CENTER FOR INTERNET SECURITY

CIS Oracle MySQL Community Server 5.7

v1.0.0 - 12-29-2015

http://benchmarks.cisecurity.org

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 5.7 Benchmark, provides prescriptive guidance for establishing a secure configuration posture for MySQL Community Server 5.7. This guide was tested against MySQL Community Server 5.7 running on Ubuntu Linux 14.04, but applies to other linux distributions as well. 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 5.7.

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 5.7 running on Linux 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 5.7 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 5.7 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 system-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 system-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:

Contributor
Adam Montville
Robert Thomas

Editor
Binod Bista
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 Place Databases on Non-System Partitions (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

It is generally accepted that host operating systems should include different filesystem partitions for different purposes. One set of filesystems are typically called "system partitions", and are generally reserved for host system/application operation. The other set of filesystems are typically called "non-system partitions", and such locations are generally reserved for storing data.

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:

Execute the following steps to assess this recommendation:

- Discover the datadir by executing the following SQL statement

```
show variables where variable_name = 'datadir';
```

- Using the returned `datadir Value` from the above query, execute the following in a system terminal

```
df -h <datadir Value>
```

The output returned from the `df` command above should not include root ('/'), "/var", or "/usr".

Remediation:

Perform the following steps to remediate this setting:

1. Choose a non-system partition `new location` for the MySQL data
2. Stop `mysqld` using a command like: `service mysql stop`
3. Copy the data using a command like: `cp -rp <datadir Value> <new location>`
4. Set the `datadir` location to the `new location` in the MySQL configuration file
5. Start mysqld using a command like: service mysql start

NOTE: On some Linux distributions you may need to additionally modify `apparmor` settings. For example, on a Ubuntu 14.04.1 system edit the file `/etc/apparmor.d/usr.sbin.mysqld` so that the `datadir` access is appropriate. The original might look like this:

```
# Allow data dir access
 /var/lib/mysql/ r,
 /var/lib/mysql/** rwk,
```

Alter those two paths to be the new location you chose above. For example, if that new location were `/media/mysql`, then the `/etc/apparmor.d/usr.sbin.mysqld` file should include something like this:

```
# Allow data dir access
  /media/mysql/ r,
  /media/mysql/** rwk,
```

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.

1.2 Use Dedicated Least Privileged Account for MySQL Daemon/Service (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

As with any service installed on a host, it can be provided with its own user context. Providing a dedicated user to the service provides the ability to precisely constrain the service within the larger host context.

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:

Execute the following command at a terminal prompt to assess this recommendation:

```
ps -ef | egrep "^mysql.*$"
```

If no lines are returned, then this is a finding.

NOTE: It is assumed that the MySQL user is `mysql`. Additionally, you may consider running `sudo -l` as the MySQL user or to 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.

References:

1. http://dev.mysql.com/doc/refman/5.7/en/changing-mysql-user.html
2. http://dev.mysql.com/doc/refman/5.7/en/server-options.html#option_mysqld_user

1.3 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:

Execute the following commands to assess this recommendation:

```
find /home -name ".mysql_history"
```

For each file returned determine whether that file is symbolically linked to `/dev/null`.

Remediation:

Perform the following steps to remediate this setting:

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
```

Default Value:

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

References:

1. http://dev.mysql.com/doc/refman/5.7/en/mysql-logging.html
2. http://bugs.mysql.com/bug.php?id=72158

1.4 Verify That the MYSQL_PWD Environment Variables Is Not In Use (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

MySQL can read a default database password from an environment 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:

To assess this recommendation, use the /proc filesystem to determine if MYSQL_PWD is currently set for any process

```
grep MYSQL_PWD /proc/*/environ
```

This may return one entry for the process which is executing the grep command.

Remediation:

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

Default Value:

Not set.

References:

1. http://dev.mysql.com/doc/refman/5.7/en/environment-variables.html
2. https://blogs.oracle.com/myoraclediary/entry/how_to_check_environment_variables

1.5 Disable Interactive Login (Scored)

Profile Applicability:

- Level 2 - MySQL RDBMS on Linux

Description:

When created, the MySQL user may have interactive access to the operating system, which means that the MySQL user could login to the host as any other user would.

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. Interactive access by the MySQL user is unnecessary and should be disabled.

Audit:

Execute the following command to assess this recommendation

```
getent passwd mysql | egrep "^.*[\/bin\/false|\/sbin\/nologin]$"
```

Lack of output implies a finding.

Remediation:

Perform the following steps to remediate this setting:

- Execute one of the following commands in a terminal

```
usermod -s /bin/false
usermod -s /sbin/nologin
```

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.

1.6 Verify That 'MYSQL_PWD' Is Not Set In Users' Profiles (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

MySQL can read a default database password from an environment 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:

To assess this recommendation check if MYSQL_PWD is set in login scripts using the following command:

```
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.

Default Value:

Not set.

References:

1. http://dev.mysql.com/doc/refman/5.7/en/environment-variables.html
2. https://blogs.oracle.com/myoraclediary/entry/how_to_check_environment_variables

2 Installation and Planning

This section contains important considerations when deploying MySQL services to your production network. The recommendations made herein are not scored from a benchmark perspective and generally align with best current practices as conveyed in most control frameworks.

Note also that configuration options can be added two ways. First is using the MySQL configuration file (e.g. `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.

2.1 Backup and Disaster Recovery

This section contains recommendations related to backup and recovery

2.1.1 Backup policy in place (Not Scored)

Profile Applicability:

- Level 1 - 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.

2.1.2 Verify backups are good (Not Scored)

Profile Applicability:

- Level 1 - 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.

2.1.3 Secure backup credentials (Not Scored)

Profile Applicability:

- Level 1 - 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 gain (almost) complete access to the server.

2.1.4 The backups should be properly secured (Not Scored)

Profile Applicability:

- Level 1 - 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 readable directory?
- Is the group MySQL and/or backup specific?
  - If not: the file and directory must not be group readable
- 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 that is in the database. This is true for unencrypted backups and for encrypted backups if the encryption key is stored along with the backup.

2.1.5 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.

2.1.6 Disaster recovery plan (Not Scored)

Profile Applicability:

- Level 1 - 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.

2.1.7 Backup of configuration and related files (Not Scored)

Profile Applicability:

- Level 1 - 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.

2.2 Dedicate Machine Running MySQL (Not Scored)

Profile Applicability:

- Level 1 - 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.

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

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

When a command is executed on the command line, for example `mysql -u admin -p password`, 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/5.7/en/mysql-config-editor.html
2. http://dev.mysql.com/doc/refman/5.7/en/password-security-user.html

2.4 Do Not Reuse Usernames (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

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.

2.5 Do Not Use Default or Non-MySQL-specific Cryptographic Keys (Not Scored)

Profile Applicability:

- Level 2 - MySQL RDBMS on Linux

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.

2.6 Set a Password Expiry Policy for Specific Users (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

Password expiry for specific users provides user passwords with a unique time bounded lifetime.

Rationale:

Allows additional security factors pertinent to a specific user to provide further password security; predetermined by varying security needs and usability requirements in a system or organization [1].

Audit:

Returns all users currently using the global setting default_password_life, and hence have no specific user password expiry set.

```
SELECT user, host, password_lifetime from mysql.user from mysql.user where
password_lifetime IS NULL;
```

Remediation:

Using the user and host information from the audit procedure, set each user a password lifetime e.g.

```
ALTER USER 'jeffrey'@'localhost' PASSWORD EXPIRE INTERVAL 90 DAY;
```

Impact:

Further password security factors unique to a system or organization's security policy maybe being overlooked.

Default Value:

NULL. The user's password_lifetime takes on the value set in global default_password_lifetime variable.

References:

1. [1] http://csrc.nist.gov/publications/drafts/800-118/draft-sp800-118.pdf(ES-2)

3 File System Permissions

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

3.1 Ensure 'datadir' Has Appropriate Permissions (Scored)

Profile Applicability:

- Level 1 - 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. If someone other than the MySQL user 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 passwords. Additionally, the ability to create files can lead to denial of service, or might otherwise allow someone to gain access to specific data by manually creating a file with a view definition.

Audit:

Perform the following steps to assess this recommendation:

- Execute the following SQL statement to determine the Value of datadir

```
show variables where variable_name = 'datadir';
```

- Execute the following command at a terminal prompt

```
ls -l <datadir>/.. | egrep "^d[r|w|x]{3}------\s*.\s*mysql\s*mysql\s*\d*.*mysql"
```

Lack of output implies a finding.

Remediation:

Execute the following commands at a terminal prompt:

```
chmod 700 <datadir>
chown mysql:mysql <datadir>
```

3.2 Ensure 'log_bin_basename' Files Have Appropriate Permissions (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

MySQL can operate using a variety of log files, each used for different purposes. These are the binary log, error log, slow query log, relay log, and general log. Because these are files on the host operating system, they are subject to the permissions structure provided by the host and may be accessible by users other than the MySQL user.

Rationale:

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

Audit:

Perform the following steps to assess this recommendation:

- Identify the basename of binary log files (`log_bin_basename`)by executing the following statement

```
show variables like 'log_bin_basename';
```

- Verify permissions are `660` for `mysql:mysql` on each log file of the form `log_bin_basename.nnnnnn`.

Remediation:

Execute the following command for each log file location requiring corrected permissions:

```
chmod 660 <log file>
chown mysql:mysql <log file>
```

Impact:

Changing the permissions of the log files might have impact on monitoring tools which use a logfile adapter. Also the slow query log can be used for performance analysis by application developers.

If the permissions on the relay logs and binary log files are accidentally changed to exclude the user account which is used to run the MySQL service then this might break replication.

The binary log file can be used for point in time recovery so this can also affect backup, restore and disaster recovery procedures.

3.3 Ensure 'log_error' Has Appropriate Permissions (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

MySQL can operate using a variety of log files, each used for different purposes. These are the binary log, error log, slow query log, relay log, and general log. Because these are files on the host operating system, they are subject to the permissions structure provided by the host and may be accessible by users other than the MySQL user.

Rationale:

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

Audit:

Perform the following steps to assess this recommendation:

- Find the `log_error` value (`<error_log_path>`)by executing the following statement

```
show global variables like 'log_error';
```

- Verify permissions are `660` for `mysql:mysql` for `<error_log_path>`

Remediation:

Execute the following command for each log file location requiring corrected permissions:

```
chmod 660 <log file>
chown mysql:mysql <log file>
```

Impact:

Changing the permissions of the log files might have impact on monitoring tools which use a logfile adapter. Also the slow query log can be used for performance analysis by application developers.

If the permissions on the relay logs and binary log files are accidentally changed to exclude the user account which is used to run the MySQL service then this might break replication.

The binary log file can be used for point in time recovery so this can also affect backup, restore and disaster recovery procedures.

3.4 Ensure 'slow_query_log' Has Appropriate Permissions (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

MySQL can operate using a variety of log files, each used for different purposes. These are the binary log, error log, slow query log, relay log, and general log. Because these are files on the host operating system, they are subject to the permissions structure provided by the host and may be accessible by users other than the MySQL user.

Rationale:

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

Audit:

Perform the following steps to assess this recommendation:

- Find the `slow_query_log` value (`<slow_query_log_path>`)by executing the following statement

```
show variables like 'slow_query_log_file';
```

- Verify permissions are `660` for `mysql:mysql` for `<slow_query_log_path>`

Remediation:

Execute the following command for each log file location requiring corrected permissions:

```
chmod 660 <log file>
chown mysql:mysql <log file>
```

Impact:

Changing the permissions of the log files might have impact on monitoring tools which use a logfile adapter. Also the slow query log can be used for performance analysis by application developers.

If the permissions on the relay logs and binary log files are accidentally changed to exclude the user account which is used to run the MySQL service then this might break replication.

The binary log file can be used for point in time recovery so this can also affect backup, restore and disaster recovery procedures.

3.5 Ensure 'relay_log_basename' Files Have Appropriate Permissions (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

MySQL can operate using a variety of log files, each used for different purposes. These are the binary log, error log, slow query log, relay log, and general log. Because these are files on the host operating system, they are subject to the permissions structure provided by the host and may be accessible by users other than the MySQL user.

Rationale:

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

Audit:

Perform the following steps to assess this recommendation:

Find the relay_log_basename value by executing the following statement

```
show variables like 'relay_log_basename';
```

- Verify permissions are `660` for `mysql:mysql` for each file of the form `<relay_log_basename>`

Remediation:

Execute the following command for each log file location requiring corrected permissions:

```
chmod 660 <log file>
chown mysql:mysql <log file>
```

Impact:

Changing the permissions of the log files might have impact on monitoring tools which use a logfile adapter. Also the slow query log can be used for performance analysis by application developers.

If the permissions on the relay logs and binary log files are accidentally changed to exclude the user account which is used to run the MySQL service then this might break replication.

The binary log file can be used for point in time recovery so this can also affect backup, restore and disaster recovery procedures.

3.6 Ensure 'general_log_file' Has Appropriate Permissions (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

MySQL can operate using a variety of log files, each used for different purposes. These are the binary log, error log, slow query log, relay log, and general log. Because these are files on the host operating system, they are subject to the permissions structure provided by the host and may be accessible by users other than the MySQL user.

Rationale:

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

Audit:

Perform the following steps to assess this recommendation:

- Find the `general_log_file` value by executing the following statement

```
show variables like 'general_log_file';
```

- Verify permissions are `660` for `mysql:mysql` for the indicated `general_log_file`.

Remediation:

Execute the following command for each log file location requiring corrected permissions:

```
chmod 660 <log file>
chown mysql:mysql <log file>
```

Impact:

Changing the permissions of the log files might have impact on monitoring tools which use a logfile adapter. Also the slow query log can be used for performance analysis by application developers.

If the permissions on the relay logs and binary log files are accidentally changed to exclude the user account which is used to run the MySQL service then this might break replication.

The binary log file can be used for point in time recovery so this can also affect backup, restore and disaster recovery procedures.

3.7 Ensure SSL Key Files Have Appropriate Permissions (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

When configured to use SSL/TLS, MySQL relies on key files, which are stored on the host's filesystem. These key files are subject to the host's permissions structure.

Rationale:

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

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

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

Audit:

To assess this recommendation, locate the SSL key in use by executing the following SQL statement to get the Value of ssl_key:

```
show variables where variable_name = 'ssl_key';
```

Then, execute the following command to assess the permissions of the Value:

```
ls -l <ssl_key Value> | egrep "^-r--------[ \t]*.[ \t]*mysql[ \t]*mysql.*$"
```

Lack of output from the above command implies a finding.

Remediation:

Execute the following commands at a terminal prompt to remediate this setting using the Value from the audit procedure:

```
chown mysql:mysql <ssl_key Value>
chmod 400 <ssl_key Value>
```

Impact:

If the permissions for the key file are changed incorrectly this can cause SSL to be disabled when MySQL is restarted or can cause MySQL not to start at all.

If other applications are using the same keypair then changing the permissions of the key file will affect this application. If this is the case then a new keypair must be generated for MySQL.

References:

1. http://dev.mysql.com/doc/refman/5.7/en/ssl-connections.html

3.8 Ensure Plugin Directory Has Appropriate Permissions (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

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

Rationale:

Limiting the accessibility of these objects will protect the confidentiality, integrity, and availability of the MySQL database. If someone can modify plugins then these plugins might be loaded when the server starts and the code will get executed.

Audit:

To assess this recommendation, execute the following SQL statement to discover the Value of plugin_dir:

```
show variables where variable_name = 'plugin_dir';
```

Then, execute the following command at a terminal prompt (using the discovered `plugin_dir Value`) to determine the permissions.

```
 ls -l <plugin_dir Value>/.. | egrep "^drwxr[-w]xr[-w]x[ \t]*[0-9][ \t]*mysql[
\t]*mysql.*plugin.*$"
```

Lack of output implies a finding.

NOTE: Permissions are intended to be either 775 or 755.

Remediation:

To remediate this setting, execute the following commands at a terminal prompt using the `plugin_dir Value` from the audit procedure.

```
 chmod 775 <plugin_dir Value> (or use 755)
 chown mysql:mysql <plugin_dir Value>
```

Impact:

Users other than the mysql user will no longer be able to update and add/remove plugins unless they're able to switch to the mysql user;

References:

1. http://dev.mysql.com/doc/refman/5.7/en/install-plugin.html

4 General

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

4.1 Ensure Latest Security Patches Are Applied (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

Periodically, updates to MySQL server are released to resolve bugs, mitigate vulnerabilities, and provide new features. It is recommended that MySQL installations are up to date with the latest security updates.

Rationale:

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

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

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:

To update the MySQL server a restart is required.

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

4.2 Ensure the 'test' Database Is Not Installed (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

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

Rationale:

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

References:

1. http://dev.mysql.com/doc/refman/5.7/en/mysql-secure-installation.html

4.3 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 set to `FALSE` in the MySQL configuration.

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.

Default Value:

`FALSE`

References:

1. http://dev.mysql.com/doc/refman/5.7/en/udf-security.html
2. http://dev.mysql.com/doc/refman/5.7/en/server-options.html#option_mysqld_allow-suspicious-udfs

4.4 Ensure 'local_infile' Is Disabled (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/5.7/en/string-functions.html#function_load-file
2. http://dev.mysql.com/doc/refman/5.7/en/load-data.html

4.5 Ensure 'mysqld' Is Not Started with '--skip-grant-tables' (Scored)

Profile Applicability:

- Level 1 - 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:

- Open the MySQL configuration (e.g. `my.cnf`) file and search for skip-grant-tables
- Ensure `skip-grant-tables` is set to `FALSE`

Remediation:

Perform the following to establish the recommended state:

- Open the MySQL configuration (e.g. `my.cnf`) file and set:

```
skip-grant-tables = FALSE
```

References:

1. http://dev.mysql.com/doc/refman/5.7/en/server-options.html#option_mysqld_skip-grant-tables

4.6 Ensure '--skip-symbolic-links' Is Enabled (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

The symbolic-links and skip-symbolic-links options for MySQL determine whether symbolic link support is available. When use of symbolic links are enabled, they have different effects depending on the host platform. When symbolic links are disabled, then symbolic links stored in files or entries in tables are not used by the database.

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

Audit:

Execute the following SQL statement to assess this recommendation:

```
SHOW variables LIKE 'have_symlink';
```

Ensure the `Value` returned is `DISABLED`.

Remediation:

Perform the following actions to remediate this setting:

- Open the MySQL configuration file (`my.cnf`)
- Locate `skip_symbolic_links` in the configuration
- Set the `skip_symbolic_links` to `YES`

NOTE: If `skip_symbolic_links` does not exist, add it to the configuration file in the `mysqld` section.

References:

1. http://dev.mysql.com/doc/refman/5.7/en/symbolic-links.html

2. http://dev.mysql.com/doc/refman/5.7/en/server-options.html#option_mysqld_symbolic-links

4.7 Ensure the 'daemon_memcached' Plugin Is Disabled (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

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

Rationale:

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. However, not all data is exposed by default.

Audit:

Execute the following SQL statement to assess this recommendation:

```
SELECT * FROM information_schema.plugins WHERE PLUGIN_NAME='daemon_memcached'
```

Ensure that no rows are returned.

Remediation:

To remediate this setting, issue the following command in the MySQL command-line client:

```
uninstall plugin daemon_memcached;
```

This uninstalls the memcached plugin from the MySQL server.

Default Value:

disabled

References:

1. http://dev.mysql.com/doc/refman/5.7/en/innodb-memcached-security.html

4.8 Ensure 'secure_file_priv' Is Not Empty (Scored)

Profile Applicability:

- Level 1 - 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 GLOBAL VARIABLES WHERE Variable_name = 'secure_file_priv' AND Value<>'';
```

Note: The Value should contain a valid path.

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/5.7/en/server-system-variables.html#sysvar_secure_file_priv

4.9 Ensure 'sql_mode' Contains 'STRICT_ALL_TABLES' (Scored)

Profile Applicability:

- Level 2 - MySQL RDBMS

Description:

When data changing statements are made (i.e. `INSERT`, `UPDATE`), MySQL can handle invalid or missing values differently depending on whether strict SQL mode is enabled. When strict SQL mode is enabled, data may not be truncated or otherwise "adjusted" to make the data changing statement work.

Rationale:

Without strict mode the server tries to do proceed with the action when an error might have been a more secure choice. For example, by default MySQL will truncate data if it does not fit in a field, which can lead to unknown behavior, or be leveraged by an attacker to circumvent data validation.

Audit:

To audit for this recommendation execute the following query:

```
SHOW VARIABLES LIKE 'sql_mode';
```

Ensure that STRICT_ALL_TABLES is in the list returned.

Remediation:

Perform the following actions to remediate this setting:

1. Add `STRICT_ALL_TABLES` to the `sql_mode` in the server's configuration file

Impact:

Applications relying on the MySQL database should be aware that `STRICT_ALL_TABLES` is in use, such that error conditions are handled appropriately.

References:

1. http://dev.mysql.com/doc/refman/5.7/en/sql-mode.html

5 MySQL Permissions

This section contains recommendations about user privileges.

5.1 Ensure Only Administrative Users Have Full Database Access (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

The `mysql.user` and `mysql.db` tables list a variety of privileges that can be granted (or denied) to MySQL users. Some of the privileges of concern include: `Select_priv`, `Insert_priv`, `Update_priv`, `Delete_priv`, `Drop_priv`, and so on. Typically, these privileges should not be available to every MySQL user and often are reserved for administrative use only.

Rationale:

Limiting the accessibility of the '`mysql`' database will protect the confidentiality, integrity, and availability of the data housed within MySQL. A user which has direct access to `mysql.*` might view password hashes, change permissions, or alter or destroy information intentionally or unintentionally.

Audit:

Execute the following SQL statement(s) to assess this recommendation:

```
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');
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'));
```

Ensure all users returned are administrative users.

Remediation:

Perform the following actions to remediate this setting:

1. Enumerate non-administrative users resulting from the audit procedure
2. For each non-administrative user, use the `REVOKE` statement to remove privileges as appropriate

Impact:

Consideration should be made for which privileges are required by each user requiring interactive database access.

5.2 Ensure 'file_priv' Is Not Set to 'Y' for Non-Administrative Users (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

The `File_priv` privilege found in the `mysql.user` table is used to allow or disallow a user from reading and writing files on the server host. Any user with the `File_priv` right granted has the ability to:

- Read files from the local file system that are readable by the MySQL server (this includes world-readable files)
- Write files to the local file system where the MySQL server has write access

Rationale:

The `File_priv` right 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. It should be noted that the MySQL server should not overwrite existing files.

Audit:

Execute the following SQL statement to audit this setting

```
select user, host from mysql.user where File_priv = 'Y';
```
Ensure only administrative users are returned in the result set.

Remediation:

Perform the following steps to remediate this setting:

1. Enumerate the non-administrative users found in the result set of the audit procedure
2. For each user, issue the following SQL statement (replace "<user>" with the non-administrative user:

```
REVOKE FILE ON *.* FROM '<user>';
```

References:

1. http://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#priv_file

5.3 Ensure 'process_priv' Is Not Set to 'Y' for Non-Administrative Users (Scored)

Profile Applicability:

- Level 2 - MySQL RDBMS

Description:

The PROCESS privilege found in the mysql.user table determines whether a given user can see statement execution information for all sessions.

Rationale:

The `PROCESS` privilege allows principals to view currently executing MySQL statements beyond their own, including statements used to manage passwords. This may be leveraged by an attacker to compromise MySQL or to gain access to potentially sensitive data.

Audit:

Execute the following SQL statement to audit this setting:

```
select user, host from mysql.user where Process_priv = 'Y';
```
Ensure only administrative users are returned in the result set.

Remediation:

Perform the following steps to remediate this setting:

1. Enumerate the non-administrative users found in the result set of the audit procedure
2. For each user, issue the following SQL statement (replace "`<user>`" with the non-administrative user:

```
REVOKE PROCESS ON *.* FROM '<user>';
```

Impact:

Users denied the `PROCESS` privilege may also be denied use of `SHOW ENGINE`.

References:

1. http://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#priv_process

5.4 Ensure 'super_priv' Is Not Set to 'Y' for Non-Administrative Users (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

The `SUPER` privilege found in the `mysql.user` table governs the use of a variety of MySQL features. These features include, `CHANGE MASTER TO`, `KILL`, `mysqladmin kill` option, `PURGE BINARY LOGS`, `SET GLOBAL`, `mysqladmin debug` option, logging control, and more.

Rationale:

The `SUPER` privilege allows principals to perform many actions, including view and terminate currently executing MySQL statements (including statements used to manage passwords). This privilege also provides the ability to configure MySQL, such as enable/disable logging, alter data, disable/enable features. Limiting the accounts that have the `SUPER` privilege reduces the chances that an attacker can exploit these capabilities.

Audit:

Execute the following SQL statement to audit this setting:

```
select user, host from mysql.user where Super_priv = 'Y';
```

Ensure only administrative users are returned in the result set.

Remediation:

Perform the following steps to remediate this setting:

1. Enumerate the non-administrative users found in the result set of the audit procedure
2. For each user, issue the following SQL statement (replace "`<user>`" with the non-administrative user:

```
REVOKE SUPER ON *.* FROM '<user>';
```

Impact:

When the `SUPER` privilege is denied to a given user, that user will be unable to take advantage of certain capabilities, such as certain `mysqladmin` options.

References:

1. http://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#priv_super

5.5 Ensure 'shutdown_priv' Is Not Set to 'Y' for Non-Administrative Users (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

The `SHUTDOWN` privilege simply enables use of the `shutdown` option to the `mysqladmin` command, which allows a user with the `SHUTDOWN` privilege the ability to shut down the MySQL server.

Rationale:

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

Audit:

Execute the following SQL statement to audit this setting:

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

Ensure only administrative users are returned in the result set.

Remediation:

Perform the following steps to remediate this setting:

1. Enumerate the non-administrative users found in the result set of the audit procedure
2. For each user, issue the following SQL statement (replace "`<user>`" with the non-administrative user):

```
REVOKE SHUTDOWN ON *.* FROM '<user>';
```

References:

1. http://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#priv_shutdown

5.6 Ensure 'create_user_priv' Is Not Set to 'Y' for Non-Administrative Users (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

The CREATE USER privilege governs the right of a given user to add or remove users, change existing users' names, or revoke existing users' privileges.

Rationale:

Reducing the number of users granted the `CREATE USER` right minimizes the number of users able to add/drop users, alter existing users' names, and manipulate existing users' privileges.

Audit:

Execute the following SQL statement to audit this setting:

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

Ensure only administrative users are returned in the result set.

Remediation:

Perform the following steps to remediate this setting:

1. Enumerate the non-administrative users found in the result set of the audit procedure
2. For each user, issue the following SQL statement (replace "<user>" with the non-administrative user):

```
REVOKE CREATE USER ON *.* FROM '<user>';
```

Impact:

Users that are denied the CREATE USER privilege will not only be unable to create a user, but they may be unable to drop a user, rename a user, or otherwise revoke a given user's privileges.

5.7 Ensure 'grant_priv' Is Not Set to 'Y' for Non-Administrative Users (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

The `GRANT OPTION` privilege exists in different contexts (`mysql.user`, `mysql.db`) for the purpose of governing the ability of a privileged user to manipulate the privileges of other users.

Rationale:

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

Audit:

Execute the following SQL statements to audit this setting:

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

Ensure only administrative users are returned in the result set.

Remediation:

Perform the following steps to remediate this setting:

1. Enumerate the non-administrative users found in the result sets of the audit procedure
2. For each user, issue the following SQL statement (replace "`<user>`" with the non-administrative user:

```
REVOKE GRANT OPTION ON *.* FROM <user>;
```

References:

1. http://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#priv_grant-option

5.8 Ensure 'repl_slave_priv' Is Not Set to 'Y' for Non-Slave Users (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

The `REPLICATION SLAVE` privilege governs whether a given user (in the context of the master server) can request updates that have been made on the master server.

Rationale:

The `REPLICATION SLAVE` privilege allows a principal to fetch binlog files containing all data changing statements and/or changes in table data from the master. This may be used by an attacker to read/fetch sensitive data from MySQL.

Audit:

Execute the following SQL statement to audit this setting:

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

Ensure only accounts designated for slave users are granted this privilege.

Remediation:

Perform the following steps to remediate this setting:

1. Enumerate the non-slave users found in the result set of the audit procedure

2. For each user, issue the following SQL statement (replace "<user>" with the non-slave user):

```
REVOKE REPLICATION SLAVE ON *.* FROM <user>;
```

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

References:

1. http://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#priv_replication-slave

5.9 Ensure DML/DDL Grants Are Limited to Specific Databases and Users (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

DML/DDL includes the set of privileges used to modify or create data structures. This includes `INSERT`, `SELECT`, `UPDATE`, `DELETE`, `DROP`, `CREATE`, and `ALTER` privileges.

Rationale:

`INSERT`, `SELECT`, `UPDATE`, `DELETE`, `DROP`, `CREATE`, and `ALTER` are powerful privileges in any database. Such privileges should be limited only to those users requiring such rights. By limiting the users with these rights and ensuring that they are limited to specific databases, the attack surface of the database is reduced.

Audit:

Execute the following SQL statement to audit this setting:

```
SELECT User,Host,Db
FROM mysql.db
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';
```

Ensure all users returned should have these privileges on the indicated databases.

NOTE: Global grants are covered in Recommendation 4.1.

Remediation:

Perform the following steps to remediate this setting:

1. Enumerate the unauthorized users, hosts, and databases returned in the result set of the audit procedure
2. For each user, issue the following SQL statement (replace "`<user>`" with the unauthorized user, "`<host>`" with host name, and "`<database>`" with the database name):

```
REVOKE SELECT ON <host>.<database> FROM <user>;
REVOKE INSERT ON <host>.<database> FROM <user>;
REVOKE UPDATE ON <host>.<database> FROM <user>;
REVOKE DELETE ON <host>.<database> FROM <user>;
REVOKE CREATE ON <host>.<database> FROM <user>;
REVOKE DROP ON <host>.<database> FROM <user>;
REVOKE ALTER ON <host>.<database> FROM <user>;
```

6 Auditing and Logging

This section provides guidance with respect to MySQL's logging behavior.

6.1 Ensure 'log_error' Is Not Empty (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

The error log contains information about events such as `mysqld` starting and stopping, when a table needs to be checked or repaired, and, depending on the host operating system, stack traces when `mysqld` fails.

Rationale:

Enabling error logging may increase the ability to detect malicious attempts against MySQL, and other critical messages, such as if the error log is not enabled then connection error might go unnoticed.

Audit:

Execute the following SQL statement to audit this setting:

```
SHOW variables LIKE 'log_error';
```

Ensure the `Value` returned is not empty.

Remediation:

Perform the following actions to remediate this setting:

1. Open the MySQL configuration file (`my.cnf` or `my.ini`)
2. Set the `log-error` option to the path for the error log

References:

1. http://dev.mysql.com/doc/refman/5.7/en/error-log.html

6.2 Ensure Log Files Are Stored on a Non-System Partition (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

MySQL log files can be set in the MySQL configuration to exist anywhere on the filesystem. It is common practice to ensure that the system filesystem is left uncluttered by application logs. System filesystems include the root, `/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:

Execute the following SQL statement to assess this recommendation:

```
SELECT @@global.log_bin_basename;
```

Ensure the value returned does not indicate root (`'/'`), `/var`, or `/usr`.

Remediation:

Perform the following actions to remediate this setting:

1. Open the MySQL configuration file (`my.cnf`)
2. Locate the `log-bin` entry and set it to a file not on root (`'/'`), `/var`, or `/usr`

References:

1. http://dev.mysql.com/doc/refman/5.7/en/binary-log.html
2. http://dev.mysql.com/doc/refman/5.7/en/replication-options-binary-log.html

6.3 Ensure 'log_error_verbosity' Is Not Set to '1' (Scored)

Profile Applicability:

- Level 2 - MySQL RDBMS

Description:

The log_error_verbosity system variable provides additional information to the MySQL log. A value of 1 enables logging of error messages. A value of 2 enables logging of error and warning messages, and a value of 3 enables logging of error, warning and note messages.

Rationale:

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

Audit:

Execute the following SQL statement to assess this recommendation:

```
SHOW GLOBAL VARIABLES LIKE 'log_error_verbosity';
```

Ensure the `Value` returned equals `2` or `3`.

Remediation:

Perform the following actions to remediate this setting:

- Open the MySQL configuration file (`my.cnf`)
- Ensure the following line is found in the `mysqld` section

```
log_error_verbosity = 2
```

The value can be 2 or 3.

Default Value:

The option is error+warning (2) by default.

References:

1. https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html#sysvar_log_error_verbosity

6.4 Ensure Audit Logging Is Enabled (Not Scored)

Profile Applicability:

- Level 2 - MySQL RDBMS

Description:

Audit logging is not really included in the Community Edition of MySQL - only the general log. Using the general log is possible, but not practical, because it grows quickly and has an adverse impact on server performance.

Nevertheless, enabling audit logging is an important consideration for a production environment, and third-party tools do exist to help with this. Enable audit logging for

- Interactive user sessions
- Application sessions (optional)

Rationale:

Audit logging helps to identify who changed what and when. The audit log might be used as evidence in investigations. It might also help to identify what an attacker was able to accomplish.

Audit:

Verify that a third-party tool is installed and configured to enable logging for interactive user sessions and (optionally) applications sessions.

Remediation:

Acquire a third-party MySQL logging solution as available from a variety of sources including, but not necessarily limited to, the following:

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

References:

1. http://dev.mysql.com/doc/refman/5.7/en/query-log.html
2. http://dev.mysql.com/doc/refman/5.7/en/mysql-enterprise-audit.html
3. https://mariadb.com/kb/en/server_audit-mariadb-audit-plugin/
4. https://github.com/mcafee/mysql-audit

6.5 Ensure 'log-raw' Is Set to 'OFF' (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

The log-raw MySQL option determines whether passwords are rewritten by the server so as not to appear in log files as plain text. If log-raw is enabled, then passwords are written to the various log files (general query log, slow query log, and binary log) in plain text.

Rationale:

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

Audit:

Perform the following actions to assess this recommendation:

- Open the MySQL configuration file (`my.cnf`)
- Ensure the `log-raw` entry is present
- Ensure the `log-raw` entry is set to `OFF`

Remediation:

Perform the following actions to remediate this setting:

- Open the MySQL configuration file (`my.cnf`)
- Find the `log-raw` entry and set it as follows

```
log-raw = OFF
```

Default Value:

OFF

References:

1. http://dev.mysql.com/doc/refman/5.7/en/password-logging.html
2. http://dev.mysql.com/doc/refman/5.7/en/server-options.html#option_mysqld_log-raw

7 Authentication

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

7.1 Ensure Passwords Are Not Stored 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 `user` and `password` to be used. Verify the `password` option is not used in the global configuration file (`my.cnf`).

Rationale:

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

Audit:

To assess this recommendation, perform the following steps:

- Open the MySQL configuration file (e.g. `my.cnf`)
- Examine the `[client]` section of the MySQL configuration file and ensure `password` is not employed.

Remediation:

Use the `mysql_config_editor` to store authentication 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/5.7/en/mysql-config-editor.html

7.2 Ensure 'sql_mode' Contains 'NO_AUTO_CREATE_USER' (Scored)

Profile Applicability:

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

Description:

`NO_AUTO_CREATE_USER` is an option for `sql_mode` that prevents a `GRANT` statement from automatically creating a user when authentication information is not provided.

Rationale:

Blank passwords negate the benefits provided by authentication mechanisms. Without this setting an administrative user might accidentally create a user without a password.

Audit:

Execute the following SQL statements to assess this recommendation:

```
SELECT @@global.sql_mode;
SELECT @@session.sql_mode;
```

Ensure that each result contains `NO_AUTO_CREATE_USER`.

Remediation:

Perform the following actions to remediate this setting:

1. Open the MySQL configuration file (`my.cnf`)
2. Find the `sql_mode` setting in the `[mysqld]` area
3. Add the `NO_AUTO_CREATE_USER` to the `sql_mode` setting

7.3 Ensure Passwords Are Set for All MySQL Accounts (Scored)

Profile Applicability:

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

Description:

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

Rationale:

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. This, in effect, bypasses authentication mechanisms.

Audit:

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

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

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

Remediation:

For each row returned from the audit procedure, set a password for the given user using the following statement (as an example):

```
 SET PASSWORD FOR <user>@'<host>' = '<clear password>'
```

NOTE: Replace `<user>`, `<host>`, and `<clear password>` with appropriate values.

References:

1. https://dev.mysql.com/doc/refman/5.7/en/assigning-passwords.html

7.4 Ensure 'default_password_lifetime' Is Less Than Or Equal To '90' (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

Password expiry provides passwords with a time bounded lifetime.

Rationale:

This benchmark prevents a password being set for an indefinite period, therefore reducing the time available a compromised password is known to an attacker.

Audit:

Level 1 MySQL RDBMS:

```
SHOW VARIABLES LIKE 'default_password_lifetime';
```

`default_password_lifetime` should be less than or equal to 90.

Remediation:

Level 1 MySQL RDBMS: global policy

```
SET GLOBAL default_password_lifetime=90
```

and in the configuration file: `default_password_lifetime=90`

As part of Installation and Planning consider set an expiry policy for specific users. Doing this will take precedence over the setting specified in default_password_lifetime. For example:

```
ALTER USER 'jeffrey'@'localhost' PASSWORD EXPIRE INTERVAL 90 DAY;
```

Impact:

Scripted clients or users dependent on automated login in a controlled environment will need to consider their authentication procedures. The server will accept the user but the user is placed in restricted mode.

In restricted mode, operations performed within the session result in an error until the user establishes a new account password.

Default Value:

global policy:
`default_password_lifetime=360`
per user policy:
DEFAULT - as per default_password_lifetime value.

References:

1. https://dev.mysql.com/doc/refman/5.7/en/password-expiration-policy.html
2. https://dev.mysql.com/doc/refman/5.7/en/password-expiration-sandbox-mode.html

7.5 Ensure Password Complexity Is in Place (Scored)

Profile Applicability:

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

Description:

Password complexity includes password characteristics such as length, case, length, and character sets.

Rationale:

Complex passwords help mitigate dictionary, brute forcing, and other password attacks. This recommendation prevents users from choosing weak passwords which can easily be guessed.

Audit:

Execute the following SQL statements to assess this recommendation:

```
SHOW VARIABLES LIKE 'validate_password%';
```

The result set from the above statement should show:

- `validate_password_length` should be 14 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`

The following lines should be present 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:

```
SELECT user,authentication_string,host FROM mysql.user
WHERE authentication_string=CONCAT('*', UPPER(SHA1(UNHEX(SHA1(user)))));
```

NOTE: This method is only capable of checking the post-4.1 password format which is also known as mysql_native_password.

Remediation:

Add to the global configuration:

```
plugin-load=validate_password.so
validate-password=FORCE_PLUS_PERMANENT
validate_password_length=14
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:

Remediation for this recommendation requires a server restart.

References:

1. http://dev.mysql.com/doc/refman/5.7/en/validate-password-plugin.html

7.6 Ensure No Users Have Wildcard Hostnames (Scored)

Profile Applicability:

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

Description:

MySQL can make use of host wildcards when granting permissions to users on specific databases. For example, you may grant a given privilege to `'<user>'@'%'`.

Rationale:

Avoiding the use of wildcards within hostnames helps control the specific locations from which a given user may connect to and interact with the database.

Audit:

Execute the following SQL statement to assess this recommendation:

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

Ensure no rows are returned.

Remediation:

Perform the following actions to remediate this setting:

1. Enumerate all users returned after running the audit procedure
2. Either ALTER the user's host to be specific or DROP the user

7.7 Ensure No Anonymous Accounts Exist (Scored)

Profile Applicability:

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

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:

Perform the following actions to remediate this setting:

1. Enumerate the anonymous users returned from executing the audit procedure
2. For each anonymous user, `DROP` or assign them a name

NOTE: As an alternative, you may execute the `mysql_secure_installation` utility.

Impact:

Any applications relying on anonymous database access will be adversely affected by this change.

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/5.7/en/mysql-secure-installation.html
2. https://dev.mysql.com/doc/refman/5.6/en/default-privileges.html

8 Network

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

8.1 Ensure 'have_ssl' Is Set to 'YES' (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

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

Rationale:

The SSL/TLS-protected MySQL protocol helps to prevent eavesdropping and man-in-the-middle attacks.

Audit:

Execute the following SQL statements to assess this recommendation:

```
SHOW variables WHERE variable_name = 'have_ssl';
```

Ensure the `Value` returned is `YES`.

NOTE: `have_openssl` is an alias for `have_ssl` as of MySQL 5.0.38. MySQL can be build with OpenSSL or YaSSL.

Remediation:

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

Impact:

Enabling SSL will allow clients to encrypt network traffic and verify the identity of the server. This could have impact on network traffic inspection.

Default Value:

DISABLED

References:

1. http://dev.mysql.com/doc/refman/5.7/en/ssl-connections.html
2. http://dev.mysql.com/doc/refman/5.7/en/ssl-options.html

8.2 Ensure 'ssl_type' Is Set to 'ANY', 'X509', or 'SPECIFIED' for All Remote Users (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

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

SSL/TLS should be enforced on a per-user basis for users which enter the system through the network.

Rationale:

The SSL/TLS-protected MySQL protocol helps to prevent eavesdropping and man-in-the-middle attacks.

Audit:

Execute the following SQL statements to assess this recommendation:

```
SELECT user, host, ssl_type FROM mysql.user
WHERE NOT HOST IN ('::1', '127.0.0.1', 'localhost');
```

Ensure the `ssl_type` for each user returned is equal to `ANY`, `X509`, or `SPECIFIED`.

NOTE: `have_openssl` is an alias for `have_ssl` as of MySQL 5.0.38. MySQL can be build with OpenSSL or YaSSL.

Remediation:

Use the GRANT statement to require the use of SSL:

```
GRANT USAGE ON *.* TO 'my_user'@'app1.example.com' REQUIRE SSL;
```

Note that REQUIRE SSL only enforces SSL. There are options like REQUIRE X509, REQUIRE ISSUER, REQUIRE SUBJECT which can be used to further restrict connection options.

Impact:

When SSL/TLS is enforced then clients which do not use SSL will not be able to connect. If the server is not configured for SSL/TLS then accounts for which SSL/TLS is mandatory will not be able to connect

Default Value:

Not enforced (`ssl_type` is empty)

References:

1. http://dev.mysql.com/doc/refman/5.7/en/ssl-connections.html
2. http://dev.mysql.com/doc/refman/5.7/en/grant.html

9 Replication

Everything related to replicating data from one server to another.

9.1 Ensure Replication Traffic Is Secured (Not Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

The replication traffic between servers should be secured.

Rationale:

The replication traffic should be secured as it gives access to all transferred 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.

9.2 Ensure 'MASTER_SSL_VERIFY_SERVER_CERT' Is Set to 'YES' or '1' (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

In the MySQL slave context the setting `MASTER_SSL_VERIFY_SERVER_CERT` indicates whether the slave should verify the master's certificate. This configuration item may be set to `Yes` or `No`, and unless SSL has been enabled on the slave, the value will be ignored.

Rationale:

When SSL is in use certificate verification is important to authenticate the party to which a connection is being made. In this case, the slave (client) should verify the master's (server's) certificate to authenticate the master prior to continuing the connection.

Audit:

To assess this recommendation, issue the following statement:

```
select ssl_verify_server_cert from mysql.slave_master_info;
```

Verify the value of `ssl_verify_server_cert` is `1`.

Remediation:

To remediate this setting you must use the `CHANGE MASTER TO` command.

```
STOP SLAVE; -- required if replication was already running
CHANGE MASTER TO MASTER_SSL_VERIFY_SERVER_CERT=1;
START SLAVE; -- required if you want to restart replication
```

Impact:

When using `CHANGE MASTER TO`, be aware of the following:

- Slave processes need to be stopped prior to executing `CHANGE MASTER TO`
- Use of `CHANGE MASTER TO` starts new relay logs without keeping the old ones unless explicitly told to keep them
- When `CHANGE MASTER TO` is invoked, some information is dumped to the error log (previous values for `MASTER_HOST, MASTER_PORT, MASTER_LOG_FILE,` and `MASTER_LOG_POS`)
- Invoking `CHANGE MASTER TO` will implicitly commit any ongoing transactions

References:

1. https://dev.mysql.com/doc/refman/5.6/en/change-master-to.html

9.3 Ensure 'master_info_repository' Is Set to 'TABLE' (Scored)

Profile Applicability:

- Level 2 - MySQL RDBMS

Description:

The `master_info_repository` setting determines to where a slave logs master status and connection information. The options are `FILE` or `TABLE`. Note also that this setting is associated with the `sync_master_info` setting as well.

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:

Execute the following SQL statement to assess this recommendation:

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

The result should be `TABLE` instead of `FILE`.

NOTE: There also should not be a master.info file in the datadir.

Remediation:

Perform the following actions to remediate this setting:

1. Open the MySQL configuration file (`my.cnf`)
2. Locate `master_info_repository`
3. Set the `master_info_repository` value to `TABLE`

NOTE: If `master_info_repository` does not exist, add it to the configuration file.

Default Value:

FILE

References:

1. http://dev.mysql.com/doc/refman/5.7/en/replication-options-slave.html#sysvar_master_info_repository

9.4 Ensure 'super_priv' Is Not Set to 'Y' for Replication Users (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

The `SUPER` privilege found in the `mysql.user` table governs the use of a variety of MySQL features. These features include, `CHANGE MASTER TO`, `KILL`, `mysqladmin kill` option, `PURGE BINARY LOGS`, `SET GLOBAL`, `mysqladmin debug` option, logging control, and more.

Rationale:

The `SUPER` privilege allows principals to perform many actions, including view and terminate currently executing MySQL statements (including statements used to manage passwords). This privilege also provides the ability to configure MySQL, such as enable/disable logging, alter data, disable/enable features. Limiting the accounts that have the `SUPER` privilege reduces the chances that an attacker can exploit these capabilities.

Audit:

Execute the following SQL statement to audit this setting:

```
select user, host from mysql.user where user='repl' and Super_priv = 'Y';
```

No rows should be returned.

NOTE: Substitute your replication user's name for `repl` in the above query.

Remediation:

Execute the following steps to remediate this setting:

1. Enumerate the replication users found in the result set of the audit procedure
2. For each replication user, issue the following SQL statement (replace "`repl`" with your replication user's name):

```
REVOKE SUPER ON *.* FROM 'repl';
```

Impact:

When the `SUPER` privilege is denied to a given user, that user will be unable to take advantage of certain capabilities, such as certain `mysqladmin` options.

References:

1. http://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#priv_super

9.5 Ensure No Replication Users Have Wildcard Hostnames (Scored)

Profile Applicability:

- Level 1 - MySQL RDBMS

Description:

MySQL can make use of host wildcards when granting permissions to users on specific databases. For example, you may grant a given privilege to `'<user>'@'%'`.

Rationale:

Avoiding the use of wildcards within hostnames helps control the specific locations from which a given user may connect to and interact with the database.

Audit:

Execute the following SQL statement to assess this recommendation:

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

Ensure no rows are returned.

Remediation:

Perform the following actions to remediate this setting:

1. Enumerate all users returned after running the audit procedure
2. Either ALTER the user's host to be specific or DROP the user

Control		Set Correctly	
		Yes	No
1	**Operating System Level Configuration**		
1.1	Place Databases on Non-System Partitions (Scored)	☐	☐
1.2	Use Dedicated Least Privileged Account for MySQL Daemon/Service (Scored)	☐	☐
1.3	Disable MySQL Command History (Scored)	☐	☐
1.4	Verify That the MYSQL_PWD Environment Variables Is Not In Use (Scored)	☐	☐
1.5	Disable Interactive Login (Scored)	☐	☐
1.6	Verify That 'MYSQL_PWD' Is Not Set In Users' Profiles (Scored)	☐	☐
2	**Installation and Planning**		
2.1	**Backup and Disaster Recovery**		
2.1.1	Backup policy in place (Not Scored)	☐	☐
2.1.2	Verify backups are good (Not Scored)	☐	☐
2.1.3	Secure backup credentials (Not Scored)	☐	☐
2.1.4	The backups should be properly secured (Not Scored)	☐	☐
2.1.5	Point in time recovery (Not Scored)	☐	☐
2.1.6	Disaster recovery plan (Not Scored)	☐	☐
2.1.7	Backup of configuration and related files (Not Scored)	☐	☐
2.2	Dedicate Machine Running MySQL (Not Scored)	☐	☐
2.3	Do Not Specify Passwords in Command Line (Not Scored)	☐	☐
2.4	Do Not Reuse Usernames (Not Scored)	☐	☐
2.5	Do Not Use Default or Non-MySQL-specific Cryptographic Keys (Not Scored)	☐	☐
2.6	Set a Password Expiry Policy for Specific Users (Not Scored)	☐	☐
3	**File System Permissions**		
3.1	Ensure 'datadir' Has Appropriate Permissions (Scored)	☐	☐
3.2	Ensure 'log_bin_basename' Files Have Appropriate Permissions (Scored)	☐	☐
3.3	Ensure 'log_error' Has Appropriate Permissions (Scored)	☐	☐
3.4	Ensure 'slow_query_log' Has Appropriate Permissions (Scored)	☐	☐
3.5	Ensure 'relay_log_basename' Files Have Appropriate Permissions (Scored)	☐	☐
3.6	Ensure 'general_log_file' Has Appropriate Permissions (Scored)	☐	☐
3.7	Ensure SSL Key Files Have Appropriate Permissions (Scored)	☐	☐
3.8	Ensure Plugin Directory Has Appropriate Permissions (Scored)	☐	☐
4	**General**		
4.1	Ensure Latest Security Patches Are Applied (Not Scored)	☐	☐
4.2	Ensure the 'test' Database Is Not Installed (Scored)	☐	☐
4.3	Ensure 'allow-suspicious-udfs' Is Set to 'FALSE' (Scored)	☐	☐
4.4	Ensure 'local_infile' Is Disabled (Scored)	☐	☐

4.5	Ensure 'mysqld' Is Not Started with '--skip-grant-tables' (Scored)	☐	☐
4.6	Ensure '--skip-symbolic-links' Is Enabled (Scored)	☐	☐
4.7	Ensure the 'daemon_memcached' Plugin Is Disabled (Scored)	☐	☐
4.8	Ensure 'secure_file_priv' Is Not Empty (Scored)	☐	☐
4.9	Ensure 'sql_mode' Contains 'STRICT_ALL_TABLES' (Scored)	☐	☐
5	**MySQL Permissions**		
5.1	Ensure Only Administrative Users Have Full Database Access (Scored)	☐	☐
5.2	Ensure 'file_priv' Is Not Set to 'Y' for Non-Administrative Users (Scored)	☐	☐
5.3	Ensure 'process_priv' Is Not Set to 'Y' for Non-Administrative Users (Scored)	☐	☐
5.4	Ensure 'super_priv' Is Not Set to 'Y' for Non-Administrative Users (Scored)	☐	☐
5.5	Ensure 'shutdown_priv' Is Not Set to 'Y' for Non-Administrative Users (Scored)	☐	☐
5.6	Ensure 'create_user_priv' Is Not Set to 'Y' for Non-Administrative Users (Scored)	☐	☐
5.7	Ensure 'grant_priv' Is Not Set to 'Y' for Non-Administrative Users (Scored)	☐	☐
5.8	Ensure 'repl_slave_priv' Is Not Set to 'Y' for Non-Slave Users (Scored)	☐	☐
5.9	Ensure DML/DDL Grants Are Limited to Specific Databases and Users (Scored)	☐	☐
6	**Auditing and Logging**		
6.1	Ensure 'log_error' Is Not Empty (Scored)	☐	☐
6.2	Ensure Log Files Are Stored on a Non-System Partition (Scored)	☐	☐
6.3	Ensure 'log_error_verbosity' Is Not Set to '1' (Scored)	☐	☐
6.4	Ensure Audit Logging Is Enabled (Not Scored)	☐	☐
6.5	Ensure 'log-raw' Is Set to 'OFF' (Scored)	☐	☐
7	**Authentication**		
7.1	Ensure Passwords Are Not Stored in the Global Configuration (Scored)	☐	☐
7.2	Ensure 'sql_mode' Contains 'NO_AUTO_CREATE_USER' (Scored)	☐	☐
7.3	Ensure Passwords Are Set for All MySQL Accounts (Scored)	☐	☐
7.4	Ensure 'default_password_lifetime' Is Less Than Or Equal To '90' (Scored)	☐	☐
7.5	Ensure Password Complexity Is in Place (Scored)	☐	☐
7.6	Ensure No Users Have Wildcard Hostnames (Scored)	☐	☐
7.7	Ensure No Anonymous Accounts Exist (Scored)	☐	☐
8	**Network**		
8.1	Ensure 'have_ssl' Is Set to 'YES' (Scored)	☐	☐
8.2	Ensure 'ssl_type' Is Set to 'ANY', 'X509', or 'SPECIFIED' for All	☐	☐

	Remote Users (Scored)		
9	**Replication**		
9.1	Ensure Replication Traffic Is Secured (Not Scored)	☐	☐
9.2	Ensure 'MASTER_SSL_VERIFY_SERVER_CERT' Is Set to 'YES' or '1' (Scored)	☐	☐
9.3	Ensure 'master_info_repository' Is Set to 'TABLE' (Scored)	☐	☐
9.4	Ensure 'super_priv' Is Not Set to 'Y' for Replication Users (Scored)	☐	☐
9.5	Ensure No Replication Users Have Wildcard Hostnames (Scored)	☐	☐

Appendix: Change History

Date	Version	Changes for this version
12-29-2015	1.0.0	Initial public release.