Linux: mininet-2.2.1-150420-ubuntu-14.04-server

HTTP server: Apache2

mysql v: 5.5.62

secure mysql installation with $sudo /usr/bin/mysql\_secure\_installation command

PHP V: 5.5.9

Unit test: PHP Unit 4

ORM: RedBeanPHP 5.5

First I took some time to setup the server.

I got as webserver my good old mininet virtual machine, based on ubuntu 14, used once for openflow SDN studies and experiments.

I decided to use this as my server aiming to gain some time setting up the scenario.

It turned up to be kind a bad decision. It took me far more time to update and configure this server. Anyways, I finally set everything up and running to start writing tests and modelling the appication.

Apache 2 was already installed.

PHP 5.5 already installed

So, install mysql server and, secure it.

mysql\_secure\_installation

install PHP Unit 4 (for PHP 5.5)

install redBeanPHP 5.5 (compatible with PHP 5.5)

the project is hosted at

/var/www/html/Seedbox

./Seedbox : contains the main PHP file (index.php) simulating the user interacrions

./Seedbox/src : contains the classes

./Seedbox/test : contaisn the tests

./Seedbox/database : contains the MySQL project dump

./Seedbox/ssl : contains the server's cert, key and ca

Securing the connection between the client and the mysql server via SSL:

Create a new local certificate authority

sudo openssl genrsa -out /etc/mysql/ca-key.pem 2048

sudo chmod 600 /etc/mysql/ca-key.pem

sudo openssl req -new -key /etc/mysql/ca-key.pem -out /etc/mysql/ca-csr.pem -subj /CN=mysql-CA/

sudo openssl x509 -req -in /etc/mysql/ca-csr.pem -out /etc/mysql/cacert.pem -signkey /etc/mysql/ca-key.pem -days 3650

sudo echo 01 > /etc/mysql/cacert.srl

Generate Server Certificate and Key

sudo openssl genrsa -out /etc/mysql/server-key.pem 2048

sudo chmod 600 /etc/mysql/server-key.pem

sudo openssl req -new -key /etc/mysql/server-key.pem -out /etc/mysql/server-csr.pem -subj /CN=mysql/

sudo openssl x509 -req -in /etc/mysql/server-csr.pem -out /etc/mysql/server-cert.pem -CA /etc/mysql/cacert.pem -CAkey /etc/mysql/ca-key.pem -days 365

Generate Clients Certificate and Key

sudo openssl genrsa -out /etc/mysql/client-key.pem 2048

sudo chmod 600 /etc/mysql/client-key.pem

sudo openssl req -new -key /etc/mysql/client-key.pem -out /etc/mysql/client-csr.pem -subj /CN=mysql/

sudo openssl x509 -req -in /etc/mysql/client-csr.pem -out /etc/mysql/client-cert.pem -CA /etc/mysql/cacert.pem -CAkey /etc/mysql/ca-key.pem -days 365

If you want to have a CA signed SSL certificate, you can purchase a trusted SSL certificate here.

copy the server-side files to the system's default cert directory. In my case: /etc/ssl/certs

Enable MySQL Server SSL Connections

vi /etc/mysql/my.cnf

add:

ssl-ca=/etc/ ssl/certs /cacert.pem

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

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

Save the file and restart the MySQL service:

sudo service restart mysql

To verify that SSL is enabled, login to the MySQL server

mysql -uroot -p

and run the following command:

mysql> show variables LIKE '%ssl%';

The output should be similar to the following one:

+---------------+----------------------------+

| Variable\_name | Value |

+---------------+----------------------------+

| have\_openssl | YES |

| have\_ssl | YES |

| ssl\_ca | /etc/mysql/cacert.pem |

| ssl\_capath | |

| ssl\_cert | /etc/mysql/server-cert.pem |

| ssl\_cipher | |

| ssl\_crl | |

| ssl\_crlpath | |

| ssl\_key | /etc/mysql/server-key.pem |

+---------------+----------------------------+

9 rows in set (0.15 sec)

Enable Remote Connections via SSL

vi /etc/mysql/my.cnf

and change

bind-address = 127.0.0.1

to

bind-address = 0.0.0.0

In order for the changes to take effect, restart the MySQL server again

Verify that your MySQL server listens on all interfaces:

sudo netstat -anp | grep 3306

the output should be similar to the following one:

tcp 0 0 0.0.0.0:3306 0.0.0.0:\* LISTEN 938/mysqld

To enable your MySQL client to connect to the MySQL server -> grant the database user access to the database on the remote server.

GRANT ALL ON database\_name.\* TO dbuser@192.168.1.10 IDENTIFIED BY 'dbuserpassword' REQUIRE SSL;

where 192.168.1.10 is your MySQL client machine IP address.

Copy the following files from your MySQL server to your MySQL client machine:

/etc/mysql/cacert.pem

/etc/mysql/client-cert.pem

/etc/mysql/client-key.pem

Use these files to connect via PHP server.

In my case, as the code was hosted in the same server, i did not use these files in connection string but, in the other hand, i tryed to connect to mysql from an unsecured host and got refused. SSL should secure data transfes against sniffing (tcpdump for exemple).

Unit test command line

/var/www/html/Seedbox# ./phpunit-4.8.36.phar --bootstrap ./src/Email.php ./tests/EmailTest.php

>TestDox

/var/www/html/Seedbox# ./phpunit-4.8.36.phar --bootstrap ./src/Email.php --testdox ./tests/EmailTest.php