Homework 5

Never use your actual GASPAR password on Com-402 exercises (if the login form is not Tequila).

As you might notice, the solutions are available inside the docker. The goal is of course to solve the exercises without looking at the solutions.

Exercise 1: [attack] P0wn it

You can't imagine the number of bad developers out there. We just found one and we'd like you to hack his website!

Setup

You must first download the docker image containing the website alongside with the database. If you remember the lecture, you should know that having the website and database on the same server is already a sign of trouble...

Use your secret agent email address and run the web server using the following command: `docker run --rm -it -p 80:80 --name hw5ex1 com402/hw5ex1 johnny@english.mi5`

Check that you can access the (rudimentary) website through your navigator. Its IP address is given when running the previous command. Since the website is super rudimentary, here are the three URLs that you can try:

`http://<ip>/`
`http://<ip>/personalities`
`http://<ip>/messages`

Try to access the container directly: `docker exec -it hw5ex1 /bin/bash`

And, look at `/root` to see what you can find ! It seems that SQL injections are possible, right ?! Please notice that in a real scenario a lot of these information are not visible to you. It would make your task harder but this is what it takes to be a real little hacker :)

Exercise 1.1:

You are Johnny English a kindhearted but inept MI5 employee working a desk job while having dreams of being an agent. A 'real' super secret agent, which also happens to be your friend, wrote a super secret message in this website and hopes that you will be able to retrieve it. This message contains a list of persons that he thinks are suspicious. He used his email address:"james@bond.mi5". Unfortunately, and for security reasons, this message was not directly addressed to you and you now have to find it.

You must write a *python3 script* that connects to the IP address of the docker in your machine and *outputs the secret message in stdout*. Your script should use the libraries `requests` (https://realpython.com/python-requests/) and `BeautifulSoup` (https://www.crummy.com/software/BeautifulSoup/bs4/doc/). The former is a library that allows you to send organic, grass-fed HTTP/1.1 requests, without the need for manual labor. The latter is a HTML parsing library enabling you to quickly find the right information in a HTML file. To install them: `pip3 install requests` and `pip3 install beautifulsoup4`.

Hint: You dont always need a form for SQL injections ;)

To verify that you retrieved the correct message compare the result with:
`/root/solutions/PascalSauvage.txt`

Exercise 1.2:

Now that you have a list of suspects, you want to access to the police database and start your research. You happen to know that one famous police inspector, *inspector Derrick*, is also using this website. *Inspector Derrick* is not a big fan of IT technologies and is known for his bad memory. You therefore suspect that he is using the same password for this website and to access criminal records, and you are going to find this password. He is registered in the database under the name "inspector_derrick" but we need access to his password. Your script should connect to `127.0.0.1` and output the password. Please notice that you do not have access to the login page. You should therefore look inside the website files to find how the database that contains passwords is created. You can then reuse the libraries proposed in the first part of the exercise.

To verify that you retrieved the correct password compare the result with:
`/root/solutions/password.txt`

Exercise 2: [defense] No SQL Injection !

You're being asked to build a super minimal website using some data stored in a MySQL database. Of course, you're starting to worry about the security nightmare this project might have.

Well not really, so many libraries do the job for us now.

Pull and run the Docker image:
`docker run -it -p 80:80 --rm --name hw5ex2 com402/hw5ex2 bash`

You will find inside the skeleton script `site.py` where you have to fill in two endpoints `/messages` and `/users`. All information needed is included in `site.py`. The goal is to make sure your script is not susceptible to SQLi attacks. In order to modify the file `site.py`, you have to copy it to your computer, modify it and copy it back to the docker. The copy from the docker can be done with: `docker cp hw5ex2:site.py site.py`. The same command can be used to copy the file back to the docker.

Hint: check MySQL cursor.

Afterwards, execute the modified script:

```
chmod +x site.py
./site.py
```

To verify your solution, run the verify script in another bash shell inside the container:
`docker exec -it hw5ex2 /bin/bash`
to open a new bash shell and then:
`./verif.py randomseed`
If it runs and displays `GOOD`, you are done, congrats !