CENTER FOR INTERNET SECURITY

CIS Oracle MySQL Community Server 5.6

v1.0.0 - 09-16-2014 DRAFT

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

CIS SECURITY BENCHMARKS TERMS OF USE

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

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

UNDER THE FOLLOWING TERMS AND CONDITIONS:

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

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

Table of Contents

Overview

This document, CIS Oracle MySQL Community Server 5.6 Benchmark, provides prescriptive guidance for establishing a secure configuration posture for MySQL Community Server 5.6. This guide was tested against MySQL Community Server 5.6 running on Linux. To obtain the latest version of this guide, please visit http://benchmarks.cisecurity.org. If you have questions, comments, or have identified ways to improve this guide, please write us at feedback@cisecurity.org.

Intended Audience

This document is intended for system and application administrators, security specialists, auditors, help desk, and platform deployment personnel who plan to develop, deploy, assess, or secure solutions that incorporate Oracle MySQL Community Server 5.6.

Consensus Guidance

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

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

Typographical Conventions

The following typographical conventions are used throughout this guide:

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

Scoring Information

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

Scored

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

Not Scored

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

Profile Definitions

The following configuration profiles are defined by this Benchmark:

- **Level 1 - MySQL RDBMS on Linux**

  Items in this profile apply to MySQL Community Server 5.6 running on Linux/UNIX and intend to:

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

- **Level 2 - MySQL RDBMS on Linux**

  This profile extends the "Level 1 - MySQL RDBMS on Linux" profile. Items in this profile apply to MySQL Community Server 5.6 running on Linux and exhibit one or more of the following characteristics:

  - are intended for environments or use cases where security is paramount
  - acts as defense in depth measure
  - may negatively inhibit the utility or performance of the technology.

- **Level 1 - MySQL RDBMS**

  Items in this profile apply to MySQL Community Server 5.6 and intend to:

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

  Note: the intent of this profile is to include checks that can be assessed by remotely connecting to a MySQL RDBMS. Therefore, file sytem-related checks are not contained in this profile.

- **Level 2 - MySQL RDBMS**

  This profile extends the "Level 1 - MySQL RDBMS" profile and exhibit one or more of the following characteristics:

  - are intended for environments or use cases where security is paramount
  - acts as defense in depth measure
  - may negatively inhibit the utility or performance of the technology.

Note: the intent of this profile is to include checks that can be assessed by remotely connecting to a MySQL RDBMS. Therefore, file sytem-related checks are not contained in this profile.

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:

Editors

Daniël van Eeden

Recommendations

1 Operating System Level Configuration

This section contains recommendations related to the Operating System on which the MySQL database server is running.

1.1 Dedicate Machine Running MySQL (Not Scored)

Profile Applicability:

- Level 2 - MySQL RDBMS on Linux

Description:

It is recommended that MySQL Server software be installed on a dedicated server. This architectural consideration affords flexibility in that the database server can be placed on a separate zone allowing access only from particular hosts and over particular protocols.

Rationale:

The attack surface is reduced on a server with only the underlying operating system, MySQL server software, and any security or operational tooling that may be additionally installed. A smaller attack surface reduces the probability of the data within MySQL being compromised.

Audit:

Verify there are no other roles enabled for the underlying operating system and that no additional applications or services unrelated to the proper operation of the MySQL server software are installed.

Remediation:

Remove excess applications or services and/or remove unnecessary roles from the underlying operating system.

Impact:

Care must be taken that applications or services that are required for the proper operation of the operating system are not removed.

Custom applications may need to be modified to accommodate database connections over the network rather than on the use (e.g., using TCP/IP connections).

Additional hardware and operating system licenses may be required to make the architectural change.

1.2 Use Dedicated Least Privileged Account for MySQL Daemon/Service (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

The MySQL user should have a limited set of permissions.

Rationale:

Utilizing a least privilege account for MySQL to execute as may reduce the impact of a MySQL-born vulnerability. A restricted account will be unable to access resources unrelated to MySQL, such as operating system configurations.

Audit:

The user account must only be used for MySQL, not for other services.

On Linux:

ps -ef | grep mysqld

Then check if the user for this process is not root.

Also run "sudo -l" as the MySQL user or check the sudoers file.

Remediation:

Create a user which is only used for running MySQL and directly related processes. This user must not have administrative rights to the system.

Impact:

If a administrative user is used a successful attack on MySQL might allow an attacker to gain access to the whole system. If the same user is used for multiple services an succesful attack on one service will result in access to multiple services.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/changing-mysql-user.html
2. http://dev.mysql.com/doc/refman/5.6/en/server-options.html#option_mysqld_user

1.3 Place Databases on Non-System Partitions (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

The database should not be located on a system partition. It should not be on the common or root (/) file system

Rationale:

Moving the database off the system partition will reduce the probability of denial of service via the exhaustion of available disk space to the operating system.

Audit:

1. Get data folder name "`show variables like 'datadir';`"
2. Verify that the database is not located on the root or system partition
    1. `df -h <datadir>` should not return "/", "/var" or "/usr"

Remediation:

Move the database to a non-system partition.

Impact:

Moving the database to a non-system partition may be difficult depending on whether there was only a single partition when the operating system was set up and whether there are additional storage available.

Default Value:

Not Applicable.

1.4 Disable MySQL Command History (Scored)

Profile Applicability:

- Level 2 - MySQL RDBMS on Linux

Description:

On Linux/UNIX, the MySQL client logs statements executed interactively to a history file. By default, this file is named `.mysql_history` in the user's home directory. Most interactive commands run in the MySQL client application are saved to a history file. The MySQL command history should be disabled.

Rationale:

Disabling the MySQL command history reduces the probability of exposing sensitive information, such as passwords and encryption keys.

Audit:

Check the existence of `$HOME/.mysql_history` on all users in the operating system where the MySQL console application is installed. If it exists, check that it is symbolically linked to `/dev/null`.

Remediation:

1. Remove `.mysql_history` if it exists.
2. Use either of the techniques below to prevent it from being created again:
    1. Set the `MYSQL_HISTFILE` environment variable to `/dev/null`. This will need to be placed in the shell's startup script.
    2. Create `$HOME/.mysql_history` as a symbolic to `/dev/null`.

```
> ln -s /dev/null $HOME/.mysql_history
```

Impact:

By disabling MySQL command history, one will not be able to access the interactive MySQL access history of that user.

Default Value:

By default, the MySQL command history file is located in `$HOME/.mysql_history`.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/mysql-logging.html
2. http://bugs.mysql.com/bug.php?id=72158

1.5 Verify MYSQL_PWD environmental variable not used (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux
- Level 2 - MySQL RDBMS on Linux

Description:

MySQL can read the database password from an environmental variable called `MYSQL_PWD`.

Rationale:

The use of the `MYSQL_PWD` environment variable implies the clear text storage of MySQL credentials. Avoiding this may increase assurance that the confidentiality of MySQL credentials is preserved.

Audit:

grep MYSQL_PWD /proc/*/environ

grep MYSQL_PWD /home/*/.{bashrc,profile,bash_profile}

Remediation:

Check which users and/or scripts are setting MYSQL_PWD and change them to use a more secure method.

Impact:

Someone with access to /proc/*/environ might learn the password.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/environment-variables.html

2. https://blogs.oracle.com/myoracklediary/entry/how_to_check_environment_variables

1.6 Disable Interactive Login (Scored)

Profile Applicability:

- Level 2 - MySQL RDBMS on Linux

Description:

When the MySQL user is created, the user may have interactive access to the operating system. Interactive access by the MySQL user is unnecessary and should be disabled.

Rationale:

Preventing the MySQL user from logging in interactively may reduce the impact of a compromised MySQL account. There is also more accountability as accessing the operating system where the MySQL server lies will require the user's own account.

Audit:

Run `getent passwd mysql` and check which shell is being used for the mysql user.

Remediation:

Linux/UNIX: Set the user's shell to `/sbin/nologin`, or similar.

Impact:

This setting will prevent the MySQL administrator from interactively logging into the operating system using the MySQL user. Instead, the administrator will need to log in using one's own account.

2 File System Permissions

The File System Permissions are critical for keeping the data and configuration of the MySQL server secure.

2.1 Data Directory is Read/Write by MySQL User Only (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

- Level 2 - MySQL RDBMS on Linux

Description:

The data directory is the location of the MySQL databases.

Rationale:

Limiting the accessibility of these objects will protect the confidentiality, integrity, and availability of the MySQL database.

Audit:

1. Identify the location of the data directory using SQL: "`show variables like 'datadir';`"
2. Verify permissions of the data directory.

Remediation:

Change the ownership of the data directory and its sub directories to the mysql user.

Impact:

If someone is allowed to read files from the data directory he or she might be able to read data from the mysql.user table which contains the passwords.

Creating files might lead to denial of service or might allow someone to gain access to specific data by manually creating a file with a view definition or a similar method.

2.2 Permission log files to be readable and writeable by MySQL user and authorized administrators only (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

Log files might contain sensitive data and should be protected.

Rationale:

Limiting the accessibility of these objects will protect the confidentiality, integrity, and availability of the MySQL logs.

Audit:

1. Find `log_bin` entry in configuration file (contains path to logs)
2. Find `log_error` entry in configuration file (contains path to logs)
3. Find `slow_query_log_file` entry in configuration file (contains path to logs)
4. Find `relay-log` entry in configuration file (contains path to logs)
5. Find `general_log_file` entry in configuration file (contains path to logs)
6. Verify permissions

Remediation:

Change permissions of the files and directories.

Impact:

If an attacker has access to one or more log files he might be able to:

1. tamper with log events to delete evidence
2. gain access to sensitive data
3. break replication and thus create a Denial-of-service attack on the MySQL infrastructure.

2.3 SSL key files should be readable by MySQL user. No other read or write permissions (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux
- Level 2 - MySQL RDBMS on Linux

Description:

The private SSL key is located in the SSL key file and might be used to protect the network communication.

Rationale:

Limiting the accessibility of these objects will protect the confidentiality, integrity, and availability of the MySQL database.

Audit:

1. If the global variables have_ssl or have_openssl have the value "YES"
    1. Locate files using the following SQL: "`show variables like 'ssl_key';`"

2. Verify permissions

Remediation:

Change the SSL key file permissions so that only the MySQL user has read access.

Impact:

If the contents of the SSL key file is known to an attacker he or she might pretent to be the server. This can be used for a man-in-the-midddle attack.

Depending on the SSL ciphersuite the key might also be used to decipher previously captured network traffic.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/ssl-connections.html

2.4 Plugin Directory is Read/Write by MySQL User Only (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux
- Level 2 - MySQL RDBMS on Linux

Description:

The plugin directory is the location of the MySQL plugins. Plugins are storage engines and user defined functions (UDFs)

Rationale:

Limiting the accessibility of these objects will protect the confidentiality, integrity, and availability of the MySQL database.

Audit:

1. Identify the location of the data directory using SQL: "`show variables like 'plugin_dir';`"
2. The owner should be the MySQL user and group.
3. The permissions should be 755 or 775

Remediation:

Change the ownership and permissions on the plugin_dir.

Impact:

If someone can modify plugins then these plugins might be loaded when the server starts and the code will get executed.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/install-plugin.html

3 General

This section contains recommendations related to various parts of the database server.

3.1 Ensure latest security patches are applied (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

Periodically, updates to MySQL server are released to resolve bugs, mitigation vulnerabilities, and provide new features. It is recommended that MySQL server installations maintain currently with available updates.

Rationale:

Maintaining currency with MySQL patches will help reduce risk associated with known vulnerabilities present in the MySQL server.

Audit:

Execute the following SQL statement to identify the MySQL server version:

```
SHOW VARIABLES WHERE Variable_name LIKE "version";
```

Now compare the version with the security announcements from Oracle and/or the OS if the OS packages are used.

Remediation:

Install the latest patches for your version or upgrade to the latest version.

Impact:

Without the latest security patches MySQL might have known vulnerabilities which might be used by an attacker to gain access.

References:

1. http://www.oracle.com/technetwork/topics/security/alerts-086861.html
2. http://dev.mysql.com/doc/relnotes/mysql/5.6/en/
3. http://web.nvd.nist.gov/view/vuln/search-results?adv_search=true&cves=on&cpe_vendor=cpe%3a%2f%3aoracle&cpe_product=cpe%3a%2f%3aoracle%3amysql&cpe_version=cpe%3a%2f%3aoracle%3amysql%3a5.6.0

3.2 Drop the 'test' database (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

The default MySQL installation comes with an unused database called `test`. It is recommended that the `test` database be dropped.

Rationale:

Dropping the `test` database will reduce the attack surface of the MySQL server.

Audit:

Execute the following SQL statement to determine if the test database is present:

```
SHOW DATABASES LIKE 'test';
```

The above SQL statement will return zero rows

Remediation:

Execute the following SQL statement to drop the `test` database:

```
DROP DATABASE "test";
```

Note: `mysql_secure_installation` performs this operation as well as other security-related activities.

Impact:

The test database can be accessed by all users and can be used to consume system resources.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/mysql-secure-installation.html

3.3 Ensure 'allow-suspicious-udfs' is set to 'FALSE' (Scored)

Profile Applicability:

- Level 2 - MySQL RDBMS

Description:

This option prevents attaching arbitrary shared library functions as user-defined functions by checking for at least one corresponding method named `_init`, `_deinit`, `_reset`, `_clear`, or `_add`.

Rationale:

Preventing shared libraries that do not contain user-defined functions from loading will reduce the attack surface of the server.

Audit:

Perform the following to determine if the recommended state is in place:

- Ensure `--allow-suspicious-udfs` is not specified in the the `mysqld` start up command line.
- Ensure `allow-suspicious-udfs` is not specified in the MySQL option file.

Remediation:

Perform the following to establish the recommended state:

- Remove `--allow-suspicious-udfs` from the `mysqld` start up command line.
- Remove `allow-suspicious-udfs` from the MySQL option file.

Impact:

This prevents someone from loading a library which was not designed to be a MySQL UDF.

Default Value:

`FALSE`

References:

1. http://dev.mysql.com/doc/refman/5.6/en/udf-security.html
2. http://dev.mysql.com/doc/refman/5.6/en/server-options.html#option_mysqld_allow-suspicious-udfs

3.4 Disable 'local_infile' (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

The `local_infile` parameter dictates whether files located on the MySQL client's computer can be loaded or selected via `LOAD DATA INFILE` or `SELECT local_file`.

Rationale:

Disabling `local_infile` reduces an attacker's ability to read sensitive files off the affected server via a SQL injection vulnerability.

Audit:

Execute the following SQL statement and ensure the Value field is set to `OFF`:

```
SHOW VARIABLES WHERE Variable_name = 'local_infile';
```

Remediation:

Add the following line to the `[mysqld]` section of the MySQL configuration file and restart the MySQL service:

```
local-infile=0
```

Impact:

Disabling `local_infile` will impact the functionality of solutions that rely on it.

Default Value:

`ON`

References:

1. http://dev.mysql.com/doc/refman/5.6/en/string-functions.html#function_load-file
2. http://dev.mysql.com/doc/refman/5.6/en/load-data.html

3.5 Ensure mysqld is not started with '--skip-grant-tables' (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

This option causes `mysqld` to start without using the privilege system.

Rationale:

If this option is used, all clients of the affected server will have unrestricted access to all databases.

Audit:

Perform the following to determine if the recommended state is in place:

- Ensure `--skip-grant-tables` is not specified in the the `mysqld` start up command line.

Remediation:

Perform the following to establish the recommended state:

- Remove --skip-grant-tables from the `mysqld` start up command line.

Impact:

Running with skip-grant-tables completely disables authentication and everyone is allowed to access and modify data.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/server-options.html#option_mysqld_skip-grant-tables

3.6 --skip-symbolic-links (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

A symbolic link might be used to gain access to otherwise protected resources.

Rationale:

Prevents sym links being used for data base files. This is especially important when MySQL is executing as root as arbitrary files may be overwritten.

Audit:

1. SQL: "`show variables like 'have_symlink';`"
2. Verify value is "DISABLED"

Remediation:

Set `skip-symbolic-links` in the configuration.

Impact:

The symbolic-links option might allow someone to direct actions by to MySQL server to other files and/or directories.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/symbolic-links.html
2. http://dev.mysql.com/doc/refman/5.6/en/server-options.html#option_mysqld_symbolic-links

3.7 The InnoDB memcached Plugin must be disabled (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

The InnoDB memcached Plugin allows users to access data stored in InnoDB with the memcached protocol.

Rationale:

The memcached plugin doesn't use the normal authentication layers of the server.

Audit:

```
select * from information_schema.plugins where PLUGIN_NAME='daemon_memcached'\G
```

This should not return any rows.

Remediation:

```
uninstall plugin daemon_memcached;
```

Impact:

By default the plugin doesn't do authentication, which means that anyone with access to the TCP/IP port of the plugin can access and modify the data. Not all data is exposed by default.

Default Value:

disabled

References:

1. http://dev.mysql.com/doc/refman/5.6/en/innodb-memcached-security.html

3.8 Ensure 'secure_file_priv' is set (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

The secure_file_priv option restricts to paths used by LOAD DATA INFILE or SELECT local_file. It is recommended that this option be set to a file system location that contains only resources expected to be loaded by MySQL.

Rationale:

Setting secure_file_priv reduces an attacker's ability to read sensitive files off the affected server via a SQL injection vulnerability.

Audit:

Execute the following SQL statement and ensure one row is returned:

```
SHOW VARIABLES WHERE Variable_name = 'secure_file_priv' and Value RLIKE 'a-zA-Z0-9';
```

Note: the RLIKE constraint is meant to ensure Value is not empty

Remediation:

Add the following line to the [mysqld] section of the MySQL configuration file and restart the MySQL service:

```
secure_file_priv=<path_to_load_directory>
```

Impact:

Solutions that rely on loading data from various sub-directories may be negatively impacted by this change. Consider consolidating load directories under a common parent directory.

Default Value:

No value set.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/server-system-variables.html#sysvar_secure_file_priv

3.9 Enable strict SQL mode (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

The SQL mode STRICT_ALL_TABLES must be enabled.

Rationale:

By default MySQL will truncate data if it does not fit in a field. This can lead to unknown behaviour.

Audit:

Run SHOW VARIABLES LIKE 'sql_mode'; and check if STRICT_ALL_TABLES is in the list.

Remediation:

Set the sql_mode in the configuration to STRICT_ALL_TABLES.

Impact:

Without strict mode the server tries to do proceed with the action when a error might have been a more secure choice. This might be used by an attacker to truncate data and circumvent data validation.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/server-sql-mode.html

4 MySQL Permissions

This section contains recommendations about user privileges.

4.1 Only admin users should have access to the mysql database (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

Verify access by checking the user and db tables. Use the following two queries: "select user, host from mysql.user where (Select_priv = 'Y') or (Insert_priv = 'Y') or (Update_priv = 'Y') or (Delete_priv = 'Y') or (Create_priv = 'Y') or (Drop_priv = 'Y');" and "select user, host from mysql.db where db = 'mysql' and ((Select_priv = 'Y') or (Insert_priv = 'Y') or (Update_priv = 'Y') or (Delete_priv = 'Y') or (Create_priv = 'Y') or (Drop_priv = 'Y'));"

Rationale:

Limiting the accessibility of the 'mysql' database will protect the confidentiality, integrity, and availability of the data housed within MySQL.

Audit:

SQL: "select user, host from mysql.user where (Select_priv = 'Y') or (Insert_priv = 'Y') or (Update_priv = 'Y') or (Delete_priv = 'Y') or (Create_priv = 'Y') or (Drop_priv = 'Y');" and "select user, host from mysql.db where db = 'mysql' and ((Select_priv = 'Y') or Insert_priv = 'Y') or (Update_priv = 'Y') or (Delete_priv = 'Y') or (Create_priv = 'Y') or (Drop_priv = 'Y'));"

Remediation:

Use the REVOKE statement to remove access from users who shouldn't have access.

Impact:

A user which has direct access to mysql.* might view password hashes and change permissions.

4.2 Do not grant to non Admin users FILE privilege (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

Verify using following query: "select user, host from mysql.user where File_priv = 'Y';"

Rationale:

The FILE privilege allows mysql users to read files from disk and to write files to disk. This may be leveraged by an attacker to further compromise MySQL.

Audit:

1. SQL: "select user, host from mysql.user where File_priv = 'Y';"
2. Ensure proper access controls are in place, and that the principle of least privilege is enforced

Remediation:

Use the REVOKE statement to remove permissions from users who shouldn't have them.

Impact:

A user with the FILE permission might create files on various places on the server.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/privileges-provided.html#priv_file

4.3 Do not grant to non Admin users PROCESS privilege (Not Scored)

Profile Applicability:

- Level 2 - MySQL RDBMS

Description:

Verify using following query: "select user, host from mysql.user where Process_priv = 'Y';"

Rationale:

The PROCESS privilege allows principals to view currently executing MySQL statements, including statements used to manage passwords. This may be leveraged by an attacker to compromise MySQL.

Audit:

1. SQL: "select user, host from mysql.user where Process_priv = 'Y';"
2. Ensure proper access controls are in place, and that the principle of least privilege is enforced

Remediation:

Use the REVOKE statement to remove permissions from users which shouldn't have them.

Impact:

A user with the PROCESS privilege might have had access to the text of the statements running on the server. This might contain sensitive data.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/privileges-provided.html#priv_process

4.4 Do not grant to non Admin users SUPER privilege (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

Verify using following query: "select user, host from mysql.user where Super_priv = 'Y';"

Rationale:

The SUPER privilege allows principals to view and terminate currently executing MySQL statements, including statements used to manage passwords. This privilege also provides the ability to configure MySQL. This may be leveraged by an attacker to compromise MySQL.

Audit:

1. SQL: "select user, host from mysql.user where Super_priv = 'Y';"
2. Ensure proper access controls are in place, and that the principle of least privilege is enforced

Remediation:

Use the REVOKE statement to remove the SUPER privilege from users who shouldn't have it.

Impact:

The SUPER privilege grants a user many privileges including writing to a server with the read_only flag set. It might also be used to remove log files.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/privileges-provided.html#priv_super

4.5 Do not grant to non Admin users SHUTDOWN privilege (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

Verify using following query: "select user, host from mysql.user where Shutdown_priv = 'Y';"

Rationale:

The SHUTDOWN privilege allows principals to shutdown MySQL. This may be leveraged by an attacker to negatively impact the availability of MySQL.

Audit:

1. SQL: "select user, host from mysql.user where Shutdown_priv = 'Y';"

2. Ensure proper access controls are in place, and that the principle of least privilege is enforced

Remediation:

Use the REVOKE statement to remove the SUPER privilege from users who shouldn't have it.

Impact:

The shutdown privilege lets someone stop the MySQL Server. This can be done over a UNIX socket or a TCP/IP connection.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/privileges-provided.html#priv_shutdown

4.6 Do not grant to non Admin users CREATE USER privilege (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

Verify using following query: "select user, host from mysql.user where Create_user_priv = 'Y';"

Rationale:

The CREATE USER privilege allows principals to create MySQL users. This may be leveraged by an attacker to compromise MySQL.

Audit:

1. SQL: "select user, host from mysql.user where Create_user_priv = 'Y';"
2. Ensure proper access controls are in place, and that the principle of least privilege is enforced

Remediation:

Use the REVOKE statement to remove the SUPER privilege from users who shouldn't have it.

Impact:

With the CREATE USER privilege someone can create new user accounts.

4.7 Do not grant to non Admin users global GRANT privilege (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

Allows changing of permissions. Verify using following query: "select user, host from mysql.user where Grant_priv = 'Y';"

Rationale:

The GRANT privilege allows a principal to grant other principals additional privileges. This may be used by an attacker to compromise MySQL.

Audit:

1. SQL: "select user, host from mysql.user where Create_user_priv = 'Y';"
2. Ensure proper access controls are in place, and that the principle of least privilege is enforced

Remediation:

REVOKE GRANT OPTION ON *.* FROM <user>

Impact:

The GRANT option allows a user grant privileges to other users. This is limited to the privileges the granting user has. If a user has a global grant option then this user is able to grant global privileges to other users.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/privileges-provided.html#priv_grant-option

4.8 Do not grant to non Slave users global REPLICATION SLAVE privilege (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

Allows changing of permissions. Verify using following query: "select user, host from mysql.user where Repl_slave = 'Y';"

Rationale:

The REPLICATION SLAVE privilege allows a principal to fetch replication datata from the master. This may be used by an attacker to read/fetch sensitive data from MySQL.

Audit:

1. SQL: "select user, host from mysql.user where Repl_slave = 'Y';"
2. Ensure proper access controls are in place, and that the principle of least privilege is enforced

Remediation:

Use the REVOKE statement to remove the SUPER privilege from users who shouldn't have it.

Impact:

A user with the replication slave privilege can ask the server to send binlog files. The binlog files contain all data changing statements and/or changes in table data. This might include password changes and other sensitive information.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/privileges-provided.html#priv_replication-slave

4.9 Limit DML/DDL grants to specific databases (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

Verify access by checking the mysql.user table.

Rationale:

Grants should be limited to specific databases.

DML: INSERT, SELECT, UPDATE, DELETE

DDL: DROP, CREATE, ALTER

Audit:

```
SELECT User,Host FROM mysql.user WHERE Select_priv='Y' OR Insert_priv='Y' OR
Update_priv='Y' OR Delete_priv='Y' OR Create_priv='Y' OR Drop_priv='Y' or
Alter_priv='Y';
```

Remediation:

Use the REVOKE statement to remove access from users who shouldn't have access.

Impact:

A user which has direct access to mysql.* might view password hashes and change permissions.

A user with global privileges might have access to newly created databases.

5 Auditing and Logging

Configuration options can be added two ways. First is using the MySQL configuration file *my.cnf* and placing options under the proper section of "[mysqld]". Options placed in the configuration file should not prefix with a double dash "--". Options can also be placed on the command line by modifying the MySQL startup script. The startup script is system dependent based on your operating system.

5.1 The errorlog must be enabled (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

The error log must be enabled.

Rationale:

Enabling error logging may increase the ability to detect malicious attempts against MySQL.

Audit:

1. SQL: "show variables like 'log_error';"
2. Verify entry

Remediation:

Add the log_error=<errorlog> option to the global configuration.

Impact:

If the error log is not enabled then connection error might go unnoticed.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/error-log.html

5.2 Logs should be on a nonsystem partition (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux
- Level 2 - MySQL RDBMS on Linux

Description:

The logs should not be part of a system filesystem like /, /var or /usr

Rationale:

Moving the MySQL logs off the system partition will reduce the probability of denial of service via the exhaustion of available disk space to the operating system.

Audit:

The "select @@global.log_bin_basename;" command should return a location which is not on /, /var or /usr.

Remediation:

Change the location of the binlog files.

Impact:

By generating many changes an attacker might be able to fill the filesystem with binlog files. This might cause unavailability for the database or other parts of the system.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/binary-log.html
2. http://dev.mysql.com/doc/refman/5.6/en/replication-options-binary-log.html

5.3 Logging of warnings should be enabled (Not Scored)

Profile Applicability:

- Level 2 - MySQL RDBMS

Description:

Logging of warnings should be enabled by setting log_warnings=2. With values greater than 1, aborted connections are written to the error log, and access-denied errors for new connection attempts are written.

Rationale:

This might help to detect malicious behavior by logging communication errors and aborted connections.

Audit:

```
grep log_warnings /etc/my.cnf

SHOW GLOBAL VARIABLES LIKE 'log_warnings';
```

Remediation:

Add to my.cnf set use SET GLOBAL…. to activate.

Impact:

Without log_warnings set to 2 it an attack might initially go unnoticed.

Default Value:

The option is enabled (1) by default.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/server-options.html#option_mysqld_log-warnings

5.4 Enable audit logging (Not Scored)

Profile Applicability:

- Level 2 - MySQL RDBMS

Description:

Enable audit logging for

- Interactive user sessions
- Application sessions (optional)

Rationale:

Audit logging helps to identify who changed what and when.

Audit:

Check if an audit log is being created for the correct events.

Remediation:

Implement Audit Logging as available with

- The General Query Log
- MySQL Enterprise Audit
- MariaDB Audit Plugin for MySQL
- McAfee MySQL Audit

Impact:

Auditing might be used to detect illicit behaviour by authenticated users.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/query-log.html
2. http://dev.mysql.com/doc/refman/5.6/en/mysql-enterprise-audit.html
3. https://mariadb.com/kb/en/server_audit-mariadb-audit-plugin/
4. https://github.com/mcafee/mysql-audit

5.5 Raw logging of passwords should be disabled (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

Logging of passwords must be disabled by setting log-raw to OFF

Rationale:

This prevents passwords from being in plain text files unnecessary.

Audit:

Check if log-raw is set to anything else than OFF in the configuration file.

Remediation:

Remove log-raw from the configuration or set it to OFF and restart MySQL.

Impact:

With raw logging of passwords enabled someone with access to the log files might see plain text passwords.

Default Value:

OFF

References:

1. http://dev.mysql.com/doc/refman/5.6/en/password-logging.html
2. http://dev.mysql.com/doc/refman/5.6/en/server-options.html#option_mysqld_log-raw

5 Authentication

This section contains configuration recommendations that pertain to the authentication mechanisms of MySQL.

5.1 Ensure 'old_passwords' is not set to '1' or 'ON' (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

This variable controls the password hashing method used by the PASSWORD() function and for the IDENTIFIED BY clause of the CREATE USER and GRANT statements.

Before 5.6.6, the value can be 0 (or OFF), or 1 (or ON). As of 5.6.6, the following value can be one of the following:

- 0 - authenticate with the mysql_native_password plugin
- 1 - authenticate with the mysql_old_password plugin
- 2 - authenticate with the sha256_password plugin

Rationale:

The mysql_old_password plugin leverages an algorithm that can be quickly brute forced using an offline dictionary attack. See CVE-2003-1480 for additional details.

Audit:

Execute the following SQL statement and ensure the Value field is not set to 1 or ON:

```
SHOW VARIABLES WHERE Variable_name = 'old_passwords';
```

Remediation:

Configure mysql to leverage the mysql_native_password or sha256_password plugin. For more information, see:

- http://dev.mysql.com/doc/refman/5.6/en/password-hashing.html
- http://dev.mysql.com/doc/refman/5.6/en/sha256-authentication-plugin.html

Impact:

When old_passwords is set to 1 the PASSWORD() function will create password hashes with a very weak hashing algorithm which might be easy to break if captured by an attacker.

Default Value:

0

References:

1. http://dev.mysql.com/doc/refman/5.6/en/server-system-variables.html#sysvar_old_passwords
2. CVE-2003-1480

5.2 Ensure 'secure_auth' is set to 'ON' (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux
- Level 2 - MySQL RDBMS on Linux

Description:

This option dictates whether the server will deny connections by clients that attempt to use accounts that have their password stored in the mysql_old_password format.

Rationale:

Enabling this option will prevent all use of passwords employing the old format (and hence insecure communication over the network).

Audit:

Execute the following SQL statement and ensure the Value field is not set to ON:

```
SHOW VARIABLES WHERE Variable_name = 'secure_auth';
```

Remediation:

Add the following line to [mysqld] portions of the MySQL option file to establish the recommended state:

```
secure_auth=ON
```

Impact:

Accounts having credentials stored using the old password format will be unable to login. Execute the following command to identify accounts that will be impacted by implementing this setting:

```
SELECT User,Host FROM mysql.user WHERE plugin='mysql_old_password';
```

Default Value:

Before MySQL 5.6.5, this option is disabled by default. As of MySQL 5.6.5, it is enabled by default; to disable it, use --skip-secure-auth.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/server-options.html#option_mysqld_secure-auth

5.3 Do Not Specify Passwords in Command Line (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux
- Level 1 - MySQL RDBMS

Description:

When a command is executed on the command line, for example "mysql -u admin -ppassword", the password may be visible in the user's shell/command history or in the process list.

Rationale:

If the password is visible in the process list or user's shell/command history, an attacker will be able to access the MySQL database using the stolen credentials.

Audit:

Check the process or task list if the password is visible.

Check the shell or command history if the password is visible.

Remediation:

Use "-p" without password and then enter the password when prompted, use a properly secured .my.cnf file, or store authentication information in encrypted format in .mylogin.cnf.

Impact:

Depending on the remediation chosen, additional steps may need to be undertaken like:

- Entering a password when prompted;
- Ensuring the file permissions on .my.cnf is restricted yet accessible by the user;
- Using mysql_config_editor to encrypt the authentication credentials in .mylogin.cnf.

Additionally, not all scripts/applications may be able to use .mylogin.cnf.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/mysql-config-editor.html
2. http://dev.mysql.com/doc/refman/5.6/en/password-security-user.html

5.4 Do not store passwords in the global configuration (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux
- Level 2 - MySQL RDBMS on Linux

Description:

The [client] section of the MySQL configuration file allows setting a password to be used. Verify this option is not used in the global configuration file (my.cnf).

Rationale:

The use of this parameter may negatively impact the confidentiality of the user's password.

Audit:

In the global options file my.cnf, examine the [client] section of the MySQL configuration file and ensure this option is not employed.

Remediation:

Use the mysql_config_editor to store authtentication credentials in .mylogin.cnf in encrypted form.

If not possible, use the user-specific options file, .my.cnf., and restricting file access permissions to the user identity.

Impact:

The global configuration is by default readable for all users on the system. This is needed for global defaults (prompt, port, socket, etc). If a password is present in this file then all users on the system may be able to access it.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/mysql-config-editor.html

5.5 NO_AUTO_CREATE_USER or -- safe-user-create (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

Prevent GRANT from creating a new user unless a non-empty password is also specified

Rationale:

Blank passwords negate the benefits provided by authentication mechanisms.

Audit:

1. SQL: "select @@global.sql_mode;" must contain NO_AUTO_CREATE_USER
2. SQL: "select @@session.sql_mode;" must contain NO_AUTO_CREATE_USER

Remediation:

Add the NO_AUTO_CREATE_USER to the sql_mode setting in the my.cnf config file.

Impact:

Without this setting an administrative user might accidentally create a user without a password.

5.6 Ensure no MySQL accounts have a blank password (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

Blank passwords allow a user to login without using a password.

Rationale:

Blank passwords negate the benefits provided by authentication mechanisms.

Audit:

Execute the following SQL query to determine if any users have a blank password:

```
SELECT User
FROM mysql.user
WHERE (plugin IN('mysql_native_password', 'mysql_old_password')
 AND (LENGTH(Password) = 0
 OR Password IS NULL))
OR (plugin='sha256_password' AND LENGTH(authentication_string) = 0);
```

No rows will be returned if all accounts have a password set.

Remediation:

Set a password for each affected user.

Impact:

Without a password only knowing the username and the list of allowed hosts will allow someone to connect to the server and assume the identity of the user.

5.7 Enforce complex passwords (Scored)

Profile Applicability:

- Level 2 - MySQL RDBMS

Description:

A policy should be in place to require complex passwords on all database accounts.

Rationale:

Complex passwords help mitigate dictionary, brute forcing, and other password attacks.

Audit:

```
SHOW VARIABLES LIKE 'validate_password%';
validate_password_length should be 8 or more
validate_password_mixed_case_count should be 1 or more
validate_password_number_count should be 1 or more
validate_password_special_char_count should be 1 or more
validate_password_policy should be MEDIUM or STRONG

This should be in the global configuration:
plugin-load=validate_password.so
validate-password=FORCE_PLUS_PERMANENT

Check if users have a password which is identical to the username:
set old_passwords=1;
select user,host from mysql.user where password(user)=password;
set old_passwords=0;
select user,host from mysql.user where password(user)=password;
```

Remediation:

Add to the global configuration and restart the server:

```
plugin-load=validate_password.so
validate-password=FORCE_PLUS_PERMANENT
validate_password_length=8
validate_password_mixed_case_count=1
validate_password_number_count=1
validate_password_special_char_count=1
validate_password_policy=MEDIUM
```

And change passwords for users which have passwords which are identical to their username.

Impact:

This prevents users from choosing weak passwords which can easily be guessed.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/validate-password-plugin.html

5.8 Each database user should be used for single purpose/person (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

Database user accounts should not be reused for multiple applications or users.

Rationale:

Utilizing unique database accounts across applications will reduce the impact of a compromised MySQL account.

Audit:

Each user should be linked to one of these

- system accounts
- a person
- an application

Remediation:

Add/Remove users so that each user is only used for one specific purpose.

Impact:

If a user is reused then a compromise of this user will compromise multiple parts of the system and/or application.

5.9 Verify if users have wildcard ('%') in hostname (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

When possible, host parameters for users should not contain wildcards ('%'). This can be checked using "select user from mysql.user where host = '%';".

Rationale:

Avoiding the use of wildcards within hostnames will ensure that only trusted principals are capable of interacting with MySQL.

Audit:

1. SQL: "select user from mysql.user where host = '%';"
2. Verify that no results are returned

Remediation:

DROP the users which may connect from any host.

Impact:

A user which may connect from any host has a large attack surface.

5.10 Ensure no anonymous accounts exist (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

Anonymous accounts are users with empty usernames (''). Anonymous accounts have no passwords, so anyone can use them to connect to the MySQL server.

Rationale:

Removing anonymous accounts will help ensure that only identified and trusted principals are capable of interacting with MySQL.

Audit:

Execute the following SQL query to identify anonymous accounts:

```
SELECT user,host FROM mysql.user WHERE user = '';
```

The above query will return zero rows if no anonymous accounts are present.

Remediation:

Remove the anonymous users or assign them a name.

Alternatively, execute the mysql_secure_installation utility to perform this operation.

Impact:

By removing the anonymous users, it will prevent anyone from using them to connect to the MySQL server.

Default Value:

Using the standard installation script, mysql_install_db, it will create two anonymous accounts: one for the host 'localhost' and the other for the network interface's IP address.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/mysql-secure-installation.html
2. https://dev.mysql.com/doc/refman/5.6/en/default-privileges.html

6 Network

This section contains recommendations related to how the MySQL server uses the network.

6.1 Client Set To Verify Server's Certificate (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux
- Level 2 - MySQL RDBMS on Linux

Description:

This option is available for client programs only, not the server. It causes the client to check the server's Common Name (CN) against the host name the client uses for connecting to the server, and teh connection fails if there is a mismatch.

This option was added in MySQL 5.1.11.

Rationale:

Verifying the server's certificate will help protect against man in the middle attacks.

Audit:

In the [client] portion of the MySQL configuration file, check for the existence of:

```
ssl_verify_server_cert
```

Alternatively, check if the MySQL client is being run with the --ssl-verify-server-cert option.

Remediation:

Configure MySQL (either through configuration file or through a run-time option) to verify the server certificate.

Impact:

If the Common Name (CN) is different from the server's host name, it will prevent the client connection from being successful.

Default Value:

By default, server certificate verification is disabled.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/ssl-options.html#option_general_ssl-verify-server-cert

6.2 Must use SSL over untrusted networks (internet) or when restricted PII is transferred (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

All network traffic must use SSL/TLS when traveling over untrusted networks.

Rationale:

SSL will protect the confidentiality and integrity of sensitive information as it traverses untrusted networks.

Audit:

1. SQL: "show variables like 'have_openssl';" is "YES"
2. SQL: "show variables like 'ssl_cert';" is set (and file exists)
3. SQL: "show variables like 'ssl_key';" is set (and file exists)
4. SQL: "show variables like 'ssl_ca';" is set (and file exists)
5. Users are forced to use SSL by setting the mysql.user.ssl_type field to ANY, X509, or SPECIFIED

Note: have_openssl is an alias for have_ssl as of MySQL 5.0.38.

Remediation:

Follow the procedures as documented in the MySQL 5.6 Reference Manual to setup SSL.

Impact:

The SSL/TLS based version of the MySQL protocol should be used to prevent eavesdropping of the communication. Authentication in SSL/TLS should be used to prevent man-in-the-middle attacks.

References:

1. http://dev.mysql.com/doc/refman/5.6/en/ssl-connections.html

6.3 Do not use a default or example certificate. Generate a key specifically for MySQL (Not Scored)

Profile Applicability:

- Level 2 - MySQL RDBMS

Description:

The SSL certificate and key used by MySQL should be used only for MySQL and only for one instance.

Rationale:

Use of default certificates can allow an attacker to impersonate the MySQL server.

Audit:

Check if the certificate is bound to one instance of MySQL.

Remediation:

Generate a new certificate/key per MySQL instance.

Impact:

If a the key is used on multiple system then a compromise of one system leads to compromise of the network traffic of all servers which use the same key.

6.4 Use --skip-networking startup option (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

Do not allow TCP/IP connections; do not bind to a port. Use if no remote access is needed.

Rationale:

If remote access is not required, preventing MySQL from binding to a network socket may reduce the exposure of a MySQL-born vulnerability.

Audit:

1. SQL: "show variables like 'skip_networking';"
2. Verify value is "ON"

Remediation:

set "skip-networking" in the global configuration and restart the server.

Impact:

With skip-networking disabled the attack surface of the MySQL server is larger.

7 Backup and Disaster Recovery

This section contains recommendations related to backup and recovery

7.1 Backup policy in place (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux
- Level 2 - MySQL RDBMS on Linux

Description:

A backup policy should be in place.

Rationale:

Backing up MySQL databases, including 'mysql', will help ensure the availability of data in the event of an incident.

Audit:

Check with "crontab -l" if there is a backup schedule.

Remediation:

Create a backup policy and backup schedule.

Impact:

Without backups it might be hard to recover from an incident.

7.2 Verify backups are good (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux
- Level 2 - MySQL RDBMS on Linux

Description:

Backups should be validated on a regular basis.

Rationale:

Verifying that backups are occurring appropriately will help ensure the availability of data in the event of an incident.

Audit:

Check reports of backup validation tests.

Remediation:

Implement regular backup checks and document each check.

Impact:

Without a well tested backup it might be hard to recover from an incident if the backup procedure contains errors or doesn't include all required data.

7.3 Secure backup credentials (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux
- Level 2 - MySQL RDBMS on Linux

Description:

The password, certificate and any other credentials should be protected.

Rationale:

A user with full privileges is needed for backup. The credentials for this user should be protected

Audit:

Check permissions of files containing passwords and/or ssl keys.

Remediation:

Change file permissions

Impact:

When the backup credentials are not properly secured then they might be abused to gain access to the server. The backup user needs an account with many privileges, so the attacker can can gain (almost) complete access to the server.

7.4 The backups should be properly secured (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux
- Level 2 - MySQL RDBMS on Linux

Description:

The backup files will contain all data in the databases. Filesystem permissions and/or encryption should be used to prevent non authorized users from gaining access to the backups.

Rationale:

Backups should be considered sensitive information.

Audit:

Check who has access to the backup files.

- Are the files world-readable (e.g. rw-r--r-)
  - Are they stored in a world readble directory?
- Is the group MySQL and/or backup specific?
  - If not: the file and directory must not be group readble
- Are the backups stored offsite?
  - Who has access to the backups?
- Are the backups encrypted?
  - Where is the encryption key stored?
  - Does the encryption key consists of a guessable password?

Remediation:

Implement encryption or use filesystem permissions.

Impact:

If an unauthorized user can access backups then they have access to all the data which is in the database. This is true for unencrypted backups and for encrypted backups if the encryption key is stored along with the backup.

7.5 Point in time recovery (Not Scored)

Profile Applicability:

- Level 2 - MySQL RDBMS on Linux

Description:

With binlogs it is possible to implement point-in-time recovery. This makes it possible to restore the changes between the last full backup and the point-in-time.

Enabling binlogs is not sufficient, a restore procedure should be created and has to be tested.

Rationale:

This can reduce the amount of information lost.

Audit:

Check if binlogs are enabled and if there is a restore procedure.

Remediation:

Enable binlogs and create and test a restore procedure.

Impact:

Without point-in-time recovery the data which was stored between the last backup and the time of disaster might not be recoverable.

7.6 Disaster recovery plan (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux
- Level 2 - MySQL RDBMS on Linux

Description:

A disaster recovery plan should be created.

A slave in a different datacenter can be used or offsite backups. There should be information about what time a recovery will take and if the recovery site has the same capacity.

Rationale:

A disaster recovery should be planned.

Audit:

Check if there is a disaster recovery plan

Remediation:

Create a disaster recovery plan

Impact:

Without a well tested disaster recovery plan it might not be possible to recover in time.

7.7 Backup of configuration and related files (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux
- Level 2 - MySQL RDBMS on Linux

Description:

The following files should be included in the backup:

- Configuration files (my.cnf and included files)
- SSL files (certificates, keys)
- User Defined Functions (UDFs)
- Source code for customizations

Rationale:

These files are required to be able to fully restore an instance.

Audit:

Check if these files are in used and are saved in the backup.

Remediation:

Add these files to the backup

Impact:

Without a complete backup it might not be possible to fully recover.

8 Replication

Everything related to replicating data from one server to another.

8.1 Secure replication traffic (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS
- Level 2 - MySQL RDBMS

Description:

The replication traffic between servers should be secured.

Rationale:

The replication traffic should be secured as it gives access to all transfered information and might leak passwords.

Audit:

Check if the replication traffic is using

- A private network
- A VPN
- SSL/TLS
- A SSH Tunnel

Remediation:

Secure the network traffic

Impact:

When the replication traffic is not secured someone might be able to capture passwords and other sensitive information when sent to the slave.

8.2 Replication user should be limited (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux
- Level 2 - MySQL RDBMS on Linux

Description:

Any replication users should have the least possible privileges.

A slave user should be limited to one slave only.

The slave user must be limited to one host.

Rationale:

This should make it easier to drop one slave user and/or change passwords regularly. It also limits what a slave user can do in case the account is compromised.

Audit:

Any other privileges than REPLICATION SLAVE may only be granted if they serve a clear purpose. Check if the user is limited to one slave and one host.

Remediation:

remove privileges which are not needed and limit access to one host per slave user.

Impact:

As the password for the replication user is in most cases stored on the slave it has a high risk of being abused.

8.3 Use a table as master info repository (Scored)

Profile Applicability:

- Level 2 - MySQL RDBMS

Description:

Use a table to store the information the client needs to connect to the master.

Rationale:

The password which the client uses is stored in the master info repository, which by default is a plaintext file. The TABLE master info repository is a bit safer, but with filesystem access it's still possible to gain access to the password the slave is using.

Audit:

```
SHOW GLOBAL VARIABLES LIKE 'master_info_repository';
```

The result should be TABLE instead of FILE. There should not be a master.info file in the datadir.

Remediation:

Change the master_info_repository to TABLE

Impact:

If the master_info_repository is set to FILE the password is stored in a cleartext master.info file.

Default Value:

FILE

References:

1. http://dev.mysql.com/doc/refman/5.6/en/replication-options-slave.html#sysvar_master_info_repository

Appendix: Change History

Date	Version	Changes for this version