DATABASE CHECKLISTS

CIS MYSQL

1 Operating System Level Configuration
2 Installation and Planning
3 File Permissions and Ownership
4 General
5 MySQL Permissions
6 Auditing and Logging
7 Authentication
8 Network
9 Replication

1.1 1.1 Place Databases on Non-System Partitions

Audit:

Execute the following steps to assess this recommendation:

- Discover the datadir by executing the following SQL statement

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

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

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

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

Remediation:

Perform the following steps to remediate this setting:

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

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

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

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

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

Impact:

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

1.2 Use Dedicated Least Privileged Account for MySQL Daemon/Service

Audit:

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

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

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

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

Remediation:

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

1.3 Disable MySQL Command History

Audit:

Execute the following commands to assess this recommendation:

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

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

Remediation:

Perform the following steps to remediate this setting:

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

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

Default Value:

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

1.4 Verify that 'MYSQL_PWD' Is Not Set

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.

1.5 Disable Interactive Login

Audit:

Execute the following command to assess this recommendation

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

Lack of output implies a finding.

Remediation:

Perform the following steps to remediate this setting:

- Execute one of the following commands in a terminal

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

Impact:

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

2.1 Dedicate Machine Running MySQL

Audit:

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

Remediation:

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

Impact:

Care must be taken that applications or services that are required for the proper operation of the operating system are not removed.
Custom applications may need to be modified to accommodate database connections over the network rather than on the use (e.g., using TCP/IP connections).
Additional hardware and operating system licenses may be required to make the architectural change.

2.2 Do Not Specify Passwords in Command Line

Audit:

Check the process or task list if the password is visible.
Check the shell or command history if the password is visible.

Remediation:

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

Impact:

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

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

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

2.3 Do Not Reuse User Accounts

Audit:

Each user should be linked to one of these

- system accounts
- a person
- an application

Remediation:

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

2.4 Do Not Use Default or Shared Cryptographic Material

Audit:

Review all cryptographic material and check to see if any of it is default or is used for other MySQL instances or for purposes other than MySQL.

Remediation:

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

3.1 Ensure 'datadir' Has Appropriate Permissions and Ownership

Audit:

Perform the following steps to assess this recommendation:

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

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

- Execute the following command at a terminal prompt

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

Lack of output implies a finding.

Remediation:

Execute the following commands at a terminal prompt:

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

3.2 Ensure 'log_bin_basename' Files Have Appropriate Permissions and Ownership

Audit:

Perform the following steps to assess this recommendation:

•Execute the following SQL statement to determine the Value of log_bin_basename

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

•Execute the following command at a terminal prompt to list all log_bin_basename.* files

```
ls <log_bin_basename>.*
```

•For each file listed, execute the following command

```
ls -l <log_bin_basename.nnnnn> | egrep "^-[r|w]{2}-[r|w]{2}----\s*.*$"
```

Lack of output implies a finding.

Remediation:

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

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

Impact:

Changing the permissions and ownership of the log files might impact monitoring tools which use a logfile adapter.

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

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

3.3 Ensure 'log_error' Has Appropriate Permissions and Ownership

Audit:

Perform the following steps to assess this recommendation:

•Execute the following SQL statement to determine the Value of log_error

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

•Execute the following command at a terminal prompt to list all log_error.* files

```
ls <log_error>.*
```

- For each file listed, execute the following command

```
ls -l <log_error> | egrep "^-[r|w]{2}-[r|w]{2}----\s*.*$"
```

Lack of output implies a finding.

Remediation:

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

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

Impact:

Changing the permissions of the log files might impact monitoring tools which use a logfile adapter.

3.4 Ensure 'slow_query_log' Has Appropriate Permissions and Ownership

Audit:

Perform the following steps to assess this recommendation:

•Execute the following SQL statement to determine the Value of slow_query_log_file

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

•Execute the following command at a terminal prompt to list all slow_query_log_file.* files

```
ls <slow_query_log_file>.*
```

•For each file listed, execute the following command

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

Lack of output implies a finding.

Remediation:

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

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

Impact:

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

3.5 Ensure 'relay_log_basename' Files Have Appropriate Permissions and Ownership

Audit:

Execute this SQL statement to determine the value of relay_log_basename:

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

Execute the following command at a terminal prompt to list all relay_log_basename. *files:

```
ls <relay_log_basename>.*
```

For each file listed, execute the following command (Lack of output implies a finding):

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

Remediation:

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

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

3.6 Ensure 'general_log_file' Has Appropriate Permissions and Ownership

Audit:

Execute this SQL statement to determine the value of general_log_file:

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

Execute the following command at a terminal prompt to list all general_log_files. *files:

```
ls <general_log_file>.*
```

For each file listed, execute the following command (Lack of output implies a finding):

```
ls -l <general_log_file> | egrep "^-[r|w]{2}-[r|w]{2}----\s*.*$"
```

Remediation:

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

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

3.7 Ensure SSL Key Files Have Appropriate Permissions and Ownership

Audit:

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

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

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

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

Lack of output from the above command implies a finding.

Remediation:

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

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

3.8 Ensure Plugin Directory Has Appropriate Permissions and Ownership

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

Lack of output implies a finding.

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

Remediation:

To remediate these settings, execute the following commands a ta terminal prompt using the `plugin_dir Value` from the audit procedure.

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

3.9 Ensure 'audit_log_file' has Appropriate Permissions and Ownership

Audit:

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

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

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 `audit_log_file` value):

```
ls -l <audit_log_file> | egrep "^-rw[-x]rw[-x][-r][-w][-x][ \t]*[0-9][ \t]*mysql[
\t]*mysql.*$"
```

Remediation:

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

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

4.1 Ensure Latest Security Patches Are Applied

Audit:

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

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

4.2 Ensure the 'test' Database Is Not Installed

Audit:

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

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

The above SQL statement will return zero rows

Remediation:

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

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

4.4 Ensure 'local_infile' Is Disabled

Audit:

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

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

Remediation:

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

```
local-infile=0
```

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

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

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

4.6 Ensure '------skip---symbolic---links' Is Enabled

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

Ensure the `Value` returned is `DISABLED`.

4.7 Ensure the 'daemon_memcached' Plugin Is Disabled

Audit:

Execute the following SQL statement to assess this recommendation:

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

Ensure that no rows are returned.

Remediation:

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

```
uninstall plugin daemon_memcached;
```

This uninstalls the memcached plugin from the MySQL server.

4.8 Ensure the 'secure_file_priv' Is Not Empty

Audit:

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

```
SHOW GLOBAL VARIABLES WHERE Variable_name = 'secure_file_priv' AND Value<>'';
```

Note: The Value should contain a valid path.

Remediation:

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

```
secure_file_priv=<path_to_load_directory>
```

4.9 Ensure 'sql_mode' Contains 'STRICT_ALL_TABLES'

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

Ensure that STRICT_ALL_TABLES is in the list returned

Remediation:

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

5.1 Ensure Only Administrative Users Have Full Database Access

Audit:

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

```
SELECT user, host
FROM mysql.user
WHERE (Select_priv = 'Y')
OR (Insert_priv = 'Y')
OR (Update_priv = 'Y')
OR (Delete_priv = 'Y')
OR (Create_priv = 'Y')
OR (Drop_priv = 'Y');
SELECT user, host
FROM mysql.db
WHERE db = 'mysql'
AND ((Select_priv = 'Y')
OR (Insert_priv = 'Y')
OR (Update_priv = 'Y')
OR (Delete_priv = 'Y')
OR (Create_priv = 'Y')
OR (Drop_priv = 'Y'));
```

Ensure all users returned are administrative users.

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

Audit:

Execute the following SQL statement to audit this setting

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

Ensure only administrative users are returned in the result set.

Remediation:

Perform the following steps to remediate this setting:

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

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

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

Audit:

Execute the following SQL statement to audit this setting:

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

Ensure only administrative users are returned in the result set.

Remediation:

Perform the following steps to remediate this setting:

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

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

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

Audit:

Execute the following SQL statement to audit this setting:

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

Ensure only administrative users are returned in the result set.

Remediation:

Perform the following steps to remediate this setting:

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

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

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

Audit:

Execute the following SQL statement to audit this setting:

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

Ensure only administrative users are returned in the result set.

Remediation:

Perform the following steps to remediate this setting:

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

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

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

Audit:

Execute the following SQL statement to audit this setting:

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

Ensure only administrative users are returned in the result set.

Remediation:

Perform the following steps to remediate this setting:

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

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

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

Audit:

Execute the following SQL statements to audit this setting:

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

Ensure only administrative users are returned in the result set.

Remediation:

Perform the following steps to remediate this setting:

1. Enumerate the non---administrative users found in the result set 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>';
```

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

Audit:

Execute the following SQL statement to audit this setting:

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

Ensure only administrative users are returned in the result set.

Remediation:

Perform the following steps to remediate this setting:

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

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

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

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

6.1 Ensure 'log_error' Is Not Empty

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

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

`SELECT @@global.log_bin_basename;`

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

6.3 Ensure 'log_warnings' Is Set to '2'

`SHOW GLOBAL VARIABLES LIKE 'log_warnings';`

6.4Ensure 'log---raw' Is Set to 'OFF'

`log-raw = OFF`

6.5 Ensure audit_log_connection_policy is not set to'NONE'

`show variables like '%audit_log_connection_policy%';`

Ensure the value is set to either `ERRORS` or `ALL`.

Remediation:

To remediate this configuration setting, execute one of the following SQL statements:

`set global audit_log_connection_policy = ERRORS`

Or

`set global audit_log_connection_policy = ALL`

6.6 Ensure audit_log_exclude_accounts is set to NULL

`SHOW VARIABLES LIKE '%audit_log_exclude_accounts%';`

Ensure the resulting `audit_log_exclude_accounts` value is `NULL`.

Remediation:

To remediate this configuration setting, execute one of the following SQL statements:

`SET GLOBAL audit_log_exclude_accounts = NULL`

6.7 Ensure audit_log_include_accounts is set to NULL

`SHOW VARIABLES LIKE '%audit_log_include_accounts%';`

Ensure the resulting value is `NULL`.

Remediation:

To remediate this configuration setting, execute one of the following SQL statements:

`SET GLOBAL audit_log_include_accounts = NULL`

6.8 Ensure audit_log_policy is set to log logins

`SHOW GLOBAL VARIABLES LIKE 'audit_log_policy';`

6.9 Ensure audit_log_policy is set to log logins and connections

```
SHOW GLOBAL VARIABLES LIKE 'audit_log_policy';
```

6.10 Ensure audit_log_statement_policy is set to ALL

```
SHOW GLOBAL VARIABLES LIKE 'audit_log_statement_policy';
```

It must return ALL

Remediation:

Add this to the mysqld section of the mysql configuration file and restart the server:

```
audit_log_statement_policy='ALL'
```

6.11 Set audit_log_strategy to SYNCHRONOUS or SEMISYNCRONOUS

```
SHOW GLOBAL VARIABLES LIKE 'audit_log_strategy';
```

The result should be `SYNCHRONOUS` or `SEMISYNCHRONOUS`

6.12 Make sure the audit plugin can't be unloaded

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

The result must be `FORCE_PLUS_PERMANENT`

Ensure the following line is found in the `mysqld` section. If not add this to the mysqld section of the mysql configuration file and restart the server:

```
audit_log = 'FORCE_PLUS_PERMANENT'
```

7.1 Ensure 'old_passwords' Is Not Set to '1'

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

Ensure the `Value` field is not set to `1`.

7.2 Ensure 'secure_auth' is set to 'ON'

SHOW VARIABLES WHERE Variable_name = 'secure_auth';

Remediation:

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

```
secure_auth=ON
```

7.4 Ensure 'sql_mode' Contains 'NO_AUTO_CREATE_USER'

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

Ensure that each result contains **NO_AUTO_CREATE_USER.**

7.5 Ensure Passwords Are Set for All MySQL Accounts

Audit:

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

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

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

Remediation:

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

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

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

7.6 Ensure Password Policy Is in Place

Audit:

Execute the following SQL statements to assess this recommendation:

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

The result set from the above statement should show:

validate_password_length should be 14 or more

validate_password_mixed_case_count should be 1 or more

validate_password_number_count should be 1 or more

validate_password_special_char_count should be 1 or more

validate_password_policy should be `MEDIUM` or `STRONG`

The following lines should be present in the global configuration:

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

Check if users have a password which is identical to the username:

```
SELECT User,Password,Host FROM mysql.user
WHERE password=CONCAT('*', UPPER(SHA1(UNHEX(SHA1(user)))));
```

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

Remediation:

Add to the global configuration:

```
plugin-load=validate_password.so
validate-password=FORCE_PLUS_PERMANENT
validate_password_length=14
validate_password_mixed_case_count=1
validate_password_number_count=1
validate_password_special_char_count=1
validate_password_policy=MEDIUM
```

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

7.7 Ensure No Users Have Wildcard Hostnames

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

7.8 Ensure No Anonymous Accounts Exist

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

8.1 Ensure 'have_ssl' Is Set to 'YES'

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

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

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

Remediation:

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

9.2 Ensure 'MASTER_SSL_VERIFY_SERVER_CERT' Is Set to 'YES' or '1'

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

Remediation:

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

9.3 Ensure 'master_info_repository' Is Set to 'TABLE'

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

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

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

Remediation:

For each replication user, issue the following SQL statement (replace "`repl`" with your replication user's name):

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

9.5 Ensure No Replication Users Have Wildcard Hostnames

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