, 9, 9.5)

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

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

Remediation:
For MS Windows:

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

For Unix:

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

```
OS => chmod -R 744
```

Audit:
Perform the following DB2 command to obtain the value for this setting:

For MS Windows:

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

For Unix:

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

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

Default Value:
The default value for a directory is read-and-write access.

3.2.10 Establish secure log mirror location (Level 1, Scorable, 8, 9)

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

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

Remediation:

1. Connect to the DB2 database

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

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Connect to the DB2 database.

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

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

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

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

```

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

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

Default Value:
The default value for `mirrorlogpath` is null.

3.2.11 Establish retention set size for backups (Level 2, Scorable, 8, 9, 9.5)

Description:

The num_db_backups parameter specifies the number of backups to retain for a database before the old backups is marked deleted. It is recommended that this parameter be set to at least 12.

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

Remediation:

1. Connect to the DB2 database

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

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Connect to the DB2 database.

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

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

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

3. Locate the NUM_DB_BACKUPS value in the output:

```

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

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

Default Value:
The default value for num_db_backups is 12.

3.2.12 Set archive log failover retry limit (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:

Establish a failover retry time limit will ensure that the database will always have a mean to recover from an abnormal termination. This parameter should not be set to 0. The recommended value is 5.

Remediation:

1. Connect to the DB2 database

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

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Connect to the DB2 database.

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

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

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

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

```

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

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

Default Value:
The default value for `numarchretry` is 5.

3.3 Database Administration Server Settings

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

3.3.1 Establish DAS administrative group (Level 1, Scorable, 8, 9, 9.5)

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

Rationale:
The DAS is a special administrative tool that enables remote administration of DB2 servers. DASADM authority is the highest level of authority within the DAS. Restrict non-essential users from this group since it may allow malicious users to tamper with the administration of the DB2 servers.

Remediation:

1. Attach to the DB2 instance

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

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

```
db2 => update admin configuration using dasadm_group <valid system
group>
```

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Attach to the DB2 instance

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

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

```
db2 => get admin configuration
```

3. Locate this value in the output:

```

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

Note: `DASADM_GROUP` is set to `DASADM` in the above output.

Default Value:
The default value for `dasadm_group` is null.

3.3.2 Set a generic system name (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:
Exposing OS or DB revision information may provide malicious users with enough information to identify vulnerabilities that affect the platforms.

Remediation:

1. Attach to the DB2 instance

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

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Attach to the DB2 instance

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

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

```
db2 => get admin configuration
```

3. Locate this value in the output:

```

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

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

Default Value:
The default value for `db2system` is the hostname.

3.3.3 Disable DAS discoverability (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:
Administration server should not handle any type of discovery request. This will prevent a malicious user from discovering all the DB2 servers in the network.

Remediation:

1. Attach to the DB2 instance

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

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Attach to the DB2 instance

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

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

```
db2 => get admin configuration
```

3. Locate this value in the output:

```

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

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

Default Value:
The default value for `discover` is `SEARCH`.

3.3.4 Do not execute expired tasks (Level 2, Scorable, 8, 9, 9.5)

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

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

Remediation:

1. Attach to the DB2 instance

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

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Attach to the DB2 instance

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

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

```
db2 => get admin configuration
```

3. Locate this value in the output:

```

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

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

Default Value:
The default value for `Aexec_exp_task` is `NO`.

3.3.5 Secure the JDK runtime library (Level 2, Scorable, 8, 9, 9.5)

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

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

Remediation:

1. Attach to the DB2 instance

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

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Attach to the DB2 instance

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

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

```
db2 => get admin configuration
```

3. Locate this value in the output:

```

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

Note: `JDK_PATH` is set to `C:\Program Files\Java` in the above output.

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

2.3.6 Secure the JDK 64-bit runtime library (Level 2, Scorable, 8, 9, 9.5)

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

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

Remediation:

1. Attach to the DB2 instance

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

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

```
db2 => update database manager configuration using jdk_64_path 
<jdk_path>
```

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Attach to the DB2 instance

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

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

```
db2 => get admin configuration
```

3. Locate this value in the output:

```

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

Note: `AJDK_64_PATH` is set to `C:\Program Files\Java` in the above output.

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

3.3.7 Disable unused task scheduler (Level 2, Scorable, 8, 9, 9.5)

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

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

Remediation:

1. Attach to the DB2 instance

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

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

```
db2 => update admin configuration using sched_enable off
```

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Attach to the DB2 instance

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

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

```
db2 => get admin configuration
```

3. Locate this value in the output:

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

Note: `SCHED_ENABLE` is set to `OFF` in the above output.

Default Value:
The default value for `sched_enable` is `OFF`.

4. Label-Based Access Controls (LBAC)

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

4.0.1 Enforce Label-Based Access Controls Implementation (Level 2, Not Scorable, 9, 9.5)

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

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

Remediation:
Impose LBAC capability on tables and rows with sensitive data.

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

4.0.2 Review Security Rule Exemptions (Level 1, Not Scorable, 9, 9.5)

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

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

Remediation:
Review all users that have LBAC rule exemptions.

Audit:

Review and justify all rule exemption grants.

4.0.3 Review Security Label Component (Level 1, Not Scorable, 9, 9.5)

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

Rationale:
Security label component shall be implemented to provide different level of access to different sensitive data.

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

Audit:
Review and justify all security label components.

4.0.4 Review Security Label Policies (Level 1, Not Scorable, 9, 9.5)

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

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

Remediation:
Review all security label policies and ensure that it is set up properly.

Audit:
Review and justify all security label policies.

4.0.5 Review Security Labels (Level 1, Not Scorable, 9, 9.5)

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

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

Remediation:
Review all security labels and ensure that it is set up properly.

Audit:
Review and justify all security labels.

5. Database Maintenance

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

5.0.1 Enable Redundancy (Level 2, Not Scorable, 8, 9, 9.5)

Description:
Redundancy is process of ensuring that you have multiple copies of your backups. This will ensure that a single corrupted backup does not cause a complete outage of the system.

Rationale:
Redundant backups will prevent a single point of failure if one copy of your backups is corrupted.

Remediation:
Define a process to replicate your backups onto multiple locations.

Audit:
Review the replication of your backups based on company policy.

5.0.2 Protecting Backups (Level 1, Not Scorable, 8, 9, 9.5)

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

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

Remediation:
Define a security policy for all backups stored.

Audit:
Review the access of your backups based on company policy.

5.0.3 Enable Database Maintenance (Level 2, Scorable, 8, 9, 9.5)

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

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

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
```

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Connect to the DB2 database:

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

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

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

3. Locate this value in the output:

```

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

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

Default Value:
The default value for `auto_maint` is `ON`.

5.0.4 Schedule Runstat and Reorg (Level 1, Not Scorable, 8, 9, 9.5)

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

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

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

Audit:
Not Applicable

6. Securing Database Objects

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

6.0.1 Restrict Access to `SYSCAT.AUDITPOLICIES` (Level 2, Scorable, 8, 9, 9.5)

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

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

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.2 Restrict Access to `SYSCAT.AUDITUSE` (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:

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

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
```

Audit:
Perform the following DB2 command to obtain the value for this setting:

1. Connect to the DB2 database.

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

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

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

6.0.3 Restrict Access to `SYSCAT.DBAUTH` (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:
This view contains all the grants in the database and may be used as a level of exploit.

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.4 Restrict Access to `SYSCAT.COLAUTH` (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:

This view contains all the grants in the database and may be used as a level of exploit.

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.5 Restrict Access to `SYSCAT.EVENTS` (Level 2, Scorable, 8, 9, 9.5)

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

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

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.6 Restrict Access to *`SYSCAT.EVENTTABLES`* (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:
Public should not have access to see the target name of the event monitoring table.

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.7 Restrict Access to `SYSCAT.ROUTINES` (Level 2, Scorable, 8, 9, 9.5)

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

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

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.8 Restrict Access to `SYSCAT.INDEXAUTH` (Level 2, Scorable, 8, 9, 9.5)

Description:
The SYSCAT.INDEXAUTH view contains a list of user or group that has CONTROL access on an index. It is recommended that the PUBLIC role be restricted from accessing this view.

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

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.9 Restrict Access to `SYSCAT.PACKAGEAUTH` (Level 2, Scorable, 8, 9, 9.5)

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

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

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.10 Restrict Access to `SYSCAT.PACKAGES` (Level 2, Scorable, 8, 9, 9.5)

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

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

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.11 Restrict Access to `SYSCAT.PASSTHRUAUTH` (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:
The ability to see which account has pass-through privilege can allow a hacker to exploit that account to access another data source.

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.12 Restrict Access to `SYSCAT.SECURITYLABELACCESS` (Level 2, Scorable, 8, 9, 9.5)

Description:
The `SYSCAT.SECURITYLABELACCESS` view contains all accounts in the database that have a security label privilege. It is recommended that the `PUBLIC` role be restricted from accessing this view.

Rationale:
Allowing public access to view all accounts having security label privilege can lead to privilege escalation to sensitive data.

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.SECURITYLABELACCESS FROM PUBLIC
```

Audit:
Perform the following DB2 command 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 = 'SECURITYLABELACCESS' and grantee = ‘PUBLIC’
```

6.0.13 Restrict Access to `SYSCAT.SECURITYLABELCOMPONENTELEMENTS` (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:
Public should not be able to view all the elements of a security component and the database security policy.

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.SECURITYLABELCOMPONENTELEMENTS FROM
PUBLIC
```

Audit:
Perform the following DB2 command 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 = 'SECURITYLABELCOMPONENTELEMENTS' and grantee = ‘PUBLIC’
```

6.0.14 Restrict Access to `SYSCAT.SECURITYLABELCOMPONENTS` (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:
Public should not be able to view all the security components and the database security policy.

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.SECURITYLABELCOMPONENTS FROM PUBLIC
```

Audit:
Perform the following DB2 command 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 = 'SECURITYLABELCOMPONENTS' and grantee = ‘PUBLIC’
```

6.0.15 Restrict Access to `SYSCAT.SECURITYLABELS` (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:
Public should not be able to view all the security components and the database security policy.

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.SECURITYLABELS FROM PUBLIC
```

Audit:

Perform the following DB2 command 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 = 'SECURITYLABELS' and grantee = ‘PUBLIC’
```

6.0.16 Restrict Access to `SYSCAT.SECURITYPOLICIES` (Level 2, Scorable, 8, 9, 9.5)

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

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

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.17 Restrict Access to `SYSCAT.SECURITYPOLICYCOMPONENTRULES` (Level 2, Scorable, 8, 9, 9.5)

Description:

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

Rationale:
Public should not be able to view all the access rules of the database security policies.

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.SECURITYPOLICYCOMPONENTRULES FROM PUBLIC
```

Audit:
Perform the following DB2 command 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 = 'SECURITYPOLICYCOMPONENTRULES' and grantee = ‘PUBLIC’
```

6.0.18 Restrict Access to SYSCAT.SECURITYPOLICYEXEMPTIONS (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:

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

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.19 Restrict Access to `SYSCAT.SURROGATEAUTHIDS` (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:

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

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.20 Restrict Access to `SYSCAT.ROLEAUTH` (Level 2, Scorable, 9.5)

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

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

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.21 Restrict Access to `SYSCAT.ROLES` (Level 2, Scorable, 8, 9, 9.5)

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

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

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.22 Restrict Access to `SYSCAT.ROUTINEAUTH` (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:
Public should not have access to see all the grants of routines to users or groups because this could be used as a point of exploit.

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.23 Restrict Access to SYSCAT.SCHEMAAUTH (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:
Public should not have access to see all the grants of schemas to users or groups because this could be used as a point of exploit.

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.24 Restrict Access to SYSCAT.SCHEMATA (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:
Public should not have access to see all the created schemas in the database because this could be used as a point of exploit.

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.25 Restrict Access to `SYSCAT.SEQUENCEAUTH` (Level 2, Scorable, 8, 9, 9.5)

Description:
The `SYSCAT.SEQUENCEAUTH` contains user or group that has access to one or more privileges on a sequence. It is recommended that the `PUBLIC` role be restricted from accessing this view.

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

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.26 Restrict Access to `SYSCAT.STATEMENTS` (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:
Public should not have access to the source code or the SQL statements of a database package. This could lead to an exploit.

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.27 Restrict Access to `SYSCAT.PROCEDURES` (Level 2, Scorable, 8, 9, 9.5)

Description:
The `SYSCAT.PROCEDURES` contains all stored procedures in the database. It is recommended that the `PUBLIC` role be restricted from accessing this view.

Rationale:
Public should not have access to the names of the stored procedures in a database. This could lead to an exploit.

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.PROCEDURES FROM PUBLIC
```

Audit:
Perform the following DB2 command 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 = 'PROCEDURES' and grantee = ‘PUBLIC’
```

6.0.28 Restrict Access to `SYSCAT.TABAUTH` (Level 2, Scorable, 8, 9, 9.5)

Description:
The `SYSCAT.TABAUTH` contains user or group that has been granted one or more privileges on a table or view. It is recommended that the `PUBLIC` role be restricted from accessing this view.

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

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.29 Restrict Access to *`SYSCAT.TBSPACEAUTH`* (Level 2, Scorable, 8, 9, 9.5)

Description:
The `SYSCAT.TBSPACEAUTH` contains user or group that has been granted the USE privilege on a particular table space in the database. It is recommended that the `PUBLIC` role be restricted from accessing this view.

Rationale:
Public should not have access to the grants of the tablespaces in a database. This could lead to an exploit.

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
```

Audit:
Perform the following DB2 command 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’
```

6.0.30 Restrict Access to Tablespaces (Level 2, Scorable, 8, 9, 9.5)

Description:
A Tablespace is where the data is physically stored. It is recommended that usage of tablespaces be granted to authorize users only.

Rationale:

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

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
```

Audit:
Perform the following DB2 command 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 and grantee =
'PUBLIC'
```

7. Entitlements

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

7.0.1 Establish an administrator group (Level 2, Scorable, 8, 9, 9.5)

Description:
The sysadm_group parameter defines the system administrator group with SYSADM authority for the DB2 instance. It is recommended that sysadm_group group contains authorized users only .

Rationale:
Review all users belonging to the assigned group for the SYSADM authority since it is the highest level of authority within the database manager (ie, stopping/starting services, backup/recovery, and maintenance) and controls all database objects (ie, data, system objects and privileges).

Remediation:
Define a valid group name to the SYSADM group.

1. Attach to the DB2 instance

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

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

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

Audit:
Perform the following DB2 command 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:

```

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

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

Default Value:
The default value for `SYSADM_GROUP` is `NULL`.

References:

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

7.0.2 Establish system control group (Level 2, Scorable, 8, 9, 9.5)

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

Rationale:
Review all users belonging to the assigned group for the SYSCTRL authority since it has the authority to affect the resources and the operation of the operating system. Access to this group of privileges can compromise the overall system and the DB2 instance.

Remediation:
Define a valid group name to the SYSCTRL group. Note: this parameter does not apply on MS Windows.

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]
```

Audit:
Perform the following DB2 command 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:

```

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

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

Default Value:
The default value for `SYSCTRL_GROUP` is `NULL`.

References:

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

7.0.3 Establish system maintenance group (Level 1, Scorable, 8, 9, 9.5)

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

Rationale:
Review all users belonging to the assigned group for the SYSMAINT authority since it has ability to perform maintenance operations on the database instance. Access to this maintenance group can impact the performance of the database and the host.

Remediation:
Define a valid group name to the SYSMAINT group. Note: this parameter does not apply on MS Windows.

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]
```

Audit:
Perform the following DB2 command 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:

```

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

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

Default Value:
The default value for SYSMAIN_GROUP is NULL.

References:

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

7.0.4 Establish system monitoring group (Level 1, Scorable, 8, 9, 9.5)

Description:
The sysmon_group parameter defines the operating system group with system monitor (SYSMON) authority. It is recommended that sysmon_group group contains authorized users only.

Rationale:

Review all users belonging to the assigned group for the SYSMON authority since it has the ability to perform system snapshots at both the database and instance level.

Remediation:
Define a valid group name to the `SYSMON` group.

1. Attach to the DB2 instance.

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

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

```
db2 => update database manager configuration using sysmon_group [sys
monintor group name]
```

Audit:
Perform the following DB2 command 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:

```

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

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

Default Value:
The default value for `SYSMON_GROUP` is `NULL`.

References:

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

7.0.5 Secure SECADM Authority (Level 1, Scorable, 9, 9.5)

Description:
The SECADM (security administrator) role grants the authority to create, alter (where applicable), and drop roles, trusted contexts, audit policies, security label components, security policies and security labels. It is also the authority required to grant and revoke roles, security labels and exemptions, and the SETSESSIONUSER privilege. SECADM

authority has no inherent privilege to access data stored in tables. It is recommended that `secadm` role be granted to authorize users only.

Rationale:
Review all users that have access to this authority.

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>
```

Audit:
Perform the following DB2 command 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'
```

Note: Review the list of users in the above output.

References:

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

7.0.6 Secure DBADM Authority (Level 1, Scorable, 9, 9.5)

Description:
The DBADM (database administration) role grants the authority to a user to perform administrative tasks on a specific database. It is recommended that `dbadm` role be granted to authorize users only.

Rationale:
Review all users that have access to this authority.

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>
```

Audit:
Perform the following DB2 command 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'
```

Note: Review the list of users in the above output.

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

7.0.7 Secure CREATAB Authority (Level 1, Scorable, 9, 9.5)

Description:
The CREATAB (create table) role grants the authority to a user to create tables within a specific database. It is recommended that `createtab` role be granted to authorize users only.

Rationale:
Review all users that have access to this authority.

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>
```

Audit:
Perform the following DB2 command 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
createtabauth = 'Y'
```

Note: Review the list of users in the above output.

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

7.0.8 Secure BINDADD Authority (Level 1, Scorable, 9, 9.5)

Description:
The BINDADD (bind application) role grants the authority to a user to create packages on a specific database. It is recommended that `bindadd` role be granted to authorize users only.

Rationale:
Review all users that have access to this authority.

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>
```

Audit:
Perform the following DB2 command 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'
```

Note: Review the list of users in the above output.

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

7.0.9 Secure CONECT Authority (Level 1, Scorable, 9, 9.5)

Description:
The CONNECT role grants the authority to a user to connect to a specific database. It is recommended that `connect` role be granted to authorize users only.

Rationale:
Review all users that have access to this authority.

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>
```

Audit:
Perform the following DB2 command 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'
```

Note: Review the list of users in the above output.

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

7.0.10 Secure NOFENCE Authority (Level 1, Scorable, 9, 9.5)

Description:
The NOFENCE role grants the authority to a user to create user-defined functions or procedure that is not fenced in the memory block of the database. It is recommended that `nofence` role be granted to authorize users only.

Rationale:

Review all users that have access to this authority.

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_NOT_FENCED_ROUTINE ON DATABASE FROM USER
<username>
```

Audit:
Perform the following DB2 command 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
nofenceauth = 'Y'
```

Note: Review the list of users in the above output.

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

7.0.11 Secure IMPLSCHEMA Authority (Level 1, Scorable, 9, 9.5)

Description:
The IMPLSCHEMA (implicit schema) role grants the authority to a user to create objects without specifying a schema that already exists. It is recommended that `implschema` role be granted to authorize users only .

Rationale:
Review all users that have access to this authority.

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 IMPLICIT_SCHEMA ON DATABASE FROM USER <username>
```

Audit:
Perform the following DB2 command 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
implschemaauth = 'Y'
```

Note: Review the list of users in the above output.

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

7.0.12 Secure LOAD Authority (Level 1, Scorable, 9, 9.5)

Description:
The LOAD role grants the authority to a user to load data into tables. It is recommended that `load` role be granted to authorize users only .

Rationale:
Review all users that have access to this authority.

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>
```

Audit:
Perform the following DB2 command 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'
```

Note: Review the list of users in the above output.

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

7.0.13 Secure EXTERNALROUTINE Authority (Level 1, Scorable, 9, 9.5)

Description:
The EXTERNALROUTINE role grants the authority to a user to create user-defined functions and procedures in a specific database. It is recommended that `externalroutine` role be granted to authorize users only.

Rationale:
Review all users that have access to this authority.

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>
```

Audit:
Perform the following DB2 command 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'
```

Note: Review the list of users in the above output.

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

7.0.14 Secure QUIESCECONNECT Authority (Level 1, Scorable, 9, 9.5)

Description:
The QUIESCECONNECT role grants the authority to a user to access a database even in the quiesced state. It is recommended that `quiesceconnect` role be granted to authorize users only.

Rationale:
Review all users that have access to this authority.

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>
```

Audit:
Perform the following DB2 command 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'
```

Note: Review the list of users in the above output.

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

8. General Policy and Procedures

8.0.1 Start and Stop DB2 Instance (Level 1, Not Scorable, 8, 9, 9.5)

Description:
The DB2 instance manages the database environment and sets the configuration parameters. It is recommended that only administrators are allowed to start and stop the DB2 instance.

Rationale:

Only privileged users should have access to start and stop the DB2 instance. This will ensure that the DB2 instance is controlled by authorized administrators.

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

Audit:
On MS Windows: go to Start, then to the Run option. Type in `services.msc` in the command prompt. Locate the DB2 service and identify the user/group that can start and stop the service.

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

8.0.2 Start and Stop DB2 Administrator Server (Level 2, Not Scorable, 8, 9, 9.5)

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

Rationale:
Only privileged users should have access to start and stop the DB2 administration server. This will ensure that the DB2 administration server is controlled by authorized administrators.

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

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

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

8.0.3 Remove Unused Schemas (Level 1, Not Scorable, 8, 9, 9.5)

Description:
A schema is a logical grouping of database objects. It is recommended that unused schemas be removed from the database.

Rationale:
Unused schemas can be left unmonitored and may be subjected to abuse and therefore should be removed.

Remediation:
Revoke access from any unnecessary users.

1. Connect to the DB2 database.

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

2. Review unused schemas and remove if necessary
3. Run the following command from the DB2 command window:

```
db2 => drop scheme <scheme name> restrict
```

Audit:

1. Connect to the DB2 database.

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

2. Review the list of schemas
3. Run the following command from the DB2 command window:

```
db2 => select schemaname from syscat.schemata
```

8.0.4 Review System Tablespaces (Level 1, Not Scorable, 8, 9, 9.5)

Description:
System tablespaces store all system objects data within that database. It is recommended that system tablespaces are used to stored system data.

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

Remediation:
Revoke access from any unnecessary users.

1. Connect to the DB2 database.

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

2. Review unused users and user objects that are stored in the system tablespaces

Audit:

1. Connect to the DB2 database.

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

2. Review the list of system tablespaces
3. 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')
```

8.0.5 Remove Default Databases (Level 2, Scorable, 8, 9, 9.5)

Description:
A DB2 Instance may come installed with default databases. It is recommended that the `SAMPLE` database be removed.

Rationale:
Removing unused, well-known, databases will reduce the attack surface of the system.

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
```

Audit:
Perform the following DB2 command 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              =
```

Note: Identify the default databases from the output above.

8.0.6 Enable SSL communication with LDAP server (Level 2, Scorable, 9.1, 9.5)

Description:
The communication later between a DB2 instance and the LDAP server should be encrypted. It is recommended that the `ENABLE_SSL` parameter in the IBMLDAPSecurity.ini file be set to **`TRUE`**.

Rationale:
SSL should be enabled between the DB2 instance and the LDAP server to prevent userid and password be set in plain text.

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

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
```

Audit:
Perform the following command to obtain the parameter setting:

1. Connect to the DB2 host

2. Edit the IBMLDAPSecurity.ini file

3. Verify the existence of this parameter :

```
ENABLE_SSL = TRUE
```

Note: The default setting is the omission of this parameter.

8.0.7 Secure the permission of the IBMLDAPSecurity.ini file (Level 2, Scorable, 9.1, 9.5)

Description:
The IBMLDAPSecurity.ini file contains the IBM LDAP security plug-in configurations.

Rationale:
Recommended value is ready-only (RO) to Everyone/Other/Users/Domain Users. This will ensure that the parameter file is protected.

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

Remediation:
For MS Windows:

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

For Unix:

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

For MS Windows:

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

For Unix:

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

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

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

8.0.8 Secure the permission of the SSLconfig.ini file (Level 2, Scorable, 9.1, 9.5)

Description:
The SSLconfig.ini file contains the SSL configuration parameters for the DB2 instance, including the password for KeyStore.

Rationale:
Recommended value is ready-only (RO) to Everyone/Other/Users/Domain Users. This will ensure that the parameter file is protected.

Note: the file is located under `INSTANCE_HOME/cfg/`, for Unix; and `%INSTHOME%\`, for MS Windows. Only the instance owner should have access to this file.

Remediation:
For MS Windows:

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

For Unix:

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

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

Audit:
Perform the following DB2 command to obtain the value for this setting:

For MS Windows:

1. Connect to the DB2 host
2. Right-click over the file directory

3. Choose *Properties*

4. Select the *Security* tab

5. Review access from all non-administrator accounts

For Unix:

1. Connect to the DB2 host

2. Change to the file directory

3. Change the permission level of the directory

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

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

9. DB2 Utilities and Tools

9.0.1 Secure DB2 Control Center (Level 1, Not Scorable, 8, 9, 9.5)

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

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

Remediation:
Revoke access from any unnecessary users.

1. Connect to the host
2. Review users and groups that have access to start the DB2 Control Center

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

9.0.2 Secure DB2 Configuration Assistant Utility (Level 1, Not Scorable, 8, 9, 9.5)

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

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

Remediation:
Revoke access from any unnecessary users.

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

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

9.0.3 Secure DB2 Health Monitor Utility (Level 1, Not Scorable, 8, 9, 9.5)

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

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

Remediation:
Revoke access from any unnecessary users.

1. Connect to the host
2. Review users and groups that have access to start the DB2 Health Center

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

9.0.4 DB2 Activity Monitor Utility (Level 1, Not Scorable, 8, 9, 9.5)

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

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

Remediation:
Revoke access from any unnecessary users.

1. Connect to the host
2. Review users and groups that have access to start the DB2 Activity Monitor

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

Appendix A: Change History

Date	Version	Changes for this version
November 5th, 2009	1.0.0	Initial Public Release
December 31st, 2009	1.1.0	- Section 1.0.2: Updated Rationale - Section 1.0.3: Updated Description - Section 1.0.4: Added a warning note before the Remediation step - Section 2.0.1: Changed remediation section, step #3 from 744 to 740 - Section 2.0.2: Updated Rationale - Section 3.1.8: Updated Rationale - Section 3.1.13: Changed the recommended value to NO - Section 3.1.14: Page 29, Step #3 should be MAXAPPLS, and not DISCOVER_DB - Section 3.1.16: Added a note before the remediation step - Section 3.3.1: Remediation Step #2 should say admin manager configuration, not database manager configuration -Section 3.3.2: Remediate Step #2 should say admin manager configuration, not database manager configuration - Section 6: Added an additional comment - Section 6.0.26: Rationale should say “public should not” as opposed to “public should” - Section 6.0.27: Rationale should say “public should not” as opposed to “public should” - Section 6.0.28: Rationale should say “public should not” as opposed to “public should” - Section 7.0.1: Changed the description to say, “system administrator group” as opposed to the “operating system group” - Section 7.0.2: Changed the description to say, “system administrator group” as opposed to the “operating system group” - Section 7.0.3: Changed the description to say, “system administrator group” as opposed to the “operating system group” - Section 7.0.4: Changed the description to say, “system administrator group” as opposed to the “operating system group” -Section 8.0.5: Remediation, Step #2, removed the “`drop database toolsdb`” command - Added Section 8.0.6: Enable SSL communication

	with LDAP server - Added section 8.0.7: Secure the permission of the IBMLDAPSecurity.ini file - Added secion 8.0.8: Secure the permission of the SSLconfig.ini file