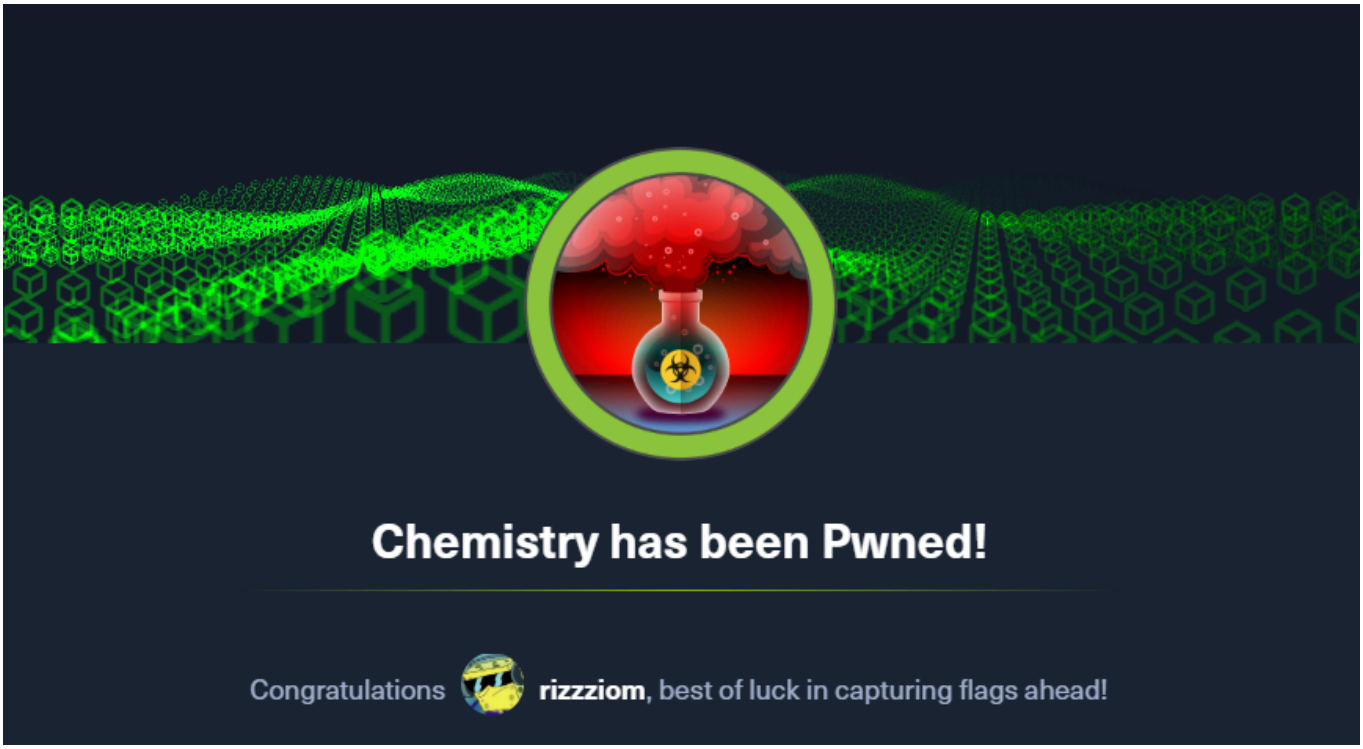
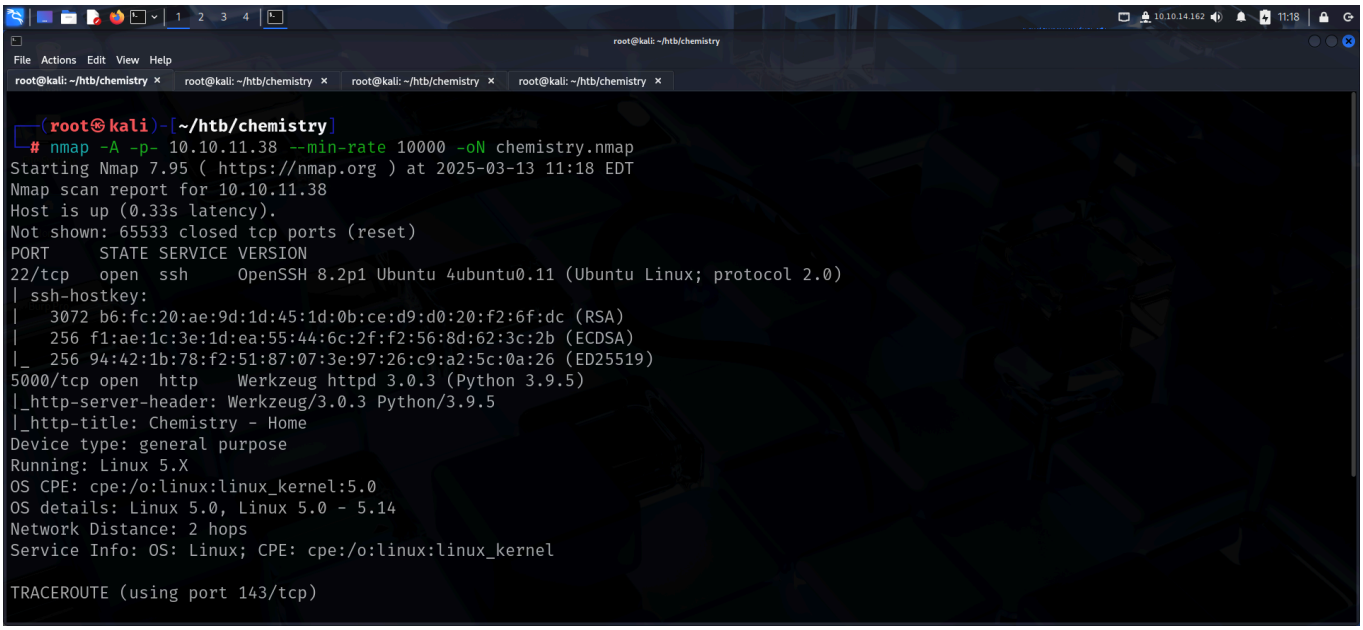
CHEMISTRY



To access the machine click on the link given below:
https://www.hackthebox.com/machines/chemistry

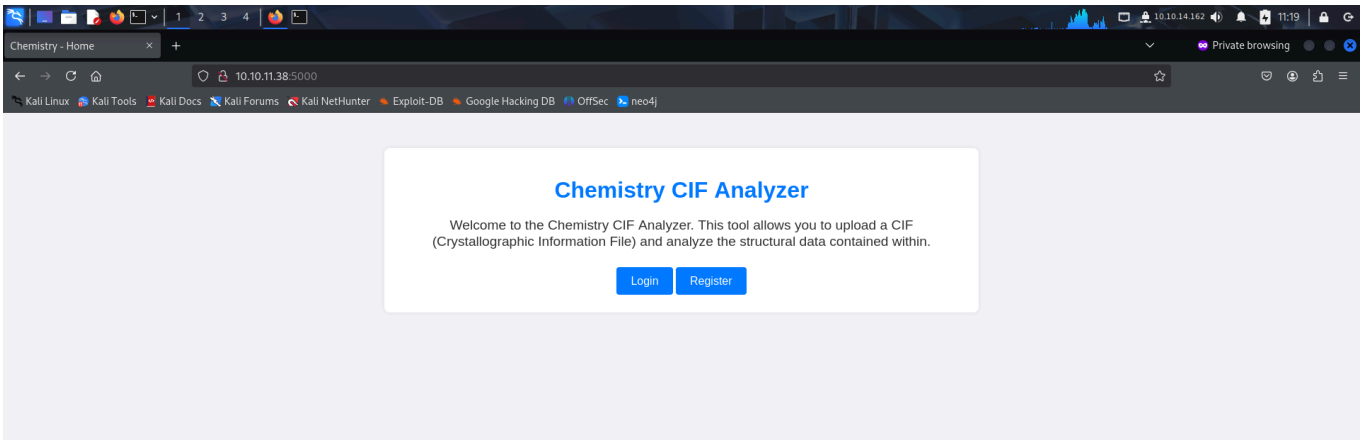
RECONAISSANCE

I performed an **nmap** aggressive scan on the target to find open ports and the services running on them.

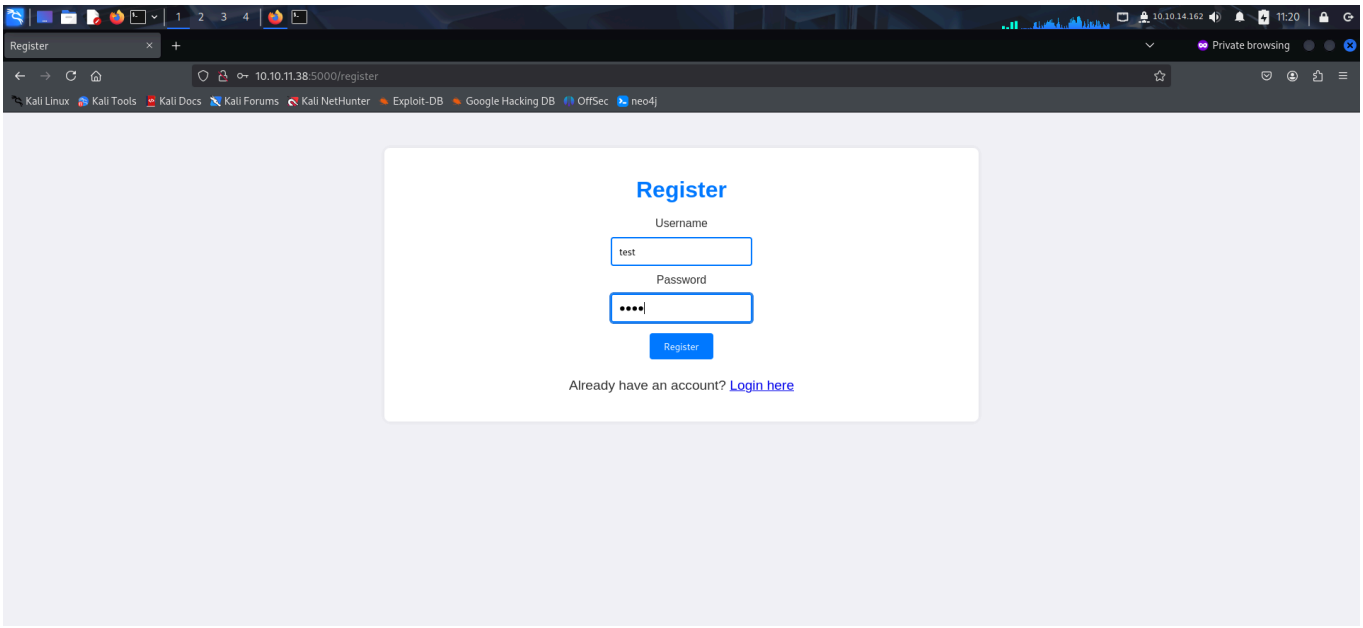


FOOTHOLD

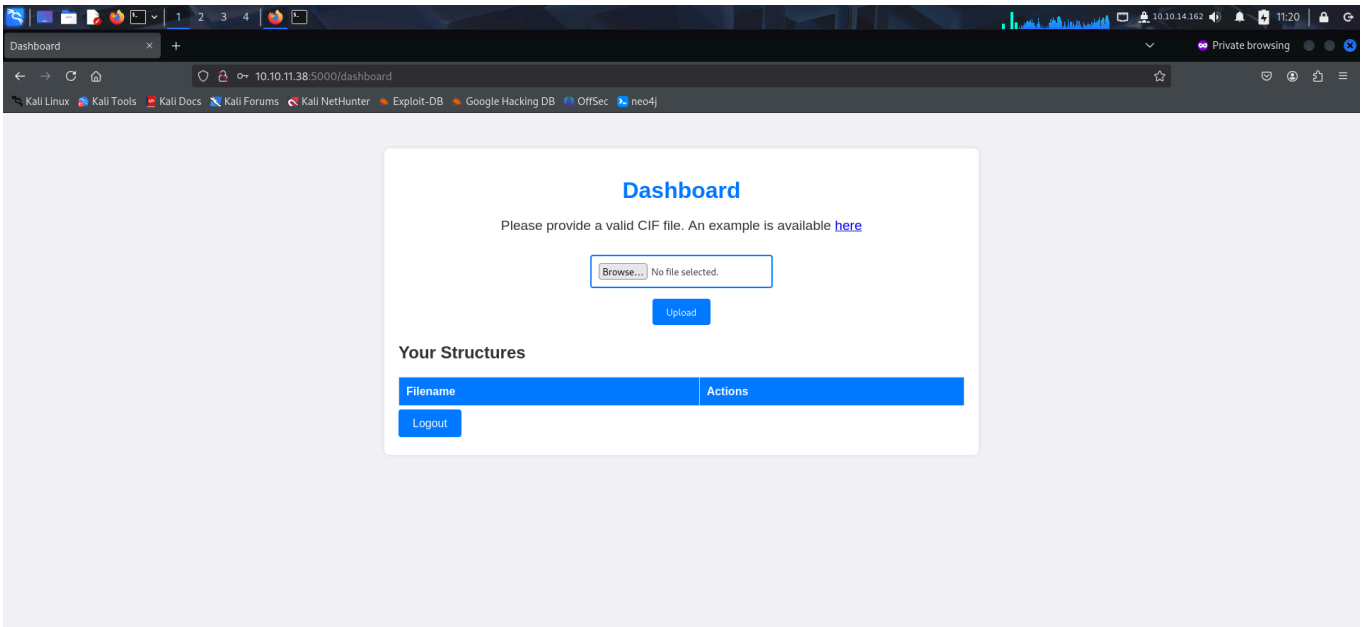
The **nmap** scan revealed only 2 ports, hence I accessed the port running **http** service using **firefox**.



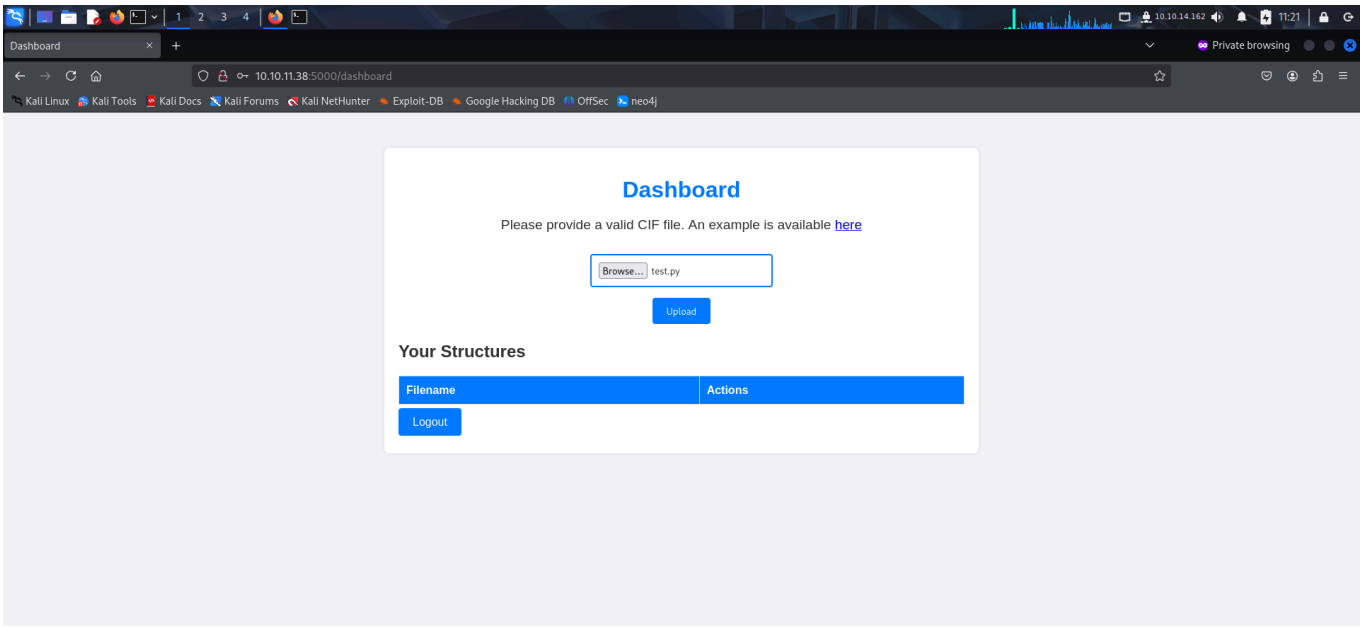
I registered using `test` : `test`

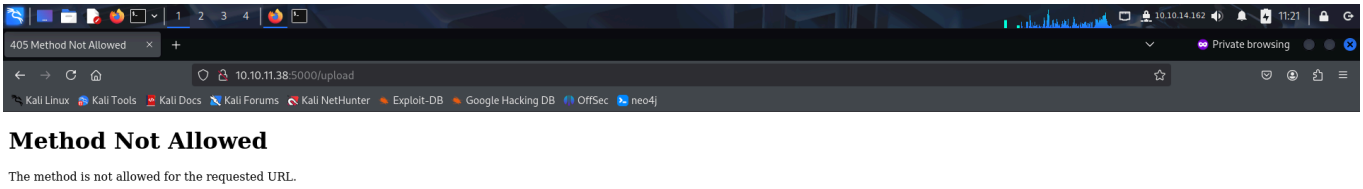


The application allowed us to upload a **CIF** file.

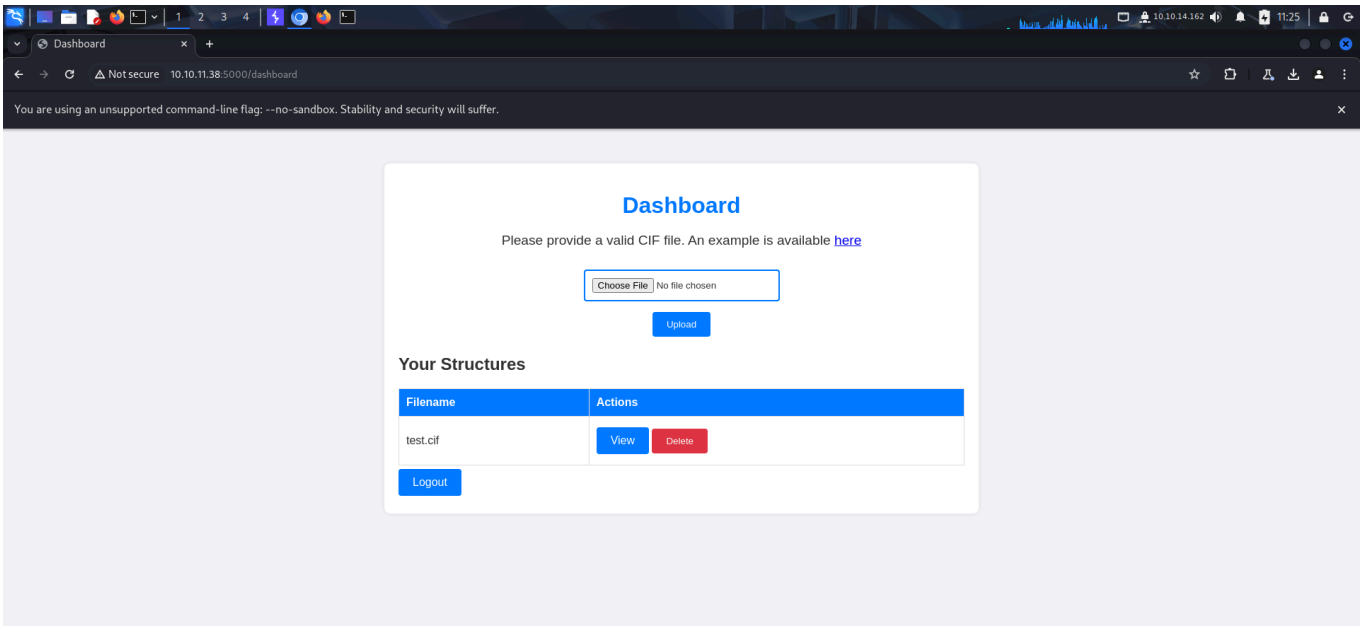


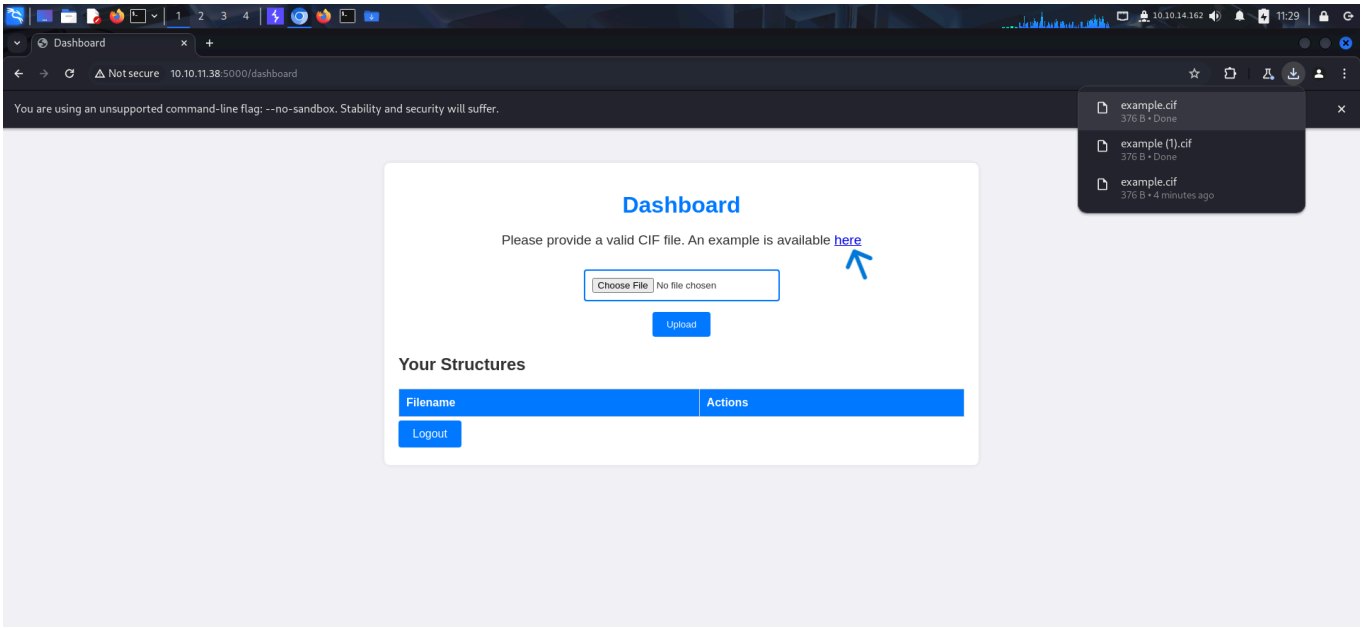
Since the **nmap** scan revealed the backend to be running on **python**, I tried uploading a **python** script but failed.

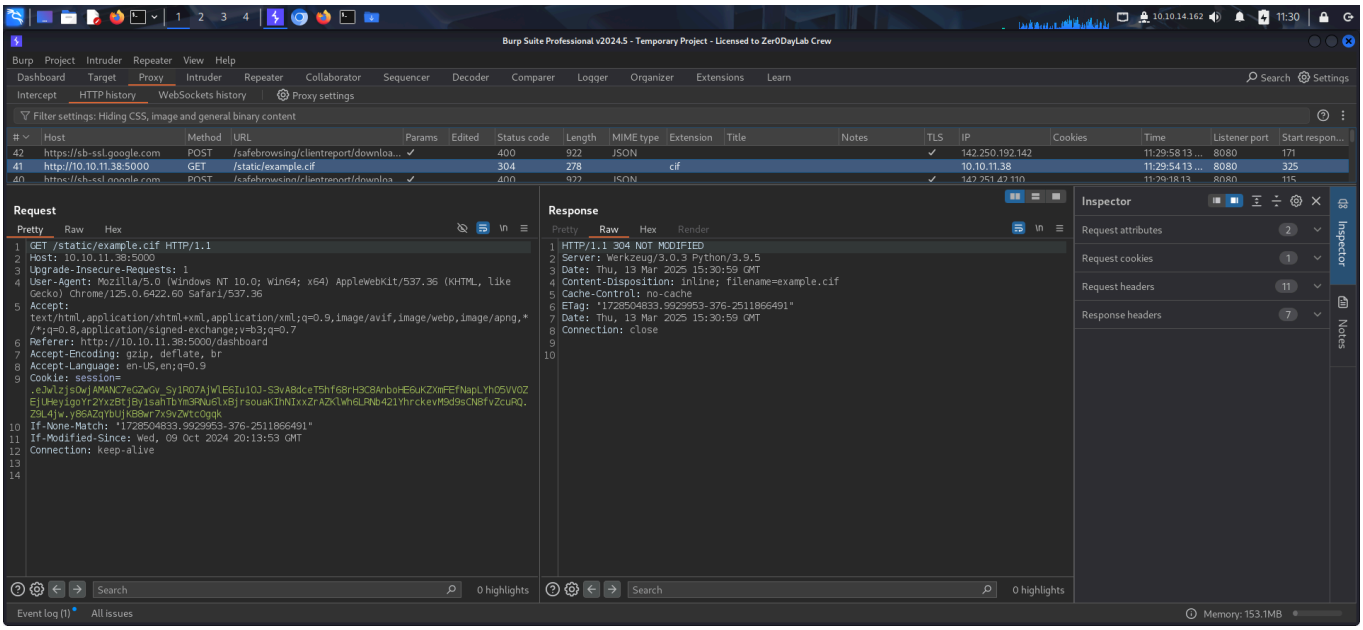




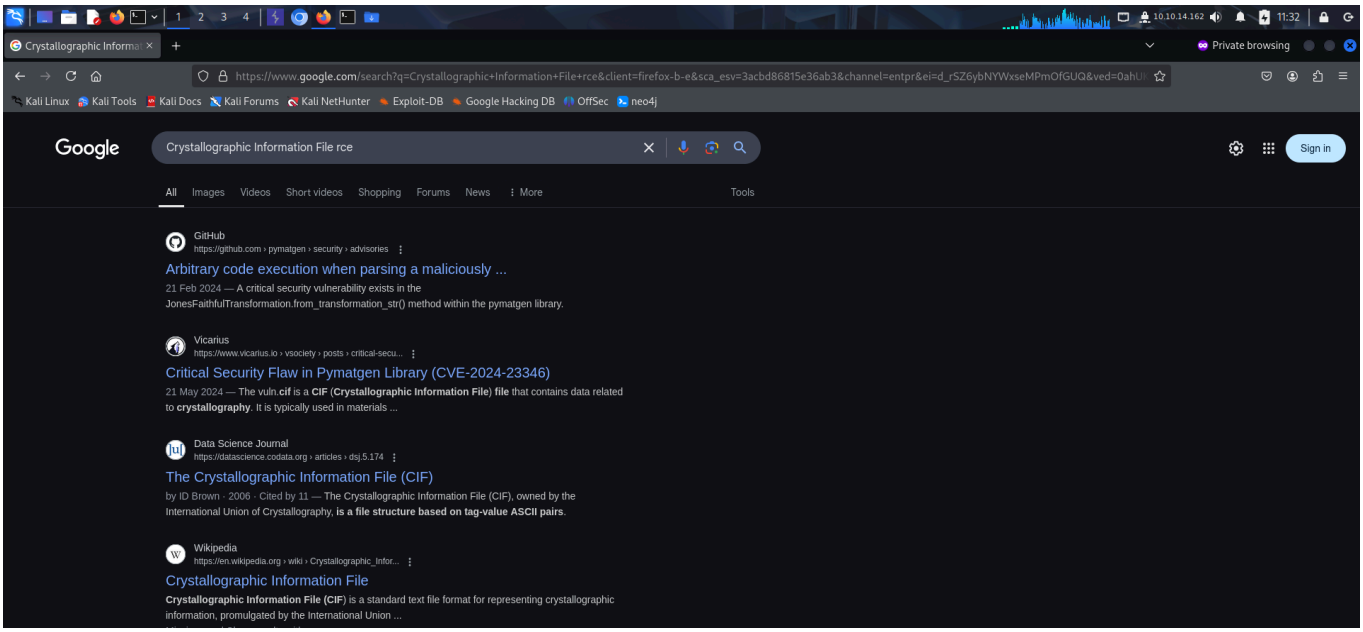
I then uploaded a **CIF** file to analyze the behavior. The application also contained a dummy CIF file for us to download, so I downloaded it.


Dashboard
Not secure 10.10.11.38:5000/dashboard
You are using an unsupported command-line flag: --no-sandbox. Stability and security will suffer.
Dashboard
Please provide a valid CIF file. An example is available here
Choose File No file chosen
Upload
Your Structures
Filename
Actions
test.cif
View
Delete
Logout

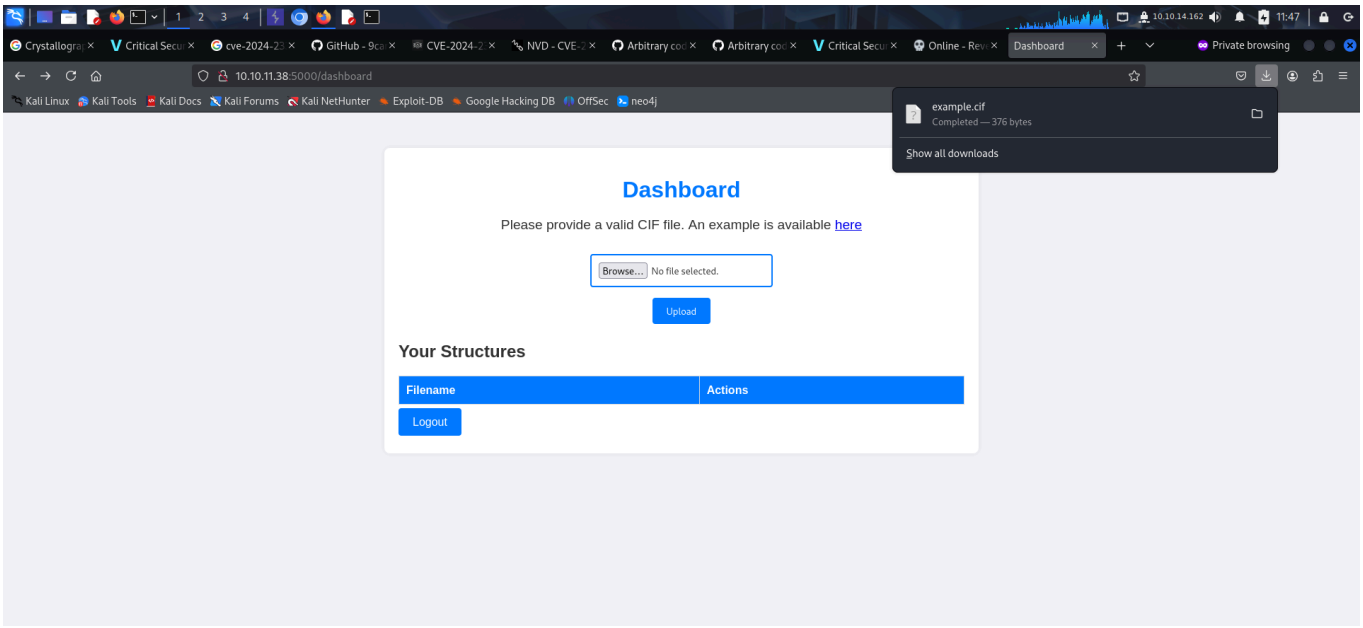

Dashboard
Not secure 10.10.11.38:5000/dashboard
You are using an unsupported command-line flag: --no-sandbox. Stability and security will suffer.
example.cif
376 B • Done
example (1).cif
376 B • Done
example.cif
376 B • 4 minutes ago
Dashboard
Please provide a valid CIF file. An example is available here
Choose File No file chosen
Upload
Your Structures
Filename
Actions
Logout



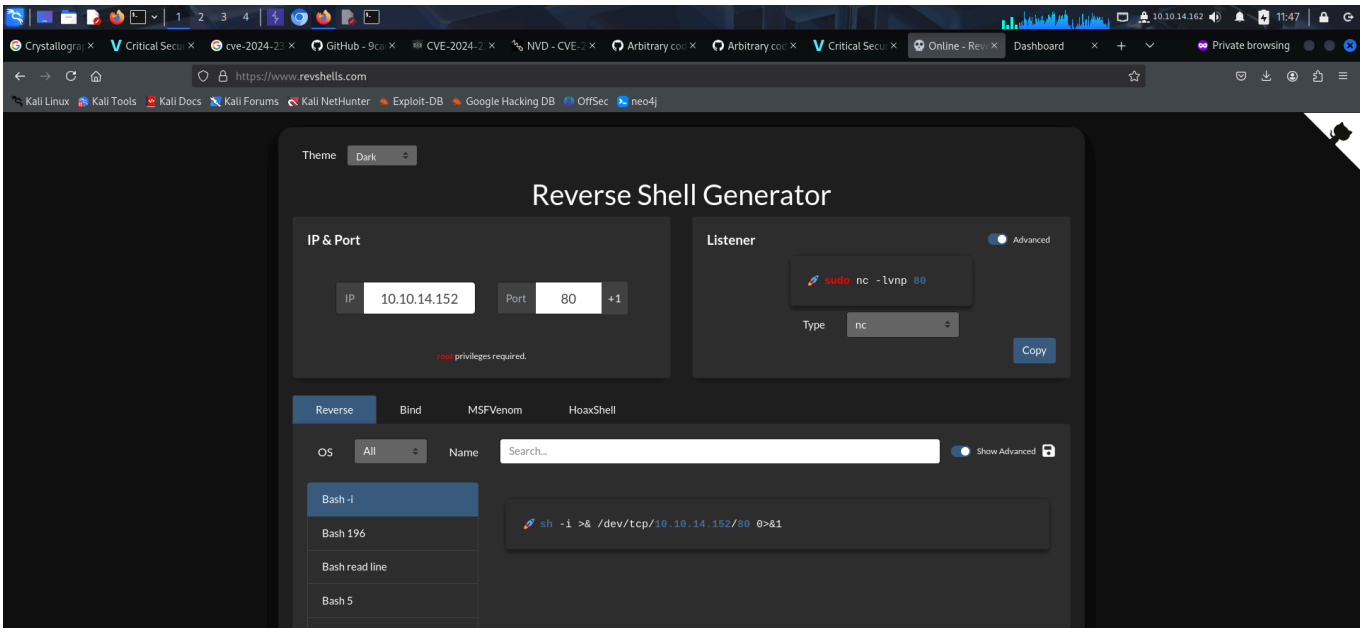
Since I found nothing of interest, I googled vulnerabilities that could be exploited and found interesting articles.



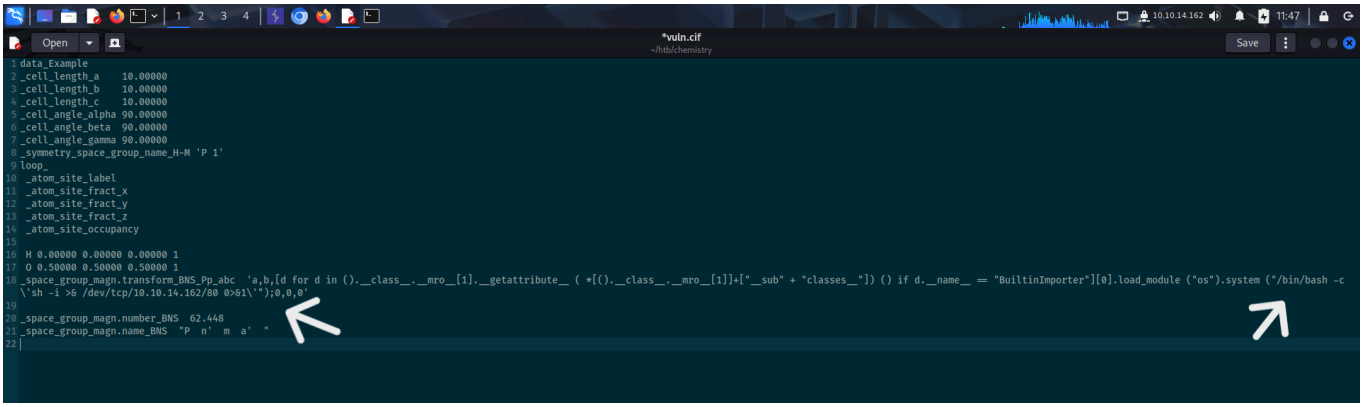
Hence I downloaded the example **cif** file.



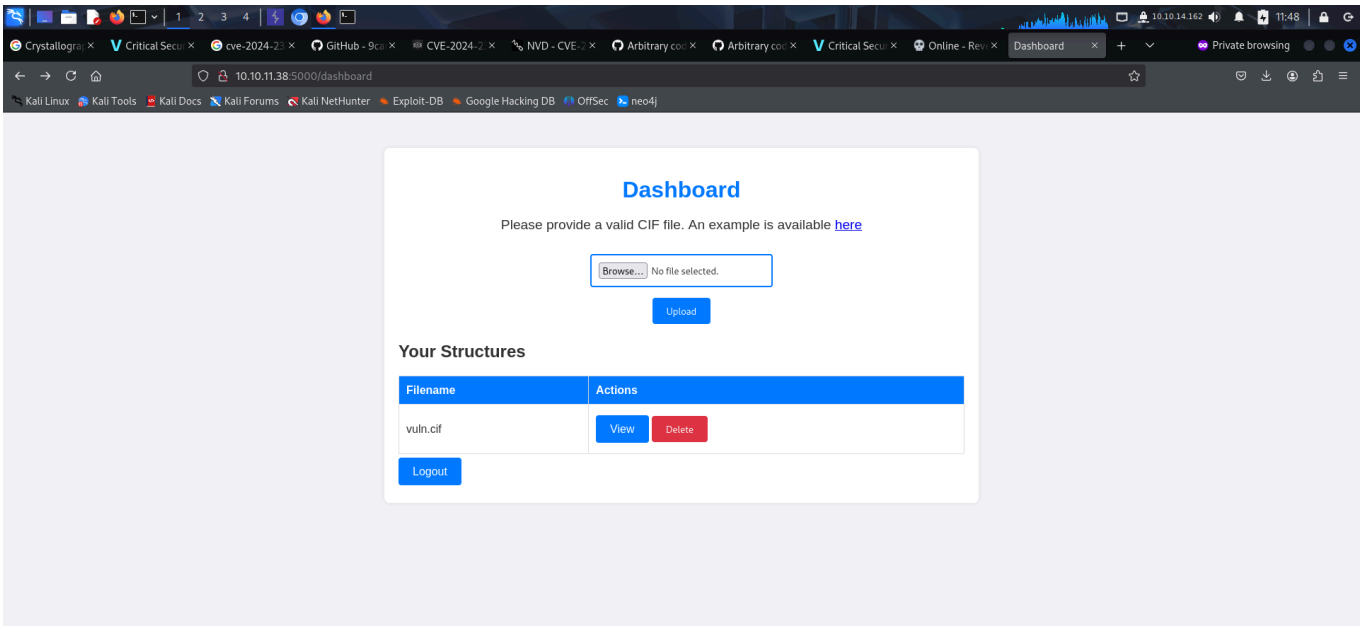
I visited **revshells** and generated a simple **bash** payload.



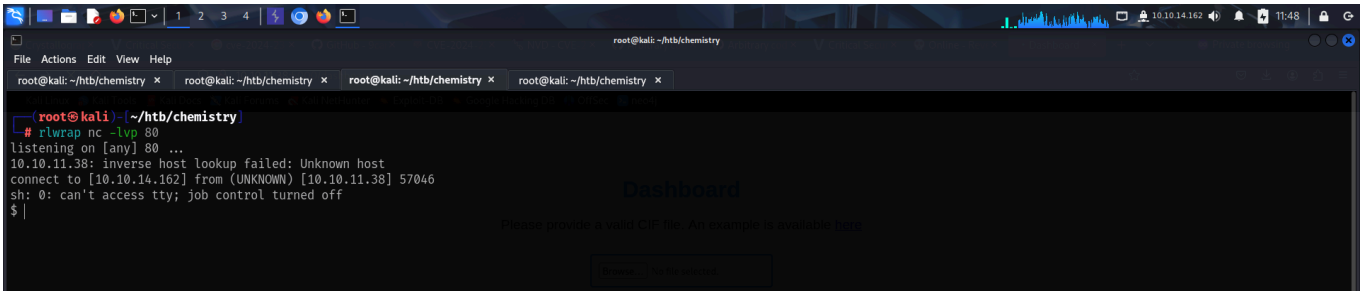
I modified the **CIF** file and added the payload in it.



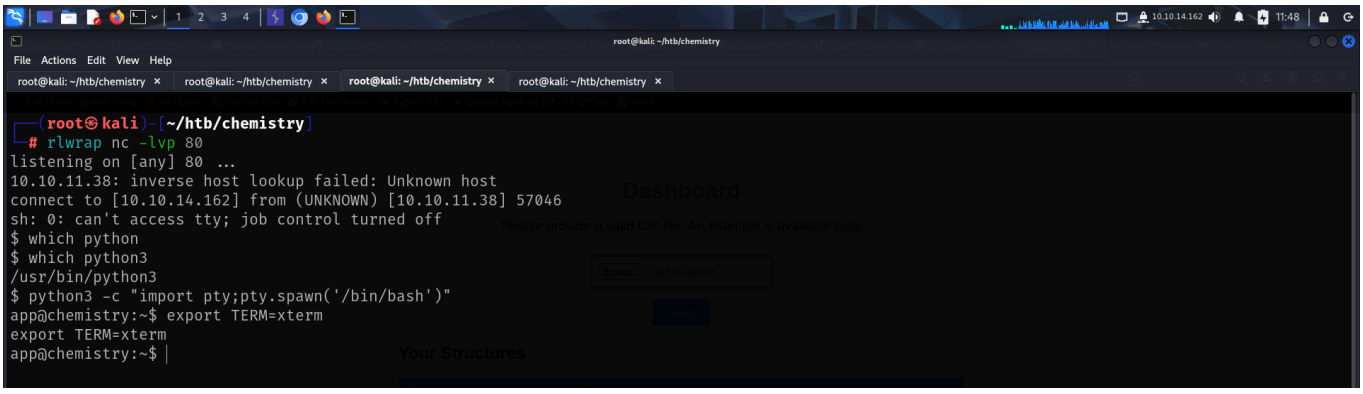
Finally I uploaded the malicious CIF file.



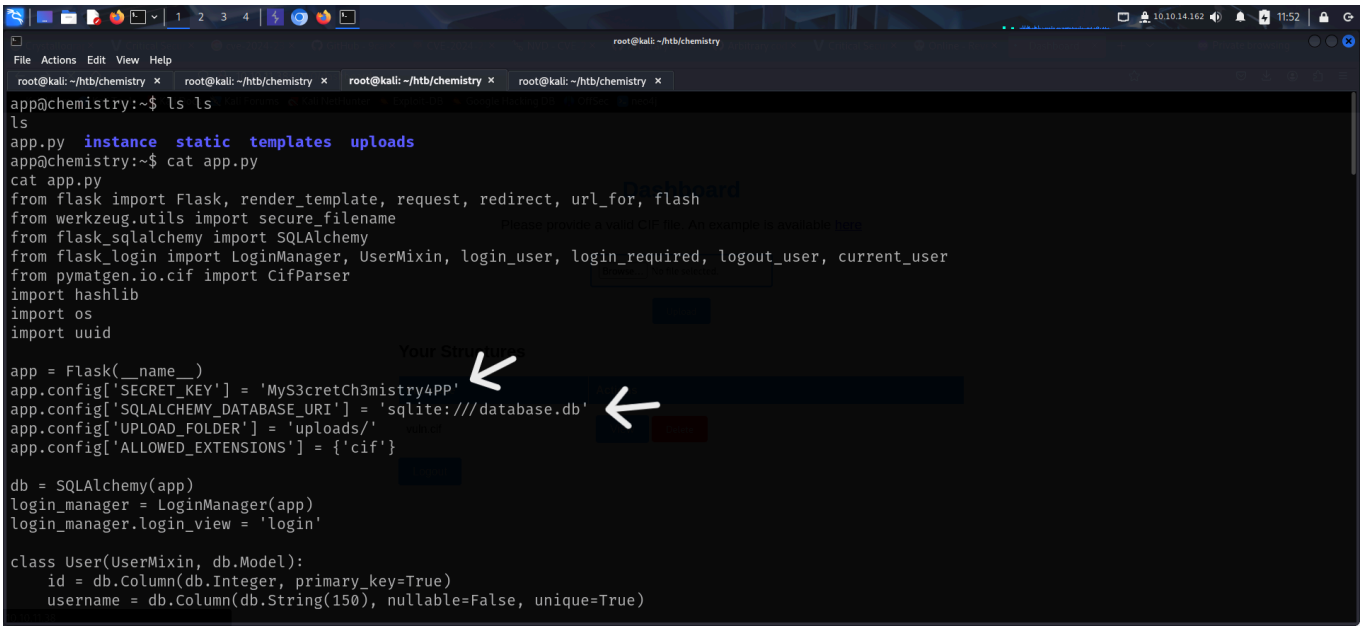
I started a reverse shell listener using **netcat** and when I executed the CIF file, I got a reverse shell.

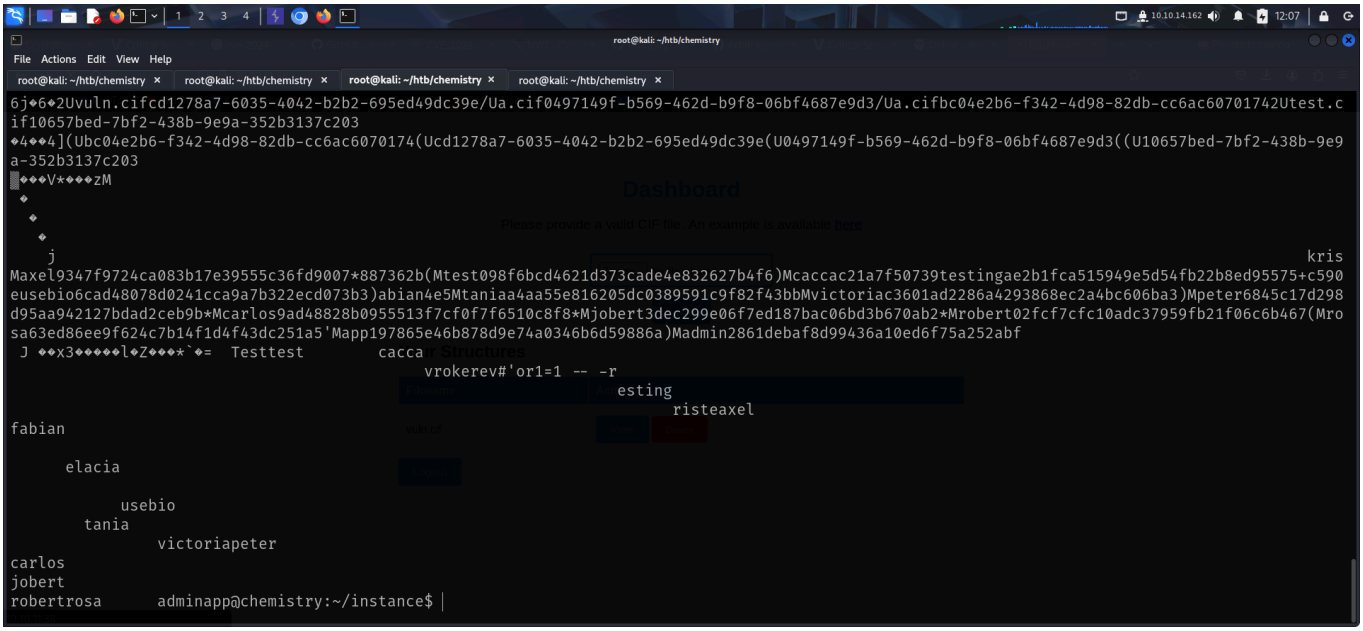


I spawned a **tty** shell and exported my terminal for better shell functionality.

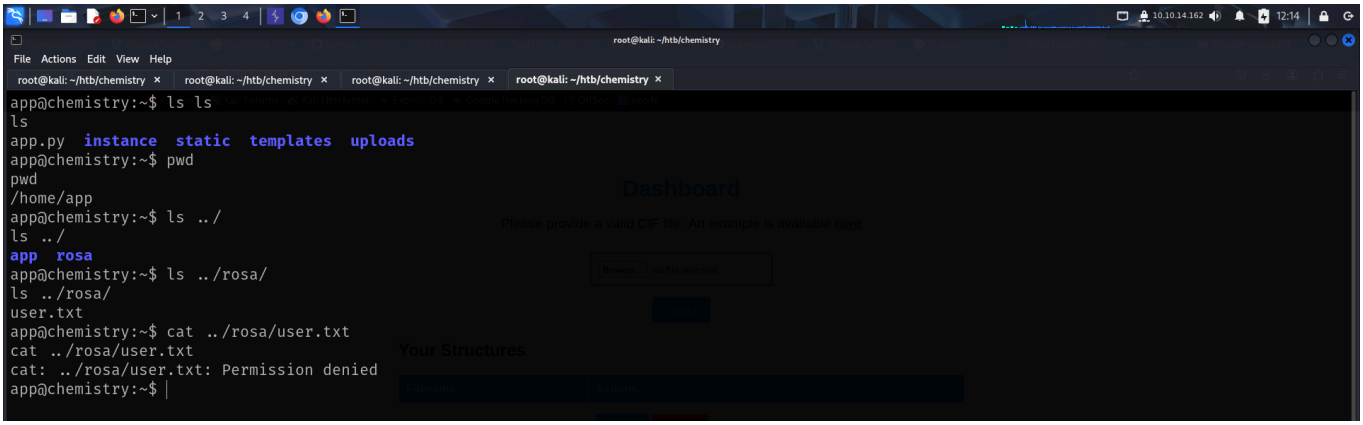


I discovered the existence of an sqlite file along with a password.



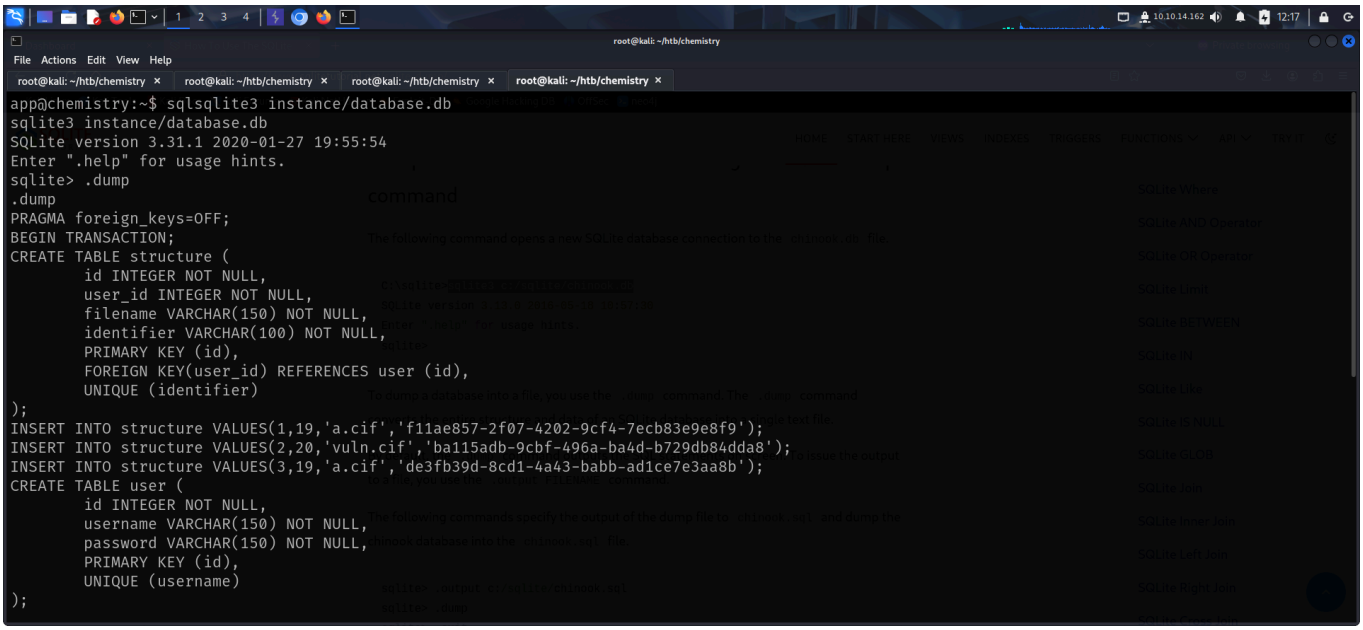


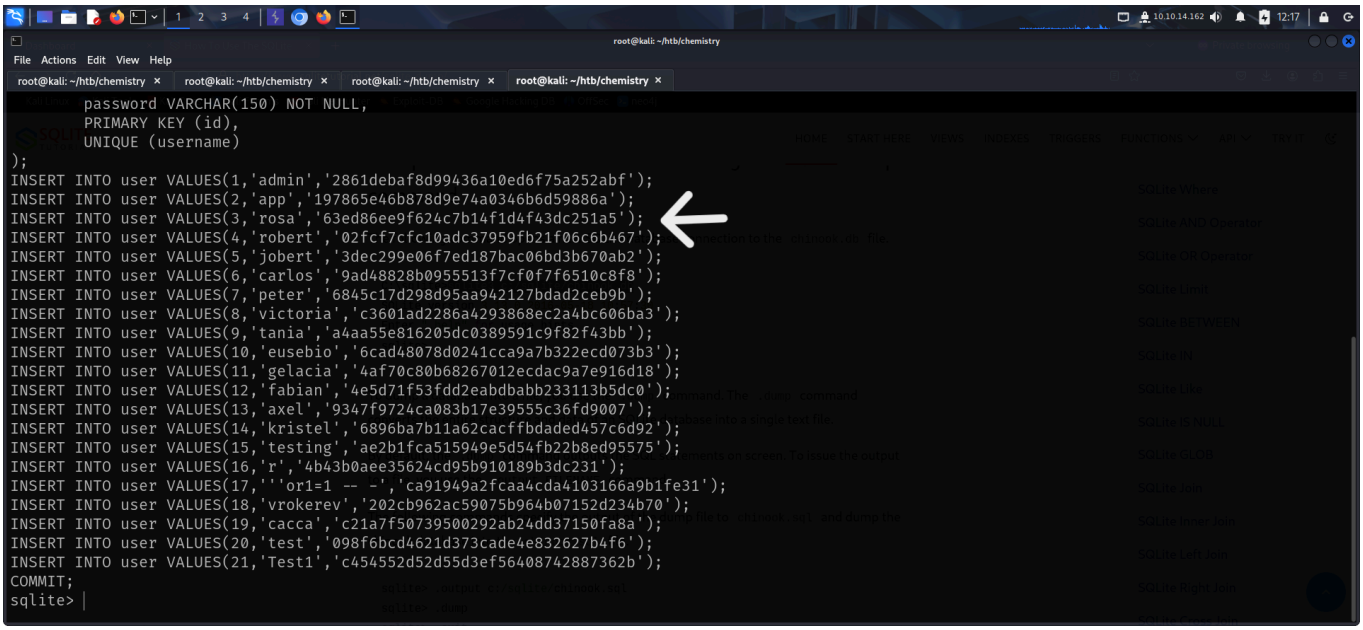
Then I tried looking for the flag. I found it in the home directory of another user called **rosa**. However, since I did not have enough permissions, I could not access it.



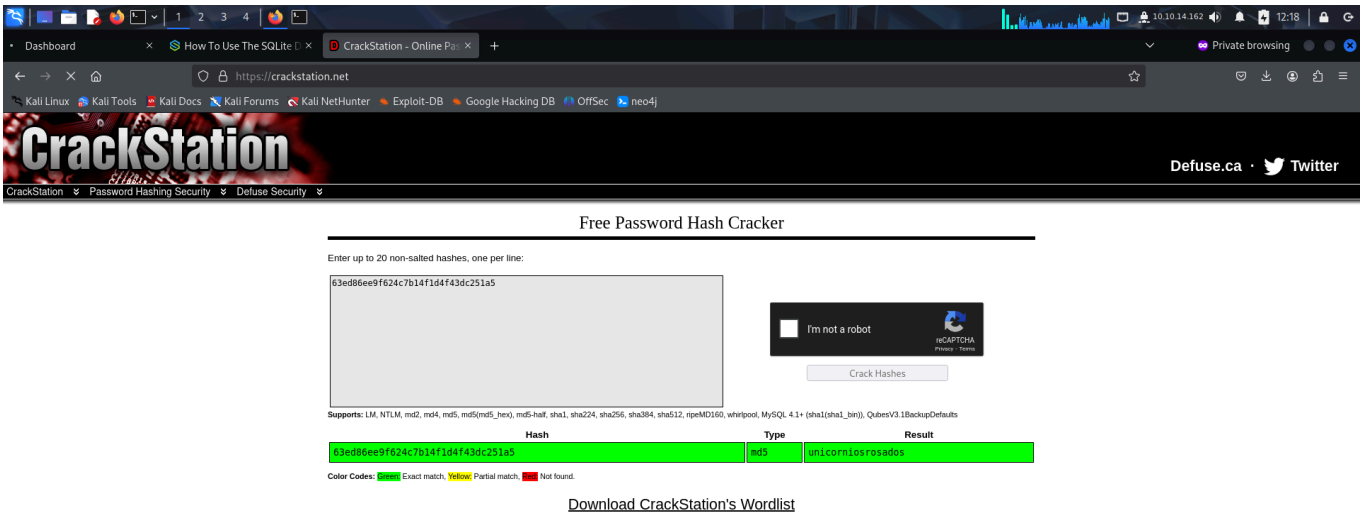
PRIVILEGE ESCALATION

I then viewed the sqlite database file and found the md5 hashes of different users including **rosa**.

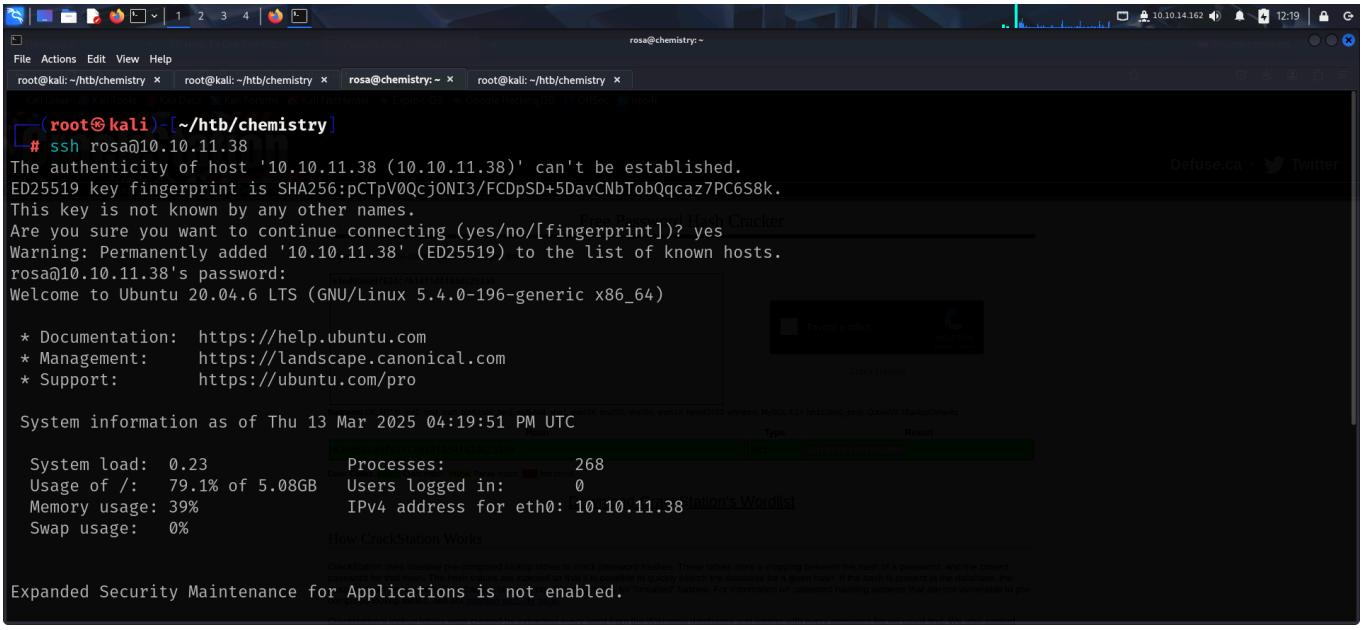


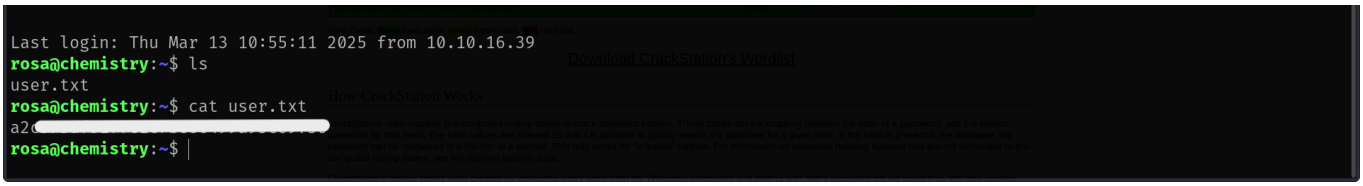


I visited **crackstation** and cracked the password hash.

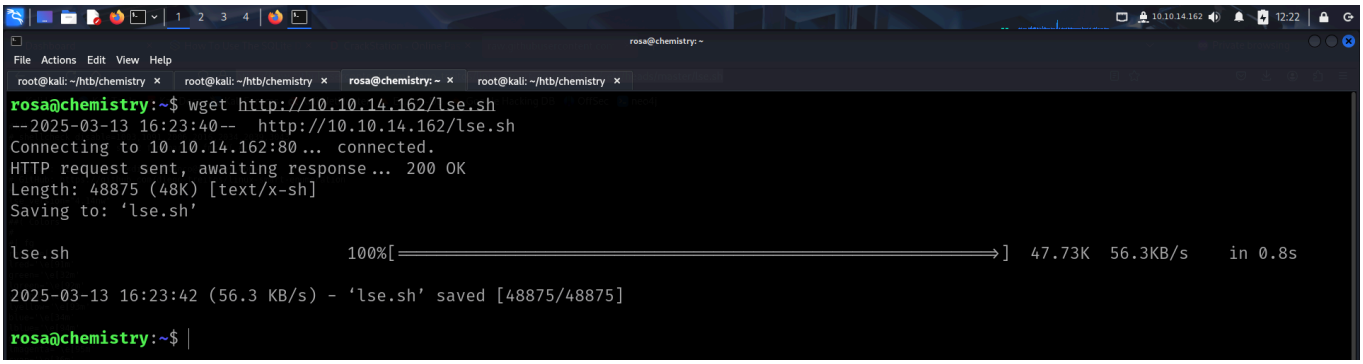


Finally, I used the credentials to log in as **rosa** and captured the user flag from the home directory.

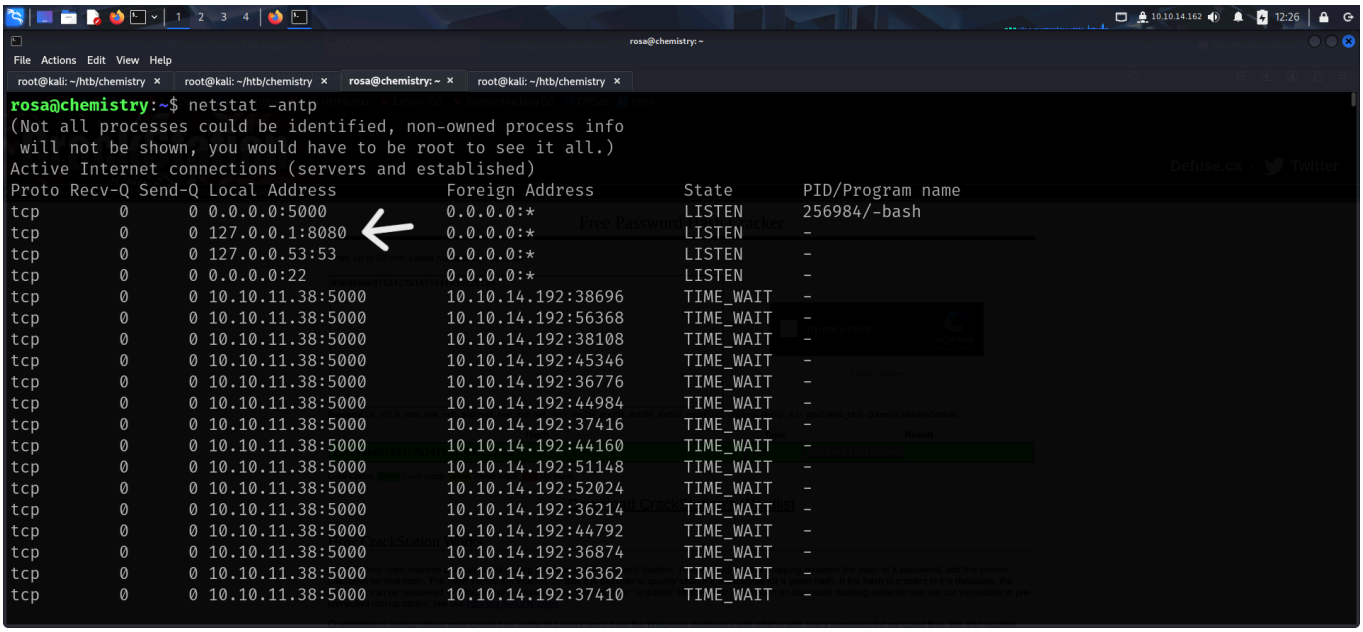




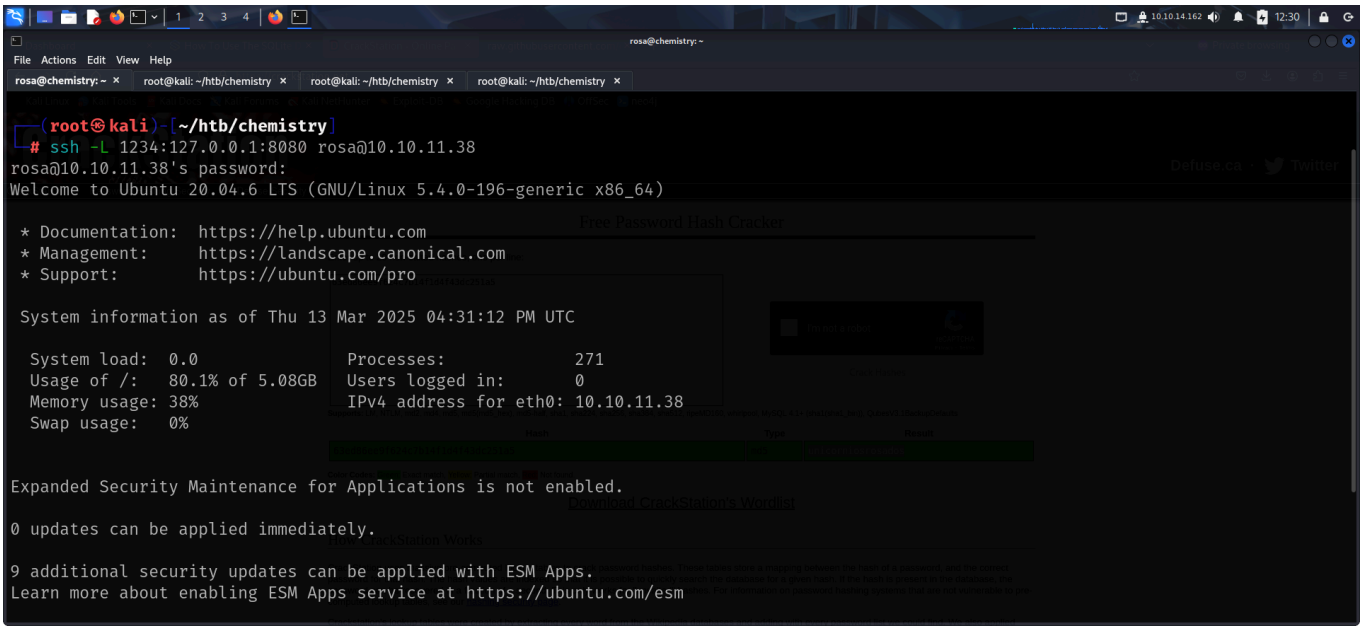
After capturing the user flag, I downloaded and ran **linux smart enumeration** script to look for ways to escalate privilege, however the script found nothing interesting.

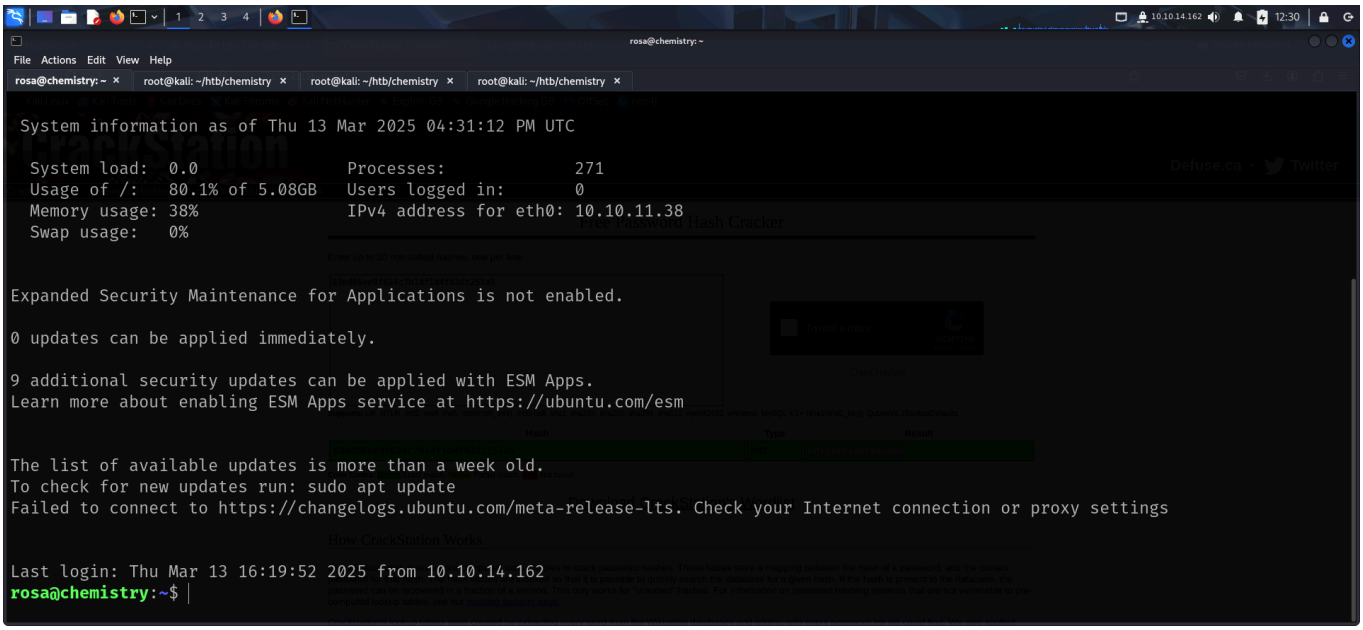


I then looked at listening ports and found port **8080** bounded to localhost on listen mode.

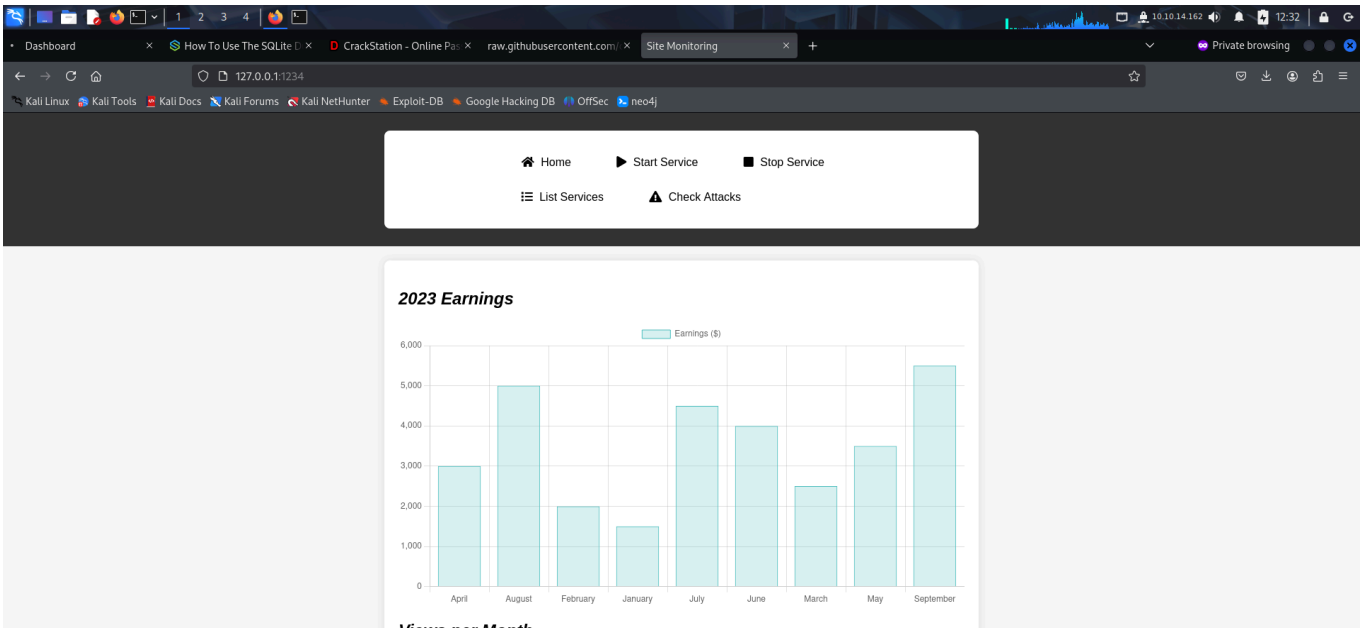


To access the internally bounded port, I performed port forwarding, binding port 1234 to the localhost port 8080 of the target.

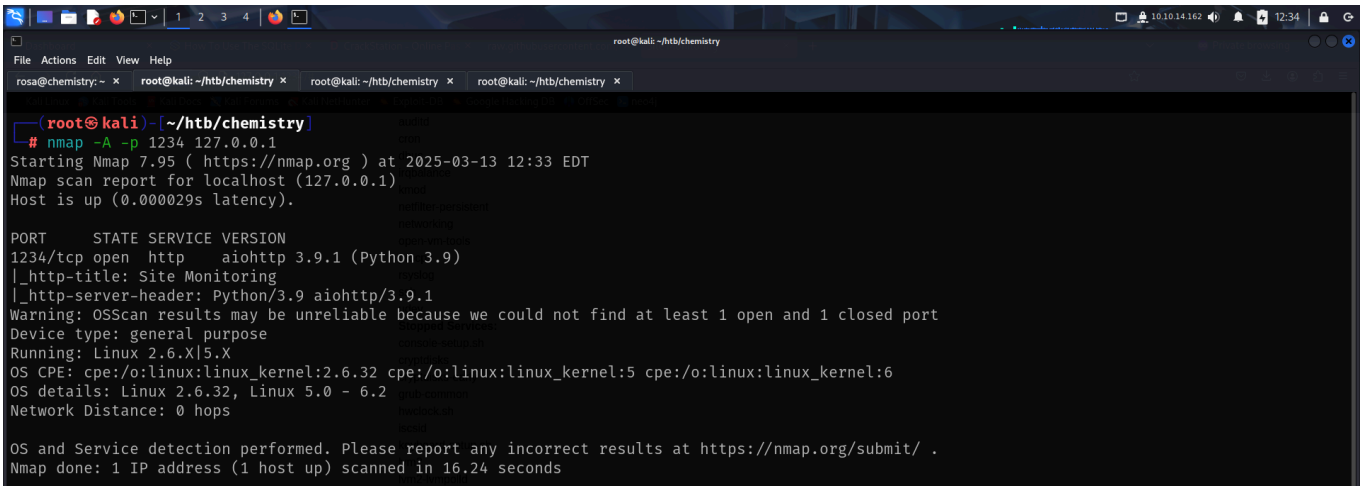




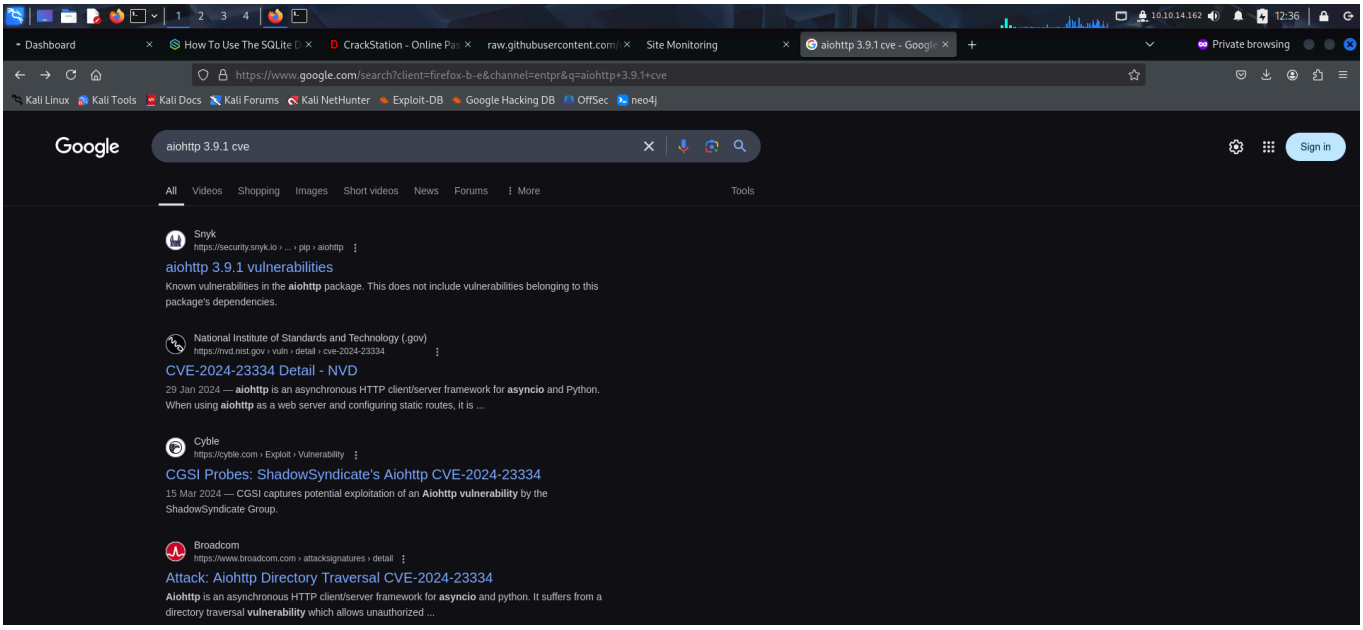
After that, I was able to access the service running on port 8080.

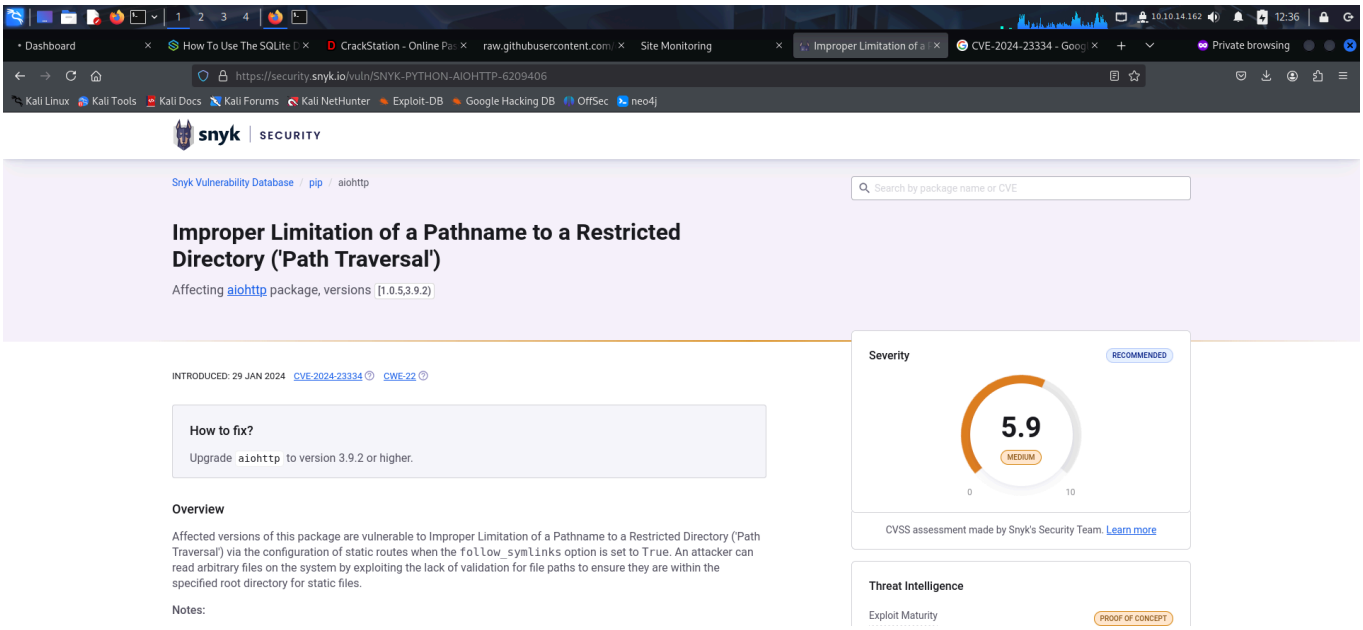


I performed an **nmap** scan on the service and found the service version.

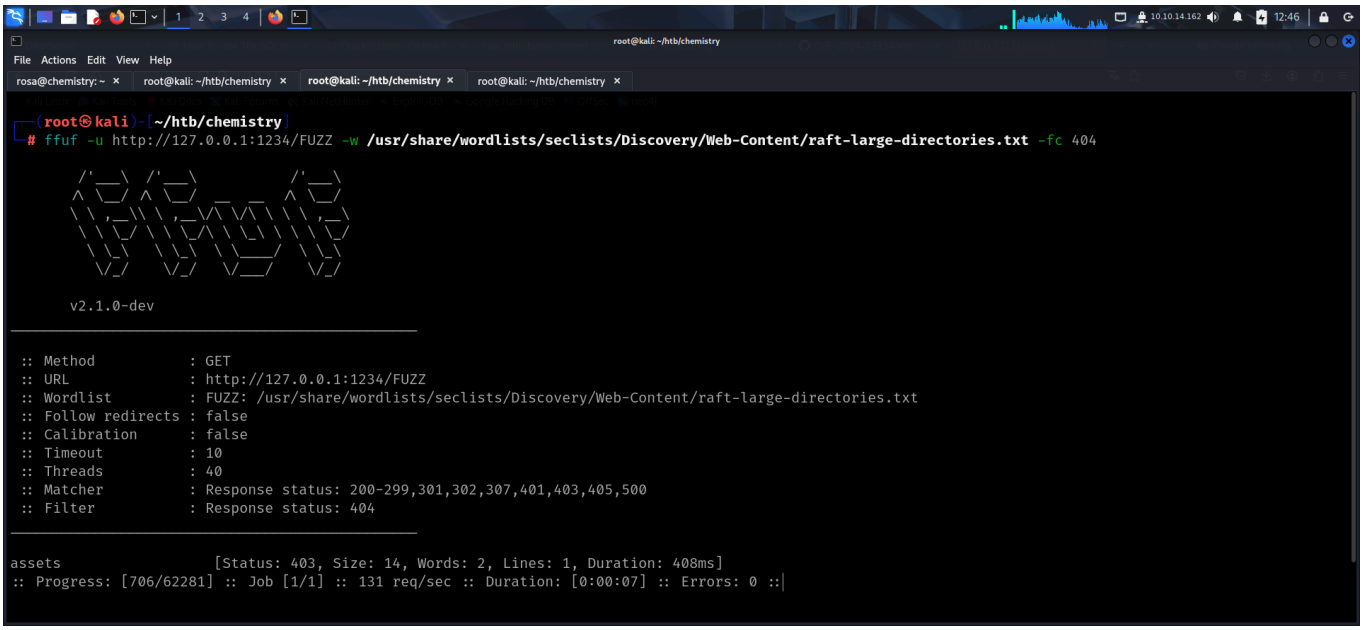


further footprinting revealed a **path traversal** vulnerability associated with this particular service version.

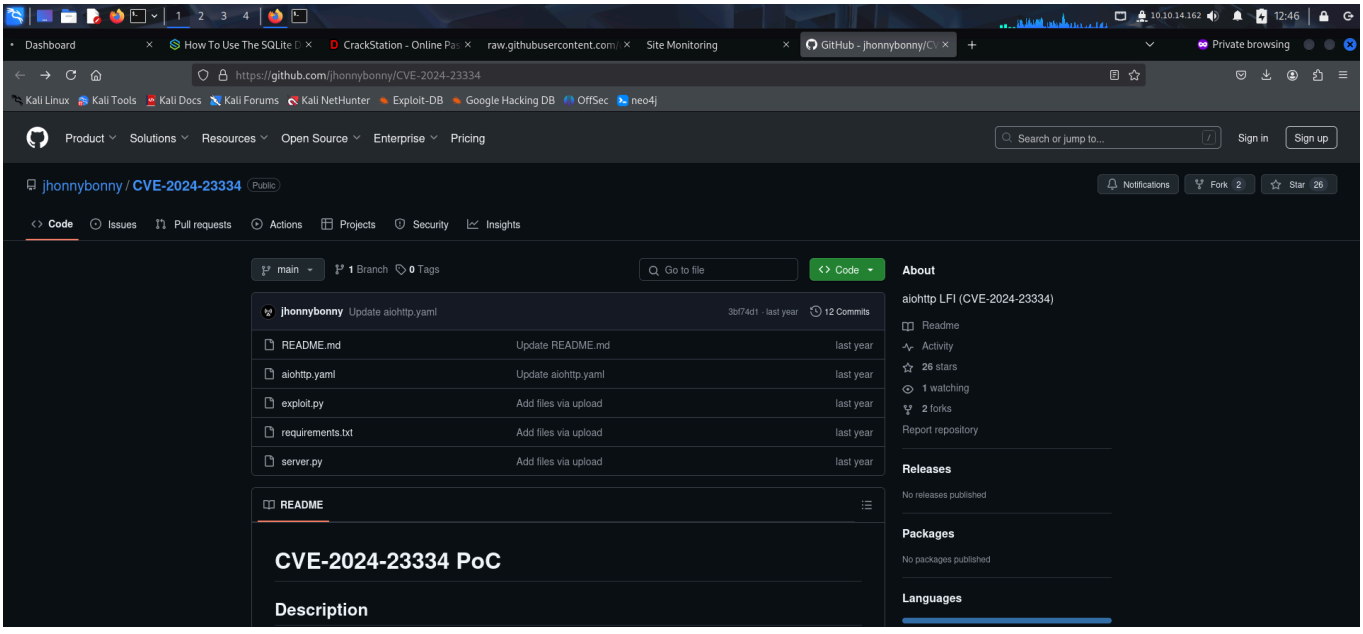


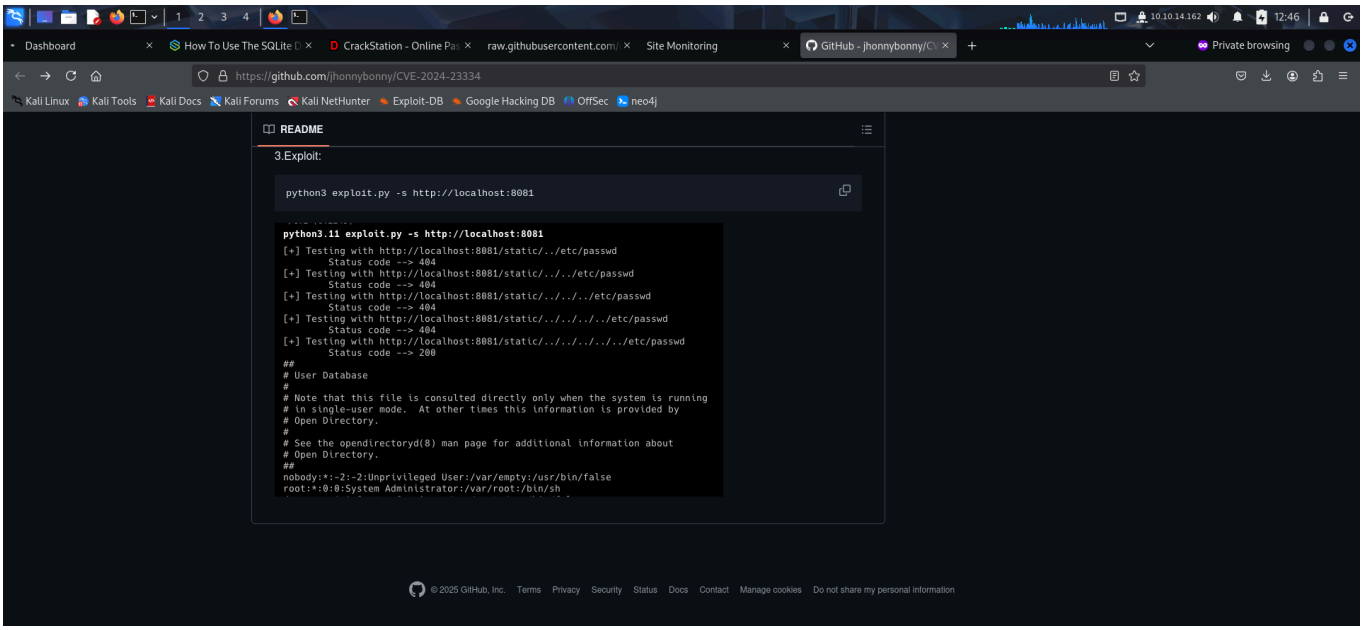


Since the vulnerability was path traversal, I looked for directories on the target using **ffuf**.

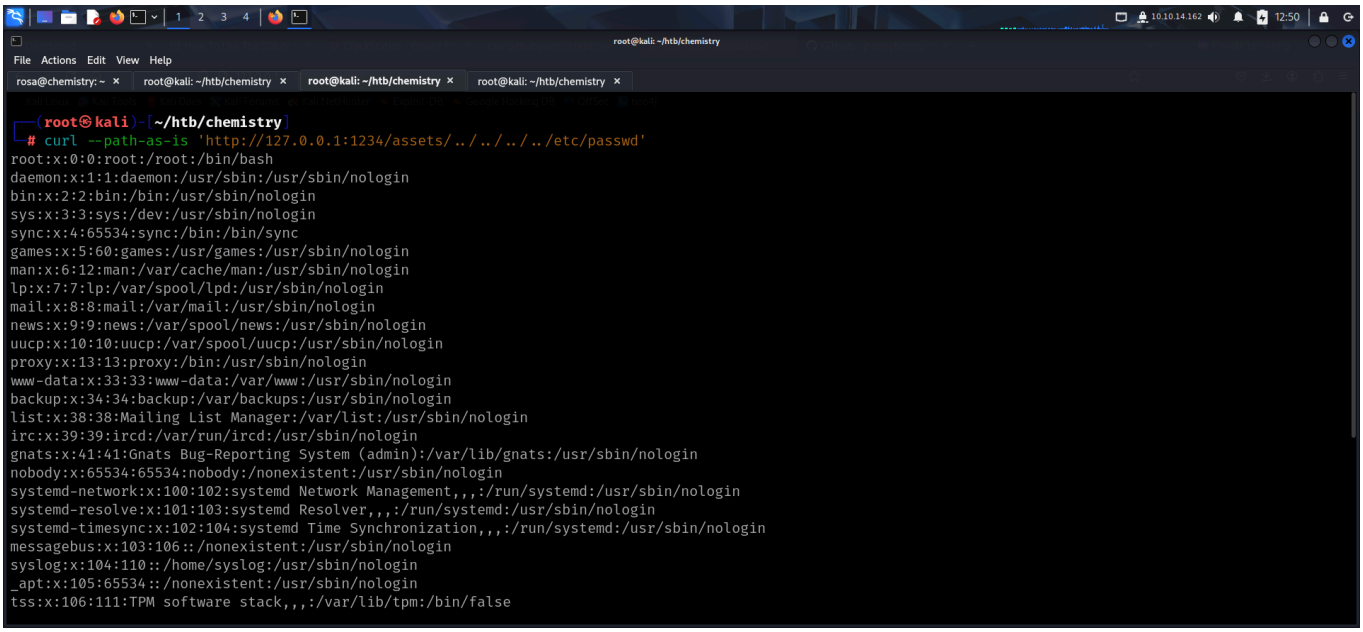


I then found a PoC of the CVE on **github** and got a reference on how the vulnerability could be explotied.

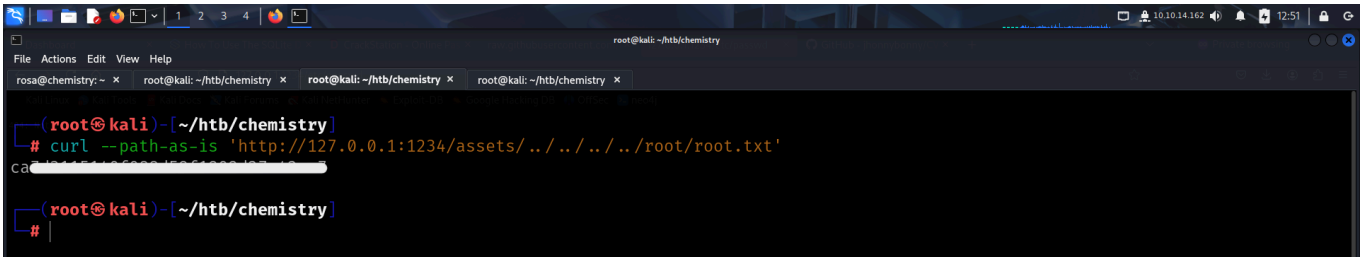




I then used **curl** and tried accessing the **/etc/passwd** file by exploiting the **path traversal** vulnerability.



Since I was able to confirm the vulnerability, I exploited it to access the root flag from the **/root** directory.



With that we successfully pwned the machine **chemistry** :)
Happy Hacking !