

How to Set MySQL MariaDB server root user to be able to connect from any host on the Internet / Solution to ' ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES) '

Author : admin



In this small article, I'll shortly explain on how I setup a Standard default package MariaDB Database server on Debian 10 Buster Linux and how I configured it to be accessible from any hostname on the Internet in order to make connection from remote Developer PC with MySQL GUI SQL administration tools such as MySQL WorkBench / HeidiSQL / Navicat / dbForge as well as the few set-backs experienced in the process (e.g. what was the reason for ' ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES) ' error and its solution.

Setting new or changing old MariaDB (MySQL) root server password

I've setup a brand new MariaDB database (The new free OpenSource software fork of MySQL) mariadb-server-10.3 on a Debian 10, right after the OS was installed with the usual apt command:

```
# apt install mariadb-server
```

Next tep was to change the root access password which was set to empty pass by default, e.g. connected with mysql CLI locally while logged via SSH on server and run:

```
MariaDB [(none)]> mysql -u root -p
```

```
use mysql;  
update user set authentication_string=PASSWORD("MyChosenNewPassword") where  
User='root';
```

There was requirement by the customer, that MySQL server is not only accessed locally but be accessed from any IP address from anywhere on the Internet, so next step was to do so.

Allowing access to MySQL server from Anywhere

Allowing access from any host to MariaDB SQL server is a bad security practice but as the **customer is the King** I've fulfilled this weird wish too, by changing the listener for MariaDB (MySQL) on **Debian 10 codenamed Buster**

changing the default listener
to be not the **default 127.0.0.1** (localhost) but any listener is done by modifying the **bind-address** directive in conf **/etc/mysql/mariadb.conf.d/50-server.cnf**:

```
root@linux:~# vim /etc/mysql/mariadb.conf.d/50-server.cnf
```

Then comment out

```
bind-address = 127.0.0.1
```

and add instead **0.0.0.0 (any listener)**

```
bind-address = 0.0.0.0  
root@linux:/etc/mysql/mariadb.conf.d# grep -i bind-address 50-server.cnf  
##bind-address = 127.0.0.1
```

```
bind-address = 0.0.0.0
```

Then to make the new change effective restart MariaDB (luckily still using the old systemV init script even though systemd is working.

```
root@linux:~# /etc/init.d/mysql restart  
[ ok ] Restarting mysql (via systemctl): mysql.service.
```

To make sure it is properly listening on MySQL defaults TCP port 3306, then as usual used **netcat**.

```
root@pritchi:~# netstat -etna |grep -i 3306  
tcp      0      0 0.0.0.0:3306      0.0.0.0:*        LISTEN    109      1479917
```

By the way the [exact mariadb.cnf used on this middle-sized front-backend server is here](#) - the server is planned to be a Apache Web server + Database host with MySQL DB of a middle range to be able to serve few thousand of simultaneous unique customers.

To make sure no firewall is preventing MariaDB to be accessed, I've checked for any reject rules **iptables** and **ipset** definitions, e.g.:

```
root@linux:~# iptables -L |gre -i rej  
...  
root@linux:~# ipset list  
...
```

Then to double make sure the MySQL is allowed to access from anywhere, used simple telnet from my Desktop Laptop PC (that also runs Debian Linux) towards the server .

```
hipo@jeremiah:~$ telnet 52.88.235.45 3306
Trying 52.88.235.45...
Connected to 52.88.235.45.
Escape character is '^]'.
[
5.5.5-10.3.15-MariaDB-1
rQ}Cs>v\ -G1[W%O>+Y^OQmysql_native_password
Connection closed by foreign host.
```

As telnet is not supporting the data encryption after TCP proto connect, in a few seconds time, remote server connection is terminated.

Setting MySQL user to be able to connect to local server MySQL from any remote hostname

I've connected locally to MariaDB server with **mysql -u root -p** and issued following set of SQL commands to **make MySQL root user be able to connect from anywhere**:

```
CREATE USER 'root'@'%' IDENTIFIED BY 'my-secret-pass';
GRANT ALL ON *.* TO 'root'@'localhost';
GRANT ALL ON *.* TO 'root'@'%';
```

Next step, I've took was to try logging in with root (admin) MariaDB superuser from MySQL **CLI (Command Line Interface) on my desktop** just to find out, I'm **facing a nasty error**.

```
hipo@jeremiah:~$ mysql -u root -H remote-server-hostname.com -p
Enter password:
```

```
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
```

My first guess was something is wrong with my root user created in MySQL's **mysql.user** table (In MySQL this is the privileges table that stores, how MySQL user credentials are handled by **mysqld** local OS running process.

Changing the MySQL root (admin) password no longer possible on Debian 10 Buster?

The standard way ot change the MySQL root password well known via a simple **dpkg-reconfigure** (provided by Debian's **debconf** is no longer working so below command produces empty output instead of triggering the good old Ncurses text based interface well-known over the years ...

```
root@linux:~# /usr/sbin/dpkg-reconfigure mariadb-server-10.3
```

```
...
```

Viewing MariaDB (MySQL) username / password set-up from the CLI

To list how this set-privileges looked like I've used following command:

MariaDB [mysql]> **select * from mysql.user where User = 'root';**

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+
+ Host      | User | Password | Select_priv | Insert_priv | Update_priv |
Delete_priv | Create_priv | Drop_priv | Reload_priv | Shutdown_priv | Process_priv | File_priv |
Grant_priv | References_priv | Index_priv | Alter_priv | Show_db_priv | Super_priv |
Create_tmp_table_priv | Lock_tables_priv | Execute_priv | Repl_slave_priv | Repl_client_priv |
Create_view_priv | Show_view_priv | Create_routine_priv | Alter_routine_priv | Create_user_priv
| Event_priv | Trigger_priv | Create_tablespace_priv | Delete_history_priv | ssl_type | ssl_cipher |
x509_issuer | x509_subject | max_questions | max_updates | max_connections |
max_user_connections | plugin | authentication_string | password_expired | is_role |
default_role | max_statement_time |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+
+ | localhost | root | *E6D338325F50177F2F6A15EDZE932D68C88B8C4F | Y | Y | Y |
Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
Y | Y | Y | Y | Y | Y | Y | Y | Y |
Y | Y | Y | Y | Y | Y | Y | Y | Y |
Y | Y | 0 | 0 | 0 |
| 0 | mysql_native_password | N | N |
0.000000 |
+ % | root | *E6D338325F50177F2F6A15EDZE932D68C88B8C4F | Y | Y |
Y | Y | Y | Y | Y | Y | Y | Y | N | Y |
Y | Y | Y | Y | Y | Y | Y | Y | Y |
Y | Y | Y | Y | Y | Y | Y | Y | Y |
Y | Y | 0 | 0 | 0 |
```

```
|          0 |          |          | N      | N      |          |          | 0.000000 |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
+
```

The hashed (encrypted) password string is being changed from the one on the server, so please don't try to hack me (decrypt it) :)

As it is visible from below output the **Host field** for root has the '%' string which means, any hostname is authorized to be able to connect and login to the MySQL server, so this was not the problem.

After quite some time on reading on **what causes**

' **ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)** ',

I've spend some time reading various forum discussions online on the err such as the one on

[StackOverflow here](#) SuperUser.com's [how to fix access denied for user 'root'@'localhost'](#) and one on

[askubuntu.com's - ERROR 1045\(28000\) : Access denied for user 'root@localhost' \(using password: no \)](#)

and after a while finally got it, thanks to a cool **IRC.FREENODE.NET** guy nicknamed, **hedenface** who pointed me I'm that, I'm trying to use the **-H** flag (**Produce HTML**) instead of **-h** (**host_name**), it seems somehow I ended up with the wrong memory that the **-H** stands for hostname, by simply using **-h** I could again login Hooray!!!

```
root@linux:~$ mysql -u root -h remote-server-host.com -p
```

```
Enter password:
```

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
```

```
Your MariaDB connection id is 14
```

```
Server version: 10.3.15-MariaDB-1 Debian 10
```

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

I've further asked the customer to confirm, he can connect also from his Microsoft Windows 10 PC situated on a different LAN network and got his confirmation. Few notes to make here is I've also installed **phpmyadmin** on the server using phpmyadmin php source code latest version, as in Debian 10 it seems the good old PHP is no longer available (as this crazy developers again made a mess and there is no phpmyadmin .deb package in Debian Buster - but that's a different story I'll perhaps try to document in some small article in future.