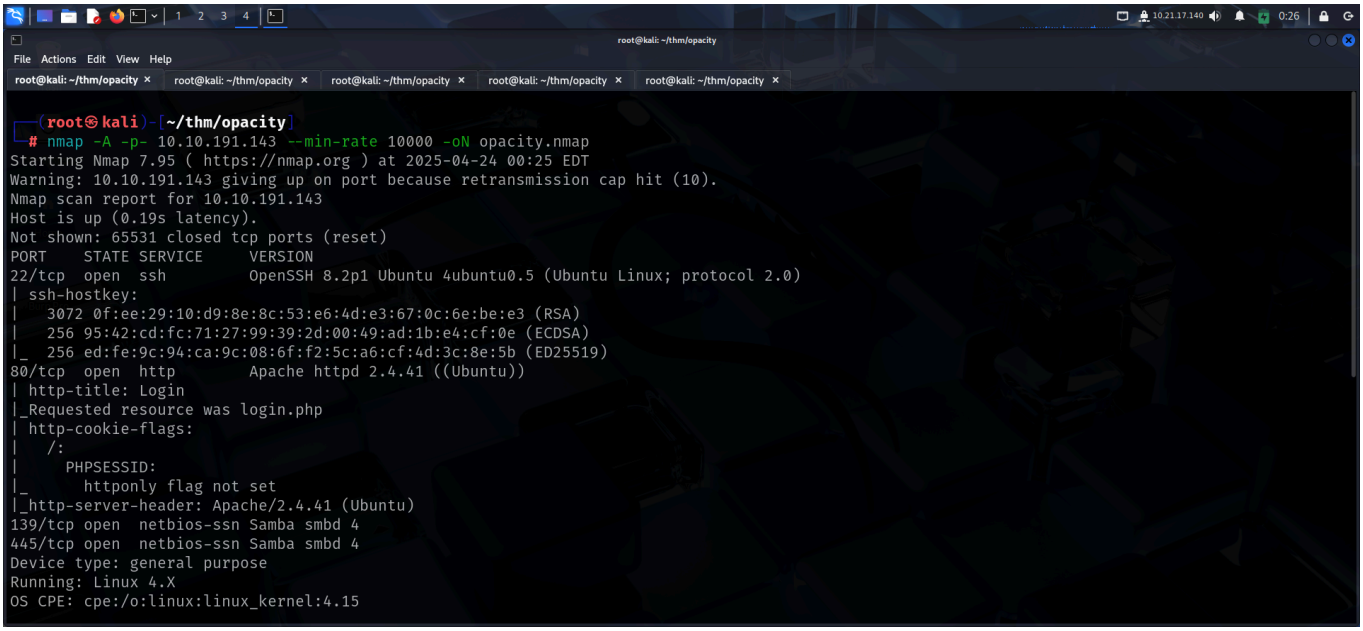
OPACITY

https://tryhackme.com/room/opacity

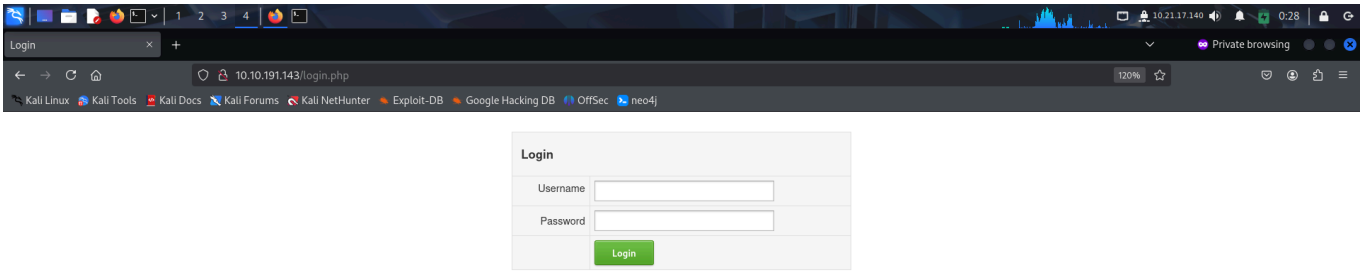
SCANNING

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

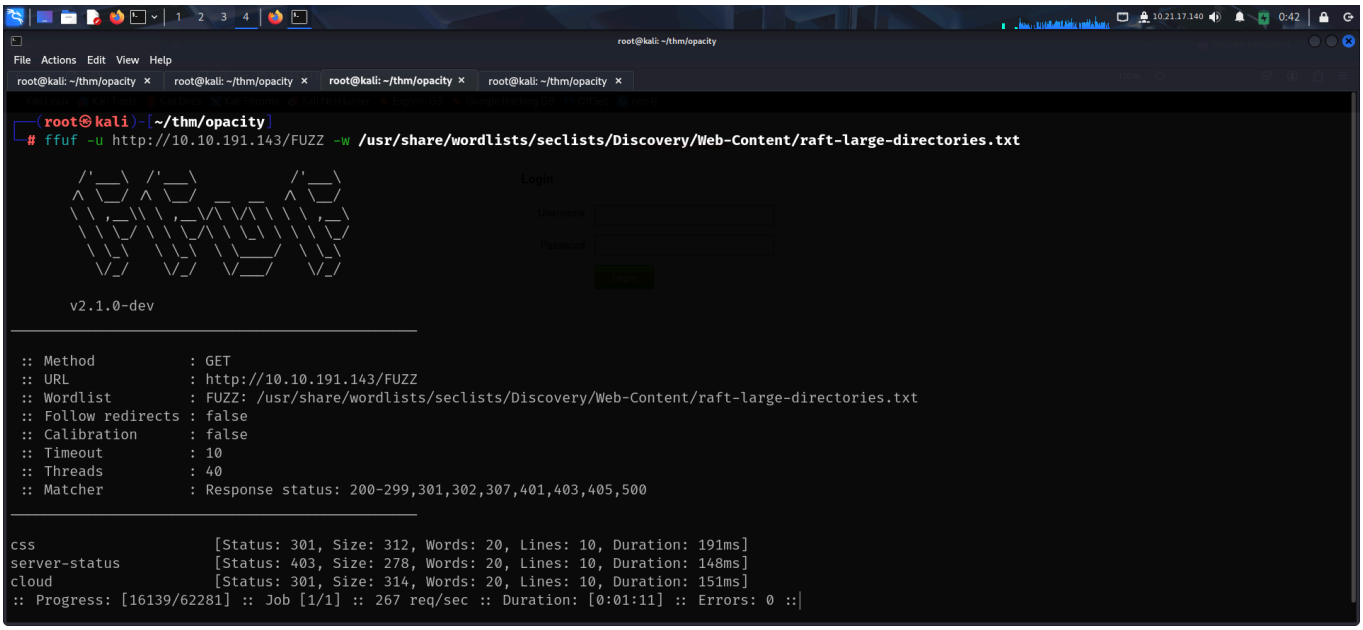


FOOTHOLD

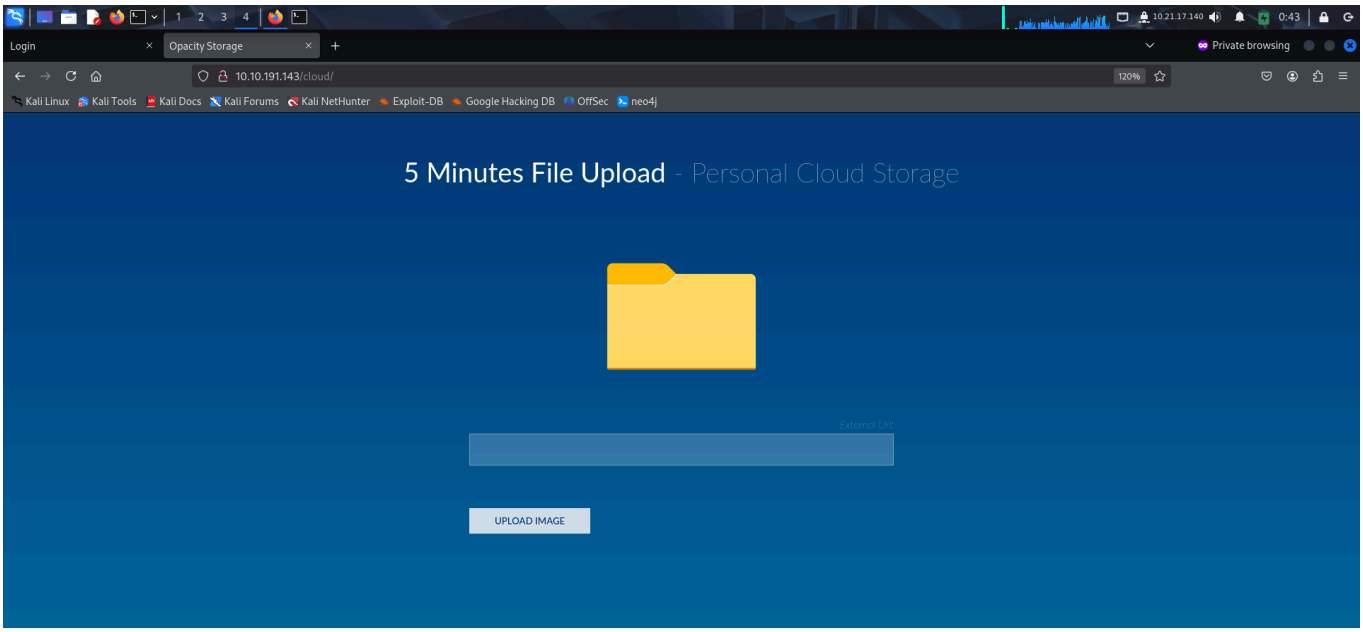
I accessed the web application running on port 80 and found a login page.



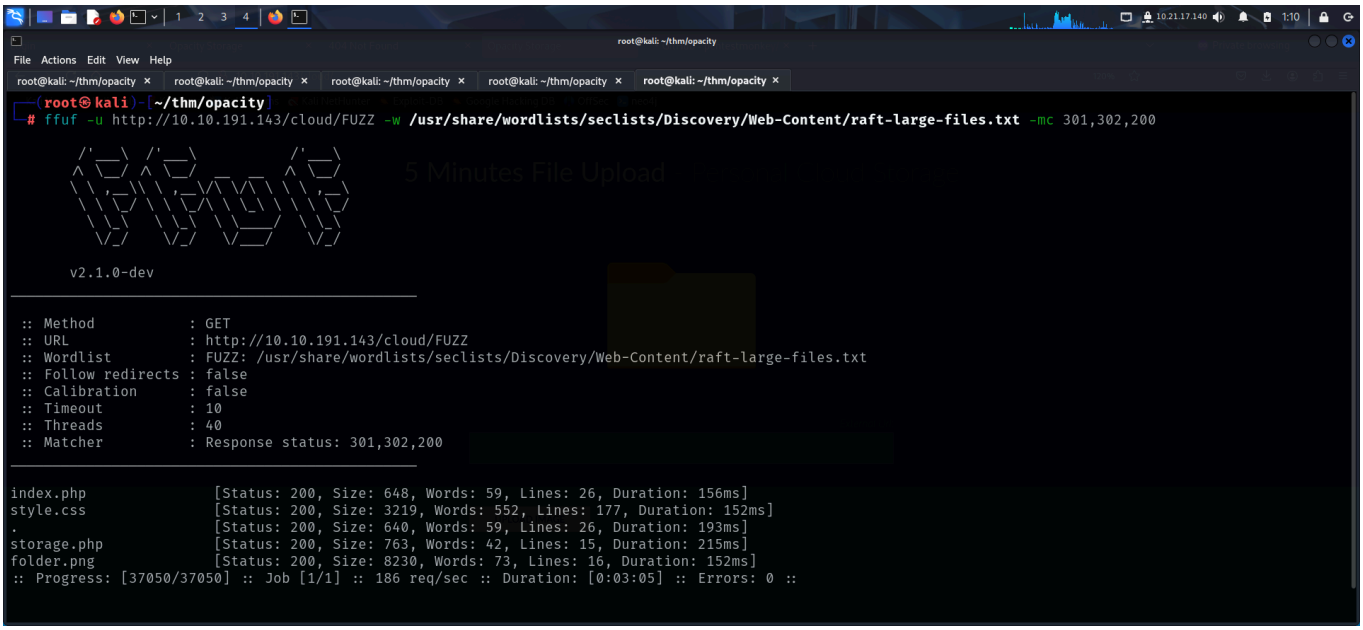
I fuzzed hidden directories and found an interesting directory called `cloud`.



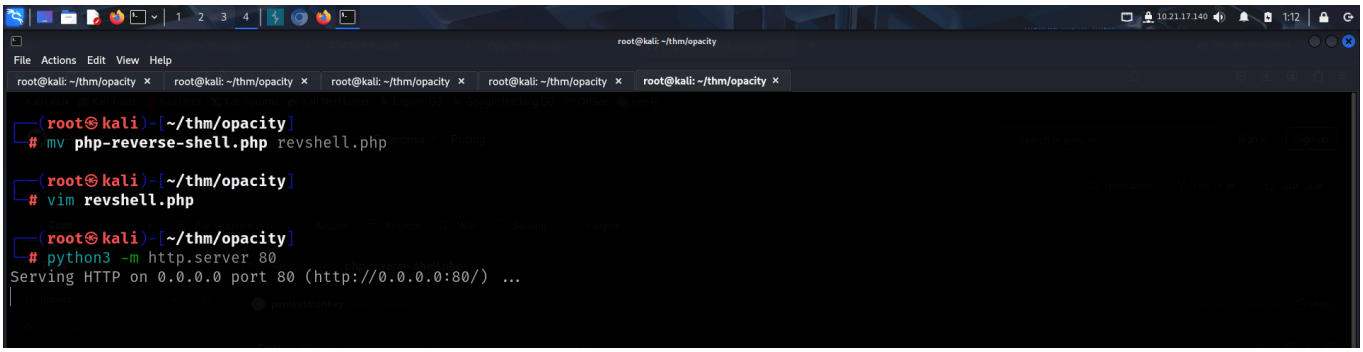
I accessed the `cloud` endpoint and found a file upload functionality.



I then fuzzed hidden files in the `/cloud` directory and found few files.

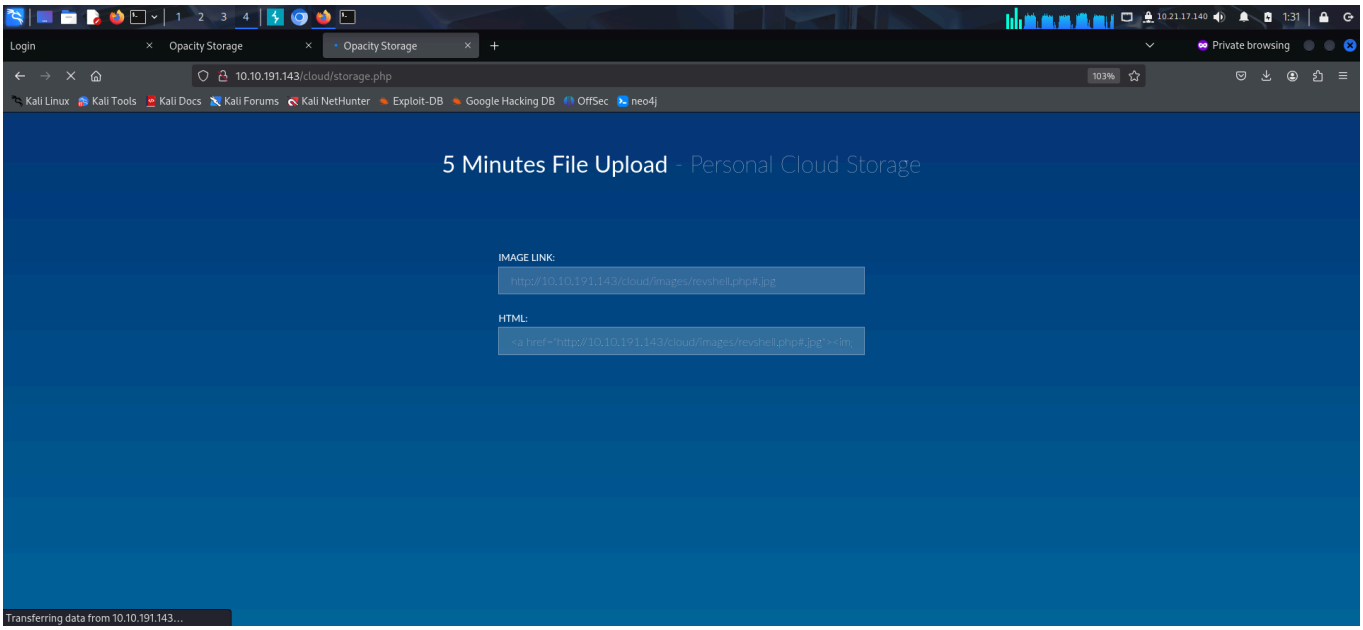


I then created a reverse shell php payload and served it using python's http server on my local system.

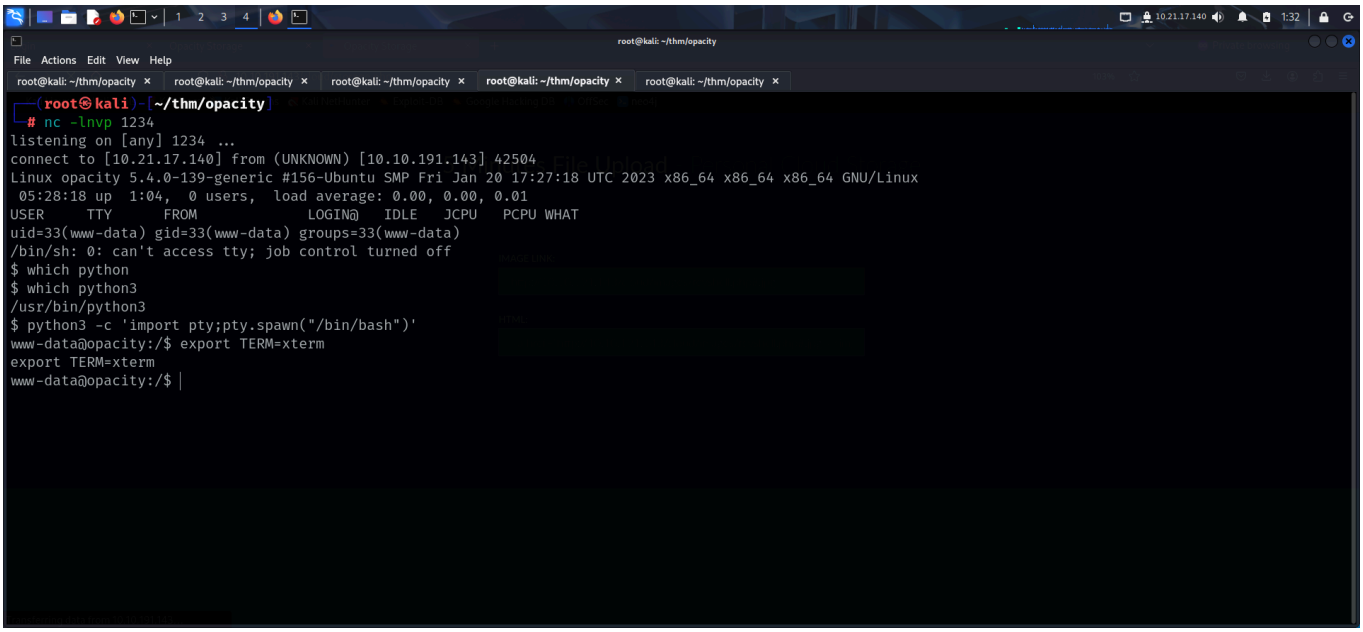


I tried uploading it directly, however the application had some sort of validation mechanism in place that only allowed image files. So I used `#` to add `.jpg` extension after my php payload. Fragment identifiers (everything after `#`) **are not sent to the server** by the browser. So when your request is sent, it removes `.jpg` and sends the php reverse shell.

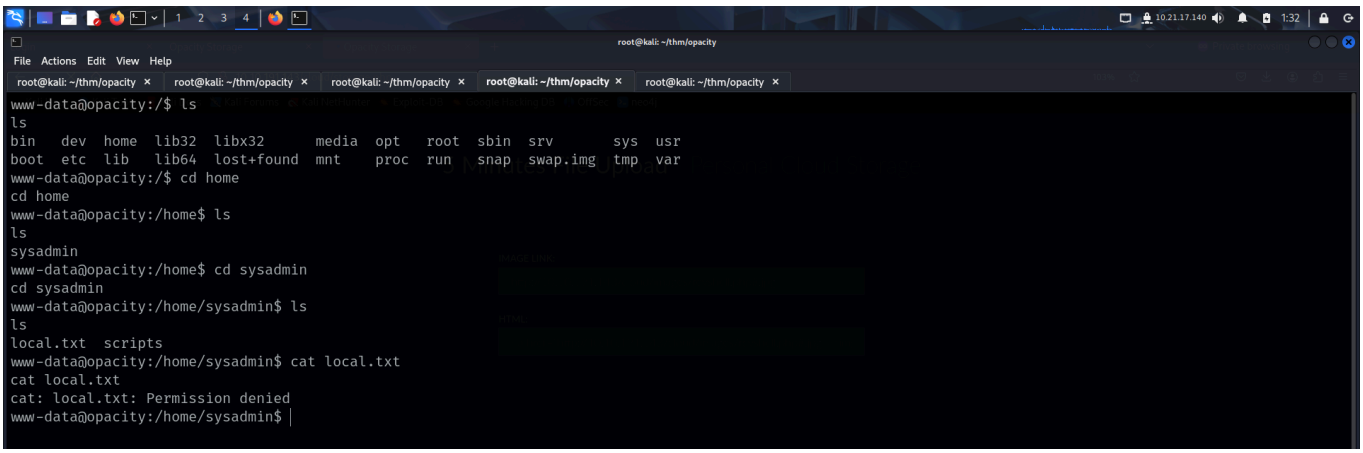
I was able to bypass the security mechanism and upload the payload.



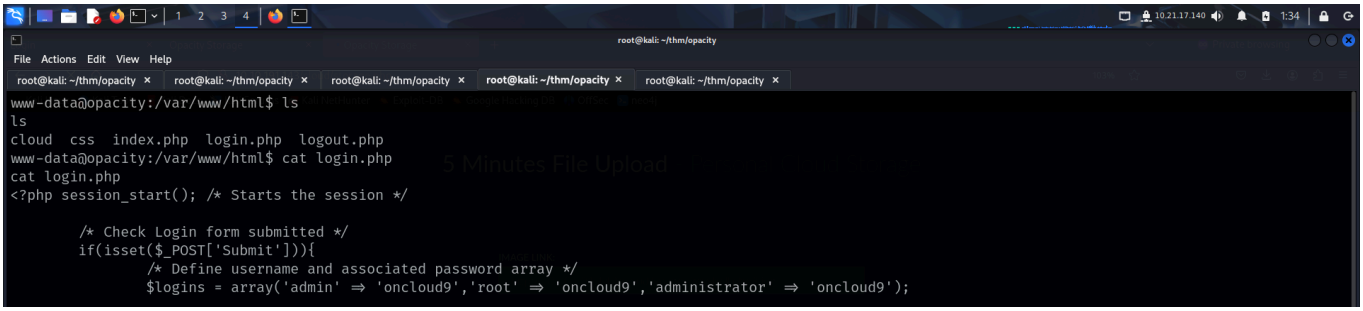
The payload got executed after being uploaded and I got a reverse shell on my **netcat** listener.



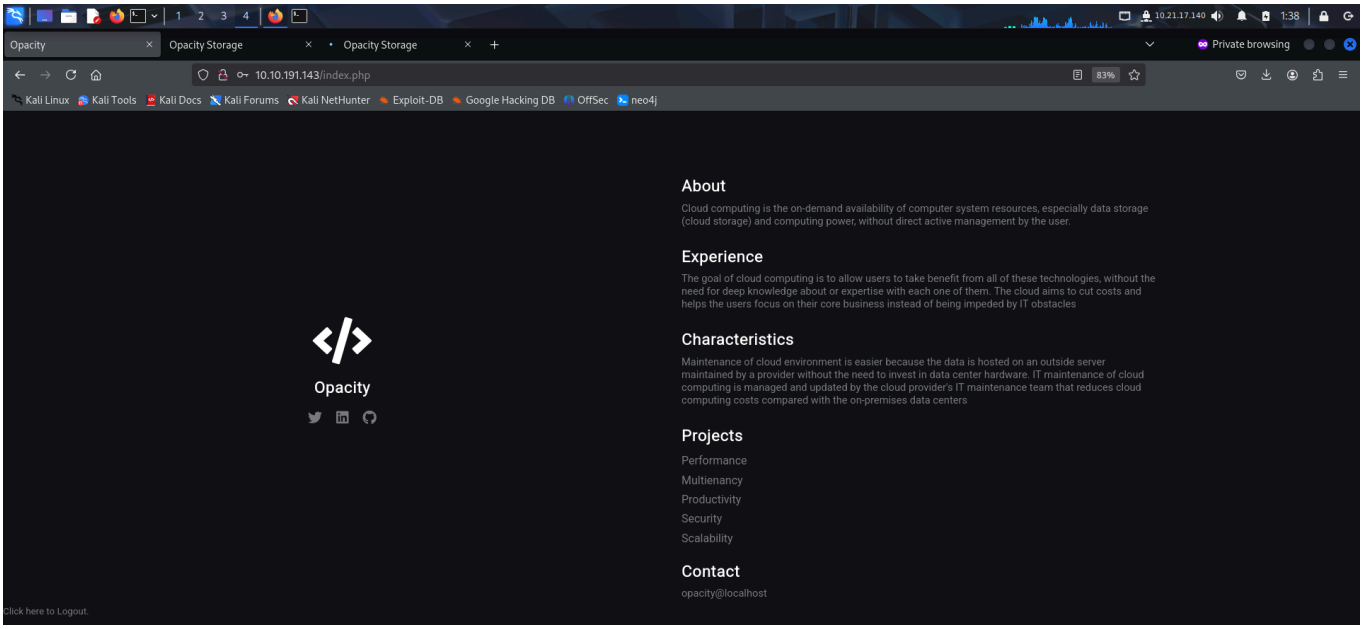
I then tried accessing the user flag from sysadmin's home directory.



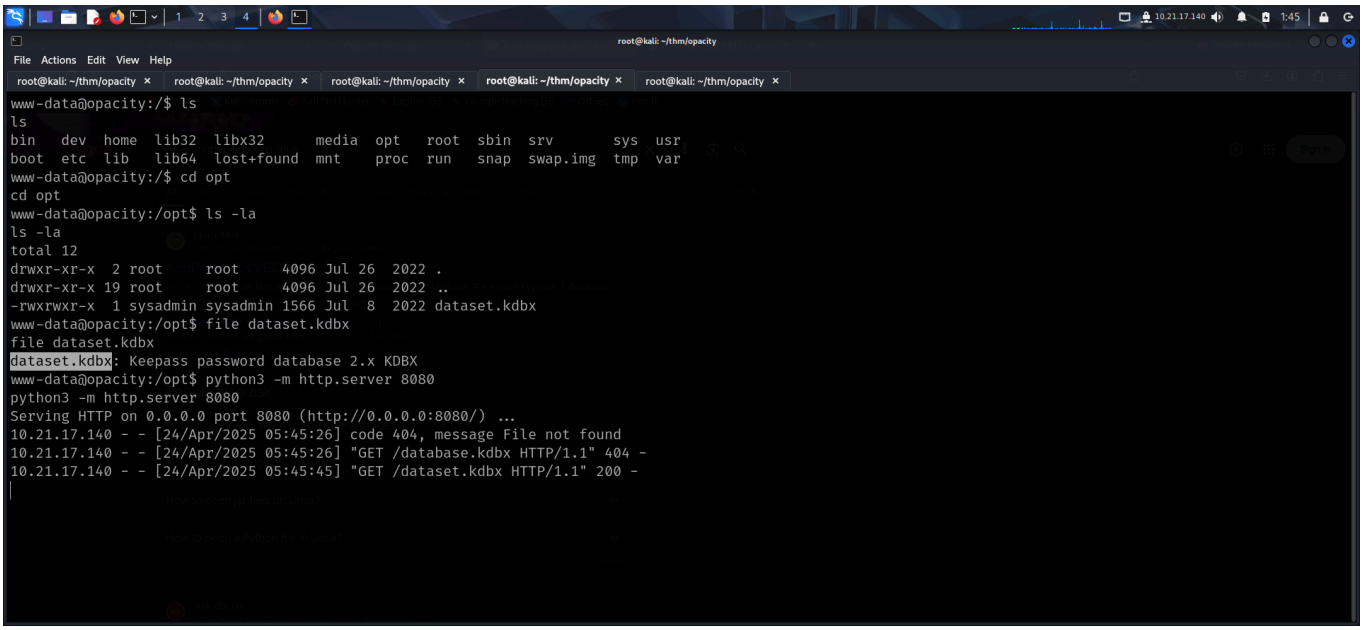
Since I did not have permissions to read the user flag, I would have to escalate my privileges. Hence I read the source code of the login page and found the login credentials.



I logged in using the credentials but found nothing interesting on the application.



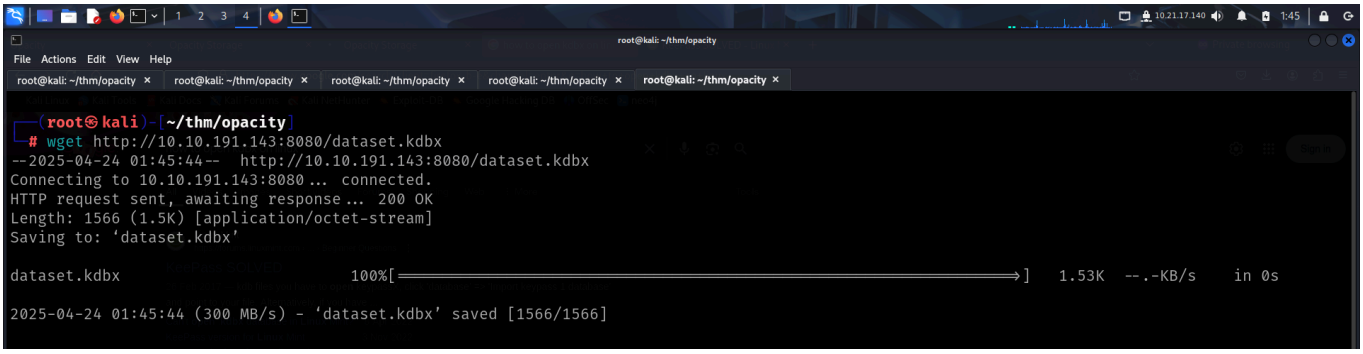
I then explored other directories and found a keepass password database file inside the `/opt` directory.



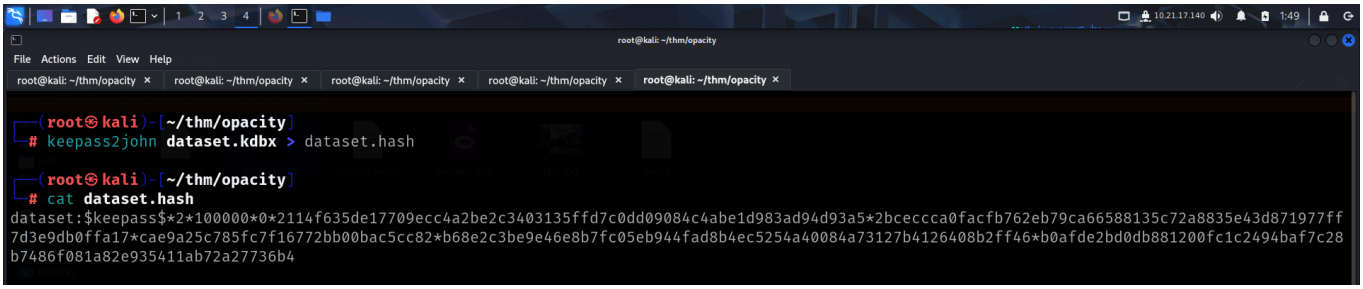
To know more about keepass, I referred to the below artciles:

- https://fileinfo.com/extension/kdbx
- https://gist.github.com/lgg/e6ccc6e212d18dd2ecd8a8c116fb1e45

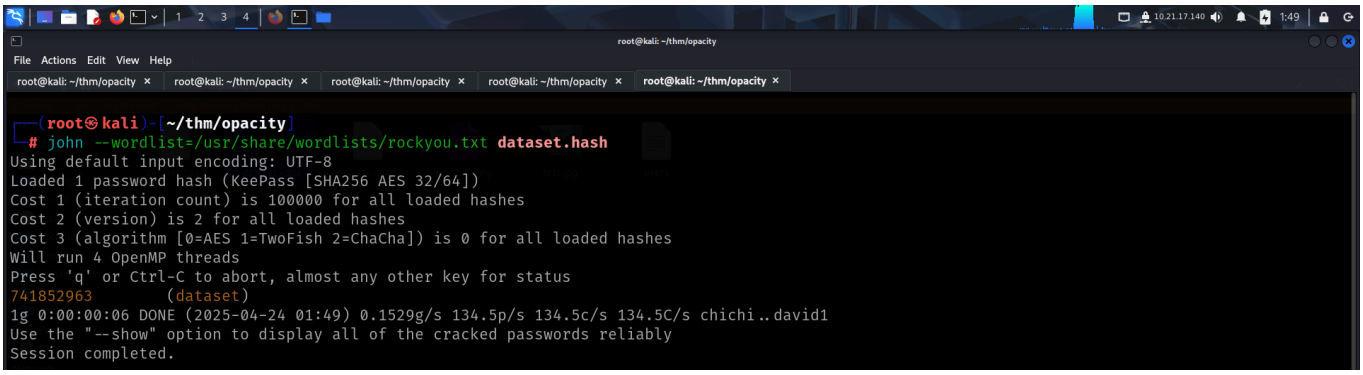
I downloaded the file on my local system.



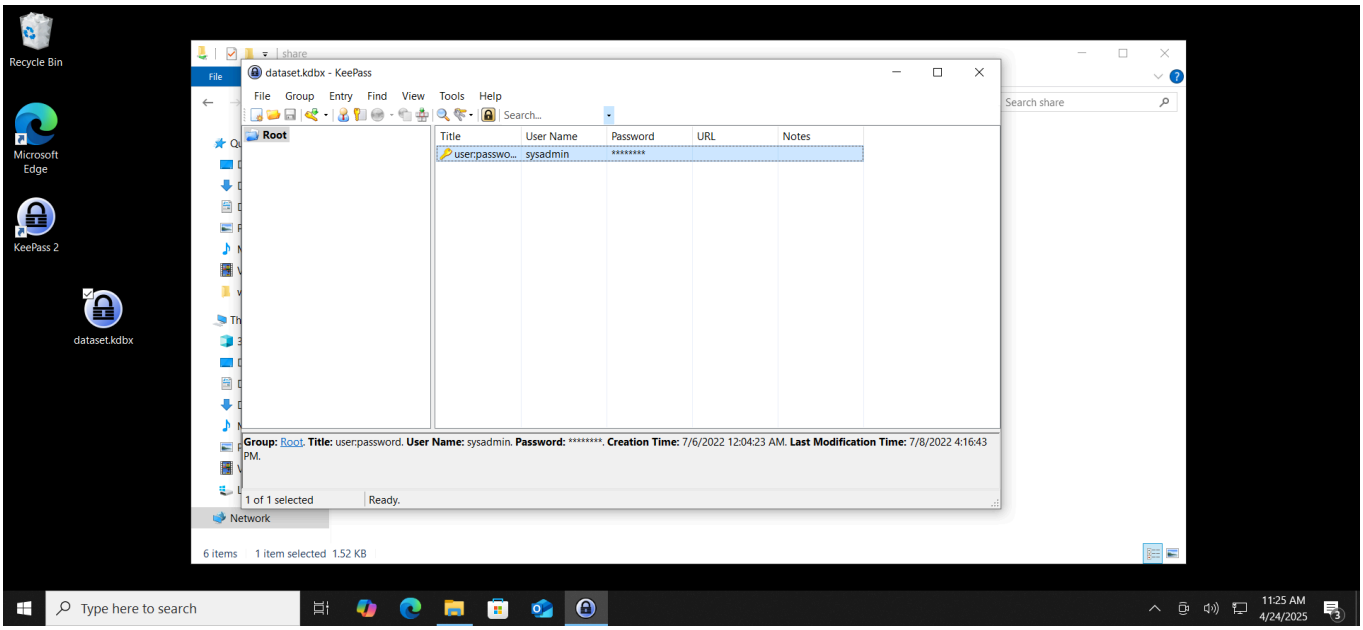
I then downloaded keepass on my windows system and tried opening it, however it was password protected. So I converted the file to john crackable format.

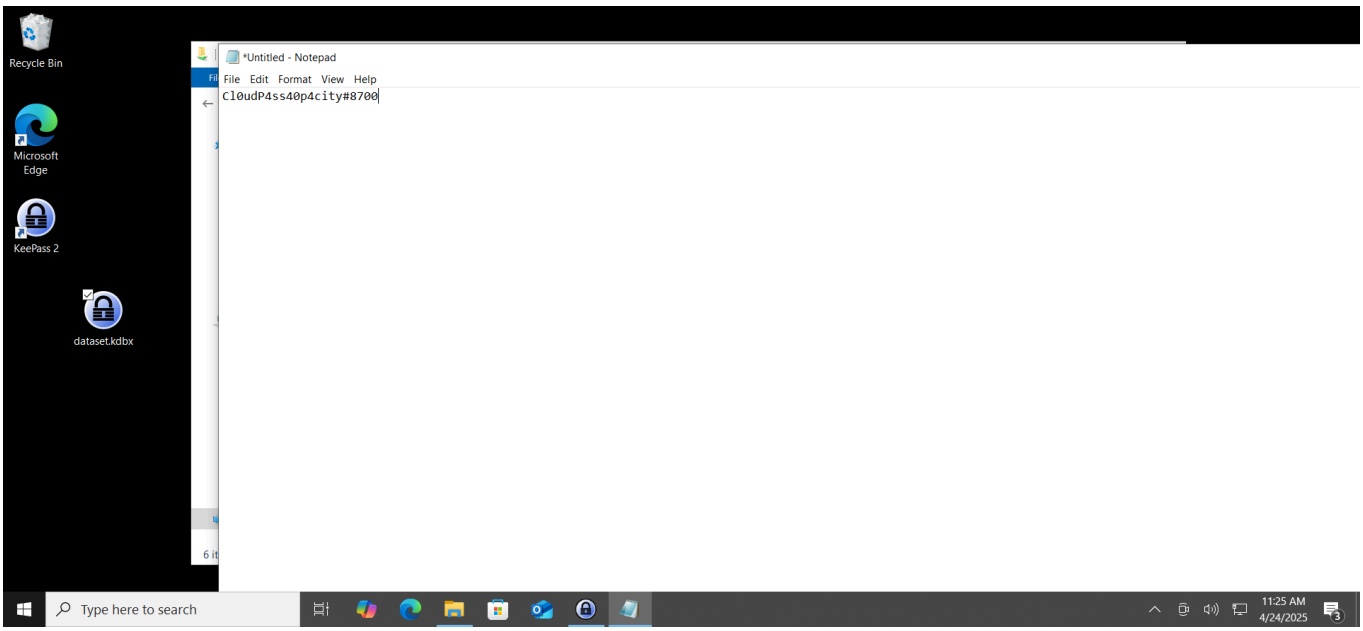


I then cracked its password using **john**.

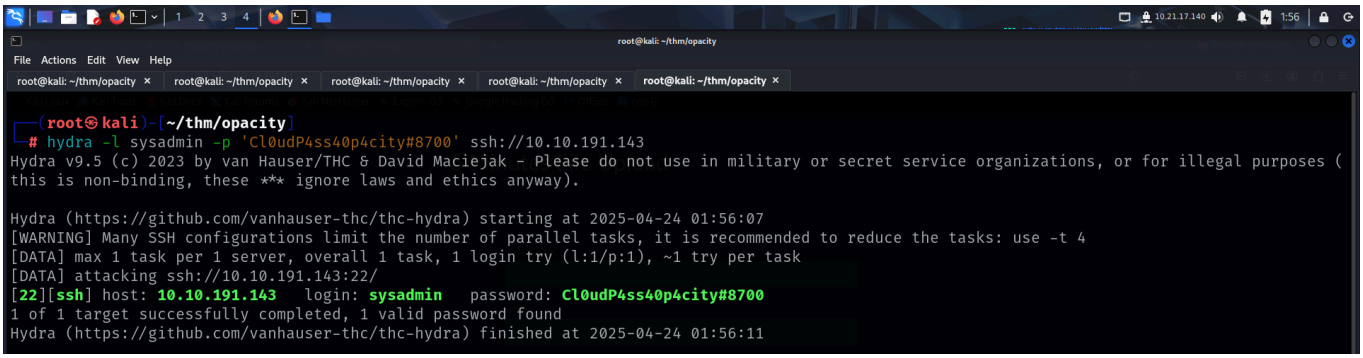


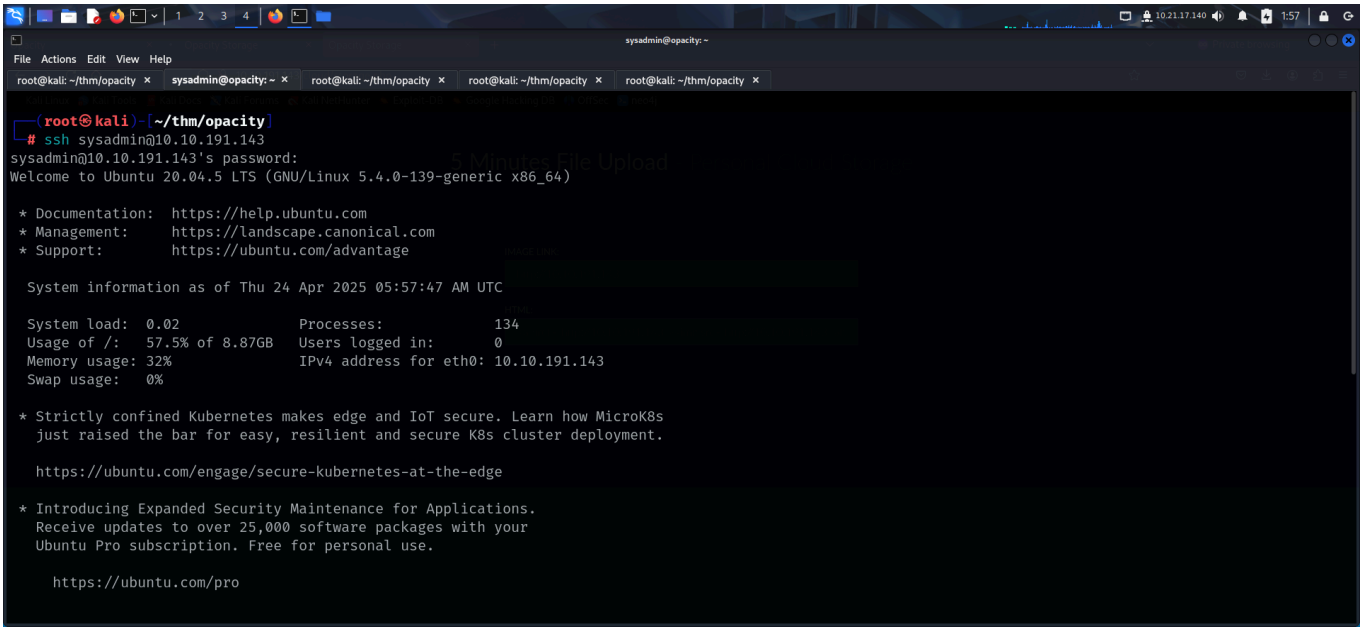
I used the password to view the contents of the database and found user credentials.





I then logged in using the newly found credentials.

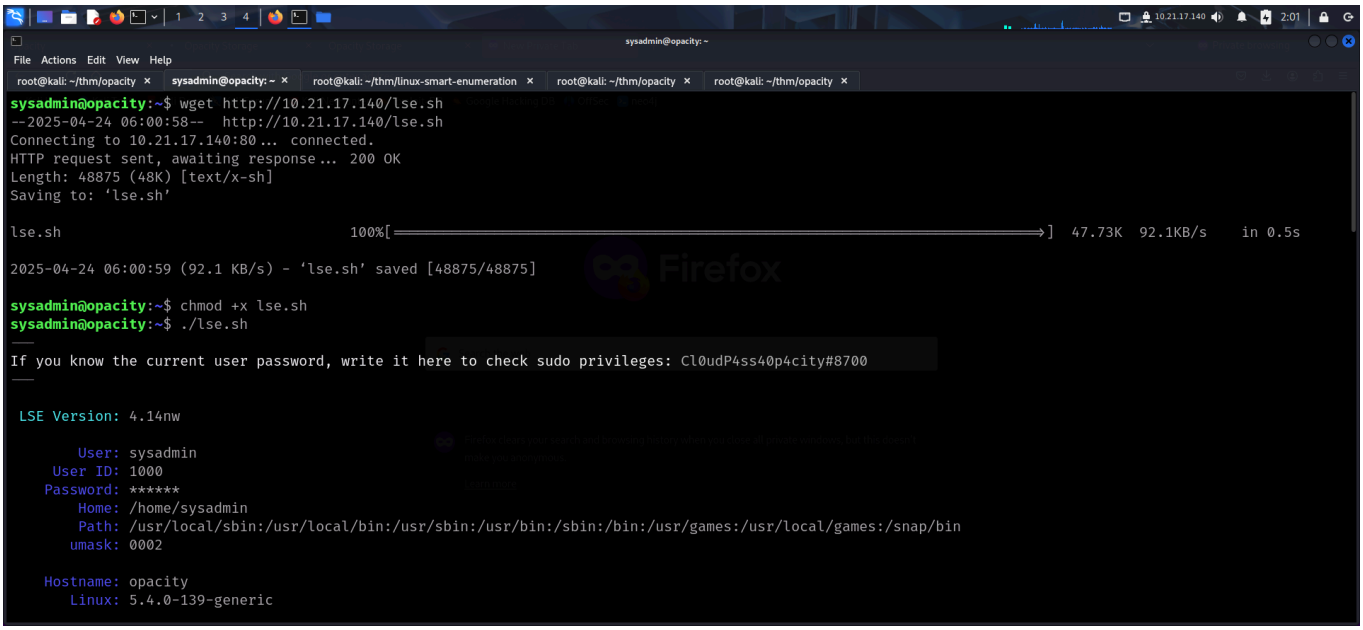


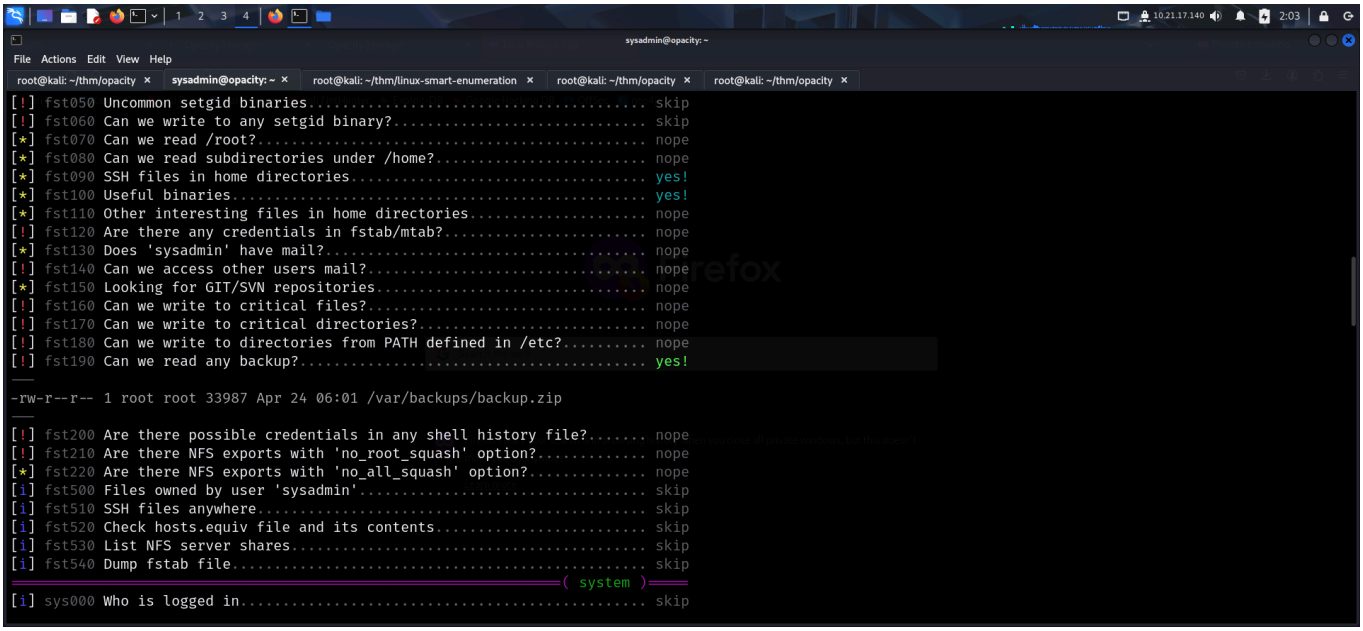


I then captured the local flag.

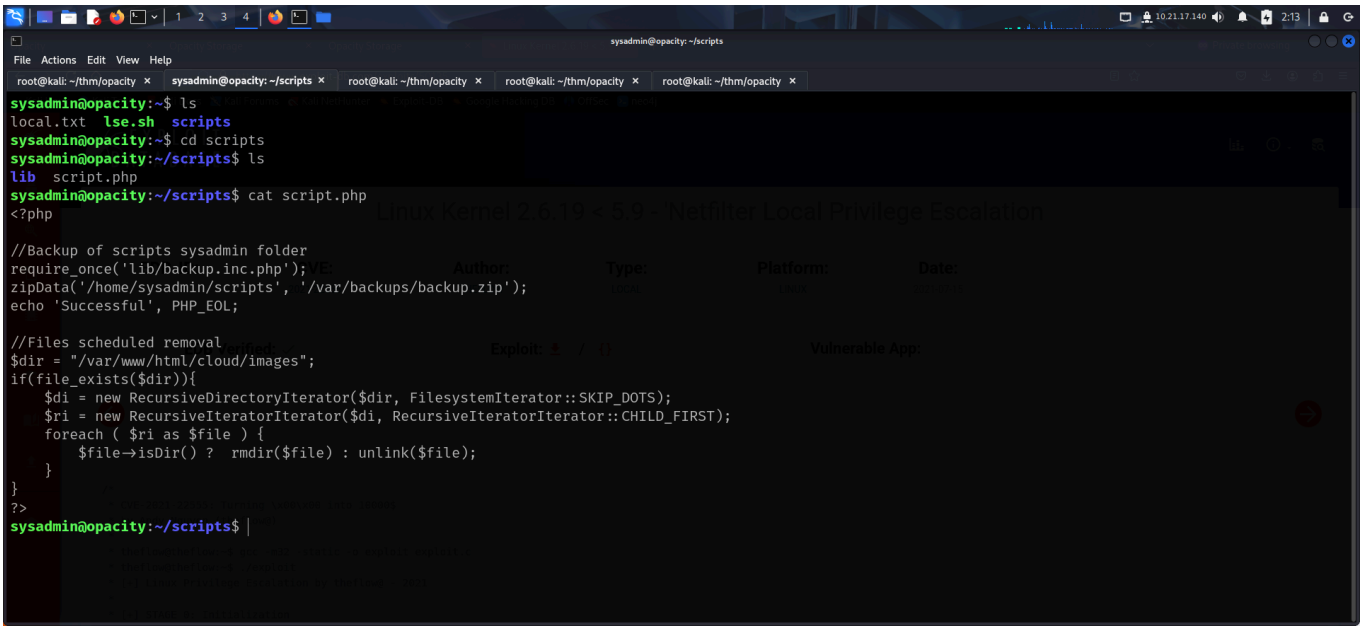
PRIVILEGE ESCALATION

I downloaded **linux smart enumeration** script on the target and ran it to find privileges escalation vectors.

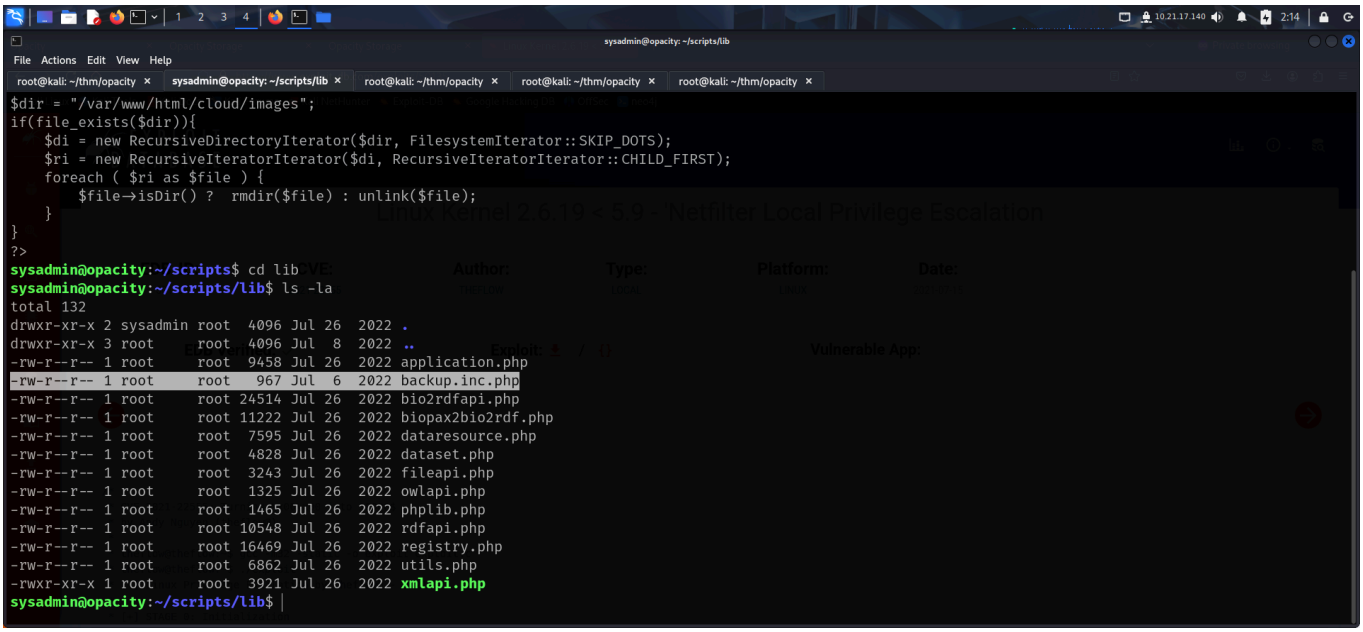


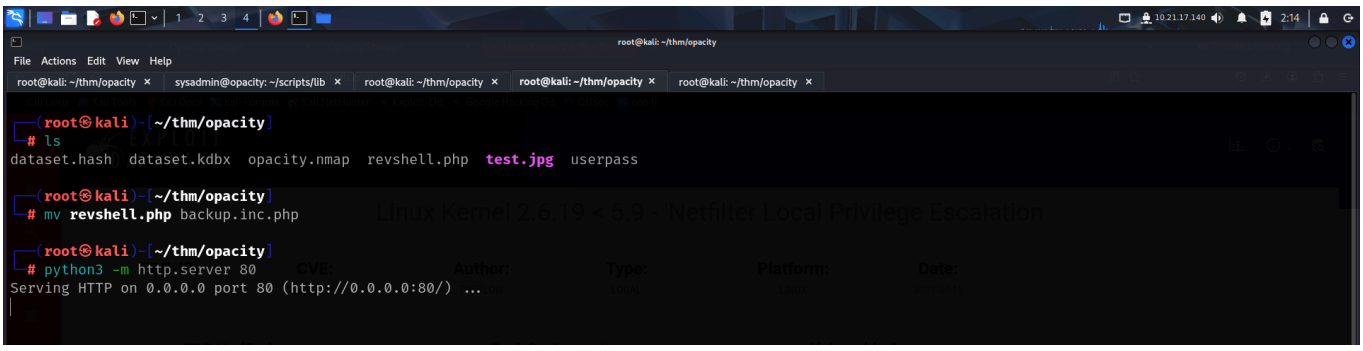


We had read access to a backup file. I also found a script in sysadmin's home directory that used the backup.zip file to save a backup of the scripts folder.

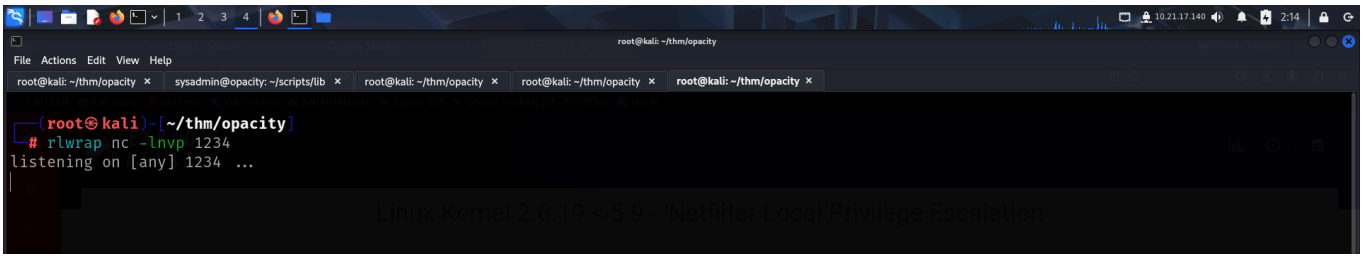


It used a file called `backup.inc.php` So I replaced that file with my php reverse shell.

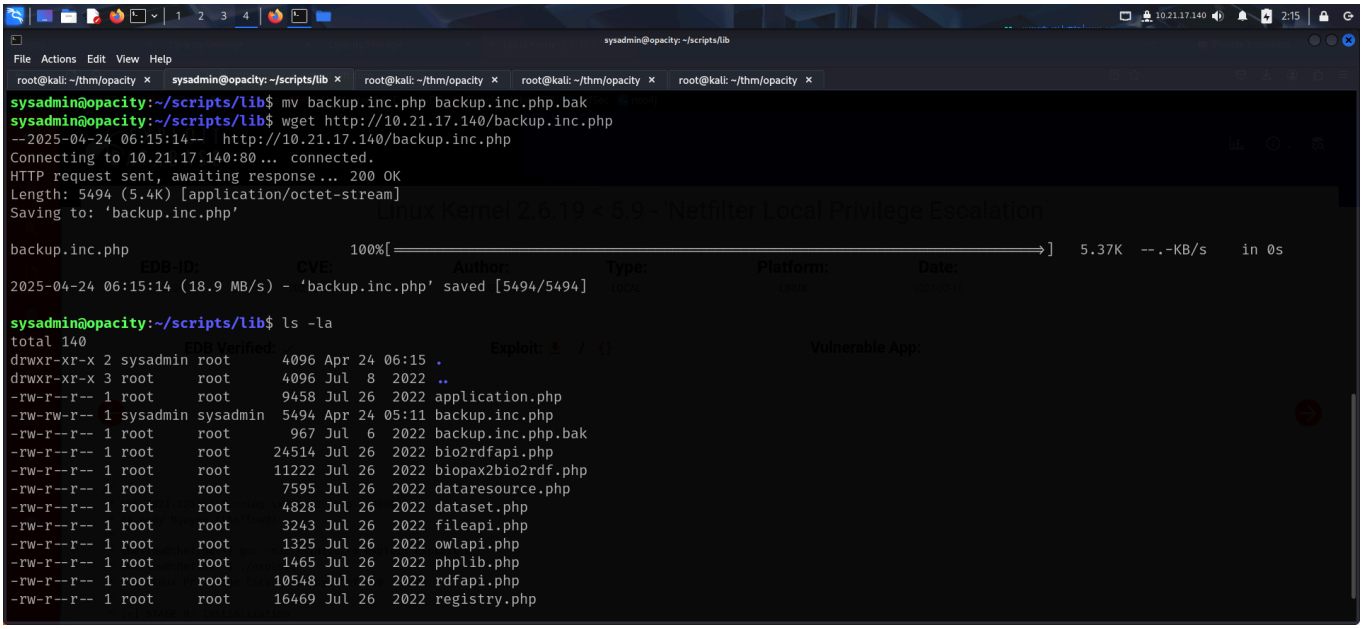




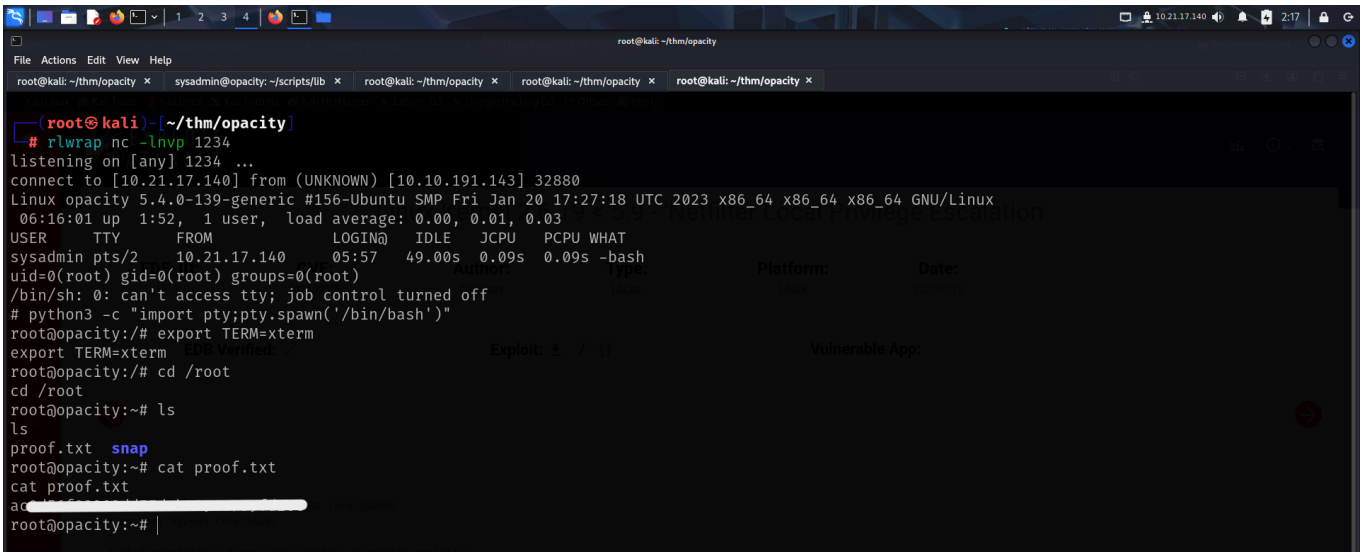
I started a netcat listener



After transferring the reverse shell with the name as `backup.inc.php`, I got a reverse shell in some time.



After gaining root access, I captured proof.txt from `/root`.



That's it from my side, until next time !