MARIA DB HARDENING:

Step 1 : Install MariaDB

sudo apt-key adv --recv-keys --keyserver hkp://keyserver.ubuntu.com:80 0xF1656F24C74CD1D8

sudo echo "deb [arch=amd64,i386,ppc64el] http://ftp.utexas.edu/mariadb/repo/10.1/ubuntu xenial main" >> /etc/apt/sources.list

sudo apt-get update -y

sudo apt-get install mariadb-server -y

sudo systemctl start mysql

sudo systemctl enable mysql

sudo mysql_secure_installation

Step 2 : Create an SSL Certificate and a Private Key for the Server

sudo mkdir /etc/mysql-ssl

sudo cd /etc/mysql-ssl

sudo openssl genrsa 2048 > ca-key.pem

sudo openssl req -new -x509 -nodes -days 365000 -key ca-key.pem -out ca-cert.pem

sudo openssl req -newkey rsa:2048 -days 365 -nodes -keyout server-key.pem -out server-req.pem

sudo sudo openssl rsa -in server-key.pem -out server-key.pem

sudo openssl x509 -req -in server-req.pem -days 365 -CA ca-cert.pem -CAkey ca-key.pem -set_serial 01 -out server-cert.pem

ca-cert.pem ca-key.pem server-cert.pem server-key.pem server-req.pem

Step 3 : Configure MariaDB Server to use SSL

sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf

Add the following lines under the [mysqld] section:

ssl-ca=/etc/mysql-ssl/ca-cert.pem

ssl-cert=/etc/mysql-ssl/server-cert.pem

ssl-key=/etc/mysql-ssl/server-key.pem

##Change this value to connect the MariaDB server from another host.

bind-address = *

sudo systemctl restart mysql

mysql -u root -p

MariaDB [(none)]> SHOW VARIABLES LIKE '%ssl%';

```
sudo systemctl restart mysql
```

Now, you can check whether the SSL configuration is working or not with the following query:

mysql -u root -p

MariaDB [(none)]> SHOW VARIABLES LIKE '%ssl%';

Step 4 : Create a User with SSL Privileges

mysql -u root -p

Next, create user remote and grant privilege to access the server over SSL.

MariaDB [(none)]>GRANT ALL PRIVILEGES ON *.* TO 'remote'@'192.168.0.191' IDENTIFIED BY 'password' REQUIRE SSL;

Then, flush the privileges with the following command:

MariaDB [(none)]>FLUSH PRIVILEGES;

MariaDB [(none)]>exit;

Step 5 : Create The Client Certificate

sudo cd /etc/mysql-ssl

sudo sudo openssl req -newkey rsa:2048 -days 365 -nodes -keyout client-key.pem -out client-req.pem

Next, process the client RSA key with the following command:

sudo openssl rsa -in client-key.pem -out client-key.pem

Finally, sign the client certificate with the following command:

sudo openssl x509 -req -in client-req.pem -days 365 -CA ca-cert.pem -CAkey ca-key.pem -set_serial 01 -out client-cert.pem

Step 6 : Configure MariaDB Client to Use SSL

First, on the client machine, download the key for MariaDB with the following command:

sudo apt-key adv --recv-keys --keyserver hkp://keyserver.ubuntu.com:80 0xF1656F24C74CD1D8

Then, add the MariaDB repository to the /etc/apt/sources.list file:

sudo echo "deb [arch=amd64,i386,ppc64el] http://ftp.utexas.edu/mariadb/repo/10.1/ubuntu xenial main" >> /etc/apt/sources.list

sudo apt-get update -y

sudo mkdir /etc/mysql-ssl

Next, copy all the client certificates from the server machine to the client machine with the following command:

sudo scp root@192.168.0.190:/etc/mysql-ssl/client-* /etc/mysql-ssl/

Then, you will need to configure the MariaDB client to use SSL. You can do this by creating a /etc/mysql/mariadb.conf.d/50-mysql-clients.cnf file:

sudo nano /etc/mysql/mariadb.conf.d/50-mysql-clients.cnf

Add the following lines:

```
[client]
ssl-ca=/etc/mysql-ssl/ca-cert.pem
ssl-cert=/etc/mysql-ssl/client-cert.pem
ssl-key=/etc/mysql-ssl/client-key.pem
```

Step 7 : Verify Remote Connections

On the client machine, run the following command to connect to the MariaDB server:

```
mysql -u remote -h 192.168.0.190 -p mysql
```

Check the status of the connection with the following command:

```
MariaDB [mysql]> status
```