# Securing MySQL Server

MySQL Server is insecure in it's default configuration. Servers with default configuration have given attackers full access to systems in the past. Securing MySQL Server is an essential part in the server deployment.

Securing MySQL Server involves the following steps.

**1) Overview**
**2) Securing root & anonymous users**
**3) Securing test Database**
**4) Removing anonymous MySQL accounts (OPTIONAL)**
**5) Removing test database (OPTIONAL)**
**6) Disabling MySQL Server history file**
**7) Disabling remote login**
**8) Changing the default MySQL Server administrator name (OPTIONAL)**
**9) Do not specify passwords on command line**

## 1) Overview

Before securing our MySQL Server, we must have an overview of the server security. In other words, we must understand what we are trying to do and why.

In the terminal, login to MySQL Server.

```
# mysql
```

We will be in the mysql shell. And this will be without a password prompt. Because there is no password set.

In the mysql shell, Type ***status***.

```
mysql> status
```

We will see that the current user is ***root@localhost***. That means, we are logged into the MySQL Server as ***root*** user without a password.

Show the existing databases.

```
mysql> show databases;
```

The output would be as below.

```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| test               |
```

```
+--------------------+
```

So we have 4 databases – ***information_schema***, ***mysql***, ***performance_schema***, ***test***

Select the database ***mysql*** to work upon

```
mysql> use mysql;
```

Show the existing tables in mysql.

```
mysql> show tables;
```

The output would be as below

```
+---------------------------+
| Tables_in_mysql           |
+---------------------------+
| columns_priv              |
| db                        |
| event                     |
| func                      |
| general_log               |
| help_category             |
| help_keyword              |
| help_relation             |
| help_topic                |
| innodb_index_stats        |
| innodb_table_stats        |
| ndb_binlog_index          |
| plugin                    |
| proc                      |
| procs_priv                |
| proxies_priv              |
| servers                   |
| slave_master_info         |
| slave_relay_log_info      |
| slave_worker_info         |
| slow_log                  |
| tables_priv               |
| time_zone                 |
| time_zone_leap_second     |
| time_zone_name            |
| time_zone_transition      |
| time_zone_transition_type |
| user                      |
+---------------------------+
```

The first table is ***db***. Let us see the contents of that table.

```
mysql> describe db;
```

The output would be as below.

```
+-----------------------+---------------+------+-----+---------+-------+
| Field                 | Type          | Null | Key | Default | Extra |
+-----------------------+---------------+------+-----+---------+-------+
| Host                  | char(60)      | NO   | PRI |         |       |
| Db                    | char(64)      | NO   | PRI |         |       |
| User                  | char(16)      | NO   | PRI |         |       |
| Select_priv           | enum('N','Y') | NO   |     | N       |       |
| Insert_priv           | enum('N','Y') | NO   |     | N       |       |
| Update_priv           | enum('N','Y') | NO   |     | N       |       |
| Delete_priv           | enum('N','Y') | NO   |     | N       |       |
| Create_priv           | enum('N','Y') | NO   |     | N       |       |
| Drop_priv             | enum('N','Y') | NO   |     | N       |       |
| Grant_priv            | enum('N','Y') | NO   |     | N       |       |
| References_priv       | enum('N','Y') | NO   |     | N       |       |
| Index_priv            | enum('N','Y') | NO   |     | N       |       |
| Alter_priv            | enum('N','Y') | NO   |     | N       |       |
| Create_tmp_table_priv | enum('N','Y') | NO   |     | N       |       |
```

```
| Lock_tables_priv      | enum('N','Y') | NO   |     | N       |       |
| Create_view_priv      | enum('N','Y') | NO   |     | N       |       |
| Show_view_priv        | enum('N','Y') | NO   |     | N       |       |
| Create_routine_priv   | enum('N','Y') | NO   |     | N       |       |
| Alter_routine_priv    | enum('N','Y') | NO   |     | N       |       |
| Execute_priv          | enum('N','Y') | NO   |     | N       |       |
| Event_priv            | enum('N','Y') | NO   |     | N       |       |
| Trigger_priv          | enum('N','Y') | NO   |     | N       |       |
+-----------------------+---------------+------+-----+---------+-------+
```

We see that the first field is ***Db***. Let us see the contents of field ***Db***.

```
mysql> select Db from mysql.db;
```

The output would be as below.

```
+---------+
| Db      |
+---------+
| test    |
| test_% |
+---------+
```

This entry permits all accounts to access the test database and other databases with names that start with ***test_***. This is true even for accounts that otherwise have no special privileges such as the default anonymous accounts.

Also, we see that the last table is ***user***. Let us see the contents of that table.

```
mysql> describe user;
```

The output would be as below.

```
+------------------------+-----------------------------------+------+-----+---------+------
| Field                  | Type                              | Null | Key | Default | Extra
+------------------------+-----------------------------------+------+-----+---------+------
| Host                   | char(60)                          | NO   | PRI |         |
| User                   | char(16)                          | NO   | PRI |         |
| Password               | char(41)                          | NO   |     |         |
| Select_priv            | enum('N','Y')                     | NO   |     | N       |
| Insert_priv            | enum('N','Y')                     | NO   |     | N       |
| Update_priv            | enum('N','Y')                     | NO   |     | N       |
| Delete_priv            | enum('N','Y')                     | NO   |     | N       |
| Create_priv            | enum('N','Y')                     | NO   |     | N       |
| Drop_priv              | enum('N','Y')                     | NO   |     | N       |
| Reload_priv            | enum('N','Y')                     | NO   |     | N       |
| Shutdown_priv          | enum('N','Y')                     | NO   |     | N       |
| Process_priv           | enum('N','Y')                     | NO   |     | N       |
| File_priv              | enum('N','Y')                     | NO   |     | N       |
| Grant_priv             | enum('N','Y')                     | NO   |     | N       |
| References_priv        | enum('N','Y')                     | NO   |     | N       |
| Index_priv             | enum('N','Y')                     | NO   |     | N       |
| Alter_priv             | enum('N','Y')                     | NO   |     | N       |
| Show_db_priv           | enum('N','Y')                     | NO   |     | N       |
| Super_priv             | enum('N','Y')                     | NO   |     | N       |
| Create_tmp_table_priv  | enum('N','Y')                     | NO   |     | N       |
| Lock_tables_priv       | enum('N','Y')                     | NO   |     | N       |
| Execute_priv           | enum('N','Y')                     | NO   |     | N       |
| Repl_slave_priv        | enum('N','Y')                     | NO   |     | N       |
| Repl_client_priv       | enum('N','Y')                     | NO   |     | N       |
| Create_view_priv       | enum('N','Y')                     | NO   |     | N       |
| Show_view_priv         | enum('N','Y')                     | NO   |     | N       |
| Create_routine_priv    | enum('N','Y')                     | NO   |     | N       |
| Alter_routine_priv     | enum('N','Y')                     | NO   |     | N       |
| Create_user_priv       | enum('N','Y')                     | NO   |     | N       |
| Event_priv             | enum('N','Y')                     | NO   |     | N       |
| Trigger_priv           | enum('N','Y')                     | NO   |     | N       |
| Create_tablespace_priv | enum('N','Y')                     | NO   |     | N       |
| ssl_type               | enum('','ANY','X509','SPECIFIED') | NO   |     |         |
| ssl_cipher             | blob                              | NO   |     | NULL    |
```

```
| x509_issuer             | blob                                | NO   |     | NULL     |
| x509_subject            | blob                                | NO   |     | NULL     |
| max_questions           | int(11) unsigned                    | NO   |     | 0        |
| max_updates             | int(11) unsigned                    | NO   |     | 0        |
| max_connections         | int(11) unsigned                    | NO   |     | 0        |
| max_user_connections    | int(11) unsigned                    | NO   |     | 0        |
| plugin                  | char(64)                            | YES  |     |          |
| authentication_string   | text                                | YES  |     | NULL     |
| password_expired        | enum('N','Y')                       | NO   |     | N        |
+-------------------------+-------------------------------------+------+-----+---------+------
```

We see that the first three fields are ***Host***, ***User*** and ***Password***. Again, let us see the contents of those fields.

```
mysql> select Host,User,Password from user;
```

The output would be as below.

**NOTE: example.com** will be replaced with the hostname of your machine.

```
+-------------+------+----------+
| Host        | User | Password |
+-------------+------+----------+
| localhost   | root |          |
| example.com | root |          |
| 127.0.0.1   | root |          |
| ::1         | root |          |
| localhost   |      |          |
| example.com |      |          |
+-------------+------+----------+
```

This shows there are 4 root and 2 anonymous accounts, all without passwords. This leaves our MySQL Server installation unprotected.
For those wondering why the 4 root accounts, they are all aliases of each other

**localhost** – Hostname of localhost
**example.com** – Hostname of machine
**127.0.0.1** – IPv4 address of localhost
**::1** – IPv6 address of localhost

Finally, there are two more things to take care of.

♦ Our home folder will contain a file named ***.mysql_history*** which contains all the commands we typed in the mysql shell; this also includes any typed passwords also.

♦ The Remote MySQL login feature is enabled; unless you want it on purpose. It can enable an attacker to target the MySQL database by direct TCP/IP connections from other hosts.

We can exit from the MySQL shell.

```
mysql> exit
```

### 2) Securing root & anonymous users

In the terminal, login to MySQL Server as root.

```
# mysql -u root
```

We will be in the mysql shell as root.

Let us assign password to all root accounts.
In the mysql shell, execute the following commands.

**NOTE:**

- Replace ***YOUR PASSWORD*** with what you want to be your password.
- Each root account is different and can have a different password. Unless you know what you are doing, set same password for all root accounts.
- The ***single quotations*** wrapping the usernames, hostnames and passwords are not necessary but will avoid errors when they contains spaces or other characters that are special to mysql interpreter.
- The ***FLUSH*** statement causes the server to reread the grant tables. Without it, the password change remains unnoticed by the server until we restart it.

```
mysql> set password for 'root'@'localhost' = password('YOUR PASSWORD');
```

```
mysql> set password for 'root'@'localhost.localdomain' = password('YOUR PASSWORD');
```

```
mysql> set password for 'root'@'127.0.0.1' = password('YOUR PASSWORD');
```

```
mysql> set password for 'root'@'::1' = password('YOUR PASSWORD');
```

```
mysql> flush privileges;
```

```
mysql> exit
```

OR

```
mysql> update mysql.user set password = password('YOUR PASSWORD') where user = 'root';
```

```
mysql> flush privileges;
```

```
mysql> exit
```

Now, let us assign password to all anonymous accounts.
In the terminal, login to the MySQL Server as root.

```
# mysql -u root -p
```

There will be a password prompt. Enter the root password.

In the mysql shell, execute the following commands.
**NOTE:**

- Replace ***YOUR PASSWORD*** with what you want to be your password.
- Each root account is different and can have a different password. Unless you know what you are doing, set same password for all root accounts.
- The ***single quotations*** wrapping the usernames, hostnames and passwords are not necessary but will avoid errors when they contains spaces or other characters that are special to mysql interpreter.
- The ***FLUSH*** statement causes the server to reread the grant tables. Without it, the password change remains unnoticed by the server until we restart it.

```
mysql> set password for ''@'localhost' = password('YOUR PASSWORD');
```

```
mysql> set password for ''@'localhost.localdomain' = password('YOUR PASSWORD');
```

```
mysql> flush privileges;
```

```
mysql> exit
```

OR

```
mysql> update mysql.user set password = password('YOUR PASSWORD') where user = '';
```

```
mysql> flush privileges;
```

```
mysql> exit
```

### 3) Securing the test Database

The ***mysql.db*** table contains rows that permit access for any user to the test database and other databases with names that start with ***test_***. These rows have an empty ***User*** column value, which for access-checking purposes matches any user name. This means that such databases can be used even by accounts that otherwise possess no privileges. So we must remove any user access to test databases.

In the terminal, login to MySQL Server as root.

```
# mysql -u root -p
```

Enter the password when prompted.

In the mysql shell, execute the following commands.

**NOTE:** The ***FLUSH*** statement causes the server to reread the grant tables. Without it, the password change remains unnoticed by the server until we restart it.

```
mysql> delete from mysql.db where Db like 'test%';
```

```
mysql> flush privileges;
```

```
mysql> exit
```

With this change, only users who have global database privileges or privileges granted explicitly for the test database, can use it.

### 4) Removing anonymous MySQL accounts (OPTIONAL)

We have password protected the anonymous accounts. But if we want to remove them for additional security, we will have to follow the below steps.

In the terminal, login to the MySQL Server as root.

```
# mysql -u root -p
```

Enter the password when prompted.

In the mysql shell, execute the following commands.

```
mysql> drop user ''@'localhost';
```

```
mysql> drop user ''@'localhost.localdomain';
```

```
mysql> exit
```

The anonymous accounts will be removed.

### 5) Removing test database (OPTIONAL)

We have secured the test databases. But if we want to remove them for additional security, we

will have to follow the below steps.

In the terminal, login to MySQL Server as root.

```
# mysql -u root -p
```

Enter the password when prompted.

In the mysql shell, execute the following commands.

```
mysql> drop database test;
```

```
mysql> exit
```

The test database will be removed.

### 6) Disabling MySQL Server history file

Mysql client writes the record of executed statements to a history file. By default this file is named ***.mysql_history*** and is created in our home directory. It will contain all the commands we have typed in mysql shell, including the account passwords. To disable this feature, follow the below steps.

**NOTE:** This is a per user change. You will have to perform the below steps for each user, for whom you wish to disable the history file.

First remove the ***.mysql_history*** file from the user's home directory.
In the terminal, execute the below command.

```
# rm $HOME/.mysql_history
```

Create a softlink for ***.mysql_history*** file to the ***null*** device.

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

In this way, all writings will be done to the null device and nothing will be recorded.

### 7) Disabling remote login

This applies only if the database is going to be used by locally installed PHP applications. This will limit possibilities of attacking the MySQL database by direct TCP/IP connections from other hosts.

```
WARNING:
 This will cause MySQL to stop listening on all TCP/IP ports including 127.0.0.1. Only
local communication will be possible, and it will be through MySQL socket.
   The interesting part comes here; On Unix, MySQL programs treat the host name localhost
specially, in a way that is likely different from what you expect compared to other
network-based programs. For connections to localhost, MySQL programs attempt to connect to
the local server by using a Unix socket file. For all other hosts including 127.0.0.1, ::1,
localhost.localdomain, example.com etc., MySQL always attempt to connect via TCP regardless
of whatever options you give.
   You can always specify the hostname as localhost in your PHP/MySQL programs. But if you
are sure you want to use another hostname, skip this procedure.
```

TIP:
If remote access to the database is required for interactive purposes; for example, typing our own MySQL commands, the SSH protocol can be used instead.

Open the ***/etc/my.cnf*** file. Add the following entry under ***[mysqld]*** section

```
skip-networking
```

Restart the mysql service

```
# service mysql restart
```

### 8) Changing the default MySQL Server administrator name (OPTIONAL)

The default administrator for MySQL Server is ***root***. Knowing this makes it easy to perform brute-force or dictionary attacks on the administrator password. But if we change the name of administrator, the intruder will have to guess not only the password but first and foremost the username. This makes the attack much more difficult. To change the default MySQL Server administrator name, follow the below steps.

In the terminal, login to MySQL Server as root.

```
# mysql -u root -p
```

Enter the password when prompted.

In the mysql shell, execute the following commands.

**NOTE:**

- Replace ***NEWADMIN*** with your new administrator name.
- The ***FLUSH*** statement causes the server to reread the grant tables. Without it, the password change remains unnoticed by the server until we restart it.

```
mysql> update mysql.user set user="NEWADMIN" where user="root";
```

```
mysql> flush privileges;
```

```
mysql> exit
```

### 9) Do not specify passwords on command line

Passwords should not be specified on the command line using the syntax ***mysql -u USERNAME -p PASSWORD***. There is a brief time window during which passwords specified this way can be viewed by other users with process status utilities such as ps.

Instead, tell the program to prompt for the password using the syntax ***mysql -u USERNAME -p***.

After securing the MySQL Server, it is recommended to view the following sections.

**Customizing MySQL Server**

**Basics of MySQL Server**

OR

You may go directly to the following section.

**Installing Apache HTTP Server**



---