Environment setup:

Graphical user interface, text, application, email

Description automatically generated

Building the docker container

Text

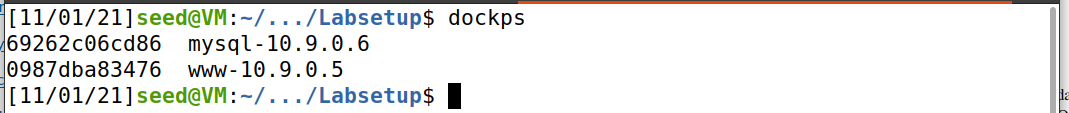
Description automatically generated

Starting up the server

Graphical user interface, application

Description automatically generated

Checking if the server is running



Task1:



Using docksh command we get into the mysql container

Text

Description automatically generated

And then we login to to the mysql using its username and password and get the mysql command line interface

Graphical user interface, text, application, Word

Description automatically generated

We then use the given database using the “use” command

A picture containing graphical user interface

Description automatically generated

Use the “show tables” command to show all available tables in the database

A picture containing chart

Description automatically generated

Then we use the mysql “select” command to view the data in the available table;

Task2:

2.1:

Graphical user interface, application

Description automatically generated

We type in the sql injection attack code ‘or Name=’Admin’;# in the username field so that we can login admin page without knowing his password

Table

Description automatically generated

As we can see the attack code is successful and we are able to login as admin and see his home page

2.2:

Graphical user interface, text, application, email

Description automatically generated

Text, letter

Description automatically generated

We use the curl 'http://www.seed-server.com/unsafe\_home.php?username=%27+or+Name%3D%27Admin%27%3B%23&Password=' command in the terminal to get the data from the server in the terminal about admin’s home page (that is login as admin and show all the details of his homepage in html format). This is sql injection attack using terminal

2.3:



Graphical user interface, application

Description automatically generated

We try to execute the attack code using 2 commands separated by a “;”

Graphical user interface, text, application

Description automatically generated with medium confidence

We can see that the attack doesn’t work as there is a countermeasure that exists which doesn’t allow multiple commands to run when separated by a “;”

Task3:

3.1:

Table

Description automatically generated

We login as alice and view her profile

Graphical user interface, application

Description automatically generated

And then we go to edit profile page to start our attack



This is the attack code that we use for this task

Graphical user interface, application

Description automatically generated Table

Description automatically generated

Then we use the attack code in any of the available fields in alice’s edit profile page and then.

And then we see that once the attack code is executed we were able to change alice’s salary to “123456001”

3.2:

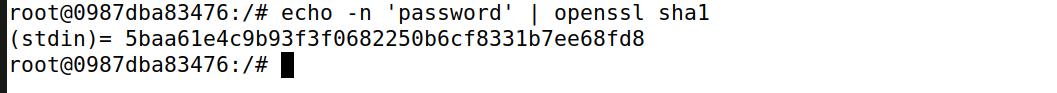
Graphical user interface, application

Description automatically generated Table

Description automatically generated

We use the same attack as in previous task but now we set salary to “1” and eid to boby’s eid that is “20000”

3.3:

In this task we first generate sha1 encrypted form of “password”



This is the attack code that we use for this task

Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated

Chart

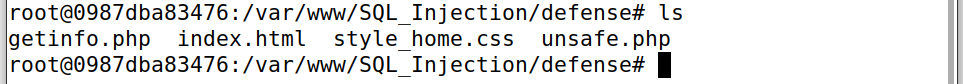
Description automatically generated

Graphical user interface, table

Description automatically generated

We can see above that boby’s password is changed to “password” after the attack code is executed

Task4:





We open the unsafe.php using nano

Text

Description automatically generated

And we edit the query to prepare statement as shown in the above image

Graphical user interface, application, website

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Now when we try to execute the attack code to login as admin without knowing his password and we can see that the attack is not successful and we are not able to retrieve admin’s information as we have used the prepare statement in the code and all the code is sent in code channel and data is sent in data channel. Where as before the code was edited all of the code and data both were being sent in code channel

Graphical user interface, application

Description automatically generated

We try to login as admin using admin’s password aswell

Graphical user interface, text, application, email

Description automatically generated

We see the we are able to get admin’s data when we use admin’s password.