CIS Center for Internet Security®

CIS Benchmarks™

CIS Oracle MySQL Community Server 5.7 Benchmark

v2.0.0 - 04-20-2022

Terms of Use

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

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

Table of Contents

Overview

All CIS Benchmarks focus on technical configuration settings used to maintain and/or increase the security of the addressed technology, and they should be used in **conjunction** with other essential cyber hygiene tasks like:

- Monitoring the base operating system for vulnerabilities and quickly updating with the latest security patches
- Monitoring applications and libraries for vulnerabilities and quickly updating with the latest security patches

In the end, the CIS Benchmarks are designed as a key **component** of a comprehensive cybersecurity program.

This is the final release of the CIS Benchmark for Oracle MySQL Community Server 5.7. CIS encourages you to migrate to a more recent, supported version of this technology.

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, 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 CIS Benchmark was created using a consensus review process comprised of a global community of subject matter experts. The process combines real world experience with data-based information to create technology specific guidance to assist users to secure their environments. 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://workbench.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

Recommendation Definitions

The following defines the various components included in a CIS recommendation as applicable. If any of the components are not applicable it will be noted or the component will not be included in the recommendation.

Title

Concise description for the recommendation's intended configuration.

Assessment Status

An assessment status is included for every recommendation. The assessment status indicates whether the given recommendation can be automated or requires manual steps to implement. Both statuses are equally important and are determined and supported as defined below:

Automated

Represents recommendations for which assessment of a technical control can be fully automated and validated to a pass/fail state. Recommendations will include the necessary information to implement automation.

Manual

Represents recommendations for which assessment of a technical control cannot be fully automated and requires all or some manual steps to validate that the configured state is set as expected. The expected state can vary depending on the environment.

Profile

A collection of recommendations for securing a technology or a supporting platform. Most benchmarks include at least a Level 1 and Level 2 Profile. Level 2 extends Level 1 recommendations and is not a standalone profile. The Profile Definitions section in the benchmark provides the definitions as they pertain to the recommendations included for the technology.

Description

Detailed information pertaining to the setting with which the recommendation is concerned. In some cases, the description will include the recommended value.

Rationale Statement

Detailed reasoning for the recommendation to provide the user a clear and concise understanding on the importance of the recommendation.

Impact Statement

Any security, functionality, or operational consequences that can result from following the recommendation.

Audit Procedure

Systematic instructions for determining if the target system complies with the recommendation

Remediation Procedure

Systematic instructions for applying recommendations to the target system to bring it into compliance according to the recommendation.

Default Value

Default value for the given setting in this recommendation, if known. If not known, either not configured or not defined will be applied.

References

Additional documentation relative to the recommendation.

CIS Critical Security Controls® (CIS Controls®)

The mapping between a recommendation and the CIS Controls is organized by CIS Controls version, Safeguard, and Implementation Group (IG). The Benchmark in its entirety addresses the CIS Controls safeguards of (v7) “5.1 - Establish Secure Configurations” and (v8) '4.1 - Establish and Maintain a Secure Configuration Process” so individual recommendations will not be mapped to these safeguards.

Additional Information

Supplementary information that does not correspond to any other field but may be useful to the user.

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
Matthew Woods
David Van der Ploeg

Editor
Tim Harrison, Center for Internet Security

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 (Manual)

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 is typically called system partitions, and these are generally reserved for host system/application operation. The other set of filesystems is 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 caused by exhaustion of available disk space to the operating system.

Impact:

Moving database files and directories 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 non-system partitions available.

Audit:

Execute the following steps to assess this recommendation:

- Obtain the location of the `datadir` and other MySQL database files by executing the following SQL statement:

```
SELECT VARIABLE_NAME, VARIABLE_VALUE
FROM performance_schema.global_variables
WHERE (VARIABLE_NAME LIKE '%dir' or VARIABLE_NAME LIKE '%file') and
(VARIABLE_NAME NOT LIKE '%core%'
   AND VARIABLE_NAME <> 'local_infile' AND VARIABLE_NAME <>
'relay_log_info_file') order by
   VARIABLE_NAME;
```

- Using the value returned for the `datadir`, and other results from the above query, execute the following in a system terminal:

```
df -h <directory>
```

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

Remediation:

Perform the following steps to remediate this setting for the `datadir`:

1. Backup the database.
2. Choose a non-system partition `new location` for MySQL data.
3. Stop `mysqld` using a command like: `service mysql stop`.
4. Copy the data using a command like: `cp -rp <datadir Value> <new location>`.
5. Set the `datadir` location to the `new location` in the MySQL configuration file.
6. 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,
```

Default Value:

Not Applicable.

References:

1. https://dev.mysql.com/doc/mysql-secure-deployment-guide/5.7/en/secure-deployment-permissions.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.12 Segment Data Processing and Storage Based on Sensitivity Segment data processing and storage based on the sensitivity of the data. Do not process sensitive data on enterprise assets intended for lower sensitivity data.		●	●
v7	2.10 Physically or Logically Segregate High Risk Applications Physically or logically segregated systems should be used to isolate and run software that is required for business operations but incur higher risk for the organization.			●

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

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 needed 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 fail.
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. Additionally, it's best to avoid providing shell access to such an account.
Shell access can be removed using the following command at a terminal prompt:

```
/usr/sbin/groupadd -g 27 -o -r mysql >/dev/null 2>&1 || :
/usr/sbin/useradd -M -N -g mysql -o -r -d /var/lib/mysql -s /bin/false \
    -c "MySQL Server" -u 27 mysql >/dev/null 2>&1 || :
```

References:

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

Additional Information:

The root user may be used to start the MySQL service on Linux/UNIX, but then it must be configured to drop privileges by specifying a service specific user in the `my.cnf` or `my.ini` file.

1.3 Disable MySQL Command History (Automated)

Profile Applicability:

- Level 2 - MySQL RDBMS on Linux

Description:

On Linux/UNIX, the MySQL client and MySQL Shell log statements executed interactively to a history file. The default MySQL Client file is named `.mysql_history` in the user's home directory. The files are split by language and named `history.sql`, `history.js` and `history.py`. Most interactive commands run in the MySQL client application are saved to a history file. The MySQL command history should be disabled. By default, the MySQL Shell does not save history between sessions.

Rationale:

Disabling the MySQL Client and MySQL Shell command history reduces the probability of exposing sensitive information, such as passwords, encryption keys, or other sensitive data or information.

Audit:

Execute the following commands to assess this recommendation:

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

For MySQL Shell

```
ls -d .??*/* | egrep history | grep mysql
```

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:
   - Set the `MYSQL_HISTFILE` environment variable to `/dev/null`. This will need to be placed in the shell's startup script.
   - Create `$HOME/.mysql_history` as a symbolic to `/dev/null`.

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

Another way to prevent history from being recorded is to use `--batch` option.

Default Value:

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

References:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.5 Securely Dispose of Data Securely dispose of data as outlined in the enterprise's data management process. Ensure the disposal process and method are commensurate with the data sensitivity.	●	●	●
v7	13.2 Remove Sensitive Data or Systems Not Regularly Accessed by Organization Remove sensitive data or systems not regularly accessed by the organization from the network. These systems shall only be used as stand alone systems (disconnected from the network) by the business unit needing to occasionally use the system or completely virtualized and powered off until needed.	●	●	●

1.4 Verify That the MYSQL_PWD Environment Variable Is Not In Use (Automated)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

MySQL can read a default database password from an environment variable called `MYSQL_PWD`. Avoiding use of this environment variable can better safeguard the confidentiality of MySQL credentials.

Rationale:

Using the `MYSQL_PWD` environment variable implies MySQL credentials are stored as clear text.

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.
For unattended logins you should consider:

1. MySQL Configuration Editor
2. Different authentication methods (e.g., X509 certificate verification)
3. Use MySQL Enterprise LDAP plugin with Kerberos or SASL tokens.

Default Value:

Not set.

References:

1. https://dev.mysql.com/doc/refman/5.7/en/environment-variables.html
2. https://dev.mysql.com/doc/refman/5.7/en/mysql-config-editor.html
3. https://dev.mysql.com/doc/refman/5.7/en/pluggable-authentication.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 Encrypt Sensitive Data at Rest Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●
v7	16.4 Encrypt or Hash all Authentication Credentials Encrypt or hash with a salt all authentication credentials when stored.		●	●

1.5 Ensure Interactive Login is Disabled (Automated)

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.

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.

Audit:

Execute the following command to assess this recommendation:

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

Lack of output implies a fail.

Remediation:

Execute one of the following commands in a terminal:

```
usermod -s /bin/false mysql
```

Or

```
usermod -s /sbin/nologin mysql
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.	●	●	●
v7	4.3 Ensure the Use of Dedicated Administrative Accounts Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities.	●	●	●

1.6 Verify That 'MYSQL_PWD' is Not Set in Users' Profiles (Automated)

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.

References:

1. https://dev.mysql.com/doc/refman/5.7/en/environment-variables.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 Encrypt Sensitive Data at Rest Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●
v7	16.4 Encrypt or Hash all Authentication Credentials Encrypt or hash with a salt all authentication credentials when stored.		●	●

2 Installation and Planning

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

An important consideration is related to the configuration options via the MySQL configuration file (e.g., `my.cnf`) and placing options under the proper section of `[mysqld]`. Options placed in the `my.cnf` configuration file should not prefix with a double dash (`--`). On Linux systems, `my.cnf` is located in the `/etc/` directory.

Finally, configuration options can also be placed on the command line by modifying the MySQL startup script. The startup script is system dependent and 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 (Manual)

Profile Applicability:

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

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

Audit:

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

Remediation:

Create a backup policy and backup schedule.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	11.2 Perform Automated Backups Perform automated backups of in-scope enterprise assets. Run backups weekly, or more frequently, based on the sensitivity of the data.	●	●	●

2.1.2 Verify Backups are Good (Manual)

Profile Applicability:

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

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. 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.

Audit:

Check reports of backup validation tests.

Remediation:

Implement regular backup checks and document each check.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	11.5 Test Data Recovery Test backup recovery quarterly, or more frequently, for a sampling of in-scope enterprise assets.		●	●
v7	10.3 Test Data on Backup Media Test data integrity on backup media on a regular basis by performing a data restoration process to ensure that the backup is properly working.		●	●

2.1.3 Secure Backup Credentials (Manual)

Profile Applicability:

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

Description:

A database user with the least amount of privileges required to perform backup is needed for backup. The credentials for this user should be protected. The password, certificate and any other credentials should be protected.

Rationale:

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.

Audit:

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

Remediation:

Change file permissions.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v8	11.3 Protect Recovery Data Protect recovery data with equivalent controls to the original data. Reference encryption or data separation, based on requirements.	●	●	●
v7	10.4 Ensure Protection of Backups Ensure that backups are properly protected via physical security or encryption when they are stored, as well as when they are moved across the network. This includes remote backups and cloud services.	●	●	●

2.1.4 The Backups Should be Properly Secured (Manual)

Profile Applicability:

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

Description:

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

Rationale:

Backups should be considered sensitive information. If an unauthorized user can access backups, then they have access to all data in the database. This is true for unencrypted backups and for encrypted backups if the encryption key is stored along with the backup.

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 consist of a guessable password?

Remediation:

Implement encryption, properly restrict filesystem permissions, protect and backup encryption keys.

References:

1. https://dev.mysql.com/doc/refman/5.7/en/innodb-backup.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	11.3 Protect Recovery Data Protect recovery data with equivalent controls to the original data. Reference encryption or data separation, based on requirements.	●	●	●
v7	10.4 Ensure Protection of Backups Ensure that backups are properly protected via physical security or encryption when they are stored, as well as when they are moved across the network. This includes remote backups and cloud services.	●	●	●

2.1.5 Point-in-Time Recovery (Manual)

Profile Applicability:

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

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. The binlogs need to be backed up to separate media. The restore procedure should be created and tested. The data in the binlog files may contain sensitive information which needs be stored in the proper location with restrictive permissions and may require encryption.

Rationale:

Using binlogs can reduce the amount of information lost when recovering from a backup.

Impact:

Binlogs can grow quite large and take up a large amount of space so auto remove needs to be put into place.

Audit:

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

Remediation:

Enable binlogs and create and test a restore procedure.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	11.2 Perform Automated Backups Perform automated backups of in-scope enterprise assets. Run backups weekly, or more frequently, based on the sensitivity of the data.	●	●	●
v7	10.2 Perform Complete System Backups Ensure that each of the organization's key systems are backed up as a complete system, through processes such as imaging, to enable the quick recovery of an entire system.	●	●	●

2.1.6 Disaster Recovery (DR) Plan (Manual)

Profile Applicability:

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

Description:

A disaster recovery plan should be created.

MySQL Cluster (group replication), MySQL Replica Sets (asynchronous replication) or both may be used.

A slave in a different data center and offsite backups may be used. There should be information regarding the Recovery Time Objective (RTO), i.e., how long recovery will take, and if the recovery site has the same capacity. Additionally, delayed replicas can be a valuable part of a DR plan. Network (default) and at rest encryption should be used to protect data.

Rationale:

A disaster recovery strategy should be planned and formalized. Without a well tested disaster recovery plan it might not be possible to recover in time.

Audit:

Check if there is a disaster recovery plan

Remediation:

Create a disaster recovery plan

References:

1. https://dev.mysql.com/doc/refman/5.7/en/group-replication-security.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	11.1 Establish and Maintain a Data Recovery Process Establish and maintain a data recovery process. In the process, address the scope of data recovery activities, recovery prioritization, and the security of backup data. Review and update documentation annually, or when significant enterprise changes occur that could impact this Safeguard.	●	●	●
v7	10 Data Recovery Capabilities Data Recovery Capabilities			

2.1.7 Backup of Configuration and Related Files (Manual)

Profile Applicability:

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

Description:

It is important to include configuration, log, key, certificates, and customized files in backups.

Rationale:

Including all configuration, log, key, certificates, and customized files in any backup will ensure the backup can fully restore an instance.

Audit:

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

- Edited Configuration files (`my.cnf` and included files)
- Files related to Key Management and Keyring (KMIP, other Key Management Services)
- Audit Log Files (if not handled by other methods)
- SSL files (certificates, keys)
- User Defined Functions (UDFs)
- Source code for customizations

Remediation:

Add any omitted files to the backup.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	11.2 Perform Automated Backups Perform automated backups of in-scope enterprise assets. Run backups weekly, or more frequently, based on the sensitivity of the data.	●	●	●
v7	10.2 Perform Complete System Backups Ensure that each of the organization's key systems are backed up as a complete system, through processes such as imaging, to enable the quick recovery of an entire system.	●	●	●

2.2 Dedicate the Machine Running MySQL (Manual)

Profile Applicability:

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

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.

Impact:

Care must be taken that to ensure applications or services that are required for 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.

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.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.12 Segment Data Processing and Storage Based on Sensitivity Segment data processing and storage based on the sensitivity of the data. Do not process sensitive data on enterprise assets intended for lower sensitivity data.		●	●
v7	2.10 Physically or Logically Segregate High Risk Applications Physically or logically segregated systems should be used to isolate and run software that is required for business operations but incur higher risk for the organization.			●

2.3 Do Not Specify Passwords in Command Line (Manual)

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.

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`.

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`.

References:

1. https://dev.mysql.com/doc/refman/5.7/en/mysql-config-editor.html
2. https://dev.mysql.com/doc/refman/5.7/en/password-security-user.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 Encrypt Sensitive Data in Transit Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).		●	●
v7	16.4 Encrypt or Hash all Authentication Credentials Encrypt or hash with a salt all authentication credentials when stored.		●	●

2.4 Do Not Reuse Usernames (Manual)

Profile Applicability:

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

Description:

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

Rationale:

Utilizing unique database accounts across applications will reduce the impact of a compromised MySQL account. If a user is reused, then a compromise of this user will compromise multiple parts of the system and/or application.

Audit:

Each user (excluding mysql reserved users) should be linked to one of these:

- system accounts
- a person
- an application

To list users (and exclude mysql reserved users)

```
SELECT host, user, plugin,
  IF(plugin = 'mysql_native_password',
 'WEAK SHA1', 'STRONG SHA2') AS HASHTYPE
FROM mysql.user WHERE user NOT IN
  ('mysql.infoschema', 'mysql.session', 'mysql.sys') AND
  plugin NOT LIKE 'auth%' AND plugin <> 'mysql_no_login' AND
  LENGTH(authentication_string) > 0
ORDER BY plugin;
```

Remediation:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.1 Establish and Maintain an Inventory of Accounts Establish and maintain an inventory of all accounts managed in the enterprise. The inventory must include both user and administrator accounts. The inventory, at a minimum, should contain the person's name, username, start/stop dates, and department. Validate that all active accounts are authorized, on a recurring schedule at a minimum quarterly, or more frequently.	●	●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	4.3 Ensure the Use of Dedicated Administrative Accounts Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities.	●	●	●

2.5 Ensure Non-Default, Unique Cryptographic Material is in Use (Manual)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

The cryptographic material used by MySQL, such as digital certificates and encryption keys, should be used only for MySQL and only for one instance. Default cryptographic material should not be used since it is not unique to the instance.

Rationale:

If a cryptographic material is used on multiple MySQL instances and/or systems then a compromise of one may lead to the compromise of the network traffic of all servers which use the same cryptographic material. If an attacker gains access to shared cryptographic material, including default material, the attacker can reuse that material to impersonate the MySQL server or otherwise compromise its operations.

Audit:

Review all cryptographic material. If it is default, used for other MySQL instances and/or for purposes other than MySQL then this is a finding.
Review the server certificate by running

```
cd <data_dir and/or ssl_cert>
sudo openssl x509 -in server-cert.pem -subject -noout | grep
Auto_Generated_Server_Certificate
```

The output for the auto generated pem will look something like:

```
subject= /CN=MySQL_Server_5.7.36_Auto_Generated_Server_Certificate
```

If no rows return, the check is a pass since the certificate is not MySQL auto-generated.

Remediation:

Generate new certificates, keys, and other cryptographic material as needed for each affected MySQL instance.

References:

1. https://dev.mysql.com/doc/refman/5.7/en/using-encrypted-connections.html

2.6 Ensure 'password_lifetime' is Less Than or Equal to '365' (Automated)

Profile Applicability:

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

Description:

Ensure 'password_lifetime' is Less Than or Equal to '365'

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.

Audit:

The global password lifetime is set using `default_password_lifetime`. If the value of `default_password_lifetime` is greater than `0`, it indicates the permitted password lifetime.

Execute the following command to check the global password lifetime:

```
SELECT VARIABLE_NAME, VARIABLE_VALUE
FROM performance_schema.global_variables where VARIABLE_NAME like
'default_password_lifetime';
```

A value greater than `365` implies a fail.

When the global password lifetime is less than or equal to `365`, or not configured, each user account shall be checked by executing the following command:

```
SELECT user, host, password_lifetime from mysql.user where password_lifetime
= 0 OR password_lifetime >= 365;
```

A lack of results implies compliance.

Note: A value of `0` implies the password never expires.

Remediation:

To configure the global password lifetime to `365` by executing the following command:

```
set global default_password_lifetime = 365;
```

Alternatively, configure the password lifetime for each user returned by the audit procedure by executing the following command:

```
ALTER USER '<username>'@'<localhost>' PASSWORD EXPIRE INTERVAL 365 DAY;
```

Default Value:

NULL

References:

1. https://csrc.nist.gov/csrc/media/publications/sp/800-118/archive/2009-04-21/documents/draft-sp800-118.pdf
2. https://dev.mysql.com/doc/refman/5.7/en/validate-password.html

Additional Information:

When a user's `password_lifetime` is set to `NULL` it takes on the value set in global `default_password_lifetime` variable.

2.7 Ensure Password Complexity is Configured (Automated)

Profile Applicability:

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

Description:

Passwords that are too complex in nature make it harder for users to remember, leading to bad practices. In addition, composition requirements provide no defense against common attack types such as social engineering or insecure password storage. In keeping with the overall goal of having users create a password that is not overly weak, it's best to have at least 14 characters for a password only account.

Rationale:

Malicious actors regularly attempt to compromise databases by attacking or guessing passwords. Stolen credentials may be used to gain access to steal information, engage in financial fraud, and more.

By enforcing practical and secure policies, end user cooperation grows. In general, longer passwords are better (harder to crack), but a forced password length requirement can cause user behavior that is predictable and undesirable. Having a reasonable minimum length with no maximum character limit increases the resulting average password length used and thus increases the security of that password.

Impact:

Enforcing too much complexity or length may be difficult for users to memorize. This may cause users to use predictable patterns or other bad practices, resulting in weaker passwords.

Audit:

Inspect the password policy settings.

```
SELECT VARIABLE_NAME, VARIABLE_VALUE
FROM performance_schema.global_variables
WHERE VARIABLE_NAME like 'valid%password%';
```

Compare the results to the table below.

```
+--------------------------------------+---------------------------------+
| Variable_name                        | Value                           |
+--------------------------------------+---------------------------------+
| validate_password_check_user_name    | ON                              |
| validate_password_dictionary_file    | '<FILENAME OF DICTIONARY FILE>' |
| validate_password_length             | 14                              |
| validate_password_mixed_case_count   | 1                               |
| validate_password_number_count       | 1                               |
| validate_password_policy             | STRONG                          |
| validate_password_special_char_count | 1                               |
+--------------------------------------+---------------------------------+
```

Remediation:

If not already present in the `my.cnf`, add this line:

```
plugin-load=validate_password.so
```

Set password policies in accordance with the organizationally defined policy and security best practices:

```
set global validate_password_check_user_name='ON';
set global validate_password_dictionary_file='<FILENAME OF DICTIONARY FILE>';
set global validate_password_length=14;
set global validate_password_policy='STRONG';
```

Use with care. Passwords that are too complex in nature make it harder for users to remember, leading to bad practices.

```
set global validate_password_mixed_case_count=1;
set global validate_password_special_char_count=1;
set global validate_password_number_count=1;
```

Default Value:

The MySQL validate password complexity plugin is not used by default.

References:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.2 Use Unique Passwords Use unique passwords for all enterprise assets. Best practice implementation includes, at a minimum, an 8-character password for accounts using MFA and a 14-character password for accounts not using MFA.	●	●	●

2.8 Lock Out Accounts if Not Currently in Use (Manual)

Profile Applicability:

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

Description:

If users with accounts will not be using their account for some time, to reduce the risk of attacks or inappropriate account usage or if suspicions exist that an account might be under attack, disabling the account will secure it and once it's ready to resume use it can easily be re-enabled.

Rationale:

Only have active accounts that will be used.

Audit:

Review the locked status of accounts:

```
select user, host, account_locked from mysql.user;
```

Accounts not in use and MySQL Reserved accounts should show as locked (`Y`).

Remediation:

To lock accounts - example:

```
ALTER USER 'jeffrey'@'localhost' ACCOUNT LOCK;
```

To unlock accounts - example

```
ALTER USER 'jeffrey'@'localhost' ACCOUNT UNLOCK;
```

Note: Works for `CREATE` as well. It is good practice to `LOCK` an account if created ahead of time.

Default Value:

Accounts are unlocked by default.

References:

1. https://dev.mysql.com/doc/refman/5.7/en/account-locking.html

Additional Information:

When a client attempts to connect to a locked account, the attempt fails.

```
Access denied for user 'user_name'@'host_name'.
Account is locked.
```

The server increments the `Locked_connects` status variable that indicates the number of attempts to connect to a locked account. To view the `Locked_conects` execute this query:

```
show global status like 'Locked_connects';
```

The error log will contain the message `ER_ACCOUNT_HAS_BEEN_LOCKED`.

The Oracle MySQL documentation, at the referenced link, provides this additional information:

> "The account-locking capability depends on the presence of the account_locked column in the mysql.user system table. For upgrades from MySQL versions older than 5.7.6, perform the MySQL upgrade procedure to ensure that this column exists. See Section 2.11, “Upgrading MySQL”. For nonupgraded installations that have no account_locked column, the server treats all accounts as unlocked, and using the ACCOUNT LOCK or ACCOUNT UNLOCK clauses produces an error."

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.3 Disable Dormant Accounts Delete or disable any dormant accounts after a period of 45 days of inactivity, where supported.	●	●	●
v7	16.9 Disable Dormant Accounts Automatically disable dormant accounts after a set period of inactivity.	●	●	●

2.9 Ensure AES Encryption Mode for AES_ENCRYPT/AES_DECRYPT is Configured Correctly (Automated)

Profile Applicability:

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

Description:

A block encryption mode with a Cipher Block Chaining (CBC) mode value and key length of 256 is recommended when using the `AES_ENCRYPT()` and `AES_DECRYPT()` functions for encryption.

Rationale:

The default for backward compatibility on upgraded MySQL databases is `aes-128-ecb`. Using 128-bit keys does not provide sufficient security. Regardless of whether breaking the lowest level is beyond existing technology, larger key sizes are needed to better protect data and satisfy regulations.

Impact:

Configuring a key length of 256 may impact backwards compatibility.

Audit:

Run the following statement:

```
select @@block_encryption_mode;
```

A value other than `aes-256-*` is a fail.

Where `*` is one of the following - `ECB`, `CBC`, `CFB1`, `CFB8`, `CFB128`, `OFB`

Remediation:

Add the following lines to the MySQL server's `/etc/my.cnf`:

For example, if Block Encryption Mode for aes-256 CBC

```
block_encryption_mode=aes-256-cbc
```

Default Value:

`aes-128-ecb`

References:

1. https://dev.mysql.com/doc/mysql-secure-deployment-guide/5.7/en/secure-deployment-block-encryption-mode.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.11 Leverage Vetted Modules or Services for Application Security Components Leverage vetted modules or services for application security components, such as identity management, encryption, and auditing and logging. Using platform features in critical security functions will reduce developers' workload and minimize the likelihood of design or implementation errors. Modern operating systems provide effective mechanisms for identification, authentication, and authorization and make those mechanisms available to applications. Use only standardized, currently accepted, and extensively reviewed encryption algorithms. Operating systems also provide mechanisms to create and maintain secure audit logs.		●	●
v7	18.5 Use Only Standardized and Extensively Reviewed Encryption Algorithms Use only standardized and extensively reviewed encryption algorithms.		●	●

2.10 Ensure Socket Peer-Credential Authentication is Used Appropriately (Manual)

Profile Applicability:

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

Description:

The server-side `auth_socket` authentication plugin, authenticates clients that connect to the MySQL server from the local host through the Unix socket file. Users authenticated by the `auth_socket` need not specify a password when connecting to the server. However, users authenticated by the `auth_socket` plugin are restricted from connecting remotely; they can only connect from the local host through the Unix socket file. This method is only suitable in situations where the server administrator OS account access is restricted.

Rationale:

This method may be desirable in specific cases, including:

- The Linux system where MySQL is running is dedicated to the MySQL server and only the MySQL DBA and OS Admin have access.
- When control over user authentication is centralized in the operating system.
- It is desirable that audit trails in the database and operating system can use the same user names.
- For certain other narrow installation use cases `auth_socket` may be desirable.
- Only local connections for a user.

Impact:

Things to consider when using the operating system to authenticate users:

- The user must have an operating system account on the computer which must be accessed.
- If a user has logged in using this method and steps away from the terminal, another user could easily log in because this user does not need any passwords or credentials. This could pose a serious security problem.
- When an operating system is used to authenticate database users, managing distributed database environments and database links requires special care. Special care must also be taken not to leave such a terminal unlocked and unattended. Hence, we recommend that you carefully evaluate your requirements before opting for `auth_socket`.
- This will not work where distributed connections are required.

Audit:

To assess this recommendation run the following:

```
SELECT PLUGIN_NAME, PLUGIN_STATUS
       FROM INFORMATION_SCHEMA.PLUGINS
       WHERE PLUGIN_NAME LIKE 'auth%';
```

To determine users who can use auth socket:

```
select user, host, plugin from mysql.user where plugin = 'auth_socket';
```

If this is enabled and the organization does not allow use of this feature, this is a fail.

If `host` is not the localhost or an unauthorized user is listed, this is a fail.

Remediation:

Add these options under the `[mysqld]` option group in the MySQL `/etc/my.cnf`:

```
plugin-load-add=auth_socket.so
auth_socket=FORCE_PLUS_PERMANENT
```

For example:

For an OS user which can login to MySQL using `auth_socket`:

```
CREATE USER '  i'@'localhost' IDENTIFIED WITH auth_socket;
```

The user can then login using:

```
mysql -u <user>
```

References:

1. https://dev.mysql.com/doc/mysql-secure-deployment-guide/5.7/en/secure-deployment-configure-authentication.html#secure-deployment-auth-socket

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.6 Establish and Maintain an Inventory of Authentication and Authorization Systems Establish and maintain an inventory of the enterprise's authentication and authorization systems, including those hosted on-site or at a remote service provider. Review and update the inventory, at a minimum, annually, or more frequently.		●	●

2.11 Ensure MySQL is Bound to an IP Address (Automated)

Profile Applicability:

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

Description:

By default, the MySQL server accepts TCP/IP connections from MySQL user accounts on all server host IPv6 and IPv4 interfaces. You can make this configuration more restrictive by setting the `bind_address` configuration option to a specific IPv4 or IPv6 address so that the server only accepts TCP/IP connections on that address.

Rationale:

Limiting the IP address provides additional controls and restrictions on how client applications can connect to MySQL. If not configured to a specific IP all IPs for this server can be used to connect to MySQL.

Audit:

Run the following statement:

```
SELECT VARIABLE_NAME, VARIABLE_VALUE
FROM performance_schema.global_variables
WHERE VARIABLE_NAME = 'bind_address';
```

No rows returned implies a fail.

Remediation:

For example, to have the MySQL server only accept connections on a specific IPv4 address, add an entry similar to this under the `[mysqld]` option group in the MySQL `/etc/my.cnf`:

```
bind_address=192.0.2.24
```

In this case, clients can connect to the server using `--host=192.0.2.24`. Connections on other server host addresses are not permitted.

Default Value:

Not set.

References:

1. https://dev.mysql.com/doc/mysql-secure-deployment-guide/5.7/en/secure-deployment-secure-connections.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.10 Apply Secure Design Principles in Application Architectures Apply secure design principles in application architectures. Secure design principles include the concept of least privilege and enforcing mediation to validate every operation that the user makes, promoting the concept of "never trust user input." Examples include ensuring that explicit error checking is performed and documented for all input, including for size, data type, and acceptable ranges or formats. Secure design also means minimizing the application infrastructure attack surface, such as turning off unprotected ports and services, removing unnecessary programs and files, and renaming or removing default accounts.		●	●

2.12 Limit Accepted Transport Layer Security (TLS) Versions (Automated)

Profile Applicability:

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

Description:

MySQL supports multiple protocols of TLS. The higher the version the stronger the security and/or better the performance.

Rationale:

Requiring clients attempting to connect to MySQL to use higher versions of TLS to better protect data in transit.

Impact:

Connections attempting to use an unsupported version of TLS or Cipher will fail.

Audit:

To list the versions of TLS the server accepts, run the following statement:

```
select @@tls_version;
```

If the list includes `TLSv1` and/or `TLSv1.1`, this is a fail.

To view current connections and the version of SSL in use run:

```
select * from performance_schema.status_by_thread where VARIABLE_NAME like
'ssl_version';
```

If the list includes, `TLSv1` and/or `TLSv1.1`, this is a fail.

MySQL negotiates to the highest version of TLS, if connections are using older TLS versions, those clients will need to be upgraded to newer MySQL Connectors or community drivers that support newer versions of TLS.

Remediation:

Set the version(s) of TLS you wish to accept in `mysql.conf` specify TLS and Ciphers.

For example to only accept TLS 1.3 set `tls_version` in `my.conf`:

```
tls_version=TLSv1.3
```

If TLS 1.3 is not supported on the Operating System then set to TLS 1.2:

```
tls_version=TLSv1.2
```

Note: with this setting, only clients that support the specified TLS version(s) are able to establish an encrypted connection to the server.

Default Value:

All TLS and cipher versions are enabled by default.

References:

1. https://dev.mysql.com/doc/mysql-secure-deployment-guide/5.7/en/secure-deployment-secure-connections.html
2. https://dev.mysql.com/doc/refman/5.7/en/encrypted-connection-protocols-ciphers.html#encrypted-connection-protocol-configuration

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 Encrypt Sensitive Data in Transit Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).		●	●
v8	16.11 Leverage Vetted Modules or Services for Application Security Components Leverage vetted modules or services for application security components, such as identity management, encryption, and auditing and logging. Using platform features in critical security functions will reduce developers' workload and minimize the likelihood of design or implementation errors. Modern operating systems provide effective mechanisms for identification, authentication, and authorization and make those mechanisms available to applications. Use only standardized, currently accepted, and extensively reviewed encryption algorithms. Operating systems also provide mechanisms to create and maintain secure audit logs.		●	●
v7	14.4 Encrypt All Sensitive Information in Transit Encrypt all sensitive information in transit.		●	●
v7	18.5 Use Only Standardized and Extensively Reviewed Encryption Algorithms Use only standardized and extensively reviewed encryption algorithms.		●	●

2.13 Require Client-Side Certificates (X.509) (Automated)

Profile Applicability:

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

Description:

Client-side certificates may be used as proof of identity as well as to encrypt data in transit.

Rationale:

Requiring client-side certificates provides additional validation of a user's identity, thus increasing the level of security, while also providing strong encryption.

Audit:

Run the following statement

```
select user, host, ssl_type from mysql.user;
```

If `ssl_type` returns `X509` or `SSL`, client-side certificate details must be provided to connect.

Remediation:

Create or Alter users using the `REQUIRE X509`.

For example:

```
CREATE USER 'newuser2'@'%' IDENTIFIED BY <password> require x509;
```

For accounts created with a `REQUIRE X509` clause, clients must specify at least `--ssl-cert` and `--ssl-key`. In addition, `--ssl-ca` (or `--ssl-capath`) is recommended so that the public certificate provided by the server can be verified.

For example:

```
mysql --ssl-ca=ca.pem \
      --ssl-cert=client-cert.pem \
      --ssl-key=client-key.pem
```

Additional Information:

The audit procedure excludes these internal user accounts from evaluation because, by default, they are created with an invalid password and are locked to disallow access.

- `'mysql.infoschema'@'localhost'`
- `'mysql.session'@'localhost'`
- `'mysql.sys'@'localhost'`

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 Encrypt Sensitive Data in Transit Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).		●	●
v7	14.4 Encrypt All Sensitive Information in Transit Encrypt all sensitive information in transit.		●	●

2.14 Ensure Only Approved Ciphers are Used (Automated)

Profile Applicability:

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

Description:

MySQL supports multiple encryption ciphers. Ciphers can vary in strength, speed and overhead.

Rationale:

Requiring clients attempting to connect to MySQL to use strong ciphers protects data in transit.

Impact:

Connections attempting to use an unsupported cipher will fail.

Audit:

Run the following statement:

```
SELECT VARIABLE_NAME, VARIABLE_VALUE
FROM performance_schema.global_variables
WHERE VARIABLE_NAME='ssl_cipher';
```

If `ssl_cipher` is not set to `ECDHE-ECDSA-AES128-GCM-SHA256`, this is a fail.

Remediation:

Set `ssl_cipher` in the `my.cnf` to an approved cipher suite:

```
ssl_cipher='ECDHE-ECDSA-AES128-GCM-SHA256'
```

References:

1. https://dev.mysql.com/doc/refman/5.7/en/encrypted-connection-protocols-ciphers.html#encrypted-connection-cipher-configuration

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.11 Leverage Vetted Modules or Services for Application Security Components Leverage vetted modules or services for application security components, such as identity management, encryption, and auditing and logging. Using platform features in critical security functions will reduce developers' workload and minimize the likelihood of design or implementation errors. Modern operating systems provide effective mechanisms for identification, authentication, and authorization and make those mechanisms available to applications. Use only standardized, currently accepted, and extensively reviewed encryption algorithms. Operating systems also provide mechanisms to create and maintain secure audit logs.		●	●
v7	18.5 Use Only Standardized and Extensively Reviewed Encryption Algorithms Use only standardized and extensively reviewed encryption algorithms.		●	●

2.15 Implement Connection Delays to Limit Failed Login Attempts (Automated)

Profile Applicability:

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

Description:

MySQL Server can enable administrators to introduce an increasing delay in server response to clients after a certain number of consecutive failed connection attempts.

Rationale:

Delaying connection attempts provides a deterrent that slows down brute force attacks that attempt to access MySQL user accounts.

Audit:

Determine if the plugins for delaying connections are installed.

```
SELECT PLUGIN_NAME, PLUGIN_STATUS
       FROM INFORMATION_SCHEMA.PLUGINS
       WHERE PLUGIN_NAME LIKE 'connection%';
```

Two rows should be returned showing `ACTIVE` status.

```
CONNECTION_CONTROL                | ACTIVE
CONNECTION_CONTROL_FAILED_LOGIN_ATTEMPTS | ACTIVE
```

If both plugins are not active, this is a fail.
Next assess the setting for the connection controls by running

```
SELECT VARIABLE_NAME, VARIABLE_VALUE
       FROM performance_schema.global_variables WHERE VARIABLE_NAME LIKE
'connection_control%';
```

Time doubling throttling (in minutes) between each retry (`0`, `1`, `2`, `4`, `8`, etc.) with a permanent account lockout (IT reset required) after 12 retries.

If `connection_control_failed_connections_threshold` is less than `5` (attempts), this is a fail.

If `connection_control_min_connection_delay` is less than `60000` (ms - 1 minute), this is a fail.

Max delay `connection_control_max_connection_delay` is `0` or less than `1920000` (ms, 32 minutes) a, this is a fail.

Finally, assess the failed login attempts.

```
select host, user, JSON_EXTRACT(user_attributes,
'$.Password_locking.failed_login_attempts') as failed_login_attempts from
mysql.user;
```

If failed login attempts is less than `12` this is a fail.

Remediation:

Add the following lines to `my.cnf`:

```
[mysqld]
plugin-load-add=connection_control.so
connection-control=FORCE_PLUS_PERMANENT
connection-control-failed-login-attempts=FORCE_PLUS_PERMANENT
connection_control_failed_connections_threshold=5
connection_control_min_connection_delay=60000
connection_control_max_connection_delay=1920000
```

Delays are in milliseconds for server response to failed connection attempt.

- `60000` (ms - 1 minute)
- `1920000` (ms, 32 minutes)

For each user set

```
ALTER USER <user> FAILED_LOGIN_ATTEMPTS 12;
```

References:

1. https://dev.mysql.com/doc/refman/5.7/en/connection-control.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v7	16 Account Monitoring and Control Account Monitoring and Control			

3 File Permissions

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

3.1 Ensure 'datadir' Has Appropriate Permissions (Automated)

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, it may be possible 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';
```

Or

```
SELECT VARIABLE_NAME, VARIABLE_VALUE
FROM performance_schema.global_variables
WHERE VARIABLE_NAME LIKE 'datadir';
```

- Execute the following command at a terminal prompt

```
sudo ls -ld <datadir> | grep "drwxr-x---.*mysql.*mysql"
```

Lack of output implies a fail.

Remediation:

Execute the following commands at a terminal prompt:

```
chmod 750 <datadir>
chown mysql:mysql <datadir>
```

References:

1. https://dev.mysql.com/doc/mysql-secure-deployment-guide/5.7/en/secure-deployment-permissions.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

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

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 (which can be encrypted), error log, slow query log, relay log, general log, and in the enterprise edition, the audit log (which can be encrypted). Because these are files on the host operating system, they are subject to the permissions and ownership structure provided by the host and may be accessible by users other than the MySQL user. Additionally, using secure key management and at rest MySQL encryption can further protect data from OS users.

Rationale:

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

Impact:

Changing the permissions and ownership of the relay logs and binary log files might have impact on external tools.

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.

Audit:

Perform the following steps to assess this recommendation:

1. Execute the following SQL statement to determine the Value of `log_bin_basename`

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

2. Execute the following command at a terminal prompt to list all non-compliant `log_bin_basename.*` file permissions

```
ls -l | egrep "^-(?![r|w]{2}-[r|w]{2}----
.\*mysql\s\*mysql).\*<log_bin_basename>.\*$
```

Lack of output implies compliance.

Remediation:

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

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

References:

1. https://dev.mysql.com/doc/refman/5.7/en/password-logging.html
2. https://dev.mysql.com/doc/mysql-secure-deployment-guide/5.7/en/secure-deployment-permissions.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

3.3 Ensure 'log_error' Has Appropriate Permissions (Automated)

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 (which can be encrypted), error log, slow query log, relay log, general log, and in the enterprise edition, the audit log (which can be encrypted). Because these are files on the host operating system, they are subject to the permissions and ownership structure provided by the host and may be accessible by users other than the MySQL user. Additionally, using secure key management and at rest MySQL encryption can further protect data from OS users.

Much of the information about the state of MySQL exists in MySQL, the MySQL `performance_schema` or `informations_schema`. In cases where the information you need is within a running MySQL, use these methods as they are more secure as do not require OS login and access.

Rationale:

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

Impact:

Changing the permissions of the error log files might have impact on monitoring tools which use an error log file adapter.

Audit:

Perform the following steps to assess this recommendation:

1. Execute the following SQL statement to determine the Value of `log_error`:

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

2. Execute the following command at a terminal prompt to list all non-compliant *`<log_error>`*`.*` file permissions:

```
ls -l /usr/local/mysql/data/mysqld.local.err | grep '^-rw-------
.*mysql.*mysql.*$'
```

   Lack of output implies a fail.

Remediation:

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

```
chmod 600 <log file>
chown mysql:mysql <log file>
```

References:

1. https://dev.mysql.com/doc/refman/5.7/en/error-log.html
2. https://dev.mysql.com/doc/mysql-secure-deployment-guide/5.7/en/secure-deployment-permissions.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

3.4 Ensure 'slow_query_log' Has Appropriate Permissions (Automated)

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 (which can be encrypted), error log, slow query log, relay log, general log, and in the enterprise edition, the audit log (which can be encrypted). Because these are files on the host operating system, they are subject to the permissions and ownership structure provided by the host and may be accessible by users other than the MySQL user. Additionally, using secure key management and at rest MySQL encryption can further protect data from OS users.

Much of the information about the state of MySQL exists in MySQL, the MySQL `performance_schema` or `informations_schema`. If you can get the information you need from within MySQL that is more secure as it does not require OS access. If you are not going to use log files it is best to first disable (don't enable) and remove any prior logs.

Rationale:

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

Impact:

Changing the permissions of the log files may impact monitoring tools which use a log file adapter. Also, the slow query log can be used for performance analysis by application developers.

The information about the performance exists in MySQL `performance_schema` or `sys` schema views. In cases where the information you need is within a running MySQL, disable the slow query log and instead use these methods as they are more secure and do not require OS login and access.

Audit:

Perform the following steps to assess this recommendation:

1. Execute the following SQL statement to determine the Value of `slow_query_log`:

```
show variables like 'slow_query_log';
```

   Best for the slow query log to be disabled indicated by OFF.

2. Execute the following SQL statement to determine the location of `slow_query_log_file`:

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

3. Execute the following command at a terminal prompt to list non-compliant *`<slow_query_log_file>.*`* file permissions:

```
ls -l | egrep "^-(?![r|w]{2}-[r|w]{2}----
.*mysql\s*mysql).*<slow_query_log_file>.*$
```

If the slow query log is enabled, lack of output implies compliance.
If the slow query log is disabled, remove any old slow query log files.

Remediation:

Set slow query log to OFF (instead use `SYS` schema views or query `Performance_Schema`)

```
SET @@GLOBAL.slow_query_log = OFF;
```

If slow query is enabled, execute the following command to correct permissions and ownership:

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

Default Value:

Slow query log is off by default.

References:

1. https://dev.mysql.com/doc/refman/5.7/en/slow-query-log.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

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

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 (which can be encrypted), error log, slow query log, relay log (which can be encrypted), general log, and in the enterprise edition, the audit log (which can be encrypted). Because these are files on the host operating system, they are subject to the permissions and ownership structure provided by the host and may be accessible by users other than the MySQL user. Additionally, using secure key management and at rest MySQL encryption can further protect data from OS users.

Rationale:

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

Impact:

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.

Audit:

Perform the following steps to assess this recommendation:

1. Execute the following SQL statement to determine the Value of `relay_log_basename`:

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

2. Execute the following command at a terminal prompt to list non-compliant *`<relay_log_basename>.*`* file permissions:

```
ls -l | egrep "^-(?![r|w]{2}-[r|w]{2}----
.*mysql\s*mysql).*<relay_log_basename>.*$
```

   Lack of output implies compliance.

Remediation:

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

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

Default Value:

<datadir> + '/' + *<hostname>* + '-relay-bin'

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

3.6 Ensure 'general_log_file' Has Appropriate Permissions (Automated)

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 (which can be encrypted), error log, slow query log, relay log (which can be encrypted), general log, and in the enterprise edition, the audit log (which can be encrypted). Because these are files on the host operating system, they are subject to the permissions and ownership structure provided by the host and may be accessible by users other than the MySQL user. Additionally, using secure key management and at rest MySQL encryption can further protect data from OS users.

Much of the information about the state of MySQL exists in MySQL, the MySQL `performance_schema` or `informations_schema`. If you can get the information you need from within MySQL that is more secure as it does not require OS access. If you are not going to use log files it is best to first disable (don't enable) and remove any prior logs.

Rationale:

Limiting the accessibility, or existence, of these log files will protect the confidentiality, integrity, and availability of the MySQL logs.

Impact:

Changing the permissions of the general log files may impact monitoring tools which use a log file adapter.

Audit:

Perform the following steps to assess this recommendation:

1. Execute the following SQL statement to determine the Values of `general_log` and `general_log_file`:

```
select @@general_log, @@general_log_file;
```

   With a `general_log` value of `0` or `OFF`, indicates the log is disabled. If `1` or `ON` it is enabled.

2. Whether the value is `0`, `OFF`, `1` or `ON` execute the following command at a terminal prompt to list non-compliant *`<general_log_file>.*`* file permissions:

```
ls -l <general_log_file>
```

If `general_log` is `0` or `OFF` (disabled) and the log file exists, remove the old general log file.

If `general_log` is `1` or `ON` (enabled) review the permissions

```
ls -l <general_log_file> grep '^-rw-------.*mysql.*mysql'
```

Lack of output implies compliance.

Remediation:

If you can, use MySQL `SYS`, `PERFORMANCE_SCHEMA`, or MySQL Auditing as these are more secure options.

By default the `general_log` is disabled (`0` or `OFF`). It's most secure to disable the `general_log`.

To disable the `general_log_file`:

```
SET @@GLOBAL.GENERAL_LOG=OFF;
```

If you must use `general_log` then assure the permissions are correct. Execute the following command for each log file location requiring corrected permissions and ownership:

```
chmod 600 <general_log_file>
chown mysql:mysql <general_log_file>
```

Default Value:

The variable `general_log` is set to `OFF` by default. The variable `general_log_file` is set to `<host_name>.log` by default.

References:

1. https://dev.mysql.com/doc/refman/5.7/en/query-log.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

3.7 Ensure SSL Key Files Have Appropriate Permissions (Automated)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

When configured to use SSL/TLS, MySQL relies on Secure Sockets Layer (SSL) key files, which are stored on the host's filesystem. These SSL key files are subject to the host's permissions and ownership structure.

MySQL 5.7 provides ways to create the SSL certificate, SSL key files and RSA key-pair files required to support encrypted connections using SSL and secure password exchange using RSA over unencrypted connections, if those files are missing the server will attempt to autogenerate these files at startup if compiled with OpenSSL.

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 are 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 cipher suite, the key might also be used to decipher previously captured network traffic.

Impact:

If the permissions or ownership for the SSL key file are configured 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 key pair, then changing the permissions or ownership of the SSL key file will affect this application. If this were to occur a new key pair must be generated for MySQL.

Audit:

Perform the following steps to assess this recommendation:

1. Locate the SSL keys and certs in use by executing the following SQL statement. To show all ssl variables:

```
SELECT VARIABLE_VALUE FROM performance_schema.global_variables
WHERE VARIABLE_NAME RLIKE '^.*ssl_(ca|capath|cert|crl|crlpath|key)$'
AND VARIABLE_VALUE <> '';
```

Note: Any `mysqlx_%` values that are null default to the classic protocols equivalent value.

2. Execute the following commands at a terminal prompt to list non-compliant *<ssl_file>* file permissions:

```
ls -l | egrep "^-(?!r-{8}.*mysql\s*mysql).*<ssl_file>.*$"
```

Lack of output implies compliance

Remediation:

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

```
chown mysql:mysql <ssl_file>
chmod 400 <ssl_file>
```

References:

1. https://dev.mysql.com/doc/refman/5.7/en/encrypted-connections.html
2. https://dev.mysql.com/doc/refman/5.7/en/creating-ssl-rsa-files-using-mysql.html

Additional Information:

If SSL is not configured this recommendation is not applicable. By default MySQL enables SSL. Using SSL is highly recommended.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

3.8 Ensure Plugin Directory Has Appropriate Permissions (Automated)

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.

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.

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 and ownership.

```
ls -ld <plugin_dir Value> | grep "dr-xr-x---\|dr-xr-xr--" | grep "plugin"
```

Lack of output implies a fail.

Note: Permissions are intended to be either `550` or `554`.

Remediation:

To remediate these settings, execute the following commands at a terminal prompt using the `plugin_dir Value` from the audit procedure. MySQL server must not be allowed to write to this location.

```
chmod 550 <plugin_dir Value> #(or use 554)
chown mysql:mysql <plugin_dir Value>
```

References:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

3.9 Secure MySQL Keyring (Automated)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

When configured to use a Keyring plugin, internal MySQL components and plugins may securely store sensitive information for later retrieval. Associated files for the selected keyring type should have proper permissions.

Rationale:

Limiting the accessibility of these objects will protect the confidentiality, integrity, and availability of internal MySQL component and plugin information.

Audit:

Perform the following steps applicable to the plugin in use to assess this recommendation:

Keyring File Plugin (Least Secure - for pre-production testing)

1. Find the `keyring_file_data` value (*<keyring_file_data_path>*) by executing the following statement:

```
grep -Po '(?<=^keyring_file_data=).+$' /etc/mysql/my.cnf
```

2. Verify permissions are `750` for `mysql:mysql` for *<keyring_file_data_path>*

Keyring Encrypted File Plugin

1. Find the `keyring_encrypted_file_data` value (*<keyring_encrypted_file_data_path>*) by executing the following statement:

```
grep -Po '(?<=^keyring_encrypted_file_data=).+$' /etc/mysql/my.cnf
```

2. Verify permissions are `750` for `mysql:mysql` for *<keyring_encrypted_file_data_path>*
3. Verify a secure method for provisioning the passphrase for the keyring_encrypted_file_data is in place.

Keyring OKV / KMIP compatible Plugin

1. Find the `keyring_okv` value (*<keyring_okv_path>*) by executing the following statement:

```
grep -Po '(?<=^keyring_okv=).+$' /etc/mysql/my.cnf
```

2. Verify permissions are `750` for `mysql:mysql` for *<keyring_okv_path>*

AWS Key Management Service

1. Find the `keyring_aws_conf_file` and `keyring_aws_data_file` values by executing the following statement:

```
grep -Po '(?<=^keyring_aws.*=).+$' /etc/mysql/my.cnf
```

2. Verify permissions are `750` for `mysql:mysql` for `keyring_aws_conf_file` and `keyring_aws_data_file`

Additionally, if no keyring plugin or keyring file plugin is configured, this is a fail.

Remediation:

If no keyring plugin or keyring file plugin is configured, instructions for configuring a keyring plugin or keyring file plugin may found at:

- KMIP - https://dev.mysql.com/doc/refman/5.7/en/keyring-okv-plugin.html#keyring-okv-configuration
- AWS - https://dev.mysql.com/doc/refman/5.7/en/keyring-aws-plugin.html#keyring-aws-plugin-configuration

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

```
chmod 750 <keyring file>
chown mysql:mysql <keyring file>
```

References:

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

Additional Information:

Use of `keyring_file` is intended for development and testing and will not pass most security regulatory requirements.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

4 General

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

4.1 Ensure Latest Security Patches Are Applied (Manual)

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.

Impact:

To update the MySQL server a restart is required.

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.

References:

1. http://www.oracle.com/technetwork/topics/security/alerts-086861.html
2. https://dev.mysql.com/doc/relnotes/mysql/5.7/en/
3. https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aoracle&cpe_product=cpe%3A%2F%3Aoracle%3Amysql&cpe_version=cpe%3A%2F%3Aoracle%3Amysql%3A5.7.0

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	2.2 Ensure Authorized Software is Currently Supported Ensure that only currently supported software is designated as authorized in the software inventory for enterprise assets. If software is unsupported, yet necessary for the fulfillment of the enterprise's mission, document an exception detailing mitigating controls and residual risk acceptance. For any unsupported software without an exception documentation, designate as unauthorized. Review the software list to verify software support at least monthly, or more frequently.	●	●	●
v7	2.2 Ensure Software is Supported by Vendor Ensure that only software applications or operating systems currently supported by the software's vendor are added to the organization's authorized software inventory. Unsupported software should be tagged as unsupported in the inventory system.	●	●	●

4.2 Ensure Example or Test Databases are Not Installed on Production Servers (Automated)

Profile Applicability:

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

Description:

The default MySQL installation does not contain any example or test databases. However, it is a good idea to review for common example databases and ensure they have been removed from production systems.

Rationale:

Dropping example databases will reduce the attack surface of the MySQL server.

Audit:

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

```
SELECT * FROM information_schema.SCHEMATA where SCHEMA_NAME not in
('mysql','information_schema', 'sys', 'performance_schema');
```

If this is a production system, and a database name includes an example or `test` database this is a finding.
Common example database names are:

- employees
- world
- world_x
- sakila
- airportdb
- menagerie

Remediation:

Execute the following SQL statement to drop an example database:

```
DROP DATABASE <database name>;
```

Default Value:

By default, MySQL 5.7 does not contain any example or test databases.

References:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.10 Apply Secure Design Principles in Application Architectures Apply secure design principles in application architectures. Secure design principles include the concept of least privilege and enforcing mediation to validate every operation that the user makes, promoting the concept of "never trust user input." Examples include ensuring that explicit error checking is performed and documented for all input, including for size, data type, and acceptable ranges or formats. Secure design also means minimizing the application infrastructure attack surface, such as turning off unprotected ports and services, removing unnecessary programs and files, and renaming or removing default accounts.		●	●

4.3 Ensure 'allow-suspicious-udfs' Is Set to 'OFF' (Manual)

Profile Applicability:

- Level 2 - MySQL RDBMS on Linux

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 `OFF` in the MySQL configuration:

```
my_print_defaults mysqld | grep allow-suspicious-udfs
```

No results returned would be a pass.

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:

`OFF`

References:

1. https://dev.mysql.com/doc/extending-mysql/5.7/en/adding-loadable-function.html#loadable-function-security
2. https://dev.mysql.com/doc/refman/5.7/en/server-options.html#option_mysqld_allow-suspicious-udfs

Additional Information:

This option has no corresponding state in `SHOW VARIABLES`.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.10 Apply Secure Design Principles in Application Architectures Apply secure design principles in application architectures. Secure design principles include the concept of least privilege and enforcing mediation to validate every operation that the user makes, promoting the concept of "never trust user input." Examples include ensuring that explicit error checking is performed and documented for all input, including for size, data type, and acceptable ranges or formats. Secure design also means minimizing the application infrastructure attack surface, such as turning off unprotected ports and services, removing unnecessary programs and files, and renaming or removing default accounts.		●	●

4.4 Harden Usage for 'local_infile' on MySQL Clients (Automated)

Profile Applicability:

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

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 an SQL injection vulnerability.

Impact:

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

Audit:

Execute the following SQL statement:

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

Ensure the value returned is `OFF`.

Remediation:

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

```
local_infile=OFF
```

Default Value:

`ON`

References:

1. https://dev.mysql.com/doc/refman/5.7/en/string-functions.html#function_load-file
2. https://dev.mysql.com/doc/refman/5.7/en/load-data.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.		●	●
v7	4.7 Limit Access to Script Tools Limit access to scripting tools (such as Microsoft PowerShell and Python) to only administrative or development users with the need to access those capabilities.		●	●

4.5 Ensure 'mysqld' is Not Started with '--skip-grant-tables' (Manual)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

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

Additional Information:

This option has no `SHOW VARIABLES` counterpart.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

4.6 Ensure Symbolic Links are Disabled (Automated)

Profile Applicability:

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

Description:

The `symbolic-links` and `skip-symbolic-links` options for MySQL determine whether symbolic link support is available. When use of symbolic links is 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 symbolic links from being used for database 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 the 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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.10 Apply Secure Design Principles in Application Architectures Apply secure design principles in application architectures. Secure design principles include the concept of least privilege and enforcing mediation to validate every operation that the user makes, promoting the concept of "never trust user input." Examples include ensuring that explicit error checking is performed and documented for all input, including for size, data type, and acceptable ranges or formats. Secure design also means minimizing the application infrastructure attack surface, such as turning off unprotected ports and services, removing unnecessary programs and files, and renaming or removing default accounts.		●	●
v7	13 Data Protection Data Protection			

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

Profile Applicability:

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

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:

```
SHOW DATABASES LIKE 'innodb_memcache';
```

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.		●	●
v7	9.2 Ensure Only Approved Ports, Protocols and Services Are Running Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.		●	●

4.8 Ensure the 'secure_file_priv' is Configured Correctly (Automated)

Profile Applicability:

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

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. Even better, if data import/export using `LOAD DATA INFILE` or `SELECT local_file` is not used, the functionality should be disabled entirely by setting `--secure-file-priv` to `NULL`.

Rationale:

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

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.

The server checks the value of `secure_file_priv` at startup and writes a warning to the error log if the value is insecure. A non-NULL value is considered insecure if it is empty, or the value is the data directory or a subdirectory of it, or a directory that is accessible by all users.

Audit:

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

```
SHOW GLOBAL VARIABLES WHERE Variable_name = 'secure_file_priv';
```

The Value should either contain `NULL` (thus is disabled entirely) or a valid path. If set to an empty string this is a fail.

Remediation:

If you are not going to use this feature, remove `secure_file_priv` from the `[mysqld]` section of the MySQL configuration file and restart the MySQL service.

If you need this feature 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>
```

Default Value:

No value set.

References:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	13 Data Protection Data Protection			

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

Profile Applicability:

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

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 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.

Impact:

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

Audit:

To audit for this recommendation, execute the following query:

```
SHOW VARIABLES LIKE 'sql_mode';
+---------------+-------------------------------------------------------------+
| Variable_name | Value                                                       |
+---------------+-------------------------------------------------------------+
| sql_mode      | ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,     |
|               | NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,                    |
|               | NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION                  |
+---------------+-------------------------------------------------------------+
```

If `STRICT_ALL_TABLES` is not in the list returned, this is a fail.

Remediation:

Set `STRICT_ALL_TABLES` to the `sql_mode` in the server's global configuration, for example:

```
SET GLOBAL sql_mode
='STRICT_ALL_TABLES,ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO
_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUT
ION';
```

Default Value:

```
ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR
_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION
```

References:

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

Additional Information:

The `sql_mode` is a set and might contain more elements than just `STRICT_ALL_TABLES`.

There is a global `sql_mode` and a per session `sql_mode`. The per session `sql_mode` is based on the global `sql_mode` on initialization and might be changed by the application.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.10 Apply Secure Design Principles in Application Architectures Apply secure design principles in application architectures. Secure design principles include the concept of least privilege and enforcing mediation to validate every operation that the user makes, promoting the concept of "never trust user input." Examples include ensuring that explicit error checking is performed and documented for all input, including for size, data type, and acceptable ranges or formats. Secure design also means minimizing the application infrastructure attack surface, such as turning off unprotected ports and services, removing unnecessary programs and files, and renaming or removing default accounts.		●	●

5 MySQL Permissions

This section contains recommendations about user privileges.

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

Profile Applicability:

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

Description:

The `mysql.user`, `mysql.db`, and other `mysql` tables ending in `_priv` 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. The `information_schema.user_privileges` provides a consolidated view of all user privileges.

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 * from information_schema.user_privileges
where grantee not like ('\'mysql.%localhost\'');
```

Ensure all users returned are administrative users with minimal privileges required.

Note: The above query ignores MySQL internal `reserved accounts`.

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.

References:

1. https://dev.mysql.com/doc/refman/5.7/en/information-schema-user-privileges-table.html
2. https://dev.mysql.com/doc/refman/5.7/en/reserved-accounts.html

Additional Information:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.	●	●	●
v7	4.1 Maintain Inventory of Administrative Accounts Use automated tools to inventory all administrative accounts, including domain and local accounts, to ensure that only authorized individuals have elevated privileges.		●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

5.2 Ensure 'FILE' is Not Granted to Non-Administrative Users (Manual)

Profile Applicability:

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

Description:

The `FILE` privilege is used to allow or disallow a user from reading and writing files on the server host. Any user with the `FILE` 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` 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 GRANTEE FROM INFORMATION_SCHEMA.USER_PRIVILEGES
WHERE PRIVILEGE_TYPE = 'FILE';
```

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. https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#priv_file
2. https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#privileges-provided-summary

Additional Information:

See also: `secure_file_priv` system settings.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

5.3 Ensure 'PROCESS' is Not Granted to Non-Administrative Users (Manual)

Profile Applicability:

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

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.

Impact:

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

Audit:

Execute the following SQL statement to audit this setting:

```
SELECT GRANTEE FROM INFORMATION_SCHEMA.USER_PRIVILEGES
WHERE PRIVILEGE_TYPE = 'PROCESS';
```

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>';
```

References:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

5.4 Ensure 'SUPER' is Not Granted to Non-Administrative Users (Manual)

Profile Applicability:

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

Description:

The `SUPER` privilege is a powerful and far-reaching privilege and should not be granted lightly.

The `SUPER` privilege shown in the `INFORMATION_SCHEMA.USER_PRIVILEGES` 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.

It is more secure to migrate administrative users off `SUPER` and instead assign the specific and minimal set of mysql Dynamic Privileges needed to perform their tasks.

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.

Audit:

Execute the following SQL statement to audit this setting:

```
SELECT GRANTEE FROM INFORMATION_SCHEMA.USER_PRIVILEGES
WHERE PRIVILEGE_TYPE = 'SUPER';
```

Ensure only administrative users are returned in the result set.

Remediation:

Perform the following steps to remediate this setting:

2. Enumerate the non-administrative users found in the result set of the audit procedure

3. For each user, issue the following SQL statement (replace `<user>` with the non-administrative user):

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

Next minimize administrator rights

1. Assess the minimal set of Dynamic Permissions needed by a user to perform their duties.
2. For each user assign the appropriate Dynamic Permission and then revoke that `<user>` `SUPER` capability.
   For example, if administrator `'u1'@'localhost'` requires `SUPER` for binary log purging and system variable modification, these statements make the required changes to the account thus limiting rights to what is needed:

```
GRANT BINLOG_ADMIN, SYSTEM_VARIABLES_ADMIN ON *.* TO
'u1'@'localhost';
REVOKE SUPER ON *.* FROM 'u1'@'localhost';
```

References:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

5.5 Ensure 'SHUTDOWN' is Not Granted to Non-Administrative Users (Manual)

Profile Applicability:

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

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 GRANTEE FROM INFORMATION_SCHEMA.USER_PRIVILEGES
WHERE PRIVILEGE_TYPE = 'SHUTDOWN';
```

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. https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#priv_shutdown
2. https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#privileges-provided-summary

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

5.6 Ensure 'CREATE USER' is Not Granted to Non-Administrative Users (Manual)

Profile Applicability:

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

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.

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.

Audit:

Execute the following SQL statement to audit this setting:

```
SELECT GRANTEE FROM INFORMATION_SCHEMA.USER_PRIVILEGES
WHERE PRIVILEGE_TYPE = 'CREATE USER';
```

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>';
```

References:

1. https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#priv_create-user

2. https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#privileges-provided-summary

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

5.7 Ensure 'GRANT OPTION' is Not Granted to Non-Administrative Users (Manual)

Profile Applicability:

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

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 OPTION` 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 GRANTEE FROM INFORMATION_SCHEMA.USER_PRIVILEGES
WHERE PRIVILEGE_TYPE = 'GRANT OPTION';
```

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. https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#priv_grant-option
2. https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#privileges-provided-summary

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

5.8 Ensure 'REPLICATION SLAVE' is Not Granted to Non-Administrative Users (Manual)

Profile Applicability:

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

Description:

The `REPLICATION SLAVE` privilege governs whether a given user (in the context of the source server) can request updates that have been made on the source 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 GRANTEE FROM INFORMATION_SCHEMA.USER_PRIVILEGES
WHERE PRIVILEGE_TYPE = 'REPLICATION SLAVE';
```

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

Remediation:

Perform the following steps to remediate this setting:

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

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

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

References:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

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

Profile Applicability:

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

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 are permitted to have these privileges on the indicated databases.

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>;
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

5.10 Securely Define Stored Procedures and Functions DEFINER and INVOKER (Manual)

Profile Applicability:

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

Description:

Stored procedure and stored function declarations include a definition of permissions which can be used to escalate permissions. It's important to inspect these settings to ensure they do not unnecessarily escalate privileges.

Rationale:

A stored procedure or function that improperly escalates privileges may provide unintended access rights which can be improperly used.

Audit:

Run the following:

```
SHOW PROCEDURE STATUS;
SHOW FUNCTION STATUS;
```

Inspect Definer and Invoker security types.

If DEFINER is a powerful user consider that user's permissions.

If INVOKER then the rights for the stored procedure or function are that of the user executing these.

Review code using

```
SHOW CREATE PROCEDURE <name>;
SHOW CREATE FUNCTION <name>;
```

For more details on Procedures and Functions

```
SELECT * FROM information_schema.routines;
```

For more details on Procedures and Functions input and output parameters.

```
SELECT * FROM information_schema.parameters;
```

Remediation:

Drop and recreate stored procedures and functions using proper DEFINER and INVOKER settings, or other code changes.

References:

1. https://dev.mysql.com/doc/refman/5.7/en/create-procedure.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.10 Apply Secure Design Principles in Application Architectures Apply secure design principles in application architectures. Secure design principles include the concept of least privilege and enforcing mediation to validate every operation that the user makes, promoting the concept of "never trust user input." Examples include ensuring that explicit error checking is performed and documented for all input, including for size, data type, and acceptable ranges or formats. Secure design also means minimizing the application infrastructure attack surface, such as turning off unprotected ports and services, removing unnecessary programs and files, and renaming or removing default accounts.		●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

6 Auditing and Logging

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

6.1 Ensure 'log_error' is configured correctly (Automated)

Profile Applicability:

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

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 can increase the ability to detect malicious attempts against MySQL, and other critical messages. For example, if the error log is not enabled then a connection error could go unnoticed.

When configured to `stderr` MySQL will send log data to the console. Logging to the console is useful, but remember it is ephemeral. This is not recommended due to the fact that logging to console does not provide a means to force restricted access via permissions strictly to MySQL and dedicated MySQL audit accounts. This may compromise the confidentiality of the MySQL log data. Furthermore use caution if co-mingling log data from multiple sources as that can complicate log inspection. Additionally from a security auditing perspective, it's difficult and error prone to verify logging is correct using `stderr` or redirected `stderr`.

Audit:

Execute the following SQL statement to audit this setting:

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

Ensure the `Value` returned is a path to a file and not `stderr`.

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.

Default Value:

`stderr`

References:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.	●	●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●

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

Profile Applicability:

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

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.3 Ensure Adequate Audit Log Storage Ensure that logging destinations maintain adequate storage to comply with the enterprise's audit log management process.		●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	6.4 Ensure adequate storage for logs Ensure that all systems that store logs have adequate storage space for the logs generated.		●	●

6.3 Ensure 'log_error_verbosity' is Set to '2' (Automated)

Profile Applicability:

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

Description:

The `log_error_verbosity` system variable, set to `2` by default, specifies the verbosity of events sent to the MySQL error log. A value of `2` enables logging of error and warning messages, a value of `3` also includes informational logging, a value of `1` logs only errors.

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`.

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
```

Default Value:

The option is enabled (`3`) - errors, warning, and information messages are logged - by default.

References:

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

Additional Information:

`log_warnings` has been deprecated as of MySQL 5.7.2. Setting `log_warnings` will also cause `log_error_verbosity` to be set.The variable scope for `log_warnings` is global.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●

6.4 Ensure 'log-raw' is Set to 'OFF' (Automated)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.5 Securely Dispose of Data Securely dispose of data as outlined in the enterprise's data management process. Ensure the disposal process and method are commensurate with the data sensitivity.	●	●	●
v7	13.2 Remove Sensitive Data or Systems Not Regularly Accessed by Organization Remove sensitive data or systems not regularly accessed by the organization from the network. These systems shall only be used as stand alone systems (disconnected from the network) by the business unit needing to occasionally use the system or completely virtualized and powered off until needed.	●	●	●

6.5 Ensure Audit Logging Is Enabled (Manual)

Profile Applicability:

- Level 2 - MySQL RDBMS on Linux
- 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. https://dev.mysql.com/doc/refman/5.7/en/query-log.html
2. https://dev.mysql.com/doc/refman/5.7/en/mysql-enterprise-audit.html
3. https://mariadb.com/kb/en/mariadb-audit-plugin/
4. https://github.com/mcafee/mysql-audit

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.	●	●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●

7 Authentication

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

7.1 Ensure default_authentication_plugin is Set to a Secure Option (Automated)

Profile Applicability:

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

Description:

The `-default-authentication-plugin` system variable governs two things:

- Authentication plugin used by a new user account if a plugin is not specified explicitly through `CREATE USER` statement
- Initial authentication data payload generated by server in case of a new connection.

Rationale:

MySQL Native Authentication relies on the Secure Hash Algorithm 1 (SHA1) algorithm and the National Institute of Standards and Technology (NIST) has suggested to stop using it.

The MySQL Native Authentication plugin leverages this weak hashing algorithm that can be quickly brute forced.

Audit:

Execute the following SQL statement to assess this recommendation:

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

Ensure the `Value` field is not set to `mysql_native_password`.

Remediation:

Configure mysql to default to the `sha256_password` plugin.

Require `sha256_password` plugin to be used by default for new accounts.

Edit `my.cnf`, in the section `[mysqld]` add:

```
default_authentication_plugin= sha256_password
```

Determine if any users are using `mysql_native_password`.

```
select host, user, plugin from mysql.user;
```

Migrate these users from `mysql_native_password`.

```
ALTER USER user
  IDENTIFIED WITH sha256_password IDENTIFIED BY RANDOM PASSWORD PASSWORD
EXPIRE;
```

Provide users the random password value through a secure mechanism - on next login they will be forced to change the password.

Default Value:

`mysql_native_password`

References:

1. https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html#sysvar_default_authentication_plugin
2. https://dev.mysql.com/doc/refman/5.7/en/authentication-plugins.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 Encrypt Sensitive Data at Rest Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●
v7	16.4 Encrypt or Hash all Authentication Credentials Encrypt or hash with a salt all authentication credentials when stored.		●	●

7.2 Ensure Passwords are Not Stored in the Global Configuration (Automated)

Profile Applicability:

- Level 1 - 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:

Using 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.

References:

1. https://dev.mysql.com/doc/refman/5.7/en/mysql-config-editor.html

Additional Information:

There must not be a password in any of the sections of the global configuration. 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.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 Encrypt Sensitive Data at Rest Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●
v7	16.4 Encrypt or Hash all Authentication Credentials Encrypt or hash with a salt all authentication credentials when stored.		●	●

7.3 Ensure 'sql_mode' Contains 'NO_AUTO_CREATE_USER' (Automated)

Profile Applicability:

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

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

Default Value:

```
ONLY_FULL_GROUP_BY STRICT_TRANS_TABLES NO_ZERO_IN_DATE NO_ZERO_DATE
ERROR_FOR_DIVISION_BY_ZERO NO_AUTO_CREATE_USER NO_ENGINE_SUBSTITUTION
```

References:

1. https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html#sysvar_sql_mode
2. https://dev.mysql.com/doc/refman/5.7/en/sql-mode.html#sqlmode_no_auto_create_user

Additional Information:

NO_AUTO_CREATE_USER is deprecated and it is included in the default SQL mode. The documentation for this mode states:

> Previously, before NO_AUTO_CREATE_USER was deprecated, one reason not to enable it was that it was not replication safe. Now it can be enabled and replication-safe user management performed with CREATE USER IF NOT EXISTS, DROP USER IF EXISTS, and ALTER USER IF EXISTS rather than GRANT. These statements enable safe replication when replicas may have different grants than those on the source.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.10 Apply Secure Design Principles in Application Architectures Apply secure design principles in application architectures. Secure design principles include the concept of least privilege and enforcing mediation to validate every operation that the user makes, promoting the concept of "never trust user input." Examples include ensuring that explicit error checking is performed and documented for all input, including for size, data type, and acceptable ranges or formats. Secure design also means minimizing the application infrastructure attack surface, such as turning off unprotected ports and services, removing unnecessary programs and files, and renaming or removing default accounts.		●	●

7.4 Ensure Passwords are Set for All MySQL Accounts (Automated)

Profile Applicability:

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

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:
For versions prior to 5.7.6:

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

For versions 5.7.6, or later:

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

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

Remediation:

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

```
ALTER USER
       <user>@<host> IDENTIFIED BY RANDOM PASSWORD PASSWORD EXPIRE;
```

This resets the password temporarily to a RANDOM string and returns that temporary password as a result.

The user can then use this temporary password to login and is forced to set the password to one of their choosing upon login.

Note: Replace *<user>*, *<host>* with appropriate values.

References:

1. https://dev.mysql.com/doc/refman/5.7/en/assigning-passwords.html
2. https://dev.mysql.com/doc/refman/5.7/en/upgrading-from-previous-series.html#upgrade-system-table-changes

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.2 Use Unique Passwords Use unique passwords for all enterprise assets. Best practice implementation includes, at a minimum, an 8-character password for accounts using MFA and a 14-character password for accounts not using MFA.	●	●	●
v7	4.4 Use Unique Passwords Where multi-factor authentication is not supported (such as local administrator, root, or service accounts), accounts will use passwords that are unique to that system.		●	●

7.5 Set 'default_password_lifetime' to Require a Yearly Password Change (Automated)

Profile Applicability:

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

Description:

Password expiry provides passwords with a time bounded lifetime.

Rationale:

The 'default_password_lifetime' global variable prevents a password being set for an indefinite period. Excessive password expiration requirements do more harm than good, because these requirements make users select predictable passwords, composed of sequential words and numbers that are closely related to each other. More importantly, when events occur that could compromise password security account passwords should be expired immediately.

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.

Audit:

Execute the following SQL statements to assess this recommendation:

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

The result should not be `0` and less than or equal to `365`.

Remediation:

To remediate this recommendation, execute the following command:

```
SET GLOBAL default_password_lifetime=365;
```

Default Value:

From 5.7.11 on: `0`

Prior to 5.7.11: `360`

References:

1. https://dev.mysql.com/doc/refman/5.7/en/password-management.html

2. https://dev.mysql.com/doc/refman/5.7/en/expired-password-handling.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.3 Disable Dormant Accounts Delete or disable any dormant accounts after a period of 45 days of inactivity, where supported.	●	●	●
v7	16.10 Ensure All Accounts Have An Expiration Date Ensure that all accounts have an expiration date that is monitored and enforced.		●	●

7.6 Ensure Password Complexity Policies are in Place (Automated)

Profile Applicability:

- Level 1 - MySQL RDBMS on Linux

Description:

Password complexity includes password characteristics such as length, case, numerical, 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.

Impact:

Remediation for this recommendation requires a server restart.

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_check_user_name` should be `ON`
- `validate_password_policy` should be `STRONG` - checks length; numeric, lowercase/uppercase, and special characters; dictionary file

New passwords should be checked against a dictionary file that contains values known to be commonly-used, expected, or compromised. For example, the list should include, but is not limited to:

- Passwords obtained from previous breaches
- Dictionary words
- Repetitive or sequential characters (e.g., `aaaaaa`, `1234abcd`)
- Context-specific words, such as the name of the service, the username, and derivatives thereof
- `validate_password.dictionary_file` should point to a dictionary file of common words used in passwords.

The following may make the password complexity too difficult, use sparingly.

- `validate_password_mixed_case_count` not more than `1`
- `validate_password_number_count` not more than `1`
- `validate_password_special_char_count` not more than `1`

The following lines should be present in the global configuration:

```
plugin-load=validate_password.so
validate-password=FORCE_PLUS_PERMANENT
```

Remediation:

Add to the global configuration:

```
plugin-load=validate_password.so
validate-password=FORCE_PLUS_PERMANENT
validate_password_length=14
validate_password_check_user_name=ON
validate_password_dictionary_file=<path to dictionary file>
validate_password_policy=STRONG
```

Optionally set one or more of these - ensuring complexity is not overly onerous

```
validate_password_mixed_case_count=1
validate_password_number_count=1
validate_password_special_char_count=1
```

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

Default Value:

Default `component_validate_password` is not installed.

```
validate_password_length=8
validate_password_mixed_case_count=1
validate_password_number_count=1
validate_password_policy=MEDIUM
validate_password_special_char_count=1
```

References:

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

Additional Information:

The system variable `validate_password_check_user_name` is exposed by `validate_password` starting with MySQL 5.7.15.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.2 Use Unique Passwords Use unique passwords for all enterprise assets. Best practice implementation includes, at a minimum, an 8-character password for accounts using MFA and a 14-character password for accounts not using MFA.	●	●	●
v7	4.4 Use Unique Passwords Where multi-factor authentication is not supported (such as local administrator, root, or service accounts), accounts will use passwords that are unique to that system.		●	●

7.7 Ensure No Users Have Wildcard Hostnames (Automated)

Profile Applicability:

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

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.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

7.8 Ensure No Anonymous Accounts Exist (Automated)

Profile Applicability:

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

Description:

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

Rationale:

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

Impact:

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

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.

References:

1. https://dev.mysql.com/doc/refman/5.7/en/mysql-secure-installation.html
2. https://dev.mysql.com/doc/refman/5.7/en/default-privileges.html
3. https://dev.mysql.com/doc/refman/5.7/en/proxy-users.html#proxy-users-conflicts

Additional Information:

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.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.1 Establish and Maintain an Inventory of Accounts Establish and maintain an inventory of all accounts managed in the enterprise. The inventory must include both user and administrator accounts. The inventory, at a minimum, should contain the person's name, username, start/stop dates, and department. Validate that all active accounts are authorized, on a recurring schedule at a minimum quarterly, or more frequently.	●	●	●
v7	16.6 Maintain an Inventory of Accounts Maintain an inventory of all accounts organized by authentication system.		●	●

8 Network

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

8.1 Ensure 'require_secure_transport' is Set to 'ON' and/or 'have_ssl' is Set to 'YES' (Automated)

Profile Applicability:

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

Description:

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

Rationale:

Enabling Secure Sockets Layer (SSL) will allow clients to encrypt network traffic and verify the identity of the server. The SSL/TLS-protected MySQL protocol helps to prevent eavesdropping and man-in-the-middle attacks.

Impact:

Enabling SSL could have impact on network traffic inspection.

Audit:

Execute the following SQL statements to assess this recommendation:
Check the global default.

```
select @@require_secure_transport;
```

Ensure the returned `Value` is `ON` or '1'

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

Or if MySQL is built with OpenSSL:

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

Ensure the `Value` returned is `YES`.

Note: `have_openssl` is an alias for `have_ssl` when MySQL is built with OpenSSL.

Remediation:

Follow the procedures as documented in the MySQL 5.7 Reference Manual to setup SSL.

Set global policy to force SSL for all connections:

```
set require_secure_transport=ON;
```

Default Value:

`DISABLED`

References:

1. https://dev.mysql.com/doc/refman/5.7/en/using-encrypted-connections.html
2. https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html#sysvar_require_secure_transport
3. https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html#sysvar_have_openssl

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 Encrypt Sensitive Data in Transit Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).		●	●
v7	14.4 Encrypt All Sensitive Information in Transit Encrypt all sensitive information in transit.		●	●

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

Profile Applicability:

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

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.

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.

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 `X509`, or `SPECIFIED`.

Note: `ANY` means the connection must be using TLS and could optionally provide a client-side certificate.

Remediation:

Use the ALTER USER statement to require the use of SSL:

```
ALTER USER 'my_user'@'app1.example.com' REQUIRE X509;
```

Note: `REQUIRE SSL` only enforces SSL. There are additional options `REQUIRE ISSUER`, `REQUIRE SUBJECT` which can be used to further restrict the connection.

Default Value:

On the server-side SSL is `ON` by default `--ssl` (permits but does not require secure connections) and `require_secure_transport` is `OFF` (turning `ON` allows only secure connections)

References:

1. https://dev.mysql.com/doc/refman/5.7/en/using-encrypted-connections.html
2. https://dev.mysql.com/doc/refman/5.7/en/alter-user.html#alter-user-tls
3. https://dev.mysql.com/doc/refman/5.7/en/connection-options.html#option_general_ssl

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 Encrypt Sensitive Data in Transit Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).		●	●
v7	14.4 Encrypt All Sensitive Information in Transit Encrypt all sensitive information in transit.		●	●

8.3 Set Maximum Connection Limits for Server and per User (Manual)

Profile Applicability:

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

Description:

Limiting concurrent connections to a MySQL server can be used to reduce risk of Denial of Service (DoS) attacks performed by exhausting connection resources.

Rationale:

Limiting the number of concurrent sessions at the server and per user level helps to reduce the risk of DoS attacks. MySQL provides mechanisms to limit the number of simultaneous connections that can be made at the server level or by any given account.

Audit:

To check global (default) concurrent-sessions settings in the MySQL database server, to check the per user default run the query:

```
SELECT VARIABLE_NAME, VARIABLE_VALUE
FROM performance_schema.global_variables
WHERE VARIABLE_NAME LIKE 'max_%connections';
```

If the value of `max_user_connections` is `0` this means there is “no limit”.

If the value of `max_connections` is not set, there is no limit.

Also check the values on a per user basis run the following:

```
select user, host, max_user_connections from mysql.user where user not like
'mysql.%' and user not like 'root';
```

If the value is `0` this means the global value of `max_user_connections` applies.

If no limits are configured this is a fail.

Remediation:

Connect to the MySQL Database as an administrator.

For example, to set the global default per user to `50` run the command:

```
SET PERSIST max_user_connections=50;
```

To control the maximum number of clients the server permits to connect simultaneously, set the `max_connections` system variable:

```
SET PERSIST max_connections=1000;
```

Additionally, this max user connections can be set per user as well as for a given period of time period using `CREATE` or `ALTER`.

For example:

```
ALTER USER 'fred'@'localhost'
WITH MAX_CONNECTIONS_PER_HOUR 5
MAX_USER_CONNECTIONS 2;
```

Default Value:

The default value of `max_connections` is `151`, `max_user_connections` is `0` (unlimited, thus limited by `max_connections`).

References:

1. https://dev.mysql.com/doc/refman/5.7/en/user-resources.html
2. https://dev.mysql.com/doc/refman/5.7/en/connection-interfaces.html#connection-interfaces-volume-management

9 Replication

Everything related to replicating data from one server to another.

9.1 Ensure Replication Traffic is Secured (Manual)

Profile Applicability:

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

Description:

The replication traffic between servers should be secured. Security measures should include ensuring the confidentiality and integrity of the traffic, and performing mutual authentication between the servers before performing replication.

Rationale:

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

Impact:

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

Audit:

Check if the replication traffic is using one or more of the following to provide confidentiality and integrity for the traffic, and mutual authentication for the servers:

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

Remediation:

Secure the network traffic using one or more technologies to provide confidentiality and integrity for the traffic, and mutual authentication for the servers.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 Encrypt Sensitive Data in Transit Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).		●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	14.4 Encrypt All Sensitive Information in Transit Encrypt all sensitive information in transit.		●	●

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

Profile Applicability:

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

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.

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 in the session where the `CHANGE MASTER to` was run, but not all ongoing transactions on the database.

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
```

References:

1. https://dev.mysql.com/doc/refman/5.7/en/change-master-to.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	13.9 Deploy Port-Level Access Control Deploy port-level access control. Port-level access control utilizes 802.1x, or similar network access control protocols, such as certificates, and may incorporate user and/or device authentication.			●

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

Profile Applicability:

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

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. https://dev.mysql.com/doc/refman/5.7/en/replication-options-replica.html#sysvar_master_info_repository

9.4 Ensure 'super_priv' is Not Set to 'Y' for Replication Users (Automated)

Profile Applicability:

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

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.

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.

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';
```

References:

1. https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#priv_super
2. https://dev.mysql.com/doc/refman/5.7/en/show-slave-status.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.	●	●	●
v7	4.7 Limit Access to Script Tools Limit access to scripting tools (such as Microsoft PowerShell and Python) to only administrative or development users with the need to access those capabilities.		●	●

9.5 Ensure No Replication Users Have Wildcard Hostnames (Automated)

Profile Applicability:

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

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.1 Establish and Maintain an Inventory of Accounts Establish and maintain an inventory of all accounts managed in the enterprise. The inventory must include both user and administrator accounts. The inventory, at a minimum, should contain the person's name, username, start/stop dates, and department. Validate that all active accounts are authorized, on a recurring schedule at a minimum quarterly, or more frequently.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

Appendix: Summary Table

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
1	**Operating System Level Configuration**		
1.1	Place Databases on Non-System Partitions (Manual)	☐	☐
1.2	Use Dedicated Least Privileged Account for MySQL Daemon/Service (Automated)	☐	☐
1.3	Disable MySQL Command History (Automated)	☐	☐
1.4	Verify That the MYSQL_PWD Environment Variable Is Not In Use (Automated)	☐	☐
1.5	Ensure Interactive Login is Disabled (Automated)	☐	☐
1.6	Verify That 'MYSQL_PWD' is Not Set in Users' Profiles (Automated)	☐	☐
2	**Installation and Planning**		
2.1	**Backup and Disaster Recovery**		
2.1.1	Backup Policy in Place (Manual)	☐	☐
2.1.2	Verify Backups are Good (Manual)	☐	☐
2.1.3	Secure Backup Credentials (Manual)	☐	☐
2.1.4	The Backups Should be Properly Secured (Manual)	☐	☐
2.1.5	Point-in-Time Recovery (Manual)	☐	☐
2.1.6	Disaster Recovery (DR) Plan (Manual)	☐	☐
2.1.7	Backup of Configuration and Related Files (Manual)	☐	☐
2.2	Dedicate the Machine Running MySQL (Manual)	☐	☐
2.3	Do Not Specify Passwords in Command Line (Manual)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
2.4	Do Not Reuse Usernames (Manual)	☐	☐
2.5	Ensure Non-Default, Unique Cryptographic Material is in Use (Manual)	☐	☐
2.6	Ensure 'password_lifetime' is Less Than or Equal to '365' (Automated)	☐	☐
2.7	Ensure Password Complexity is Configured (Automated)	☐	☐
2.8	Lock Out Accounts if Not Currently in Use (Manual)	☐	☐
2.9	Ensure AES Encryption Mode for AES_ENCRYPT/AES_DECRYPT is Configured Correctly (Automated)	☐	☐
2.10	Ensure Socket Peer-Credential Authentication is Used Appropriately (Manual)	☐	☐
2.11	Ensure MySQL is Bound to an IP Address (Automated)	☐	☐
2.12	Limit Accepted Transport Layer Security (TLS) Versions (Automated)	☐	☐
2.13	Require Client-Side Certificates (X.509) (Automated)	☐	☐
2.14	Ensure Only Approved Ciphers are Used (Automated)	☐	☐
2.15	Implement Connection Delays to Limit Failed Login Attempts (Automated)	☐	☐
3	**File Permissions**		
3.1	Ensure 'datadir' Has Appropriate Permissions (Automated)	☐	☐
3.2	Ensure 'log_bin_basename' Files Have Appropriate Permissions (Automated)	☐	☐
3.3	Ensure 'log_error' Has Appropriate Permissions (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
3.4	Ensure 'slow_query_log' Has Appropriate Permissions (Automated)	☐	☐
3.5	Ensure 'relay_log_basename' Files Have Appropriate Permissions (Automated)	☐	☐
3.6	Ensure 'general_log_file' Has Appropriate Permissions (Automated)	☐	☐
3.7	Ensure SSL Key Files Have Appropriate Permissions (Automated)	☐	☐
3.8	Ensure Plugin Directory Has Appropriate Permissions (Automated)	☐	☐
3.9	Secure MySQL Keyring (Automated)	☐	☐
4	**General**		
4.1	Ensure Latest Security Patches Are Applied (Manual)	☐	☐
4.2	Ensure Example or Test Databases are Not Installed on Production Servers (Automated)	☐	☐
4.3	Ensure 'allow-suspicious-udfs' Is Set to 'OFF' (Manual)	☐	☐
4.4	Harden Usage for 'local_infile' on MySQL Clients (Automated)	☐	☐
4.5	Ensure 'mysqld' is Not Started with '--skip-grant-tables' (Manual)	☐	☐
4.6	Ensure Symbolic Links are Disabled (Automated)	☐	☐
4.7	Ensure the 'daemon_memcached' Plugin Is Disabled (Automated)	☐	☐
4.8	Ensure the 'secure_file_priv' is Configured Correctly (Automated)	☐	☐
4.9	Ensure 'sql_mode' Contains 'STRICT_ALL_TABLES' (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
5	**MySQL Permissions**		
5.1	Ensure Only Administrative Users Have Full Database Access (Manual)	☐	☐
5.2	Ensure 'FILE' is Not Granted to Non-Administrative Users (Manual)	☐	☐
5.3	Ensure 'PROCESS' is Not Granted to Non-Administrative Users (Manual)	☐	☐
5.4	Ensure 'SUPER' is Not Granted to Non-Administrative Users (Manual)	☐	☐
5.5	Ensure 'SHUTDOWN' is Not Granted to Non-Administrative Users (Manual)	☐	☐
5.6	Ensure 'CREATE USER' is Not Granted to Non-Administrative Users (Manual)	☐	☐
5.7	Ensure 'GRANT OPTION' is Not Granted to Non-Administrative Users (Manual)	☐	☐
5.8	Ensure 'REPLICATION SLAVE' is Not Granted to Non-Administrative Users (Manual)	☐	☐
5.9	Ensure DML/DDL Grants Are Limited to Specific Databases and Users (Manual)	☐	☐
5.10	Securely Define Stored Procedures and Functions DEFINER and INVOKER (Manual)	☐	☐
6	**Auditing and Logging**		
6.1	Ensure 'log_error' is configured correctly (Automated)	☐	☐
6.2	Ensure Log Files Are Stored on a Non-System Partition (Automated)	☐	☐
6.3	Ensure 'log_error_verbosity' is Set to '2' (Automated)	☐	☐
6.4	Ensure 'log-raw' is Set to 'OFF' (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
6.5	Ensure Audit Logging Is Enabled (Manual)	☐	☐
7	**Authentication**		
7.1	Ensure default_authentication_plugin is Set to a Secure Option (Automated)	☐	☐
7.2	Ensure Passwords are Not Stored in the Global Configuration (Automated)	☐	☐
7.3	Ensure 'sql_mode' Contains 'NO_AUTO_CREATE_USER' (Automated)	☐	☐
7.4	Ensure Passwords are Set for All MySQL Accounts (Automated)	☐	☐
7.5	Set 'default_password_lifetime' to Require a Yearly Password Change (Automated)	☐	☐
7.6	Ensure Password Complexity Policies are in Place (Automated)	☐	☐
7.7	Ensure No Users Have Wildcard Hostnames (Automated)	☐	☐
7.8	Ensure No Anonymous Accounts Exist (Automated)	☐	☐
8	**Network**		
8.1	Ensure 'require_secure_transport' is Set to 'ON' and/or 'have_ssl' is Set to 'YES' (Automated)	☐	☐
8.2	Ensure 'ssl_type' is Set to 'ANY', 'X509', or 'SPECIFIED' for All Remote Users (Automated)	☐	☐
8.3	Set Maximum Connection Limits for Server and per User (Manual)	☐	☐
9	**Replication**		
9.1	Ensure Replication Traffic is Secured (Manual)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
9.2	Ensure 'MASTER_SSL_VERIFY_SERVER_CERT' Is Set to 'YES' or '1' (Automated)	☐	☐
9.3	Ensure 'master_info_repository' Is Set to 'TABLE' (Automated)	☐	☐
9.4	Ensure 'super_priv' is Not Set to 'Y' for Replication Users (Automated)	☐	☐
9.5	Ensure No Replication Users Have Wildcard Hostnames (Automated)	☐	☐

Appendix: Change History

Date	Version	Changes for this version
Oct 9, 2018	1.1.0	_Listing Order, Status, Audit Procedure_ on **[recommendation] 3.9 Ensure 'mysql-keyring' Has Appropriate Permissions** were updated.
Sep 21, 2021	2.0.0	Revise audit query to pull from performance_schema.global_variables (Ticket 13795)
Sep 21, 2021	2.0.0	Add procedures for removing shell access to the remediation. (Ticket 13797)
Sep 21, 2021	2.0.0	Add procedures for removing shell access to the remediation. (Ticket 13796)
Sep 21, 2021	2.0.0	Add MySQL Shell procedures to the audit procedures (Ticket 13799)
Sep 22, 2021	2.0.0	Add considerations for unattended logins (Ticket 13809)
Sep 28, 2021	2.0.0	Move impact statement into rationale (Ticket 13836)
Sep 28, 2021	2.0.0	Move the impact statement for verify backups are good to the rationale. (Ticket 13837)
Sep 28, 2021	2.0.0	Move the rationale and impact statement for secure backup credentials (Ticket 13838)
Sep 28, 2021	2.0.0	Update Point-in-Time Recovery to mirror changes in Oracle MySQL EE 5.7. (Ticket 13840)

Date	Version	Changes for this version
Sep 28, 2021	2.0.0	Move the impact statement for Disaster Recovery (DR) Plan to the rationale and update discussion/rationale to match MySQL EE 5.7. (Ticket 13841)
Sep 30, 2021	2.0.0	Move the impact statement for backups should be properly secured to the rationale (Ticket 13839)
Sep 30, 2021	2.0.0	Remove the impact statement for backup of configuration & related files and revise to incorporate changes made for MySQL EE 5.7. (Ticket 13842)
Sep 30, 2021	2.0.0	Move the impact statement for do not reuse usernames and and revise the audit procedures to reflect changes made for MySQL EE 5.7. (Ticket 13876)
Sep 30, 2021	2.0.0	Move the impact statement for ensure non-default, unique cryptographic material is in use and revise to incorporate changes made for MySQL EE 5.7. (Ticket 13880)
Oct 5, 2021	2.0.0	Update ensure 'password_lifetime' is less than or equal to '365' to incorporate new guidance from MySQL EE 5.7. (Ticket 13818)
Oct 14, 2021	2.0.0	Update ensure 'datadir' has appropriate permissions to include changes from MySQL EE 5.7 (Ticket 13980)
Oct 14, 2021	2.0.0	Update ensure 'log_bin_basename' files have appropriate permissions to include changes from MySQL EE 5.7. (Ticket 13981)

Date	Version	Changes for this version
Oct 28, 2021	2.0.0	Update ensure 'log_error' has appropriate permissions to include changes from MySQL EE 5.7. (Ticket 14040)
Nov 1, 2021	2.0.0	Update ensure 'slow_query_log' has appropriate permissions to include changes from MySQL EE 5.7. (Ticket 14056)
Nov 1, 2021	2.0.0	Update ensure 'general_log_file' has appropriate permissions to include changes from MySQL EE 5.7. (Ticket 14060)
Nov 2, 2021	2.0.0	Update ensure SSL Key Files have appropriate permissions to include changes from MySQL EE 5.7. (Ticket 14065)
Nov 2, 2021	2.0.0	Update ensure Plugin Directory has appropriate permissions to include changes from MySQL EE 5.7. (Ticket 14067)
Nov 3, 2021	2.0.0	Update ensure the 'test' database is no installed to include changes from MySQL EE 5.7. (Ticket 14086)
Dec 7, 2021	2.0.0	#280 Add 'Ensure keyring file has appropriate permissions' (Ticket 4644)
Dec 7, 2021	2.0.0	#278 Update based on 5.7.15 (Ticket 4643)
Dec 7, 2021	2.0.0	Oracle MySql Community Server 5.7 benchmark is frozen and reports false positives on Ubuntu 18.04 (Ticket 10797)
Dec 7, 2021	2.0.0	the scan uses 100% CPU and runs for ever (Ticket 9690)

Date	Version	Changes for this version
Dec 7, 2021	2.0.0	Add recommendation to ensure global password complexity is configured. (Ticket 13817)
Dec 7, 2021	2.0.0	Add recommendation to ensure accounts are locked out if not currently in use. (Ticket 13819)
Dec 7, 2021	2.0.0	Add recommendation to ensure AES Encryption mode for AES_ENCRYPT/AES_DECRYPT is configured correctly. (Ticket 13820)
Dec 7, 2021	2.0.0	Add a recommendation to ensure Socket Peer-Credential Authentication is used appropriately. (Ticket 13821)
Dec 7, 2021	2.0.0	Add recommendation to ensure MySQL is bound to an IP address. (Ticket 13822)
Dec 8, 2021	2.0.0	Add a recommendation to limit accepted TLS versions. (Ticket 13823)
Dec 8, 2021	2.0.0	Add a recommendation to require client-side certificates (X.509). (Ticket 13824)
Dec 8, 2021	2.0.0	Add a recommendation to ensure only approved ciphers are used. (Ticket 13834)
Dec 8, 2021	2.0.0	Add a recommendation to ensure connection delays are implemented to limit failed login attempts. (Ticket 13833)
Dec 8, 2021	2.0.0	Revise the recommendation for 'mysql-keyring' permissions to match the updated guidance for MySQL EE 5.7. (Ticket 13921)
Dec 8, 2021	2.0.0	Update the 'allow-suspicious-udfs' recommendation to match the MySQL documentation. (Ticket 14090)

Date	Version	Changes for this version
Dec 8, 2021	2.0.0	Add a recommendation to ensure stored procedures 'DEFINER' and 'INVOKER' are securely defined. (Ticket 14263)
Dec 8, 2021	2.0.0	Revise ensure only Administrative users have full Db access to match the updated guidance for MySQL EE 5.7. (Ticket 14110)
Dec 8, 2021	2.0.0	Revise ensure 'file_priv' is not set to 'Y' for non-administrative users to match the updated guidance for MySQL EE 5.7. (Ticket 14111)
Dec 8, 2021	2.0.0	Revise ensure 'process_priv' is not set to 'Y' for non-administrative users to match the updated guidance for MySQL EE 5.7. (Ticket 14113)
Dec 10, 2021	2.0.0	Revise ensure 'super_priv' is not set to 'Y' for non-administrative users to match the updated guidance for MySQL EE 5.7. (Ticket 14117)
Dec 10, 2021	2.0.0	Revise ensure 'shutdown_priv' is not set to 'Y' for non-administrative users to match the updated guidance for MySQL EE 5.7. (Ticket 14118)
Dec 10, 2021	2.0.0	Revise tensure 'create_user_priv' is not set to 'Y' for non-administrative users to match the updated guidance for MySQL EE 5.7. (Ticket 14119)
Dec 10, 2021	2.0.0	Revise the recommendation for the 'grant_priv' privilege to match the updated guidance for MySQL EE 5.7. (Ticket 14122)
Dec 10, 2021	2.0.0	Revise the recommendation for the 'repl_slave_priv' privilege to match the updated guidance for MySQL EE 5.7. (Ticket 14123)

Date	Version	Changes for this version
Dec 14, 2021	2.0.0	Revise the ensure 'log_error_verbosity' is not set to '1' to match the new guidance in the MySQL EE 5.7 benchmark. (Ticket 14148)
Dec 14, 2021	2.0.0	Add a recommendation to ensure 'default_authentication_plugin' is configured securely. (Ticket 14151)
Dec 14, 2021	2.0.0	Remove the recommendation for the deprecated setting 'old_passwords' (Ticket 14152)
Dec 14, 2021	2.0.0	Update recommendation to ensure all accounts have passwords to reflect changes made in 5.7.6. (Ticket 14154)
Dec 14, 2021	2.0.0	Revise the recommendation for the 'have_ssl' to match the updated guidance for MySQL EE 5.7. (Ticket 14228)
Dec 14, 2021	2.0.0	Revise ensure 'default_password_lifetime' from less than or equal to 90, to at least yearly. (Ticket 14162)
Feb 9, 2022	2.0.0	Update ensure 'relay_log_basename' has appropriate permissions to include changes from MySQL EE 5.7. (Ticket 14058)
Feb 10, 2022	2.0.0	Revise the recommendation for 'local_infile' to match the updated guidance for MySQL EE 5.7. (Ticket 14091)
Feb 10, 2022	2.0.0	Update artifact for ensure 'mysqld' is not started with '--skip-grant-tables' to match MySQL EE 5.7. (Ticket 14094)

Date	Version	Changes for this version
Feb 10, 2022	2.0.0	Revise ensure the 'secure_file_priv' is Configured Correctly to state that setting this to an empty string is a fail. (Ticket 14095)
Feb 10, 2022	2.0.0	Update ensure 'sql_mode' Contains 'STRICT_ALL_TABLES' to match MySQL EE 5.7. (Ticket 14828)
Feb 14, 2022	2.0.0	Revise ensure 'log_error' is configured correctly to incorporate changes from MySQL EE 5.7. (Ticket 14150)
Feb 16, 2022	2.0.0	Update ensure password complexity policy is in place to match the updated guidance for MySQL EE 5.7. (Ticket 14207)
Feb 16, 2022	2.0.0	Add a recommendation to set maximum connections limits for server and per user. (Ticket 14209)
Mar 3, 2022	2.0.0	Remove the recommendation for the deprecated setting 'secure_auth' (Ticket 14153)