

CIS MariaDB 10.6 Benchmark

v1.1.0 - 03-29-2024

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

Terms of Use	1
Table of Contents	2
Overview	5
Intended Audience.....	5
Consensus Guidance	6
Typographical Conventions.....	7
Recommendation Definitions.....	8
Title.....	8
Assessment Status.....	8
Automated	8
Manual.....	8
Profile	8
Description.....	8
Rationale Statement	8
Impact Statement.....	9
Audit Procedure.....	9
Remediation Procedure.....	9
Default Value.....	9
References	9
CIS Critical Security Controls® (CIS Controls®).....	9
Additional Information.....	9
Profile Definitions	10
Acknowledgements	12
Recommendations	13
1 Operating System Level Configuration	13
1.1 Place Databases on Non-System Partitions (Manual)	14
1.2 Use Dedicated Least Privileged Account for MariaDB Daemon/Service (Automated)	17
1.3 Disable MariaDB Command History (Automated)	19
1.4 Verify That the MYSQL_PWD Environment Variable is Not in Use (Automated)	21
1.5 Ensure Interactive Login is Disabled (Automated)	23
1.6 Verify That 'MYSQL_PWD' is Not Set in Users' Profiles (Automated)	25
1.7 Ensure MariaDB is Run Under a Sandbox Environment (Manual)	27
2 Installation and Planning	31
2.1 Backup and Disaster Recovery	32
2.1.1 Backup Policy in Place (Manual)	33
2.1.2 Verify Backups are Good (Manual).....	34
2.1.3 Secure Backup Credentials (Manual)	35

2.1.4 The Backups Should be Properly Secured (Manual)	36
2.1.5 Point-in-Time Recovery (Automated)	38
2.1.6 Disaster Recovery (DR) Plan (Manual)	40
2.1.7 Backup of Configuration and Related Files (Manual)	42
2.2 Dedicate the Machine Running MariaDB (Manual)	43
2.3 Do Not Specify Passwords in the Command Line (Manual)	45
2.4 Do Not Reuse Usernames (Manual).....	47
2.5 Ensure Non-Default, Unique Cryptographic Material is in Use (Manual)	49
2.6 Ensure 'password_lifetime' is Less Than or Equal to '365' (Automated)	50
2.7 Lock Out Accounts if Not Currently in Use (Manual)	53
2.8 Ensure Socket Peer-Credential Authentication is Used Appropriately (Manual)	55
2.9 Ensure MariaDB is Bound to an IP Address (Automated).....	58
2.10 Limit Accepted Transport Layer Security (TLS) Versions (Automated).....	60
2.11 Require Client-Side Certificates (X.509) (Automated).....	62
2.12 Ensure Only Approved Ciphers are Used (Automated).....	64
3 File Permissions	66
3.1 Ensure 'datadir' Has Appropriate Permissions (Automated)	67
3.2 Ensure 'log_bin_basename' Files Have Appropriate Permissions (Automated)	69
3.3 Ensure 'log_error' Has Appropriate Permissions (Automated)	71
3.4 Ensure 'slow_query_log' Has Appropriate Permissions (Automated)	73
3.5 Ensure 'relay_log_basename' Files Have Appropriate Permissions (Automated)	75
3.6 Ensure 'general_log_file' Has Appropriate Permissions (Automated).....	77
3.7 Ensure SSL Key Files Have Appropriate Permissions (Automated)	79
3.8 Ensure Plugin Directory Has Appropriate Permissions (Automated)	81
3.9 Ensure 'server_audit_file_path' Has Appropriate Permissions (Automated).....	83
3.10 Ensure File Key Management Encryption Plugin files have appropriate permissions (Automated)	85
4 General	87
4.1 Ensure the Latest Security Patches are Applied (Manual)	88
4.2 Ensure Example or Test Databases are Not Installed on Production Servers (Automated)	90
4.3 Ensure 'allow-suspicious-udfs' is Set to 'OFF' (Automated)	92
4.4 Harden Usage for 'local_infile' on MariaDB Clients (Automated)	94
4.5 Ensure mariadb is Not Started With 'skip-grant-tables' (Automated)	96
4.6 Ensure Symbolic Links are Disabled (Automated)	98
4.7 Ensure the 'secure_file_priv' is Configured Correctly (Automated)	100
4.8 Ensure 'sql_mode' Contains 'STRICT_ALL_TABLES' (Automated)	102
4.9 Enable data-at-rest encryption in MariaDB (Automated).....	104
5 MariaDB Permissions	107
5.1 Ensure Only Administrative Users Have Full Database Access (Manual)	108
5.2 Ensure 'FILE' is Not Granted to Non-Administrative Users (Manual).....	110
5.3 Ensure 'PROCESS' is Not Granted to Non-Administrative Users (Manual).....	112
5.4 Ensure 'SUPER' is Not Granted to Non-Administrative Users (Manual)	114
5.5 Ensure 'SHUTDOWN' is Not Granted to Non-Administrative Users (Manual)	116
5.6 Ensure 'CREATE USER' is Not Granted to Non-Administrative Users (Manual).....	118
5.7 Ensure 'GRANT OPTION' is Not Granted to Non-Administrative Users (Manual)	120
5.8 Ensure 'REPLICATION SLAVE' is Not Granted to Non-Administrative Users (Manual)..	122
5.9 Ensure DML/DDL Grants are Limited to Specific Databases and Users (Manual)	124
5.10 Securely Define Stored Procedures and Functions DEFINER and INVOKER (Manual) 126	
6 Auditing and Logging.....	128
6.1 Ensure 'log_error' is configured correctly (Automated).....	129
6.2 Ensure Log Files are Stored on a Non-System Partition (Automated).....	131
6.3 Ensure 'log_warnings' is Set to '2' (Automated)	133

6.4 Ensure Audit Logging Is Enabled (Automated)	135
6.5 Ensure the Audit Plugin Can't be Unloaded (Automated)	137
6.6 Ensure Binary and Relay Logs are Encrypted (Automated)	139
7 Authentication.....	141
7.1 Disable use of the mysql_old_password plugin (Automated)	142
7.2 Ensure Passwords are Not Stored in the Global Configuration (Automated)	144
7.3 Ensure strong authentication is utilized for all accounts (Automated)	146
7.4 Ensure Password Complexity Policies are in Place (Automated)	149
7.5 Ensure No Users Have Wildcard Hostnames (Automated)	152
7.6 Ensure No Anonymous Accounts Exist (Automated)	153
8 Network	155
8.1 Ensure 'require_secure_transport' is Set to 'ON' and 'have_ssl' is Set to 'YES' (Automated)	156
8.2 Ensure 'ssl_type' is Set to 'ANY', 'X509', or 'SPECIFIED' for All Remote Users (Automated)	158
8.3 Set Maximum Connection Limits for Server and per User (Manual)	160
9 Replication	163
9.1 Ensure Replication Traffic is Secured (Manual)	164
9.2 Ensure 'MASTER_SSL_VERIFY_SERVER_CERT' is enabled (Automated)	166
9.3 Ensure 'super_priv' is Not Set to 'Y' for Replication Users (Automated)	168
9.4 Ensure only approved ciphers are used for Replication (Manual)	171
9.5 Ensure mutual TLS is enabled (Manual)	173
Appendix: Summary Table	175
Appendix: Change History	181

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 document, CIS MariaDB 10.6 Benchmark, provides prescriptive guidance for establishing a secure configuration posture for MariaDB 10.6. This guide was tested against MariaDB 10.6 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 MariaDB 10.6.

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
<code>Stylized Monospace font</code>	Used for blocks of code, command, and script examples. Text should be interpreted exactly as presented.
<code>Monospace font</code>	Used for inline code, commands, or examples. Text should be interpreted exactly as presented.
<i><italic font in brackets></i>	Italic texts set in angle brackets denote a variable requiring substitution for a real value.
<i>Italic font</i>	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 - MariaDB RDBMS on Linux**

Items in this profile apply to MariaDB 10.6 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 - MariaDB RDBMS on Linux**

This profile extends the "Level 1 - MariaDB RDBMS on Linux" profile. Items in this profile apply to MariaDB 10.6 running on Linux and exhibit one or more of the following characteristics:

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

- **Level 1 - MariaDB RDBMS**

Items in this profile apply to MariaDB 10.6 and intend to:

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

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

- **Level 2 - MariaDB RDBMS**

This profile extends the "Level 1 - MariaDB RDBMS" profile. Items in this profile apply to MariaDB 10.6 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 MariaDB 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

Greg MacLean

David Fuente

Pradeep Frederick

Editor

Tim Harrison CISSP, ICP, KMP, Center for Internet Security, New York

Krishna Rayavaram

Recommendations

1 Operating System Level Configuration

This section contains recommendations related to the Operating System on which MariaDB is running.

1.1 Place Databases on Non-System Partitions (Manual)

Profile Applicability:

- Level 1 - MariaDB 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 MariaDB database files by executing the following SQL statement:

```
SELECT VARIABLE_NAME, VARIABLE_VALUE
FROM information_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 MariaDB data.
3. Stop `mariadb` using a command like: `service mariadb 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 MariaDB configuration file.
6. Start `mariadb` using a command like:

```
service mariadb 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.mariadb` 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.mysql` file should include something like this:

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




Default Value:

Not Applicable.

References:

1. <https://mariadb.com/kb/en/specifying-permissions-for-schema-data-directories-and-tables/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>3.12 Segment Data Processing and Storage Based on Sensitivity</u> 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	<u>2.10 Physically or Logically Segregate High Risk Applications</u> 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 MariaDB Daemon/Service (Automated)

Profile Applicability:

- Level 1 - MariaDB 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 MariaDB to execute as needed may reduce the impact of a MariaDB-born vulnerability. A restricted account will be unable to access resources unrelated to MariaDB, 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 MariaDB user is `mysql`. Additionally, you may consider running `sudo -l` as the MariaDB user or to check the sudoers file.

Remediation:

Create a user which is only used for running MariaDB 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 "MariaDB Server" -u 27 mysql >/dev/null 2>&1 || :
```







References:

1. <https://mariadb.com/kb/en/running-mysqld-as-root/>

Additional Information:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> 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 <u>Protect Information through Access Control Lists</u> 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.			

1.3 Disable MariaDB Command History (Automated)

Profile Applicability:

- Level 2 - MariaDB RDBMS on Linux

Description:

On Unix, the mysql client writes a record of executed statements to a history file. By default, this file is named `.mysql_history` and is created in your home directory. To specify a different file, set the value of the `MYSQL_HISTFILE` environment variable.

The `.mysql_history` file should be protected with a restrictive access mode because sensitive information might be written to it, such as the text of SQL statements that contain passwords.

Rationale:

Disabling the MariaDB Client and MariaDB 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 each file returned determine whether that file is symbolically linked to `/dev/null`.

Remediation:

For MariaDB Client 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 `.mysql_history` as a symbolic to `/dev/null`.

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







Default Value:

By default, the MariaDB command history file is located in `/root/.mysql_history`.

References:

1. https://mariadb.com/kb/en/mysql-command-line-client/#the-mysql_history-file
2. <https://mariadb.com/kb/en/mysqld-options/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.5 <u>Securely Dispose of Data</u> 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 <u>Remove Sensitive Data or Systems Not Regularly Accessed by Organization</u> 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 - MariaDB RDBMS on Linux

Description:

MariaDB 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 MariaDB credentials.

Rationale:

Using the `MYSQL_PWD` environment variable implies MariaDB 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

2) Different authentication methods (e.g., X509 certificate verification)





Default Value:

Not set.

References:

1. <https://mariadb.com/kb/en/mariadb-environment-variables/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 <u>Encrypt Sensitive Data at Rest</u> 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 <u>Encrypt or Hash all Authentication Credentials</u> Encrypt or hash with a salt all authentication credentials when stored.			

1.5 Ensure Interactive Login is Disabled (Automated)

Profile Applicability:

- Level 2 - MariaDB RDBMS on Linux

Description:

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

Rationale:

Preventing the MariaDB user from logging in interactively may reduce the impact of a compromised MariaDB account. There is also more accountability, as accessing the operating system where the MariaDB server lies will require the user's own account. Interactive access by the MariaDB user is unnecessary and should be disabled.

Impact:

This setting will prevent the MariaDB administrator from interactively logging into the operating system using the MariaDB 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

```

References:

1. <https://mariadb.com/kb/en/mysql-command-line-client/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts</u> 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	<u>4.3 Ensure the Use of Dedicated Administrative Accounts</u> 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 - MariaDB RDBMS on Linux

Description:

MariaDB can read a default database password from an environment variable called `MYSQL_PWD`.

Rationale:

Use of the `MYSQL_PWD` environment variable implies MariaDB credentials are stored as clear text. Avoiding the use of this environment variable may increase assurance that the confidentiality of MariaDB credentials is preserved.

Audit:

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

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

Remediation:

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



Default Value:



Not set.

References:

1. <https://mariadb.com/kb/en/mariadb-environment-variables/>

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.			

Controls Version	Control	IG 1	IG 2	IG 3
v7	16.4 <u>Encrypt or Hash all Authentication Credentials</u> Encrypt or hash with a salt all authentication credentials when stored.			

1.7 Ensure MariaDB is Run Under a Sandbox Environment (Manual)

Profile Applicability:

- Level 2 - MariaDB RDBMS on Linux
- Level 2 - MariaDB RDBMS

Description:

Use of the `chroot()` system call at startup, `Systemd` with settings to achieve isolation, or `docker` will put MariaDB in a Sandbox environment.

Rationale:

Running MariaDB in a Sandbox environment may reduce the impact of a MariaDB-born vulnerability by making portions of the file system inaccessible to the MariaDB instance.

Impact:

Use of the `chroot` option somewhat limits `LOAD DATA INFILE` and `SELECT ... INTO OUTFILE`.

Audit:

Perform the following steps for each MariaDB instance to assess this recommendation:

1. Execute the following SQL statement to determine the value of `chroot`

```
cat /etc/mysql | egrep '(?<=^chroot=) .+$'
```

The returned value should specify a valid path which differs from the `datadir`. No results implies 'chroot' is not in use.

2. Perform the following to check `systemd`:

```
systemctl status < mariadb>.service
```

If something other than `(root)` is listed beside the `PID`, e.g. `Main PID: <PID> (root)`, this is a pass. No results implies MariaDB is not managed by `systemd`.

3. Perform the following to determine if `Docker` is installed and a MariaDB container is in use:
 1. To check for `docker` installation, execute this command:

```
$ docker -v
```

If a message stating the version of docker which is installed proceed to the next steps, otherwise no further action is needed as docker must be installed

for MariaDB to be run in docker.

2. To check if a MariaDB image exists in docker run this command:

```
$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mariadb	latest	5c284e5e8296	2 weeks ago	384MB

If a MariaDB image is listed proceed to the next steps, otherwise, no further action is needed as a MariaDB image is required for MariaDB to be run in docker.

3. Check if a MariaDB container is running:

```
$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS	NAMES	
525f7777a0e8	mariadb:latest	"docker-entrypoint.s..."	2
minutes ago	Up 2 minutes	3306/tcp mariadb-server	

If a mariadb container is listed then MariaDB is running in docker and this is a pass.

If MariaDB does not use `chroot`, `systemd`, or `docker`, this is a fail.

Remediation:

Perform one of the following steps to remediate this setting:

- Configure MariaDB to use `chroot`:
 1. Choose a non-system partition `<chroot_location>` for MariaDB
 2. Add `chroot=<chroot_location>` to the `my.cnf` option file
- Configure MariaDB to run under `systemd`:
 1. If MariaDB is managed by `systemd` and running, stop the service:

```
$ sudo systemctl stop <mysqld>.service
```

2. If a `mysql` user and group do not already exist, create them:

```
$ sudo groupadd mysql
$ sudo useradd -r -g mysql -s /bin/false mysql
```

3. Set the ownership of the base directory:

```
$ sudo chown -R mysql:mysql /usr/local/mysql/
```

4. Create or modify the `<mysqld>.service` file in `/lib/systemd/system` to include the following entries, if not already present:

```
[Unit]
Description=MariaDB Server

[Install]
WantedBy=multi-user.target

[Service]
User=mysql
Group=mysql
```

5. If MariaDB was not already managed by systemd execute this command:

```
$ sudo systemctl daemon-reload
```

6. Start the MariaDB server:

```
$ sudo systemctl start < mariadb >.service
```

7. If you would like MariaDB to automatically run at startup execute this command:




```
$ sudo systemctl enable < mariadb >.service
```

- Follow documentation in the references for standing up MariaDB in a Docker container. During setup, follow the special notes below.
 - Do not set `MARIADB_ALLOW_EMPTY_ROOT_PASSWORD` or `MYSQL_ALLOW_EMPTY_PASSWORD` to a non-empty value.
 - Ensure that passwords are not set as values of any environment variables being passed to the Docker container. For example, do not orchestrate your container by passing `MARIADB_ROOT_PASSWORD=` as an environment attribute. Instead, securely set `MARIADB_ROOT_PASSWORD` within your environment, then simply tell Docker to pass `MARIADB_ROOT_PASSWORD` (without setting the value) to the container.

References:

1. <https://mariadb.com/docs/reference/mdb/cli/mariadb/chroot/>
2. <https://mariadb.com/kb/en/installing-and-using-mariadb-via-docker/>
3. https://hub.docker.com/_/mariadb
4. <https://mariadb.com/kb/en/mariadb-docker-environment-variables/>
5. <https://mariadb.com/kb/en/mariadb-container-cheat-sheet/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>3.12 Segment Data Processing and Storage Based on Sensitivity</u> 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	<u>2.10 Physically or Logically Segregate High Risk Applications</u> 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 Installation and Planning

This section contains important considerations when deploying MariaDB 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.

The first consideration is related to the configuration options via the MariaDB configuration file (e.g., `my.cnf`) and placing options under the proper section of `[mariadb]`. 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.

The second consideration is for an administrator to connect to a MariaDB instance and change or add to the configuration options using the `SET PERSIST` command. This persists system variables in `auto.cnf` which is located in the MariaDB `datadir` by default. The file permissions on `auto.cnf` are by default more restrictive than `my.cnf` (no world permissions).

Finally, configuration options can also be placed on the command line by modifying the MariaDB 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 - MariaDB RDBMS

Description:

A backup policy should be in place.

Rationale:

Backing up MariaDB 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.

References:

1. <https://mariadb.com/kb/en/backup-and-restore-overview/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	11.2 <u>Perform Automated Backups</u> Perform automated backups of in-scope enterprise assets. Run backups weekly, or more frequently, based on the sensitivity of the data.			
v7	10.1 <u>Ensure Regular Automated Back Ups</u> Ensure that all system data is automatically backed up on regular basis.			

2.1.2 Verify Backups are Good (Manual)

Profile Applicability:

- Level 1 - MariaDB RDBMS

Description:

Backups should be validated on a regular basis.

Rationale:

Verifying that backups are occurring appropriately will help ensure data availability 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.

References:

1. <https://mariadb.com/kb/en/full-backup-and-restore-with-mariabackup/>

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 - MariaDB RDBMS

Description:

A database user with the least amount of privileges required to perform backup is needed. 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 an attacker might potentially 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 <u>Configure Data Access Control Lists</u> 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 <u>Protect Recovery Data</u> Protect recovery data with equivalent controls to the original data. Reference encryption or data separation, based on requirements.			
v7	10.4 <u>Ensure Protection of Backups</u> 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 - MariaDB 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.







```
$ mariabackup --defaults-file=/home/dbadmin/my.cnf --backup --stream=xbstream \
| openssl enc -aes-256-cbc -k mypass > backup.xb.enc
```

The example above creates an AES-encrypted backup, protected with the password "mypass" and stores it in a file "backup.xb.enc":

References:

1. <https://mariadb.com/kb/en/using-encryption-and-compression-tools-with-mariabackup/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	11.3 <u>Protect Recovery Data</u> Protect recovery data with equivalent controls to the original data. Reference encryption or data separation, based on requirements.			
v7	10.4 <u>Ensure Protection of Backups</u> 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 (Automated)

Profile Applicability:

- Level 2 - MariaDB RDBMS on Linux
- Level 2 - MariaDB 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 to 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. Check to see if `--binlog-expire-logs-seconds` is set.

```
SELECT VARIABLE_NAME, VARIABLE_VALUE, 'BINLOG - Log Expiration' as Note
FROM information_schema.global_variables where variable_name =
'binlog_expire_logs_seconds';
```

Ensure this value is not set to 0.

Remediation:

Enable binlogs, then create and test a restore procedure.







Default Value:

0

References:

1. <https://mariadb.com/kb/en/replication-and-binary-log-system-variables/>
2. <https://mariadb.com/kb/en/overview-of-the-binary-log/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	11.2 <u>Perform Automated Backups</u> Perform automated backups of in-scope enterprise assets. Run backups weekly, or more frequently, based on the sensitivity of the data.			
v7	10.2 <u>Perform Complete System Backups</u> 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 - MariaDB RDBMS

Description:

A disaster recovery plan should be created.

MariaDB Galera Cluster (group replication), MariaDB Replication (asynchronous replication) or both may be used.

A replica 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://mariadb.com/kb/en/setting-up-replication/>
2. <https://mariadb.com/kb/en/getting-started-with-mariadb-galera-cluster/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	11.1 <u>Establish and Maintain a Data Recovery Process</u> 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 <u>Data Recovery Capabilities</u> Data Recovery Capabilities			

2.1.7 Backup of Configuration and Related Files (Manual)

Profile Applicability:

- Level 1 - MariaDB 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 (`mariadb.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.

References:

1. <https://mariadb.com/kb/en/configuring-mariadb-with-option-files/>

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 MariaDB (Manual)

Profile Applicability:

- Level 1 - MariaDB RDBMS

Description:

It is recommended that MariaDB 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, MariaDB server software, and any security or operational tooling that may be additionally installed. A smaller attack surface reduces the probability of the data within MariaDB being compromised.

Impact:

Care must be taken that 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 MariaDB 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	<u>3.12 Segment Data Processing and Storage Based on Sensitivity</u> 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	<u>2.10 Physically or Logically Segregate High Risk Applications</u> 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 the Command Line (Manual)

Profile Applicability:

- Level 1 - MariaDB RDBMS on Linux

Description:

When a command is executed on the command line, for example `mariadb -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 MariaDB 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 `.mariadb.cnf` is restricted yet accessible by the user.
- Use a pluggable secure password store, e.g., a keychain.

Additionally, not all scripts/applications may be able to use `.mariadb.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:





MariaDB Client:

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

References:

1. <https://mariadb.com/kb/en/mysql-command-line-client/>

CIS Controls:

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

2.4 Do Not Reuse Usernames (Manual)

Profile Applicability:

- Level 1 - MariaDB 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 <u>Ensure the Use of Dedicated Administrative Accounts</u> 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 - MariaDB RDBMS on Linux

Description:

The cryptographic material used by MariaDB, such as digital certificates and encryption keys, should be used only for MariaDB 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 MariaDB instances and/or systems, then a compromise of one may lead to the network traffic of all servers being compromised that 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 MariaDB server or otherwise compromise its operations.

Audit:

Review all cryptographic material. If it is default, used for other MariaDB instances and/or for purposes other than MariaDB 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=MariaDB_Server_10.6.8_Auto_Generated_Server_Certificate
```

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

Remediation:

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

References:

1. <https://mariadb.com/kb/en/securing-connections-for-client-and-server/>
2. <https://mariadb.com/kb/en/secure-connections-overview/>

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

Profile Applicability:

- Level 1 - MariaDB RDBMS

Description:

Password expiration provides users with a unique time bounded password lifetime.

Rationale:

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

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 information_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:

```

WITH password_expiration_info AS (
    SELECT User, Host,
        IF(
            IFNULL(JSON_EXTRACT(Priv, '$.password_lifetime'), -1) = -1,
            @@global.default_password_lifetime,
            JSON_EXTRACT(Priv, '$.password_lifetime')
        ) AS password_lifetime,
        JSON_EXTRACT(Priv, '$.password_last_changed') AS password_last_changed
    FROM mysql.global_priv
)
SELECT pei.User, pei.Host,
    pei.password_lifetime,
    FROM_UNIXTIME(pei.password_last_changed) AS
password_last_changed_datetime,
    FROM_UNIXTIME(
        pei.password_last_changed +
        (pei.password_lifetime * 60 * 60 * 24)
    ) AS password_expiration_datetime
FROM password_expiration_info pei
WHERE pei.password_lifetime != 0
    AND pei.password_last_changed IS NOT NULL
UNION
SELECT pei.User, pei.Host,
    pei.password_lifetime,
    FROM_UNIXTIME(pei.password_last_changed) AS
password_last_changed_datetime,
    0 AS password_expiration_datetime
FROM password_expiration_info pei
WHERE pei.password_lifetime = 0
    OR pei.password_last_changed IS NULL;

```

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:

0

References:

1. <https://csrc.nist.gov/csrc/media/publications/sp/800-118/archive/2009-04-21/documents/draft-sp800-118.pdf>
2. https://mariadb.com/docs/reference/mdb/system-variables/default_password_lifetime/
3. <https://mariadb.com/kb/en/user-password-expiry/>

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 Lock Out Accounts if Not Currently in Use (Manual)

Profile Applicability:

- Level 2 - MariaDB RDBMS
- Level 2 - MariaDB RDBMS on Linux

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 CONCAT(user, '@', host, ' => ', JSON_DETAILED(priv)) FROM  
mysql.global_priv ;
```

Accounts not in use and MariaDB Reserved accounts should show as
account_locked:true

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://mariadb.com/kb/en/account-locking/>

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`.

CIS Controls:

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

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

Profile Applicability:

- Level 2 - MariaDB RDBMS on Linux
- Level 2 - MariaDB RDBMS

Description:

The server-side `unix_socket` authentication plugin authenticates clients that connect to the MariaDB server from the local host through the Unix socket file. Users authenticated using `unix_socket` need not specify a password when connecting to the server. However, users authenticated by the `unix_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 MariaDB is running is dedicated to the MariaDB server and only the MariaDB 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 `unix_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 `unix_socket`.
- This will not work where distributed connections are required.

Audit:

To determine if the `unix_socket` plugin is enabled, run:

```
SELECT PLUGIN_NAME, PLUGIN_STATUS
FROM INFORMATION_SCHEMA.PLUGINS
WHERE PLUGIN_NAME = 'unix_socket';
```

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

To determine users who can use `unix_socket`, run:

```
SELECT CONCAT(user, '@', host, ' => ', JSON_DETAILED(priv)) FROM
mysql.global_priv where JSON_CONTAINS(priv, '{"plugin":"unix_socket"}',
'$.auth_or');
```

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

Remediation:

If the plugin is active and you need to disable it in your environment, add the following option under the `[mysqld]` option group in your MariaDB configuration file, then restart MariaDB:

```
unix_socket=OFF
```

If the plugin is disabled but you seek to use it, ensure the following option is set under the `[mysqld]` option group in your MariaDB configuration file, then restart MariaDB:

```
unix_socket=ON
```

To enable an OS user to login to MariaDB using `unix_socket`, include `'unix_socket'` as an authentication plugin in your `IDENTIFIED VIA` clause of `CREATE USER` commands. For example, run:

```
CREATE USER '<user>'@'localhost' IDENTIFIED VIA unix_socket;
```

The user can then login using:

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

Default Value:

The `unix_socket` plugin is `ON` by default.



References:

1. <https://mariadb.com/kb/en/authentication-plugin-unix-socket/>
2. https://mariadb.com/kb/en/create-user/#identified-viawith-authentication_plugin

Additional Information:

You cannot dynamically install or uninstall the `unix_socket` plugin in MariaDB 10.6.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>6.6 Establish and Maintain an Inventory of Authentication and Authorization Systems</u> 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.			

2.9 Ensure MariaDB is Bound to an IP Address (Automated)

Profile Applicability:

- Level 2 - MariaDB RDBMS
- Level 2 - MariaDB RDBMS on Linux

Description:

By default, the MariaDB server accepts TCP/IP connections from MariaDB 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 MariaDB. If not configured to a specific IP all IPs for this server can be used to connect to MariaDB.

Audit:

Run the following statement:

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

A `VARIABLE_VALUE` that is empty, is `"*"`, or is `:::` implies a fail.

Remediation:

For example, to have the MariaDB server only accept connections on a specific IPv4 address, add an entry similar to this under the `[mysqld]` option group in MariaDB configuration files:

```
bind_address=192.0.2.24
```

This setting typically appears in `/etc/mysql/mariadb.conf.d/50-server.cnf`. In the case above, 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. On some linux variants (Ubuntu, Debian), `bind_address` is set to `127.0.0.1`.

References:

1. https://mariadb.com/kb/en/server-system-variables/#bind_address

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p><u>16.10 Apply Secure Design Principles in Application Architectures</u></p> <p>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.</p>			

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

Profile Applicability:

- Level 2 - MariaDB RDBMS
- Level 2 - MariaDB RDBMS on Linux

Description:

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

Rationale:

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

Impact:

Connections attempting to use an unsupported version of TLS 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.

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

Remediation:

Set the version(s) of TLS you wish to accept by setting the `tls_version` option to a comma-separated (no whitespace) string in MariaDB configuration files.

For example, to only accept TLS 1.2 or 1.3 connections, set `tls_version` likeso:

```
tls_version=TLSv1.2,TLSv1.3
```

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:









TLSv1.1,TLSv1.2,TLSv1.3

References:

1. <https://mariadb.com/kb/en/secure-connections-overview/>

2. https://mariadb.com/kb/en/ssltls-system-variables/#tls_version

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 <u>Encrypt Sensitive Data in Transit</u> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v8	16.11 <u>Leverage Vetted Modules or Services for Application Security Components</u> 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 <u>Encrypt All Sensitive Information in Transit</u> Encrypt all sensitive information in transit.			
v7	18.5 <u>Use Only Standardized and Extensively Reviewed Encryption Algorithms</u> Use only standardized and extensively reviewed encryption algorithms.			

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

Profile Applicability:

- Level 2 - MariaDB RDBMS
- Level 2 - MariaDB RDBMS on Linux

Description:

Client-side certificates may be used as proof of identity.

Rationale:

Requiring client-side certificates provides additional validation of a user's identity.

Audit:

Run the following statement

```
select user, host, ssl_type from mysql.user where user not in ('mysql',  
'root', 'mariadb.sys');
```

If `ssl_type` returns `X509`, 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
```

References:





1. <https://mariadb.com/kb/en/certificate-creation-with-openssl/>

Additional Information:

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

- `'mysql'@'localhost'`
- `'root'@'localhost'`
- `'mariadb.sys'@'localhost'`

CIS Controls:

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

2.12 Ensure Only Approved Ciphers are Used (Automated)

Profile Applicability:

- Level 2 - MariaDB RDBMS on Linux
- Level 2 - MariaDB RDBMS

Description:

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

Rationale:

Requiring clients attempting to connect to MariaDB 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 information_schema.global_variables  
WHERE VARIABLE_NAME = 'ssl_cipher';
```

If VARIABLE_VALUE is empty, or includes unapproved ciphers, this is a fail.

Remediation:

Set `ssl_cipher` to one or more approved cipher suites in your MariaDB configuration file, then restart MariaDB.

For example, set:

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

Default Value:

None

References:





1. https://mariadb.com/kb/en/ssltls-system-variables/#ssl_cipher
2. <https://mariadb.com/kb/en/secure-connections-overview/>

Additional Information:

The `ssl_cipher` option implies the `ssl` option, so you must have SSL/TLS setup to utilize this option.

`ssl_cipher` is not a dynamic variable, so it cannot be set once MariaDB is running.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>16.11 Leverage Vetted Modules or Services for Application Security Components</u> 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	<u>18.5 Use Only Standardized and Extensively Reviewed Encryption Algorithms</u> Use only standardized and extensively reviewed encryption algorithms.			

3 File Permissions

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

3.1 Ensure 'datadir' Has Appropriate Permissions (Automated)

Profile Applicability:

- Level 1 - MariaDB RDBMS on Linux

Description:

The data directory is the location of the MariaDB databases.

Rationale:

Limiting the accessibility of these objects will protect the confidentiality, integrity, and availability of the MariaDB database. If someone other than the MariaDB 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 information_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>
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> 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 <u>Protect Information through Access Control Lists</u> 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 - MariaDB RDBMS on Linux

Description:

MariaDB 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 MariaDB user. Additionally, using secure key management and at rest MariaDB encryption can further protect data from OS users.

Rationale:

Limiting the accessibility of these objects will protect the confidentiality, integrity, and availability of the MariaDB 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 MariaDB 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>
```

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 - MariaDB RDBMS on Linux

Description:

MariaDB 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 MariaDB user. Additionally, using secure key management and at rest MariaDB encryption can further protect data from OS users.

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

Rationale:

Limiting the accessibility of these objects will protect the confidentiality, integrity, and availability of the MariaDB 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 /var/log/mysql/mariadb.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://mariadb.com/kb/en/error-log/>

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	14.6 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 - MariaDB RDBMS on Linux

Description:

MariaDB 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 MariaDB user. Additionally, using secure key management and at rest MariaDB encryption can further protect data from OS users.

Much of the information about the state of MariaDB exists in MariaDB, the MariaDB `performance_schema` or `information_schema`. If you can get the information you need from within MariaDB 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 MariaDB 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 MariaDB `performance_schema` or `sys` schema views. In cases where the information you need is within a running MariaDB, 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 PERSIST 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://mariadb.com/kb/en/slow-query-log-overview/>

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 - MariaDB RDBMS on Linux

Description:

MariaDB 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 MariaDB user. Additionally, using secure key management and at rest MariaDB encryption can further protect data from OS users.

Rationale:

Limiting the accessibility of these objects will protect the confidentiality, integrity, and availability of the MariaDB 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 MariaDB 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'

References:

1. <https://mariadb.com/kb/en/mariadb-10-1-6-release-notes/>
2. <https://mariadb.com/kb/en/relay-log/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> 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 <u>Protect Information through Access Control Lists</u> 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 - MariaDB RDBMS on Linux

Description:

MariaDB 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 MariaDB user. Additionally, using secure key management and at rest MariaDB encryption can further protect data from OS users.

Much of the information about the state of MariaDB exists in MariaDB, the MariaDB `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 MariaDB 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 MariaDB `SYS`, `PERFORMANCE_SCHEMA`, or MariaDBL 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 PERSIST @@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://mariadb.com/kb/en/general-query-log/>

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.			

3.7 Ensure SSL Key Files Have Appropriate Permissions (Automated)

Profile Applicability:

- Level 1 - MariaDB RDBMS on Linux

Description:

When configured to use SSL/TLS, MariaDB 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.

MariaDB 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 MariaDB 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 MariaDB is restarted or can cause MariaDB 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 MariaDB.

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:


```
MariaDB [(none)]> SELECT * FROM information_schema.global_variables
WHERE
REGEXP_INSTR(VARIABLE_NAME, '^.*ssl_(ca|capath|cert|crl|crlpath|key)$')
AND VARIABLE_VALUE <> '';
```

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

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

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://mariadb.com/kb/en/secure-connections-overview/>

Additional Information:

If SSL is not configured this recommendation is not applicable. By default MariaDB 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 <u>Protect Information through Access Control Lists</u> 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 - MariaDB RDBMS on Linux

Description:

The plugin directory is the location of the MariaDB 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 MariaDB 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 MariaDB user will no longer be able to update and add/remove plugins unless they're able to switch to the MariaDB 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. MariaDB 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. <https://mariadb.com/kb/en/show-plugins-soname/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> 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 <u>Protect Information through Access Control Lists</u> 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 Ensure 'server_audit_file_path' Has Appropriate Permissions (Automated)

Profile Applicability:

- Level 1 - MariaDB RDBMS on Linux

Description:

MariaDB can operate using a variety of log files, each used for different purposes. These are the binary log, error log, slow query log, relay log, audit log and general log. 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.

Rationale:

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

Impact:

Changing the permissions and ownership of the audit log file may have an impact on who can access and edit the audit log. Such changes can affect monitoring tools which maybe using a log file adapter or scripted alternatives. Also, the audit log may be used for alerting by infrastructure teams which can affect real-time audit capability.

Audit:

To assess this recommendation, execute the following SQL statement to discover the `server_audit_file_path` value:

```
show global variables where variable_name='server_audit_file_path';
```

If no value is returned, auditing is not installed, and this is a fail.

Note: If you see the audit file name but no path, the default path will be the path assigned to the `datadir` variable.

Then, execute the following command at a terminal prompt (using the discovered `server_audit_file_path` value):

```
ls -l <server_audit_file_path> | egrep "^-([rw-]{2}-){2}---[ \t]*[0-9][ \t]*mysql[ \t]*mysql.*$"
```







No results implies a fail.

Remediation:

Execute the following commands for the `server_audit_file_path` discovered in the audit procedure:

```
chmod 660 <server_audit_file_path>  
chown mysql:mysql <server_audit_file_path>
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> 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 <u>Protect Information through Access Control Lists</u> 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.			

3.10 Ensure File Key Management Encryption Plugin files have appropriate permissions (Automated)

Profile Applicability:

- Level 1 - MariaDB RDBMS on Linux

Description:

Certain Key Management Encryption plugins must store sensitive information in files for later retrieval. Such files should have proper permissions.

Rationale:

Limiting the accessibility of these files will protect the confidentiality, integrity, and availability of MariaDB plugin information and the data they protect.

Audit:

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

File Key Management Plugin

1. Find the `file_key_management_filename` value by executing the following statement:

```
grep -Po '(?<=file_key_management_filename=).+$' /etc/mysql/mariadb.cnf
```

2. Verify permissions are 750 for `mysql:mysql` (or more restrictive) for `file_key_management_filename`
3. Find the `file_key_management_filekey` value by executing the following statement:

```
grep -Po '(?<=file_key_management_filekey=).+$' /etc/mysql/mariadb.cnf
```

4. Verify permissions are 750 for `mysql:mysql` (or more restrictive) for `file_key_management_filekey`

Additionally, if the File Key Management Encryption plugin is not configured (if there are no such files from steps 1 and 3 above), this is a fail.

Remediation:

If the File Key Management plugin is not configured, first implement recommendation 4.10 from this benchmark.







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

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

References:

1. <https://mariadb.com/kb/en/data-at-rest-encryption-overview/>
2. <https://mariadb.com/kb/en/file-key-management-encryption-plugin/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> 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 <u>Protect Information through Access Control Lists</u> 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 the Latest Security Patches are Applied (Manual)

Profile Applicability:

- Level 1 - MariaDB RDBMS on Linux

Description:

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

Rationale:

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

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

Impact:

To update the MariaDB server a restart is required.

Audit:

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

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

Now compare the version with the security announcements from MariaDB 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. <https://mariadb.com/kb/en/security/>
2. <https://mariadb.com/kb/en/mariadb-1060-release-notes/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>2.2 Ensure Authorized Software is Currently Supported</u> 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	<u>2.2 Ensure Software is Supported by Vendor</u> 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 - MariaDB RDBMS

Description:

The default MariaDB 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 MariaDB 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 database this is a finding.

Remediation:

Execute the following SQL statement to drop an example database:

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



Default Value:

By default, MariaDB 10.6 does not contain any example or test databases.

References:

1. <https://mariadb.com/docs/server/deploy/community-primary-cs10-6/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p><u>16.10 Apply Secure Design Principles in Application Architectures</u></p> <p>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.</p>			

4.3 Ensure 'allow-suspicious-udfs' is Set to 'OFF' (Automated)

Profile Applicability:

- Level 2 - MariaDB 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 `mariadb` start up command line.
- Ensure `allow-suspicious-udfs` is set to `OFF` in the MariaDB configuration:

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

No results returned is a pass.

Remediation:

Perform the following to establish the recommended state:

- Remove `--allow-suspicious-udfs` from the `mariadb` start up command line.
- Remove `allow-suspicious-udfs` from the MariaDB option file.

Default Value:

`OFF`



References:

1. <https://mariadb.com/docs/server/ref/mdb/cli/mariadb/allow-suspicious-udfs/>
2. <https://mariadb.com/kb/en/user-defined-functions-security/>

Additional Information:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p><u>16.10 Apply Secure Design Principles in Application Architectures</u></p> <p>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.</p>			

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

Profile Applicability:

- Level 1 - MariaDB RDBMS

Description:

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

Rationale:

For MariaDB client programs and connectors prior to 10.2.0, 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:

Check the version of MariaDB clients and connectors.
For example:

```
$ mariadb --version
```

The version should be 10.2.0 or higher.
For connectors inspect the library in use.
Most connectors provide functions which return version information.
For C - `libmysqlclient` has:

```
const char *mysql_get_client_info(void)
```

If clients have not been upgraded to 10.2.0 check the value of `local_infile`.
Execute the following SQL statement:

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

If clients are older than 10.2.0 or if `local_infile` is not in use, ensure the value returned is 0.

Remediation:

Upgrade all MariaDB clients and connectors to 10.2.0 or higher.
In the case where using `local_infile` is needed, the following changes further harden security:
On client side, secure by:
Limiting the location from where data can be read using `--load-data-local-dir`.

```
mariadb --local-infile=0 --load-data-local-dir=/my/local/data
```

Adding TLS connection to assure server identity by requiring verification.

```
mariadb --local-infile=0 --load-data-local-dir=/my/local/data --ssl-mode=VERIFY_IDENTITY
```

If `local_infile` is not in use or if clients are not upgraded - add the following line to the `[mariadb]` section of the MySQL configuration file and restart the MariaDB service:

```
local-infile=0
```





Default Value:

0 (OFF)

References:

1. <https://mariadb.com/docs/server/ref/mdb/cli/mariabdb/local-infile/>

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 mariadb is Not Started With 'skip-grant-tables' (Automated)

Profile Applicability:

- Level 1 - MariaDB RDBMS on Linux

Description:

This option causes `mariabdd` 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 MariaDB configuration (e.g., `mariadb.cnf`) file and search for `skip-grant-tables` and `skip_grant_tables`
- Ensure all occurrences of `skip-grant-tables` or `skip_grant_tables` are set to `FALSE`

Remediation:

Perform the following to establish the recommended state:

- Open the MariaDB configuration (e.g., `mariadb.cnf`) file and set:

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

- If there are any occurrences of `skip_grant_tables`, also set that to `FALSE` or remove it.







References:

1. <https://mariadb.com/docs/server/ref/mdb/cli/mariabdd/skip-grant-tables/>
2. https://mariadb.com/kb/en/server-system-variables/#skip_grant_tables
3. <https://mariadb.com/kb/en/mysqld-options/#option-prefixes>

Additional Information:

This option has no `SHOW VARIABLES` counterpart.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> 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 <u>Protect Information through Access Control Lists</u> 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 - MariaDB RDBMS

Description:

The `symbolic-links` and `skip-symbolic-links` options for MariaDB 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 MariaDB is executing as root as arbitrary files may be overwritten. The `symbolic-links` option might allow someone to direct actions by the MariaDB 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 MariaDB configuration file (`mariadb.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 `mariadb` section.

References:

1. <https://mariadb.com/docs/server/ref/mdb/cli/mariabdb/symbolic-links/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	16.10 <u>Apply Secure Design Principles in Application Architectures</u> 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 <u>Data Protection</u> Data Protection			

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

Profile Applicability:

- Level 1 - MariaDB RDBMS

Description:

The `secure_file_priv` option restricts to paths used by `LOAD DATA INFILE` or `SELECT local_file`. It is recommended that this option be set to a file system location that contains only resources expected to be loaded by MariaDB. 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 `[mariadb]` section of the MariaDB configuration file and restart the MariaDB service.

If you need this feature add the following line to the `[mariadb]` section of the MariaDB configuration file and restart the MariaDB service:

```
secure_file_priv=<path_to_load_directory>
```




Default Value:

No value set.

References:

1. https://mariadb.com/docs/server/ref/mdb/system-variables/secure_file_priv/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> 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 <u>Data Protection</u> Data Protection			

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

Profile Applicability:

- Level 2 - MariaDB RDBMS
- Level 2 - MariaDB RDBMS on Linux

Description:

When data changing statements are made (i.e., `INSERT`, `UPDATE`), MariaDB 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 MariaDB 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 MariaDB 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_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_ENGINE_SUBSTITUTION';
```

Default Value:

STRICT_TRANS_TABLES,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION

References:



1. <https://mariadb.com/docs/server/ref/mdb/cli/mariabdb/sql-mode/>
2. https://mariadb.com/docs/server/ref/mdb/sql-modes/STRICT_ALL_TABLES/
3. <https://mariadb.com/kb/en/sql-mode/>

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	<p>16.10 <u>Apply Secure Design Principles in Application Architectures</u></p> <p>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.</p>			

4.9 Enable data-at-rest encryption in MariaDB (Automated)

Profile Applicability:

- Level 2 - MariaDB RDBMS
- Level 2 - MariaDB RDBMS on Linux

Description:

Data-at-rest encryption protects the privacy of your information, prevents data breaches and helps meet regulatory requirements.

Rationale:

File system based encryption does a good job of protecting against data theft on devices unable to limit physical access. It does not, however, protect against users who have or gain access to the operating system, backups, over the network copies, etc. Encrypting data from MariaDB adds an additional layer of data protection.

Audit:

Check for the types of at-rest encryption enabled.

```
SELECT VARIABLE_NAME, VARIABLE_VALUE FROM information_schema.global_variables
where variable_name like '%ENCRYPT%';
```

OFF indicates at-rest encryption is not enabled.

To check which tablespaces are encrypted

```
SELECT SPACE,NAME FROM INFORMATION_SCHEMA.INNODB_TABLESPACES_ENCRYPTION;
```

If no data is returned or if your database has tables that are not included in this response, this is a fail.

Backup data should be encrypted at rest as well.

For example:

```
mariabackup --user=root --backup --stream=xbstream | openssl enc -aes-256-
cbc -k mypass > backup.xb.enc
```

If no encryption tool is included in the backup command, this is a fail.

Remediation:

MariaDB's data-at-rest encryption requires the use of a key management and encryption plugin.

Create the key file:

```
$ sudo mkdir -p /etc/mysql/encryption && (echo -n "1;" ; openssl rand -hex 32
) | sudo tee -a /etc/mysql/encryption/keyfile
```

Generate a random encryption password:

```
$ sudo openssl rand -hex 128 | sudo tee -a /etc/mysql/encryption/keyfile.key
```

Encrypt the key file:

```
$ sudo openssl enc -aes-256-cbc -md sha1 \  
-pass file:/etc/mysql/encryption/keyfile.key \  
-in /etc/mysql/encryption/keyfile \  
-out /etc/mysql/encryption/keyfile.enc
```

Delete the unencrypted key file:

```
$ sudo rm /etc/mysql/encryption/keyfile
```

Set permissions and ownership on the keyfile and key:

```
$ sudo chown mysql:mysql -R /etc/mysql/encryption  
$ sudo chmod 640 /etc/mysql/encryption/keyfile*
```

Edit `mariadb.cnf` to resemble the following block, optionally uncommenting

`file_key_management_encryption_algorithm = AES_CTR`:

```
[mariadb]  
...  
plugin_load_add = file_key_management  
file_key_management_filename = /etc/mysql/encryption/keyfile.enc  
file_key_management_filekey = FILE:/etc/mysql/encryption/keyfile.key  
  
# Binary Log Encryption  
encrypt_binlog = ON  
# Redo Log Encryption  
innodb_encrypt_log = ON  
# Encrypting Temporary Files  
encrypt_tmp_files = ON  
# Encrypt Temporary Tables  
innodb_encrypt_temporary_tables = ON  
  
# You can configure InnoDB encryption to automatically have all new InnoDB  
tables automatically encrypted, or specify encrypt per table.  
innodb_encrypt_tables = ON  
  
# Uncomment the line below if utilizing MariaDB built with OpenSSL  
# file_key_management_encryption_algorithm = AES_CTR
```

If needed, see References for information about

`file_key_management_encryption_algorithm` and OpenSSL usage.

Restart MariaDB:

```
$ sudo systemctl restart mariadb.service
```

Run `ALTER` to enable encryption (**Note:** This will lock the table as table is encrypted).

```
ALTER TABLE tabl  
  ENCRYPTED=YES ENCRYPTION_KEY_ID=1;
```

Revisit recommendation 3.10 after completing remediation.

Default Value:

At rest encryption is off by default.

When `innodb_encrypt_tables` is set to ON, InnoDB tables are automatically encrypted by default.




`mariadb.cnf`.

```
innodb_encrypt_tables=ON
```

References:

1. <https://mariadb.com/resources/blog/mariadb-encryption-tde-using-mariadbs-file-key-management-encryption-plugin/>
2. <https://mariadb.com/kb/en/data-at-rest-encryption-overview/>
3. <https://mariadb.com/kb/en/encrypting-binary-logs/>
4. <https://mariadb.com/kb/en/innodb-encryption-overview/>
5. <https://mariadb.com/kb/en/using-encryption-and-compression-tools-with-mariabackup/>
6. <https://mariadb.com/kb/en/encryption-key-management/>
7. <https://mariadb.com/kb/en/innodb-enabling-encryption/>
8. <https://mariadb.com/kb/en/file-key-management-encryption-plugin/#choosing-an-encryption-algorithm>
9. <https://mariadb.com/kb/en/tls-and-cryptography-libraries-used-by-mariadb/>

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	14.8 Encrypt Sensitive Information at Rest Encrypt all sensitive information at rest using a tool that requires a secondary authentication mechanism not integrated into the operating system, in order to access the information.			

5 MariaDB Permissions

This section contains recommendations about user privileges.

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

Profile Applicability:

- Level 1 - MariaDB RDBMS
- Level 1 - MariaDB 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 MariaDB 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 MariaDB. 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. The above query ignores MariaDB 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://mariadb.com/docs/server/ref/mdb/information-schema/USER_PRIVILEGES/

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	<u>5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts</u> 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	<u>4.1 Maintain Inventory of Administrative Accounts</u> Use automated tools to inventory all administrative accounts, including domain and local accounts, to ensure that only authorized individuals have elevated privileges.			
v7	<u>14.6 Protect Information through Access Control Lists</u> 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 - MariaDB RDBMS
- Level 1 - MariaDB 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 MariaDB server (this includes world-readable files).
- Write files to the local file system where the MariaDB server has write access.

Rationale:

The `FILE` right allows MariaDB users to read files from disk and to write files to disk. This may be leveraged by an attacker to further compromise MariaDB. It should be noted that the MariaDB 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://mariadb.com/docs/server/ref/mdb/system-variables/secure_file_priv/
2. <https://mariadb.com/docs/server/ref/mdb/cli/mariabdb/secure-file-priv/>

Additional Information:

See also: `secure_file_priv` system settings.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts</u> 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	<u>14.6 Protect Information through Access Control Lists</u> 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 - MariaDB RDBMS
- Level 2 - MariaDB 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 MariaDB statements beyond their own, including statements used to manage passwords. This may be leveraged by an attacker to compromise MariaDB 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://mariadb.com/docs/server/ref/mdb/information-schema/USER_PRIVILEGES/
2. <https://mariadb.com/kb/en/show-privileges/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts</u> 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	<u>14.6 Protect Information through Access Control Lists</u> 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 - MariaDB RDBMS

Description:

The `SUPER` privilege is a powerful and far-reaching privilege and should not be granted lightly. In MariaDB, `SUPER` is deprecated and will be removed in a future version of MariaDB.

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

In MariaDB, `SUPER` is deprecated and will be removed in a future version of MariaDB. Migrating Accounts from `SUPER` to Dynamic Privileges is recommended.

Rationale:

The `SUPER` privilege allows principals to perform many actions, including view and terminate currently executing MariaDB statements (including statements used to manage passwords). This privilege also provides the ability to configure MariaDB, 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:

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

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

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://mariadb.com/docs/server/ref/mdb/information-schema/USER_PRIVILEGES/
2. <https://mariadb.com/kb/en/show-privileges/>

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 - MariaDB RDBMS

Description:

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

Rationale:

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

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://mariadb.com/docs/server/ref/mdb/information-schema/USER_PRIVILEGES/
2. <https://mariadb.com/kb/en/show-privileges/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts</u> 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	<u>14.6 Protect Information through Access Control Lists</u> 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 - MariaDB RDBMS

Description:

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

Rationale:

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

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://mariadb.com/docs/server/ref/mdb/information-schema/USER_PRIVILEGES/
2. <https://mariadb.com/kb/en/show-privileges/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts</u> 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	<u>14.6 Protect Information through Access Control Lists</u> 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 - MariaDB RDBMS

Description:

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

Rationale:

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

Audit:

Execute the following SQL statements to audit this setting:

```
SELECT DISTINCT GRANTEE
FROM INFORMATION_SCHEMA.USER_PRIVILEGES
WHERE IS_GRANTABLE = 'YES';
```

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://mariadb.com/docs/server/ref/mdb/information-schema/USER_PRIVILEGES/
2. <https://mariadb.com/kb/en/show-privileges/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<u>5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts</u> 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	<u>14.6 Protect Information through Access Control Lists</u> 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 - MariaDB RDBMS

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 to table data from the source. This may be used by an attacker to read/fetch sensitive data from MariaDB.

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 replica 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://mariadb.com/docs/server/ref/mdb/privileges/REPLICATION_SLAVE/
2. <https://mariadb.com/kb/en/show-privileges/>
3. https://mariadb.com/docs/server/ref/mdb/information-schema/USER_PRIVILEGES/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> 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 <u>Protect Information through Access Control Lists</u> 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 - MariaDB RDBMS

Description:

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

Rationale:

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

Audit:

Execute the following SQL statement to audit this setting:

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

Ensure all users returned 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>;

```

References:

1. <https://mariadb.com/kb/en/data-manipulation/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 <u>Configure Data Access Control Lists</u> 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 <u>Protect Information through Access Control Lists</u> 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 - MariaDB 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://mariadb.com/kb/en/create-procedure/>
2. <https://mariadb.com/kb/en/create-function/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	<p><u>16.10 Apply Secure Design Principles in Application Architectures</u></p> <p>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.</p>			
v7	<p><u>14.6 Protect Information through Access Control Lists</u></p> <p>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.</p>			

6 Auditing and Logging

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

6.1 Ensure 'log_error' is configured correctly (Automated)

Profile Applicability:

- Level 1 - MariaDB RDBMS

Description:

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

Rationale:

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

When configured to `stderr` MariaDB 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 MariaDB and dedicated MariaDB audit accounts. This may compromise the confidentiality of the MariaDB 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.err`.

Remediation:

Perform the following actions to remediate this setting:

1. Open the MariaDB configuration file (`mariadb.cnf`).
2. Set the `log_error` option to the path for the error log.







Default Value:

`./stderr.err`

References:

1. https://mariadb.com/docs/server/ref/mdb/system-variables/log_error/
2. <https://mariadb.com/kb/en/error-log/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 <u>Collect Audit Logs</u> Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.			
v7	6.2 <u>Activate audit logging</u> 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 - MariaDB RDBMS

Description:

MariaDB log files can be set in the MariaDB 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 MariaDB 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 MariaDB configuration file (`mariadb.cnf`)
2. Locate the `log_bin` entry and set it to a file not on root (/), /var, or /usr

References:

1. https://mariadb.com/docs/server/ref/mdb/system-variables/log_bin/
2. <https://mariadb.com/kb/en/binary-log/>

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 <u>Ensure adequate storage for logs</u> Ensure that all systems that store logs have adequate storage space for the logs generated.		●	●

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

Profile Applicability:

- Level 2 - MariaDB RDBMS
- Level 2 - MariaDB RDBMS on Linux

Description:

The `log_warnings` system variable, set to 2 by default, specifies the verbosity of events sent to the MariaDB 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_warnings';
```

Ensure the `Value` returned equals 2.

Remediation:

Perform the following actions to remediate this setting:

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

```
log_warnings = 2
```





Default Value:

The option is enabled (2) - errors and warning events are logged - by default.

References:

1. https://mariadb.com/docs/server/ref/mdb/system-variables/log_warnings/
2. <https://mariadb.com/kb/en/error-log/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 <u>Collect Detailed Audit Logs</u> 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 <u>Enable Detailed Logging</u> 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 Audit Logging Is Enabled (Automated)

Profile Applicability:

- Level 2 - MariaDB RDBMS on Linux
- Level 2 - MariaDB RDBMS

Description:

Enabling audit logging is an important consideration for a production environment, and MariaDB plugin exist to help with this. Enable audit logging for

- Connect events
- Query and Table events (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 MariaDB Audit is installed and configured to enable logging for connect events and (optionally) query and table events.

```
SHOW VARIABLES LIKE '%audit%' ;
```

Remediation:

Although the plugin's shared library is distributed with MariaDB, the plugin is not actually installed by default.

Add the following to MariaDB's config file.







```
[mariadb]
...
#MariaDB plugin
plugin_load_add = server_audit
server_audit_logging=ON
server_audit_events=CONNECT
```

Reboot the instance.

References:

1. <https://mariadb.com/kb/en/mariadb-audit-plugin/>
2. <https://mariadb.com/kb/en/mariadb-audit-plugin-installation/>
3. <https://mariadb.com/kb/en/mariadb-audit-plugin-configuration/>
4. <https://mariadb.com/kb/en/mariadb-audit-plugin-log-settings/>

CIS Controls:

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

6.5 Ensure the Audit Plugin Can't be Unloaded (Automated)

Profile Applicability:

- Level 1 - MariaDB RDBMS

Description:

Set `server_audit` to `FORCE_PLUS_PERMANENT`

Rationale:

This disables unloading on the plugin.

Impact:

If someone can unload the plugin it would be possible to perform actions on the database without audit events being logged to the audit log. If the audit log plugin can be unloaded the audit log can be temporarily or permanently disabled.

Audit:

To assess this recommendation, execute the following SQL statement:

```
SELECT LOAD_OPTION FROM information_schema.plugins WHERE  
PLUGIN_NAME='SERVER_AUDIT';
```

The result must be `FORCE_PLUS_PERMANENT`

Remediation:

To remediate this setting, follow these steps:




1. Open the MariaDB configuration file (`mariadb.cnf`)
2. Ensure the following line is found in the `mariadbd` section




```
server_audit=FORCE_PLUS_PERMANENT
```

References:

1. <https://mariadb.com/kb/en/mariadb-audit-plugin-installation/>

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.			

Controls Version	Control	IG 1	IG 2	IG 3
v7	6.2 <u>Activate audit logging</u> Ensure that local logging has been enabled on all systems and networking devices.			

6.6 Ensure Binary and Relay Logs are Encrypted (Automated)

Profile Applicability:

- Level 2 - MariaDB RDBMS on Linux
- Level 2 - MariaDB RDBMS

Description:

The `encrypt_binlog` system variable may be used to configure encryption of the binary and relay logs. This may be configured to `ON` even if binary logging is not enabled in order to encrypt relay log files.

Rationale:

The database, and thus the binary and relay logs, may contain sensitive information. Encrypting the binary and relay logs protects all data stored in these logs from internal and external threats.

Audit:

To audit this setting, run the following command:

```
SELECT VARIABLE_NAME, VARIABLE_VALUE, 'BINLOG - At Rest Encryption' as Note
FROM information_schema.global_variables where variable_name like
'%ENCRYPT_LOG%';
```

Ensure it is set to `ON`.

Remediation:

Encryption of binary logs is configured by the `encrypt_binlog` system variable. To remediate misconfiguration, add `encrypt_binlog` and restart MariaDB.

```
[mariadb]
...
# Binary Log Encryption
encrypt_binlog=ON
```

Default Value:

The default Default Value: `OFF`

References:




1. <https://mariadb.com/kb/en/encrypting-binary-logs/>
2. <https://mariadb.com/kb/en/activating-the-binary-log/>
3. <https://mariadb.com/kb/en/key-management-and-encryption-plugins/>
4. <https://mariadb.com/kb/en/purge-binary-logs/>
5. <https://mariadb.com/kb/en/reset-master/>

Additional Information:

It is necessary to install a Key Management Encryption plugin prior to configuring encryption.

After enabling encryption, consider also deleting old, unencrypted logs, using the `PURGE BINARY LOGS` and `RESET MASTER` commands.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 <u>Encrypt Sensitive Data at Rest</u> 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	14.8 <u>Encrypt Sensitive Information at Rest</u> Encrypt all sensitive information at rest using a tool that requires a secondary authentication mechanism not integrated into the operating system, in order to access the information.			

7 Authentication

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

7.1 Disable use of the `mysql_old_password` plugin (Automated)

Profile Applicability:

- Level 1 - MariaDB RDBMS

Description:

The `mysql_old_password` plugin uses a cracked password routine with several flaws. It is present but not used by default in MariaDB. Steps should be taken to ensure that passwords will not be created with this plugin and that clients will not be able to authenticate to the MariaDB server using this plugin.

The `old_passwords` system variable controls whether MariaDB server uses the `mysql_old_password` plugin (instead of the stronger `mysql_native_password` plugin) when creating passwords. The `secure_auth` system variable, when enabled, will block client connections that utilize the `mysql_old_password` plugin.

Rationale:

Use of the `mysql_old_password` plugin risks disclosure of users' passwords and also permits Pass-the-Hash attacks.

Audit:

To ensure new passwords are not created using the `mysql_old_password` plugin, run:

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

Ensure the `Value` field is set to `OFF`.

To ensure connections that use the `mysql_old_password` plugin are blocked, run:

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

Ensure the `Value` field is set to `ON`.

Remediation:

If `old_passwords` was `ON`, add the following line to the `[mariadb]` section in `mariadb.cnf`:

```
old_passwords=0
```

If `'secure_auth'` was `OFF`, add the following line to the `[mariadb]` section in `mariadb.cnf`:

```
secure_auth=ON
```

Restart MariaDB.





Default Value:

`old_passwords` is `OFF` by default. `secure_auth` is `ON` by default.

References:

1. <https://mariadb.org/history-of-mysql-mariadb-authentication-protocols/>
2. <https://mariadb.com/kb/en/pluggable-authentication-overview/#default-server-authentication-plugin>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 <u>Encrypt Sensitive Data at Rest</u> 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 <u>Encrypt or Hash all Authentication Credentials</u> 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 - MariaDB RDBMS on Linux

Description:

The `[client]` section of the MariaDB configuration file allows setting a `user` and `password` to be used. Verify the `password` option is not used in the global configuration file (`mariadb.cnf`).

Rationale:

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

Impact:

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

Audit:

To assess this recommendation, perform the following steps:

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

Remediation:

Use the user-specific options file, `.mariadb.cnf.`, and restricting file access permissions to the user identity.





References:

1. https://mariadb.com/kb/en/mysql_config_editor-compatibility/

Additional Information:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 <u>Encrypt Sensitive Data at Rest</u> 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 <u>Encrypt or Hash all Authentication Credentials</u> Encrypt or hash with a salt all authentication credentials when stored.			

7.3 Ensure strong authentication is utilized for all accounts (Automated)

Profile Applicability:

- Level 1 - MariaDB RDBMS

Description:

The `mysql_native_password` and `mysql_old_password` plugins utilize weak cryptography and/or weak password routines.

In particular, the `mysql_old_password` plugin utilizes cracked password routines and is subject to Pass-the-Hash attacks. The `mysql_native_password` plugin relies on the Secure Hash Algorithm 1 (SHA-1) algorithm. The National Institute of Standards and Technology (NIST) recommends against use of SHA-1.

Additionally, these plugins allow users to set blank passwords, which allow authentication without providing a password.

All users should be using alternative, stronger plugins or be configured with invalid passwords. See Default Value section and References for more details on specific account configurations.

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.

Acquisition of user password hashes exposes users to password cracking and Pass-the-Hash attacks.

Setting a root password exposes password-based root access to remote users and to non-root linux users.

Impact:

Using the `ed25519` plugin will require installation of the plugin, and some clients may need to be configured to utilize the `client_ed25519` plugin.

Audit:

Execute the following SQL query to find any users utilizing these plugins and to find special accounts that have been configured with a password:

```
SELECT User,host
FROM mysql.user
WHERE (plugin IN('mysql_native_password', 'mysql_old_password','')
      AND NOT (User = 'root' AND authentication_string = 'invalid')
      AND NOT (User = 'mysql' and authentication_string = 'invalid'));
```

No rows will be returned if all accounts are using strong authentication mechanisms.

Remediation:

If the `root` user is returned in the audit procedure results, set that account to utilize only the `unix_socket` plugin by running the following mariadb command:

```
alter user 'root'@'localhost' identified via 'unix_socket';
```

If the `mysql` user is returned in the audit procedure results, set that account to use an invalid password by running the following mariadb command:

```
set password for 'mysql'@'localhost' = 'invalid';
```

If the `mariadb.sys` user is returned in the audit procedure results, set that account to use an invalid password by running the following mariadb command:

```
set password for 'mariadb.sys'@'localhost' = 'invalid';
```

For every other user identified by the audit procedure, use the `ALTER USER` command to configure the account to utilize one of the following authentication plugins as appropriate:

- `ed25519`
- `gssapi`
- `pam`
- `unix_socket`

Notes:

Some of these plugins will require installation if not already in use. Changing a user to utilize the `ed25519` plugin but without providing a password will make the account inaccessible. For service accounts, set a new password in MariaDB and where the service account is used. For human user accounts, set a temporary password and notify the user to change the password immediately.

If password validation plugins are already implemented, `strict_password_validation` may need to be temporarily disabled to reset `mysql` and `mariadb.sys` accounts to use invalid passwords. To do so, run `set global strict_password_validation=0;` before and `set global strict_password_validation=1;` after the `set password` commands.








Default Value:

`root` is configured to use the `unix_socket` plugin but to fallback to the `mysql_native_password` plugin. `root` and `mysql` users are created with an invalid password string, preventing password-based authentication. `mariadb.sys` is a locked account without a password set. If the account becomes unlocked, authentication without a password can occur. By default, all new users are created using the `mysql_native_password` plugin and without a password unless otherwise specified. This allows authentication without a password.

References:

1. <https://mariadb.com/kb/en/pluggable-authentication-overview/>
2. <https://mariadb.com/kb/en/authentication-from-mariadb-104/>
3. <https://mariadb.com/kb/en/set-password/>
4. <https://mariadb.com/kb/en/alter-user/>
5. <https://mariadb.com/kb/en/create-user/>
6. <https://mariadb.org/history-of-mysql-mariadb-authentication-protocols/>

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.			
v7	16.4 Encrypt or Hash all Authentication Credentials Encrypt or hash with a salt all authentication credentials when stored.			

7.4 Ensure Password Complexity Policies are in Place (Automated)

Profile Applicability:

- Level 1 - MariaDB RDBMS

Description:

Passwords should be configured with a minimum length of 14 characters and should be checked against dictionaries of common, known, and expected passwords.

Rationale:

Long passwords help mitigate dictionary, brute forcing, and other password attacks. Comparing passwords to password dictionaries prevents users from choosing known and easily guessable passwords.

Impact:

Users will not be able to directly set password hashes directly (e.g. SET PASSWORD = ") since this bypasses password validation.

Audit:

Review your mariadb configuration files for the following entries:

```
plugin_load_add = simple_password_check
simple_password_check = FORCE_PLUS_PERMANENT
plugin_load_add = cracklib_password_check
cracklib_password_check = FORCE_PLUS_PERMANENT
```

Ensure mariadb is currently running with the plugin enabled:

```
SHOW PLUGINS;
```

Verify that `simple_password_check` and `cracklib_password_check` both show `ACTIVE` status.

Execute the following SQL statements to assess password policy settings:

```
SHOW VARIABLES LIKE '%pass%';
```

The result set from the above statement should show:

- `simple_password_check_minimal_length` should be 14 or more
- `strict_password_validation` should be ON
- `cracklib_password_check_dictionary` set to an appropriate dictionary file

The dictionary file should contain 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)
- Passwords specific to times of year (e.g. seasons, months)
- Passwords specific to organization interest (e.g. organization or business names, entities, or products)
- Passwords matching usernames

Remediation:

Install the password check plugins:

```
INSTALL SONAME 'simple_password_check';  
INSTALL SONAME 'cracklib_password_check';
```

Note A supporting linux distribution package may need to be installed before installing the cracklib plugin. Follow installation guidance on the Cracklib Password Check Plugin page in the References section.

Add the following lines to MariaDB configuration files:

```
plugin_load_add = simple_password_check  
simple_password_check = FORCE_PLUS_PERMANENT  
simple_password_check_minimal_length = 14  
plugin_load_add = cracklib_password_check  
cracklib_password_check = FORCE_PLUS_PERMANENT  
strict_password_validation = ON
```

Consider customizing the password dictionary to include usernames of all MariaDB users and any other risky passwords patterns noted in the Audit Procedure.

Set `cracklib_password_check_dictionary` if using a customized password dictionary.






Default Value:

Simple Password Check Plugin **and** Cracklib Password Check Plugin **are not installed by default.** `strict_password_validation` is ON by default.

References:

1. <https://mariadb.com/kb/en/password-validation-plugins/>
2. <https://mariadb.com/kb/en/simple-password-check-plugin/>
3. <https://mariadb.com/kb/en/cracklib-password-check-plugin/>

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.2 <u>Use Unique Passwords</u> 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 <u>Use Unique Passwords</u> 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 Ensure No Users Have Wildcard Hostnames (Automated)

Profile Applicability:

- Level 1 - MariaDB RDBMS

Description:

MariaDB 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.			

7.6 Ensure No Anonymous Accounts Exist (Automated)

Profile Applicability:

- Level 1 - MariaDB RDBMS

Description:

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

Rationale:

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

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 `mariadb-secure-installation` utility.

References:

1. https://mariadb.com/kb/en/mysql_secure_installation/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.1 <u>Establish and Maintain an Inventory of Accounts</u> 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 <u>Maintain an Inventory of Accounts</u> Maintain an inventory of all accounts organized by authentication system.			

8 Network

This section contains recommendations related to how MariaDB uses the network.

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

Profile Applicability:

- Level 1 - MariaDB RDBMS

Description:

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

Rationale:

Enabling SSL/TLS will allow clients to encrypt network traffic and verify the identity of the server. SSL/TLS helps to prevent eavesdropping and man-in-the-middle attacks.

Impact:

Enabling SSL/TLS could have impact on network traffic inspection.

Audit:

Execute the following SQL statements to assess this recommendation:
Ensure the `Value` returned is '1' to guarantee insecure connections are rejected:

```
select @@require_secure_transport;
```

Ensure the `Value` returned is `YES` to guarantee that TLS is enabled:

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

Note: In MariaDB 10.0.1 and later, `have_openssl` is NOT an alias for `have_ssl`. In these releases, this variable simply indicates if MariaDB is using OpenSSL instead of MariaDB's bundled TLS library.

Remediation:

Follow the procedures as documented in the MariaDB KnowledgeBase to setup TLS.
In your MariaDB configuration file, enable `require_secure_transport`:

```
require_secure_transport=ON;
```





Default Value:

`require_secure_transport` is disabled (OFF, 0) by default. `have_ssl` defaults to `DISABLED`.

References:

1. <https://mariadb.com/kb/en/secure-connections-overview/>
2. <https://mariadb.com/kb/en/securing-connections-for-client-and-server/>
3. https://mariadb.com/kb/en/server-system-variables/#require_secure_transport
4. https://mariadb.com/kb/en/ssl-tls-system-variables/#have_openssl

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 <u>Encrypt Sensitive Data in Transit</u> Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).			
v7	14.4 <u>Encrypt All Sensitive Information in Transit</u> 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 - MariaDB RDBMS

Description:

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

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

Rationale:

SSL/TLS 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 `ANY`, `X509`, or `SPECIFIED`.

Note: `ANY` means the account must be using TLS but does not require a valid X509 certificate.

Remediation:

Use the `ALTER USER` statement to require the use of SSL/TLS:

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

Note: `REQUIRE SSL` only enforces TLS. There are additional options `REQUIRE X509`, `REQUIRE ISSUER`, `REQUIRE SUBJECT` and `REQUIRE CIPHER` which can be used to further restrict the connection.





Default Value:

The `value` of `ssl_type` defaults to an empty string, the equivalent result of using `REQUIRE NONE` with an `ALTER USER` statement.

References:

1. <https://mariadb.com/kb/en/secure-connections-overview/>
2. <https://mariadb.com/kb/en/ssltls-system-variables/>
3. <https://mariadb.com/kb/en/alter-user/#tls-options>

CIS Controls:

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

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

Profile Applicability:

- Level 1 - MariaDB RDBMS

Description:

Limiting concurrent connections to a MariaDB 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. MariaDB 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 MariaDB database server, run the query:

```
SELECT VARIABLE_NAME, VARIABLE_VALUE
FROM information_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.

To check user-specific settings, run the following:

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

A value of 0 means there is no user-specific limit, that the corresponding global setting applies.

If no limits are configured, this is a fail.

Remediation:

To persist changes to global settings, you must set these variables within MariaDB configuration files.

- To set the global default per-user connection limit, set the `max_user_connections` variable to a numeric value.
- To set the maximum number of clients the server permits to simultaneously connect, set the `max_connections` variable to a numeric value.

You may also set these variables dynamically (and only temporarily) for a running instance of MariaDB by connecting as an administrator and utilizing the commands below.

```
SET GLOBAL max_user_connections=<desired numeric value>;  
SET GLOBAL max_connections=<desired numeric value>;
```

Additionally, connections limits can be set distinctly for each user using `CREATE` or `ALTER` commands.

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`). By default, users are created without their own distinct connection limits.

References:

1. https://mariadb.com/kb/en/server-system-variables/#max_connections
2. https://mariadb.com/kb/en/server-system-variables/#max_user_connections
3. <https://mariadb.com/kb/en/handling-too-many-connections/>
4. <https://mariadb.com/kb/en/create-user/#resource-limit-options>
5. <https://mariadb.com/kb/en/configuring-mariadb-with-option-files/>

Additional Information:

Global connection limits do not apply to users with `SUPER` or `CONNECTION ADMIN` privileges. `max_user_connections` cannot be set globally if MariaDB is already running with it set to 0.

You can also limit connections to be available only for users with `SUPER` or `CONNECTION ADMIN` privilege by setting `max_user_connections` to -1.

`max_connections` and `max_user_connections` can only be set within specific ranges of values. MariaDB will accept values out of range but silently set them to the closest in-range value. For example:

```

MariaDB [(none)]> set global max_connections=0;
Query OK, 0 rows affected, 1 warning (0.000 sec)

MariaDB [(none)]> SELECT VARIABLE_NAME, VARIABLE_VALUE FROM
information_schema.global_variables WHERE VARIABLE_NAME = 'max_connections';
+-----+-----+
| VARIABLE_NAME | VARIABLE_VALUE |
+-----+-----+
| MAX_CONNECTIONS | 10              |
+-----+-----+
1 row in set (0.001 sec)

MariaDB [(none)]> set global max_connections=999999;
Query OK, 0 rows affected, 1 warning (0.000 sec)

MariaDB [(none)]> SELECT VARIABLE_NAME, VARIABLE_VALUE FROM
information_schema.global_variables WHERE VARIABLE_NAME = 'max_connections';
+-----+-----+
| VARIABLE_NAME | VARIABLE_VALUE |
+-----+-----+
| MAX_CONNECTIONS | 100000         |
+-----+-----+
1 row in set (0.001 sec)

```

9 Replication

Everything related to replicating data from one server to another.

Note that you may see the following terms used interchangeably throughout this section: *primary* (or *master*) and *replica* (or *slave*). MariaDB has historically used *master* and *slave* in replication commands and documentation, but *primary* and *replica* are now preferred terminology for MariaDB. MariaDB has begun but has not completed transitioning terminology. Where possible, this section utilizes *primary* and *replica*, but *master* and *slave* may appear in commands yet to be transitioned. More details on MariaDB's renaming initiative can be found at [MDEV-18777](#).

9.1 Ensure Replication Traffic is Secured (Manual)

Profile Applicability:

- Level 1 - MariaDB 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

If using SSL/TLS, run the following MariaDB command to ensure the REPLICA is utilizing TLS:

```
show replica status\G;
```

Verify `Master_SSL_Allowed` is set to `Yes`.

SSL/TLS mutual authentication is audited in subsequent recommendations. For other implementation options, audit mutual authentication as part of this recommendation.

Remediation:

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

If using SSL/TLS, run the following commands on the REPLICA server(s):

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

Note: The PRIMARY and REPLICA servers must already have SSL/TLS enabled and have each others' CA certificates in their trusted CA certificates files. SSL/TLS mutual authentication procedures are provided in subsequent recommendations. For other implementation options, remediate mutual authentication issues as part of this recommendation.

Default Value:

By default, replication traffic is not secured with encryption or other protections.





References:

1. <https://mariadb.com/kb/en/replication-with-secure-connections/>

Additional Information:

MariaDB provides the ability to secure replication traffic with SSL/TLS. Other security measures would be implemented outside MariaDB, either by adding technology to MariaDB servers or by implementing controls in the broader network environment.

CIS Controls:

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

9.2 Ensure 'MASTER_SSL_VERIFY_SERVER_CERT' is enabled (Automated)

Profile Applicability:

- Level 1 - MariaDB RDBMS

Description:

On the MariaDB `REPLICA`, the setting `MASTER_SSL_VERIFY_SERVER_CERT` indicates whether the `REPLICA` should perform server certificate verification of the `PRIMARY`'s certificate.

Rationale:

When establishing TLS connections, clients use certificate verification to authenticate the server as their intended peer in the connection. In this case, the `REPLICA` (client) should verify the `PRIMARY`'s (server's) certificate to authenticate the `PRIMARY` prior to continuing the connection.

Impact:

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

- `REPLICA` processes need to be stopped by running `STOP REPLICA` 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:

This audit procedure only needs to be run if replication traffic is being secured with SSL/TLS.

To assess this recommendation, issue the following statement:

```
show replica status\G;
```

Verify the value of `Master_SSL_Verify_Server_Cert` is Yes.

Remediation:

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

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

Default Value:

Disabled.


References:

1. https://mariadb.com/kb/en/change-master-to/#master_ssl_verify_server_cert
2. <https://mariadb.com/kb/en/secure-connections-overview/#server-certificate-verification>

Additional Information:

Note that this recommendation only applies if you are utilizing TLS to secure MariaDB replication traffic.

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

Profile Applicability:

- Level 1 - MariaDB RDBMS

Description:

The `SUPER` privilege found in the `mysql.user` table governs the use of a variety of MariaDB 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 MariaDB statements (including statements used to manage passwords). This privilege also provides the ability to configure MariaDB, 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
If you wish to validate permissions in more detail:

```
select * from mysql.user where user='repl'\G
```

The following columns should return `Y`.

```

***** 1. row *****
      Select_priv: Y
      Reload_priv: Y
      Shutdown_priv: Y
      Process_priv: Y
      File_priv: Y
      Grant_priv: Y
      Execute_priv: Y
      Repl_slave_priv: Y
      Repl_client_priv: Y
      Create_user_priv: Y

1 row in set (0.0007 sec)

```

Check Dynamic Privileges :

```
select PRIV from mysql.global_grants where user like 'repl'\G
```

Expected results are:

```

BACKUP_ADMIN
CLONE_ADMIN
PERSIST_RO_VARIABLES_ADMIN
REPLICATION_SLAVE_ADMIN
SYSTEM_VARIABLES_ADMIN

```

Note: Substitute your replication user's name for `repl` in the above queries.

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://mariadb.com/kb/en/grant/#super>

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.			

Controls Version	Control	IG 1	IG 2	IG 3
v7	4.7 <u>Limit Access to Script Tools</u> Limit access to scripting tools (such as Microsoft PowerShell and Python) to only administrative or development users with the need to access those capabilities.		●	●

9.4 Ensure only approved ciphers are used for Replication (Manual)

Profile Applicability:

- Level 2 - MariaDB RDBMS on Linux
- Level 2 - MariaDB RDBMS

Description:

MariaDB supports multiple encryption ciphers that can be used for TLS connections during replication. Ciphers can vary in strength, speed and overhead.

Rationale:

Requiring `REPLICA` servers to utilize strong ciphers when connecting to a `PRIMARY` server protects data in transit.

Impact:

If the `PRIMARY` and `REPLICA` servers don't support common cipher suites, replication will fail.

Audit:

This audit procedure only needs to be run if replication traffic is being secured with SSL/TLS.

To audit this recommendation, run the following command:

```
SHOW REPLICA STATUS\G;
```

Verify that `Master_Ssl_Cipher` is set to a list of approved ciphers and/or cipher suites. If this setting is empty or includes unapproved ciphers or cipher suites, this recommendation has been failed.

Remediation:

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

For example, run:

```
STOP REPLICA; -- required if replication was already running
CHANGE MASTER TO
  MASTER_SSL_CIPHER='ECDHE-ECDSA-AES128-GCM-SHA256';
START REPLICA; -- required if you want to restart replication
```

Default Value:

Empty

References:





1. <https://mariadb.com/kb/en/replication-with-secure-connections>
2. https://mariadb.com/kb/en/change-master-to/#master_ssl_cipher
3. <https://mariadb.com/kb/en/alter-user/#tls-options>

Additional Information:

Note that this recommendation only applies if you are utilizing TLS to secure MariaDB replication traffic.

Note also that you may also be able to restrict the cipher suites accepted by the `PRIMARY` for your replication users than you have set more globally via the `ssl_ciphers` variable to support all clients' cipher suite needs. To do this, run the `ALTER USER` command on your `PRIMARY`, utilizing the `REQUIRE CIPHER` option with value(s) supported by your `REPLICA` as implemented in this recommendation.

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.			

9.5 Ensure mutual TLS is enabled (Manual)

Profile Applicability:

- Level 1 - MariaDB RDBMS

Description:

Mutual TLS (a.k.a. Two-Way TLS) enhances TLS by requiring that both parties authenticate each other when establishing a connection. Mutual TLS adds a requirement (over TLS) that the client provide its certificate so the server can authenticate the client.

Rationale:

By requiring the client to authenticate to the server (in this case, the `REPLICA` to authenticate to the `PRIMARY`), the server (`PRIMARY`) prevents unauthorized clients (`REPLICAS`) from performing replication.

Impact:

The `REPLICA` will need to have TLS enabled to support mutual TLS.

Audit:

This audit procedure only needs to be run if replication traffic is being secured with SSL/TLS.

To audit this recommendation, ensure that the `REPLICA` is configured to provide a client certificate, and that the `PRIMARY` requires certificates for any replication users.

Run the following command on the `REPLICA`:

```
SHOW REPLICA STATUS\G;
```

Verify that `Master_SSL_Cert` is set to the filename where the `REPLICA`'s certificate is stored, and verify that `Master_SSL_Key` is set to the filename where the `REPLICA`'s corresponding private key is stored.

On the `PRIMARY`, run the following command for all replication users, replacing which each such username:

```
select ssl_type from mysql.user where user='<replication user>;
```

Verify that all replication users have `ssl_type` `x509`.

Remediation:

To remediate this setting, you must run the `CHANGE MASTER TO` command on the `REPLICA` with `MASTER_SSL_CERT` and `MASTER_SSL_KEY` set to the paths for the `REPLICA`'s certificate and private key files.

For example, run:

```
STOP REPLICA; -- required if replication was already running
CHANGE MASTER TO
  MASTER_SSL_CERT='/etc/mysql/mariadb.conf.d/certificates/server-cert.pem',
  MASTER_SSL_KEY='/etc/mysql/mariadb.conf.d/certificates/server-key.pem';
START REPLICA; -- required if you want to restart replication
```

If the `PRIMARY` does not require your replication users to provide X.509 certificates, use the `ALTER USER` command with `REQUIRE X509` (and/or optionally `REQUIRE SUBJECT` and/or `REQUIRE ISSUER`) for the user accounts needing remediation.

For example, run:

```
ALTER USER <replication user> REQUIRE X509;
```

Default Value:

Disabled.

References:

1. <https://mariadb.com/kb/en/replication-with-secure-connections/#enabling-two-way-tls-with-change-master>
2. <https://mariadb.com/kb/en/change-master-to/>
3. <https://mariadb.com/kb/en/alter-user/#tls-options>

Additional Information:

Note that this recommendation only applies if you are utilizing TLS to secure MariaDB replication traffic.

Although the `MASTER_SSL_CERT` and `MASTER_SSL_KEY` options imply that a certificate and key owned by the `PRIMARY` should be used, these options should actually be set to use the `REPLICA`'s certificate and key.

When running `ALTER USER`, the `REQUIRE SUBJECT` and `REQUIRE ISSUER` options enforce stronger requirements around the client certificates that a user may utilize.

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.			●
v7	1.7 Deploy Port Level Access Control Utilize port level access control, following 802.1x standards, to control which devices can authenticate to the network. The authentication system shall be tied into the hardware asset inventory data to ensure only authorized devices can connect to the network.		●	●

Appendix: Summary Table

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

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
2.3	Do Not Specify Passwords in the Command Line (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
2.4	Do Not Reuse Usernames (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
2.5	Ensure Non-Default, Unique Cryptographic Material is in Use (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
2.6	Ensure 'password_lifetime' is Less Than or Equal to '365' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
2.7	Lock Out Accounts if Not Currently in Use (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
2.8	Ensure Socket Peer-Credential Authentication is Used Appropriately (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
2.9	Ensure MariaDB is Bound to an IP Address (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
2.10	Limit Accepted Transport Layer Security (TLS) Versions (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
2.11	Require Client-Side Certificates (X.509) (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
2.12	Ensure Only Approved Ciphers are Used (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
3	File Permissions		
3.1	Ensure 'datadir' Has Appropriate Permissions (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
3.2	Ensure 'log_bin_basename' Files Have Appropriate Permissions (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
3.3	Ensure 'log_error' Has Appropriate Permissions (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
3.4	Ensure 'slow_query_log' Has Appropriate Permissions (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
3.5	Ensure 'relay_log_basename' Files Have Appropriate Permissions (Automated)	<input type="checkbox"/>	<input type="checkbox"/>

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
3.6	Ensure 'general_log_file' Has Appropriate Permissions (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
3.7	Ensure SSL Key Files Have Appropriate Permissions (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
3.8	Ensure Plugin Directory Has Appropriate Permissions (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
3.9	Ensure 'server_audit_file_path' Has Appropriate Permissions (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
3.10	Ensure File Key Management Encryption Plugin files have appropriate permissions (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4	General		
4.1	Ensure the Latest Security Patches are Applied (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
4.2	Ensure Example or Test Databases are Not Installed on Production Servers (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.3	Ensure 'allow-suspicious-udfs' is Set to 'OFF' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.4	Harden Usage for 'local_infile' on MariaDB Clients (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.5	Ensure mariadb is Not Started With 'skip-grant-tables' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.6	Ensure Symbolic Links are Disabled (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.7	Ensure the 'secure_file_priv' is Configured Correctly (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.8	Ensure 'sql_mode' Contains 'STRICT_ALL_TABLES' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
4.9	Enable data-at-rest encryption in MariaDB (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
5	MariaDB Permissions		

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
5.1	Ensure Only Administrative Users Have Full Database Access (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
5.2	Ensure 'FILE' is Not Granted to Non-Administrative Users (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
5.3	Ensure 'PROCESS' is Not Granted to Non-Administrative Users (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
5.4	Ensure 'SUPER' is Not Granted to Non-Administrative Users (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
5.5	Ensure 'SHUTDOWN' is Not Granted to Non-Administrative Users (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
5.6	Ensure 'CREATE USER' is Not Granted to Non-Administrative Users (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
5.7	Ensure 'GRANT OPTION' is Not Granted to Non-Administrative Users (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
5.8	Ensure 'REPLICATION SLAVE' is Not Granted to Non-Administrative Users (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
5.9	Ensure DML/DDL Grants are Limited to Specific Databases and Users (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
5.10	Securely Define Stored Procedures and Functions DEFINER and INVOKER (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
6	Auditing and Logging		
6.1	Ensure 'log_error' is configured correctly (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
6.2	Ensure Log Files are Stored on a Non-System Partition (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
6.3	Ensure 'log_warnings' is Set to '2' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
6.4	Ensure Audit Logging Is Enabled (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
6.5	Ensure the Audit Plugin Can't be Unloaded (Automated)	<input type="checkbox"/>	<input type="checkbox"/>

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
6.6	Ensure Binary and Relay Logs are Encrypted (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
7	Authentication		
7.1	Disable use of the mysql_old_password plugin (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
7.2	Ensure Passwords are Not Stored in the Global Configuration (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
7.3	Ensure strong authentication is utilized for all accounts (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
7.4	Ensure Password Complexity Policies are in Place (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
7.5	Ensure No Users Have Wildcard Hostnames (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
7.6	Ensure No Anonymous Accounts Exist (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
8	Network		
8.1	Ensure 'require_secure_transport' is Set to 'ON' and 'have_ssl' is Set to 'YES' (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
8.2	Ensure 'ssl_type' is Set to 'ANY', 'X509', or 'SPECIFIED' for All Remote Users (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
8.3	Set Maximum Connection Limits for Server and per User (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
9	Replication		
9.1	Ensure Replication Traffic is Secured (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
9.2	Ensure 'MASTER_SSL_VERIFY_SERVER_CERT' is enabled (Automated)	<input type="checkbox"/>	<input type="checkbox"/>
9.3	Ensure 'super_priv' is Not Set to 'Y' for Replication Users (Automated)	<input type="checkbox"/>	<input type="checkbox"/>

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
9.4	Ensure only approved ciphers are used for Replication (Manual)	<input type="checkbox"/>	<input type="checkbox"/>
9.5	Ensure mutual TLS is enabled (Manual)	<input type="checkbox"/>	<input type="checkbox"/>

Appendix: Change History

Date	Version	Changes for this version
Mar 6, 2024	1.1.0	Update Default Value (Ticket 20971)
Mar 13, 2024	1.1.0	Include docker-specific recommendations? (Ticket 21173)
Mar 13, 2024	1.1.0	bind_address accepts multiple addresses (Ticket 21174)
Mar 13, 2024	1.1.0	UPDATE 4.9 - Enable data-at-rest encryption in MariaDB (Ticket 21176)
Mar 13, 2024	1.1.0	Ensure Password Complexity Policies are in Place (Ticket 21178)
Mar 15, 2024	1.1.0	Update Default Value (Ticket 20973)