12 Рекомендаций по безопасности в MySQL / MariaDB для Linux

MySQL является самой популярной в мире системой баз данных с открытым исходным кодом, а MariaDB (форк MySQL) является самой быстрорастущей в мире системой баз данных с открытым исходным кодом. После установки сервера MySQL он небезопасен в конфигурации по умолчанию, и обеспечение его безопасности является одной из важнейших задач общего управления базой данных.

Это будет способствовать усилению и повышению общей безопасности сервера Linux, поскольку злоумышленники всегда сканируют уязвимости в любой части системы, а базы данных в прошлом были ключевыми целевыми областями. Типичным примером является грубое применение пароля **root** для базы данных MySQL.

В этом руководстве мы объясним полезные рекомендации по безопасности **MySQL / MariaDB** для Linux.

1. Безопасная установка MySQL

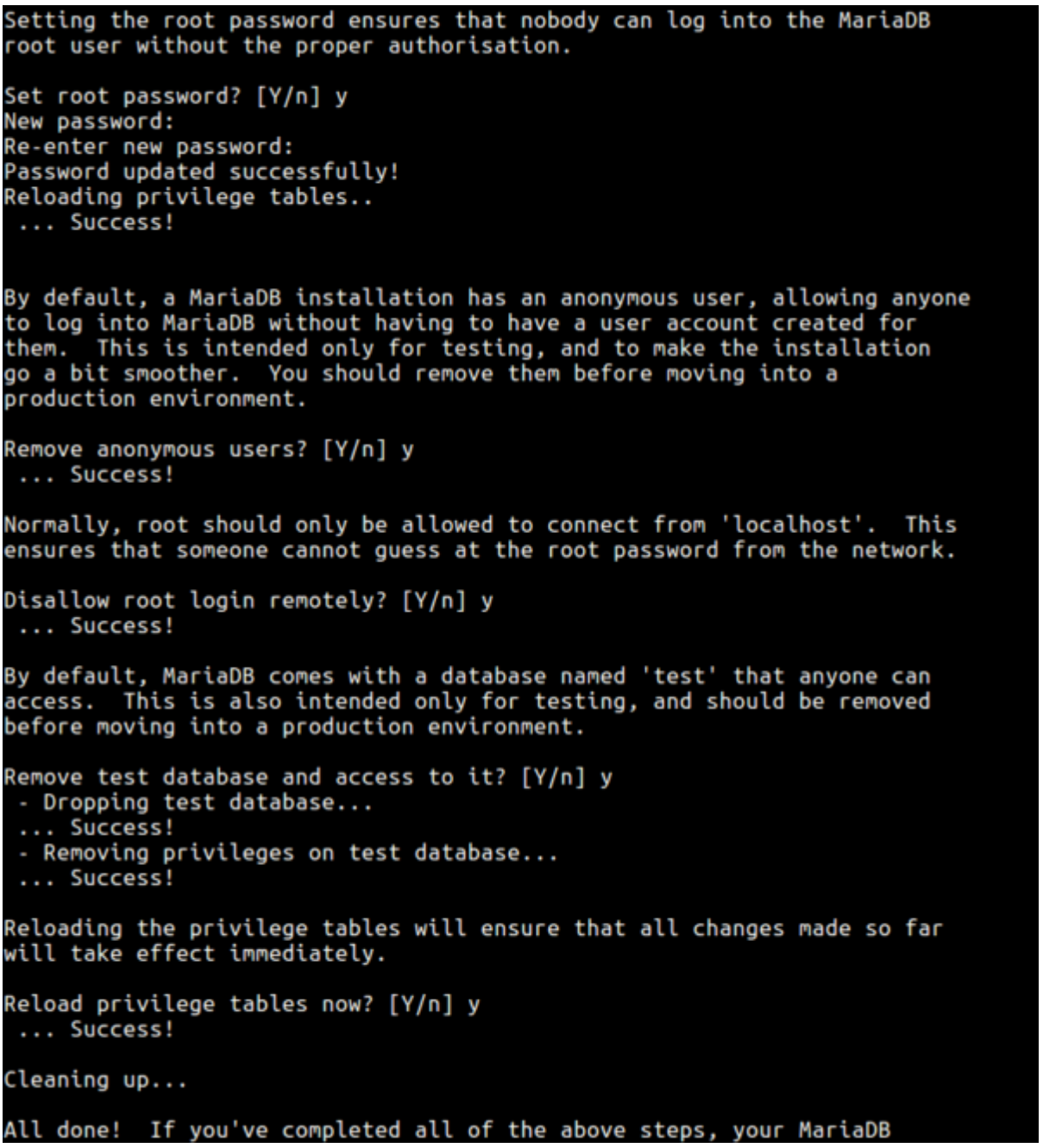
Это первый рекомендуемый шаг после установки сервера MySQL в направлении защиты сервера базы данных. Этот скрипт облегчает повышение безопасности вашего сервера **MySQL**, предлагая вам:

- установить пароль для учетной записи **root**, если вы не установили его во время установки.

- отключитье удаленный вход пользователя root, удалив доступ к учетной записи root вне локального хоста.
- удалить учетные записи анонимных пользователей и проверить базу данных, к которой по умолчанию могут получить доступ все пользователи, даже анонимные.

mysql_secure_installation

После запуска установите пароль **root** и ответьте на ряд вопросов, введя [**Да/Y**] и нажмите [**Enter**].



2. Привязка сервера базы данных к адресу localhost

Эта конфигурация ограничит доступ с удаленных машин, она говорит серверу MySQL принимать соединения только с локального хоста. Вы можете установить его в основном файле конфигурации.

vi /etc/my.cnf [RHEL/CentOS]
vi /etc/mysql/my.conf [Debian/Ubuntu]
или

vi /etc/mysql/mysql.conf.d/mysqld.cnf [Debian/Ubuntu]

Добавьте следующую строку ниже в разделе [**mysqld**].

bind-address = 127.0.0.1

3. Отключить LOCAL INFILE в MySQL

В рамках усиления безопасности необходимо отключить **local-infile**, чтобы предотвратить доступ к базовой файловой системе из **MySQL**, используя следующую директиву в разделе [mysqld].

local-infile=0

4. Измените порт MYSQL по умолчанию

Переменная Port устанавливает номер порта **MySQL**, который будет использоваться для прослушивания соединений TCP/IP. Номер порта по умолчанию — **3306**, но вы можете изменить его в разделе [**mysqld**], как показано.

Port=5000

5. Включить ведение журнала MySQL

Журналы — это один из лучших способов понять, что происходит на сервере, в случае любых атак вы можете легко увидеть любые действия, связанные с вторжением, из файлов журналов. Вы можете включить ведение журнала **MySQL**, добавив следующую переменную в разделе [**mysqld**].

log=/var/log/mysql.log

6. Установите соответствующее разрешение для файлов MySQL.

Убедитесь, что у вас установлены соответствующие разрешения для всех файлов сервера MySQL и каталогов данных. Файл **/etc/my.conf** должен быть доступен для записи только пользователю **root**. Это блокирует других пользователей от изменения конфигурации сервера базы данных.

chmod 644 /etc/my.cnf

7. Удалить историю оболочки MySQL

Все команды, которые вы выполняете в оболочке **MySQL**, хранятся клиентом **mysql** в файле истории: ~/.mysql_history. Это может быть опасно, поскольку для любых создаваемых вами учетных записей все имена пользователей и пароли, введенные в оболочке, будут записаны в файл истории.

cat /dev/null > ~/.mysql_history

8. Не запускайте команды MySQL из командной строки

Как вы уже знаете, все команды, которые вы вводите на терминале, хранятся в файле истории, в зависимости от используемой вами оболочки (например, **~/.bash_history** для bash). Злоумышленник, которому удается получить доступ к этому файлу истории, может легко увидеть любые записанные там пароли.

Настоятельно не рекомендуется вводить пароли в командной строке, примерно так:

mysql -u root -ppassword_

```
aaronkilik@tecmint ~ $ sudo mysql -u root -p=@!#@%$lab
sudo mysql -u root -p=@sudo mysql -u root -p=@@%$lab
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 46
Server version: 10.0.31-MariaDB-0ubuntu0.16.04.2 Ubuntu 16.04

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [mysql]>
```

Когда вы проверите последний раздел файла истории команд, вы увидите введенный выше пароль.

history

```
 2037  sudo mysql -u root -p
 2038  sudo mysql -u root -p=@sudo mysql -u root -p=@@%$lab
 2039  history
aaronkilik@tecmint ~ $
```

Рекомендуемый способ подключения MySQL:

mysql -u root -p
Enter password:

9. Определите пользователей базы данных приложения

Для каждого приложения, работающего на сервере, предоставьте доступ только пользователю, который отвечает за базу данных для данного приложения. Например, если у вас есть сайт **WordPress**, создайте определенного пользователя для базы данных сайта **WordPress** следующим образом.

mysql -u root -p
MariaDB [(none)]> CREATE DATABASE wordpress_db;
MariaDB [(none)]> CREATE USER 'wordpress'@'localhost' IDENTIFIED BY 'wordpress@dmin%!2';

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON wordpress_db.* TO
'wordpress'@'localhost';
MariaDB [(none)]> FLUSH PRIVILEGES;
MariaDB [(none)]> exit
```

и не забывайте всегда удалять учетные записи пользователей, которые больше не управляют какой-либо базой данных приложений на сервере.

10. Используйте дополнительные плагины безопасности и библиотеки

MySQL включает в себя ряд подключаемых модулей безопасности для: аутентификации попыток клиентов подключиться к серверу MySQL, проверки пароля и защиты хранилища для конфиденциальной информации, которые все доступны в бесплатной версии.

Вы можете найти больше здесь:
https://dev.mysql.com/doc/refman/5.7/en/security-plugins.html

11. Регулярно меняйте пароли MySQL

Это распространенный совет по безопасности информации / приложений / системы. Как часто вы будете это делать, будет полностью зависеть от вашей внутренней политики безопасности. Однако это может помешать «**шпионам**», которые могли отслеживать вашу активность в течение длительного периода времени, получить доступ к вашему серверу **mysql**.

```
MariaDB [(none)]> USE mysql;
MariaDB [(none)]> UPDATE user SET password=PASSWORD('YourPasswordHere')
WHERE User='root' AND Host = 'localhost';
MariaDB [(none)]> FLUSH PRIVILEGES;
```

12. Обновляйте MySQL Server регулярно

Настоятельно рекомендуется регулярно обновлять пакеты mysql / mariadb, чтобы быть в курсе обновлений безопасности и исправлений ошибок из репозитория

производителя. Обычно пакеты в стандартных репозиториях операционной системы устарели.

```
# yum update
# apt update
```

После внесения любых изменений в сервер **mysql/mariadb** всегда перезапускайте службу.

```
# systemctl restart mariadb #RHEL/CentOS
# systemctl restart mysql #Debian/Ubuntu
```

Спасибо за уделенное время на прочтение статьи!