1. **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

1. **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.

**3) 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.

**4) 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.

**5) 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.

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.

**6) 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

**7) 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 Serveradministrator 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

**8) Do not specify passwords on command line**

Passwords should not be specified on the command line using the syntax mysql -u USERNAME -p PASSWORD.

Instead, tell the program to prompt for the password using the syntax

mysql -u USERNAME -p.