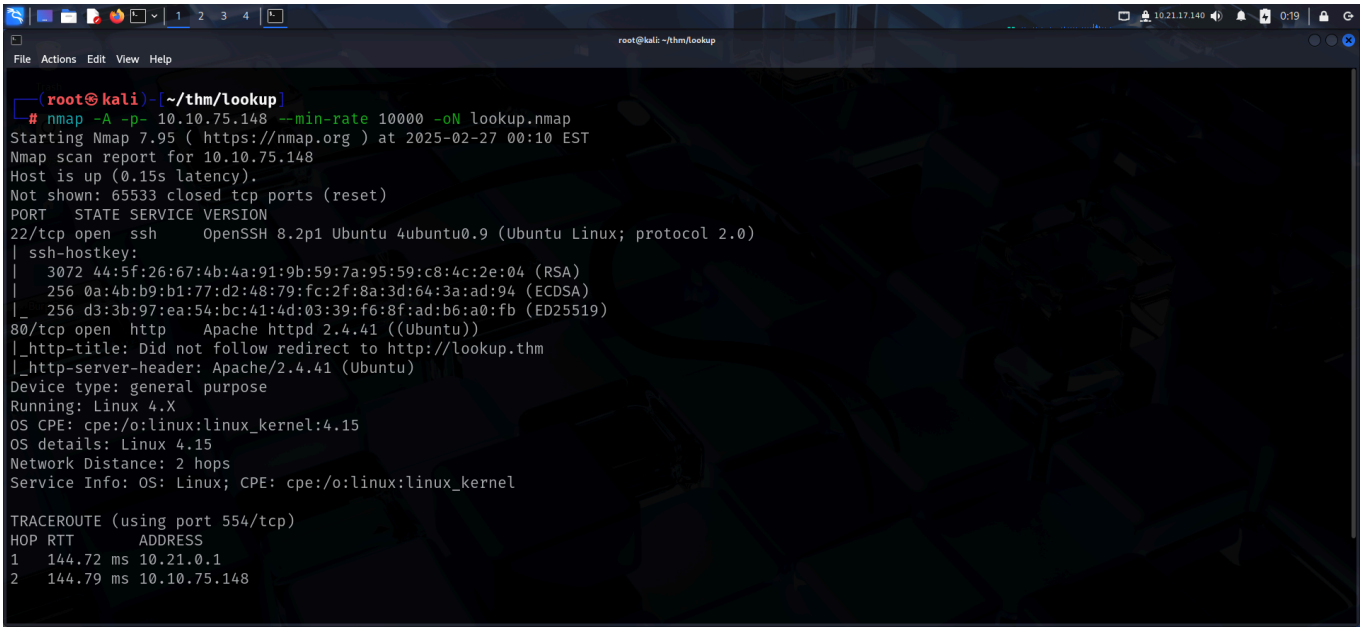
LOOKUP

Machine Link: https://tryhackme.com/room/lookup

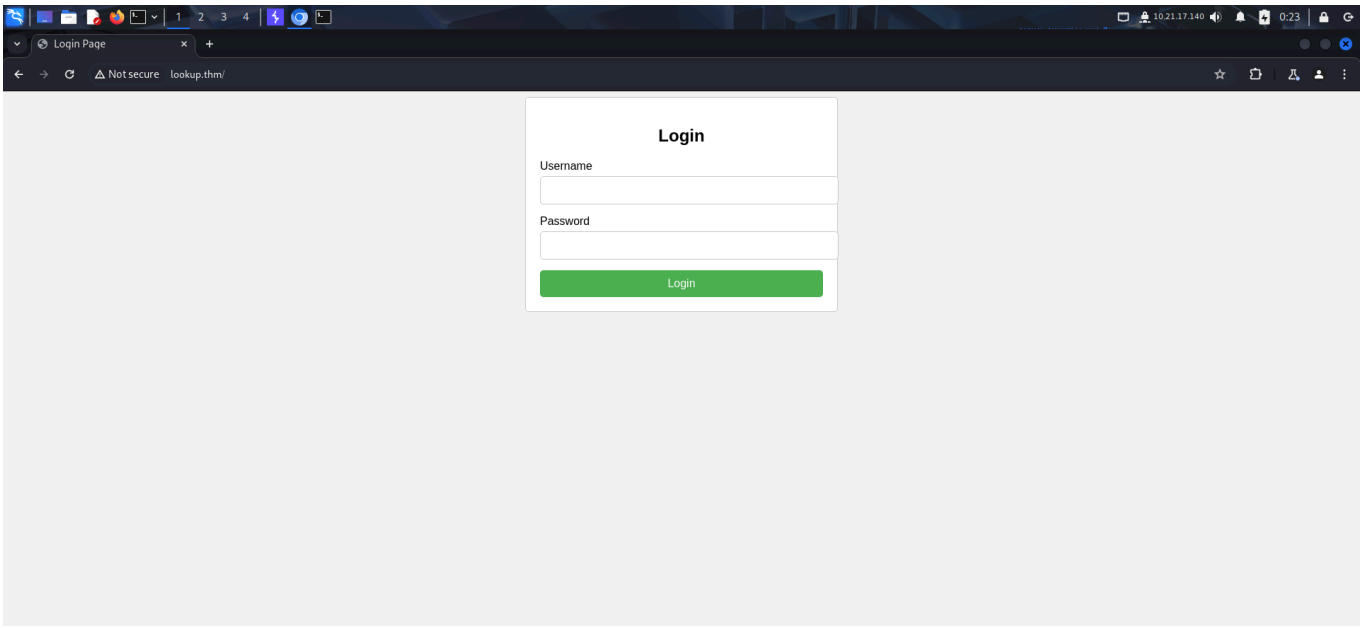
RECONNAISSANCE

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

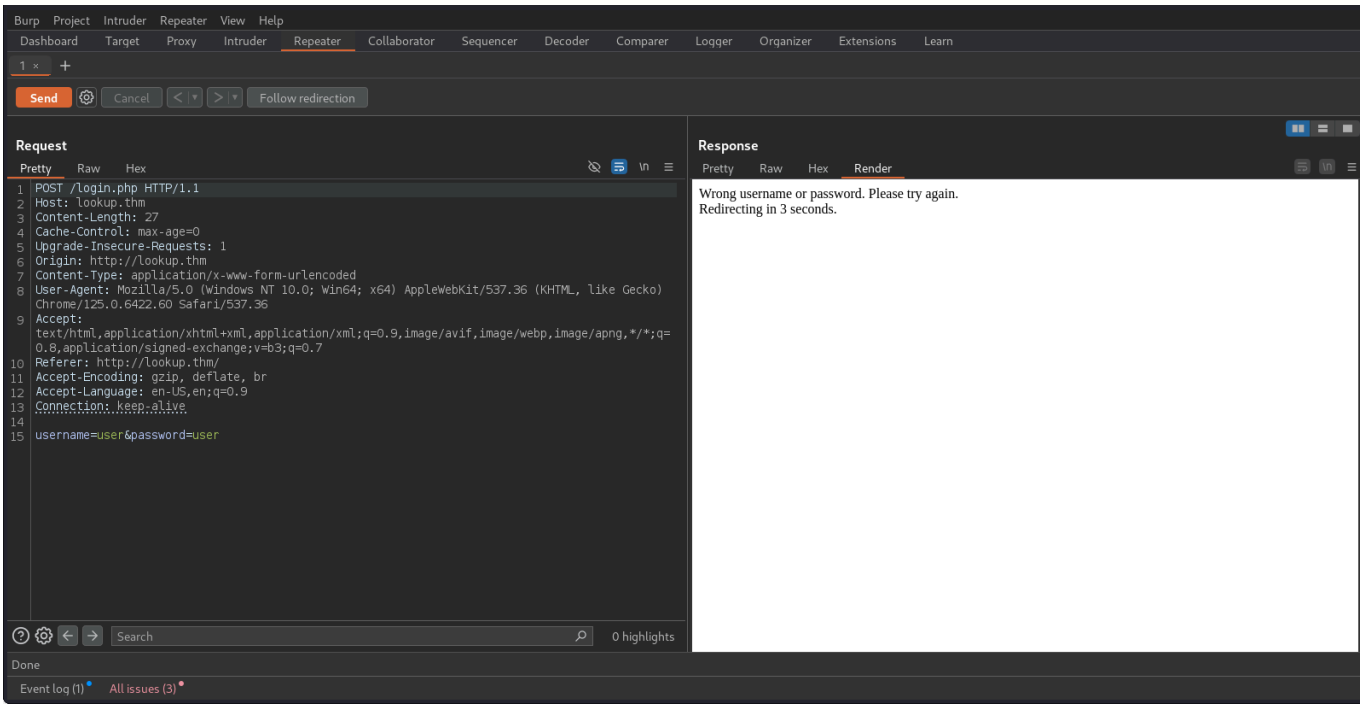


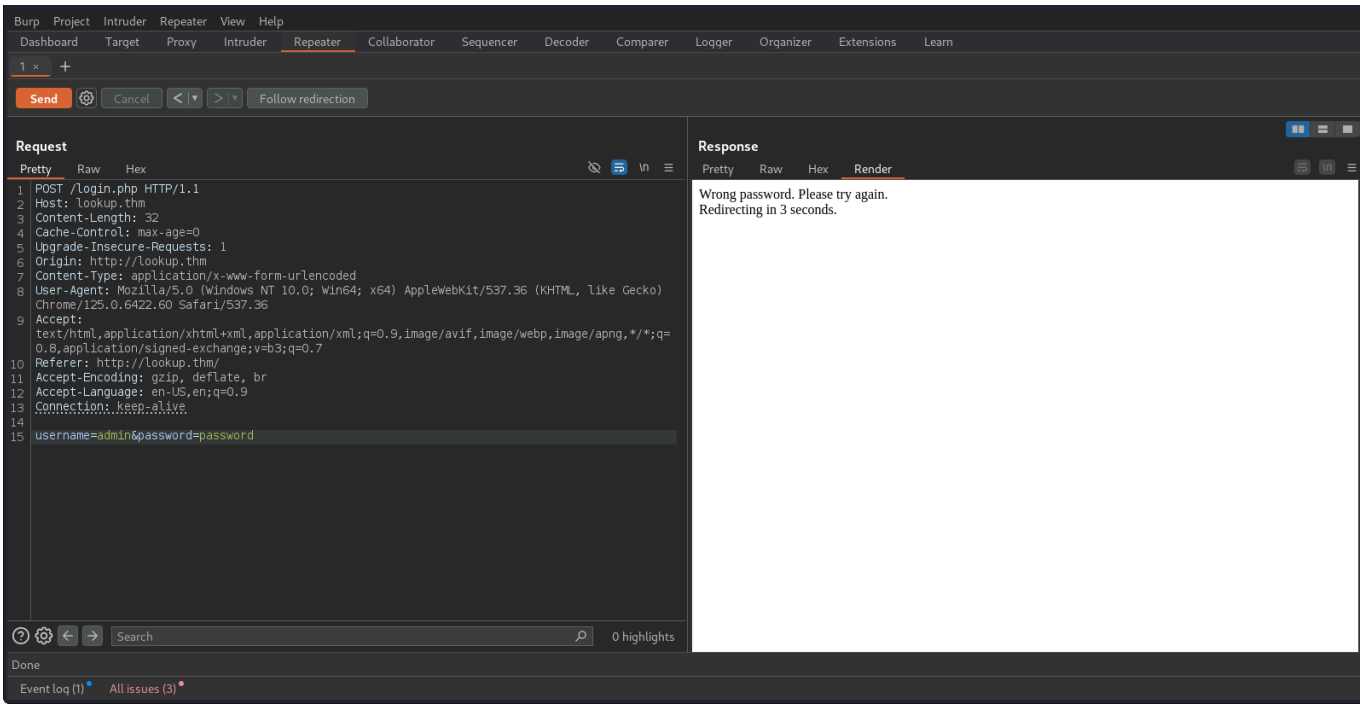
FOOTHOLD

The **nmap** scan revealed an **http** server running on the target, so I accessed it through my browser and landed on a login page.

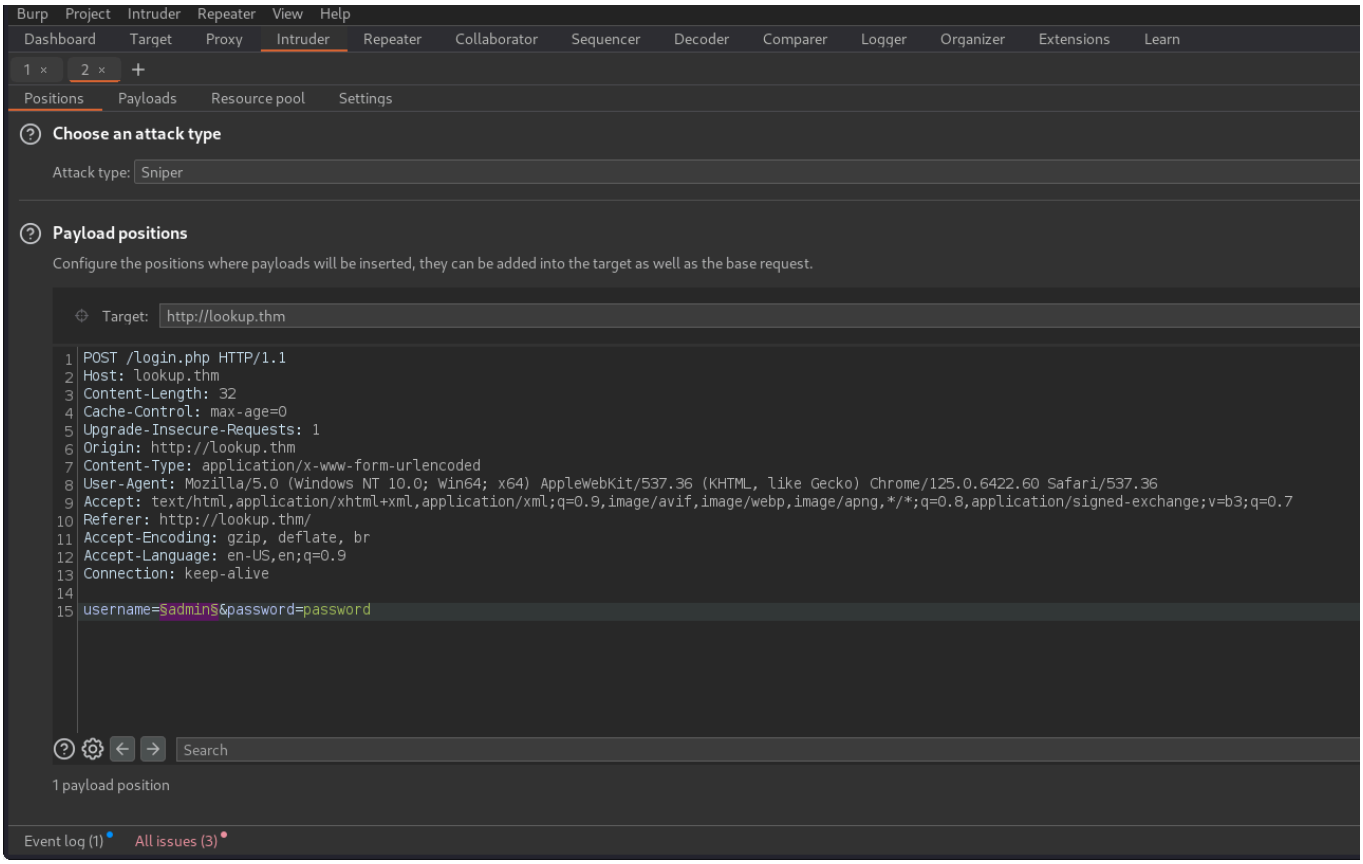


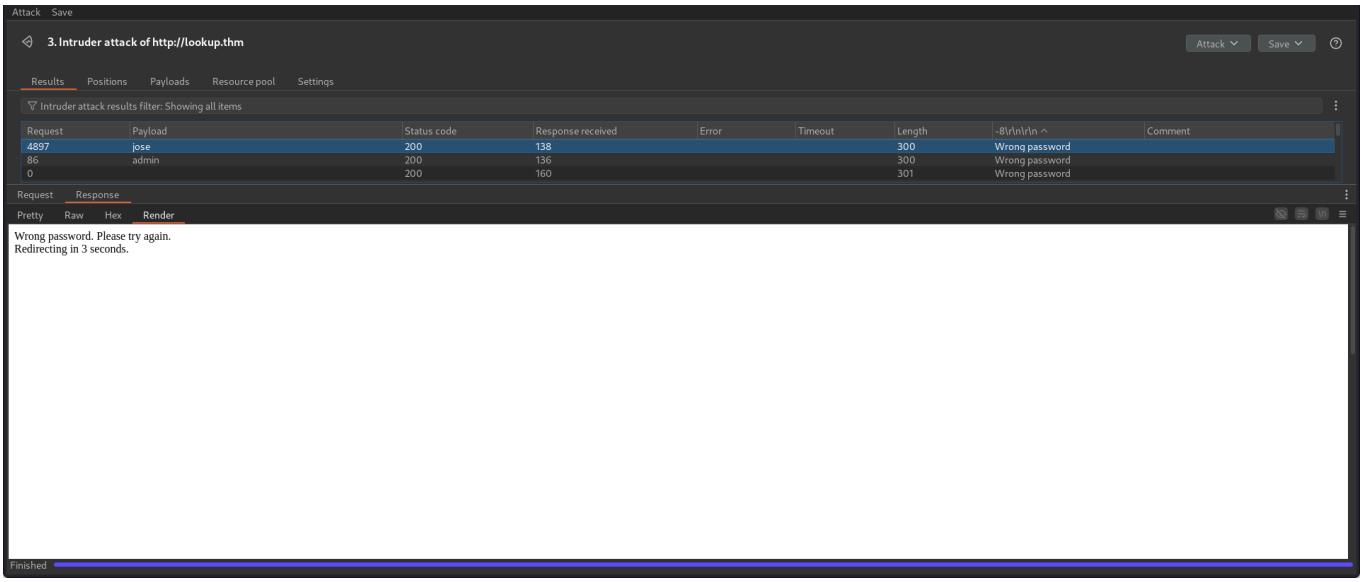
I tried logging in using some default credentials and observed a change in response when a valid username was used.



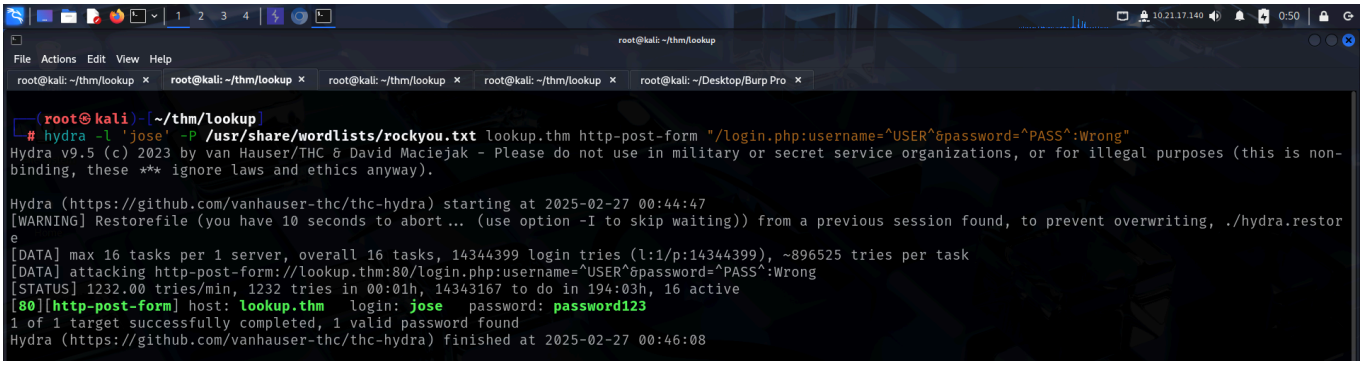


This behavior could be exploited to find valid usernames, so I bruteforced valid usernames using **burpsuite**'s **intruder** from **seclists**.

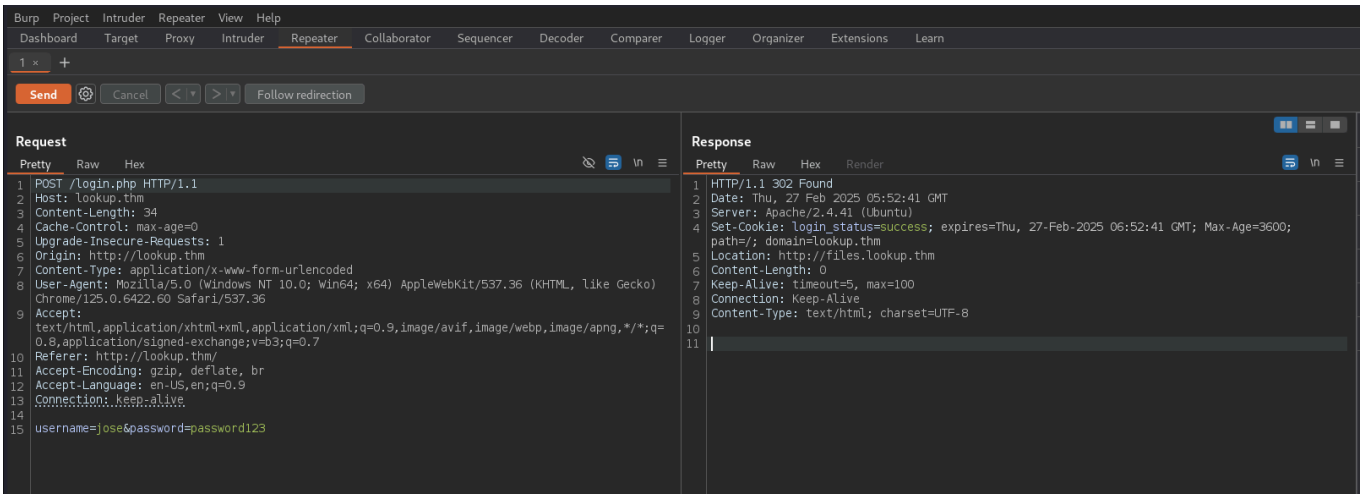




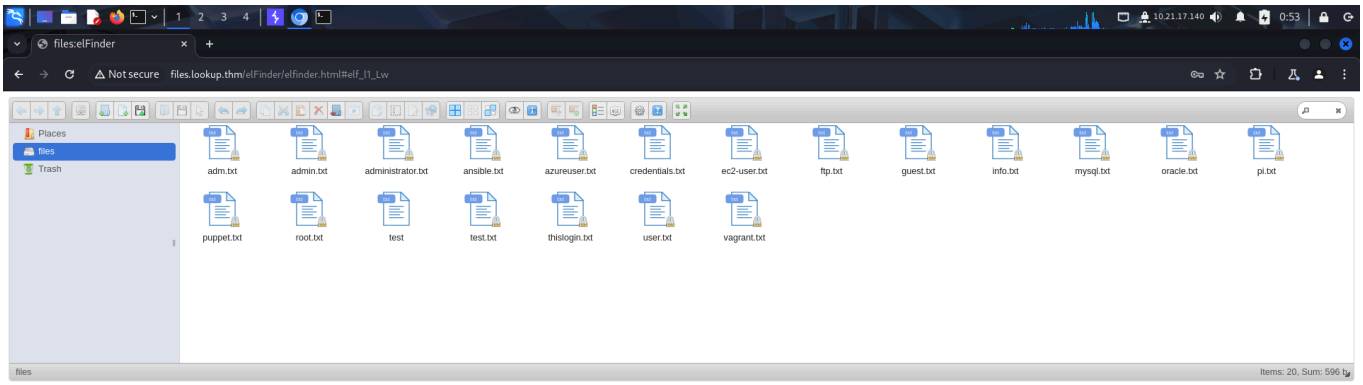
Hence I successfully found another user. I tried bruteforcing the password of admin but failed. However, I was successfully able to bruteforce the password of *jose*.



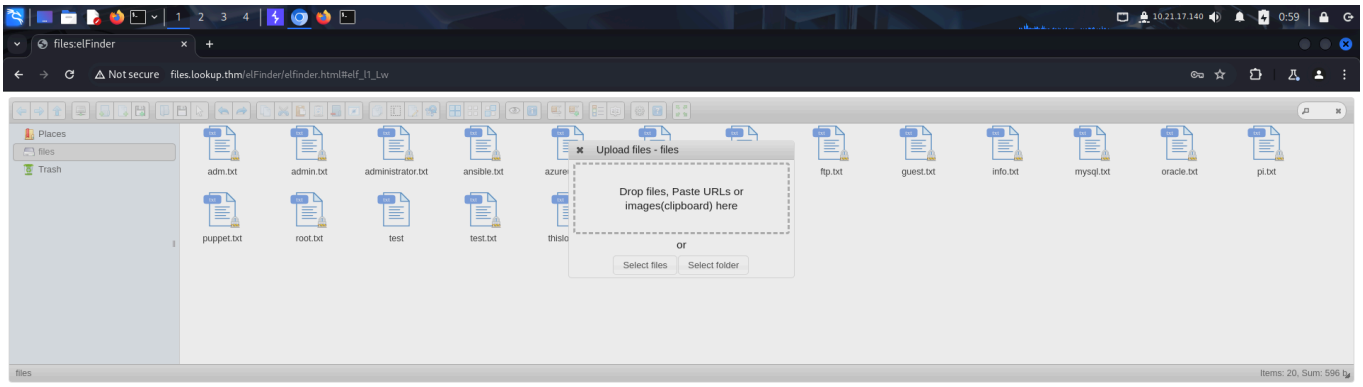
Next, I used the valid credentials to login and was redirected to a subdomain. I added the subdomain to my */etc/hosts* file for appropriate resolution.

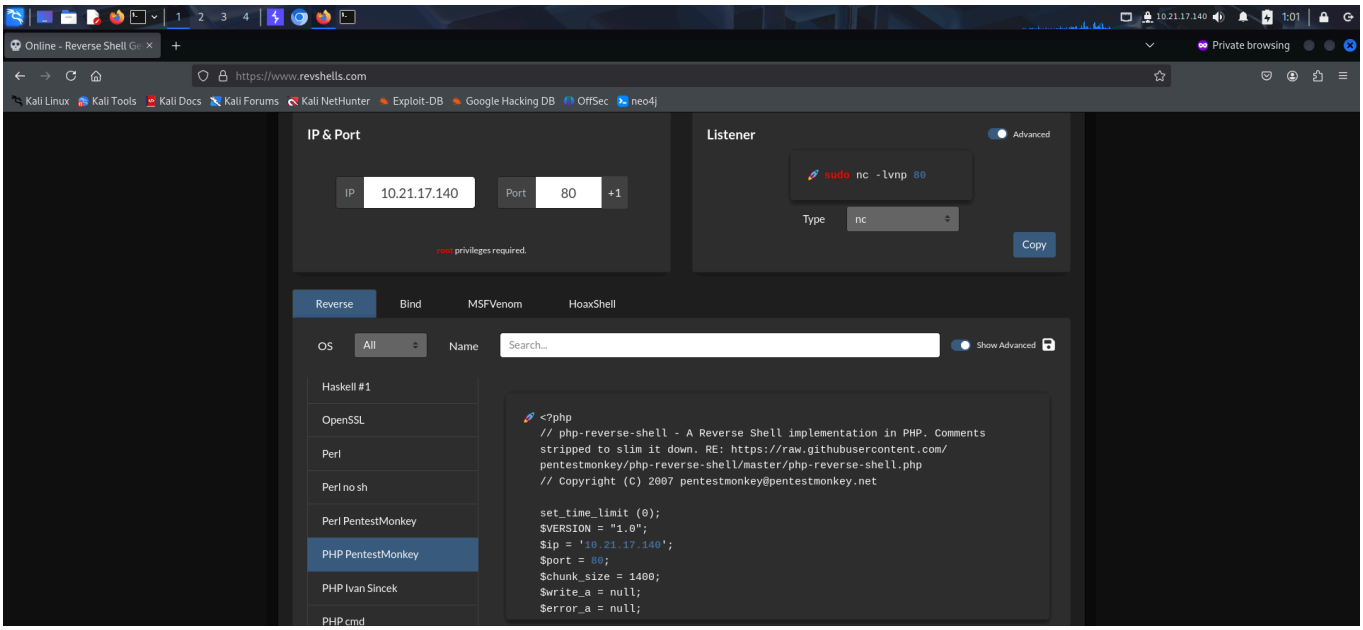


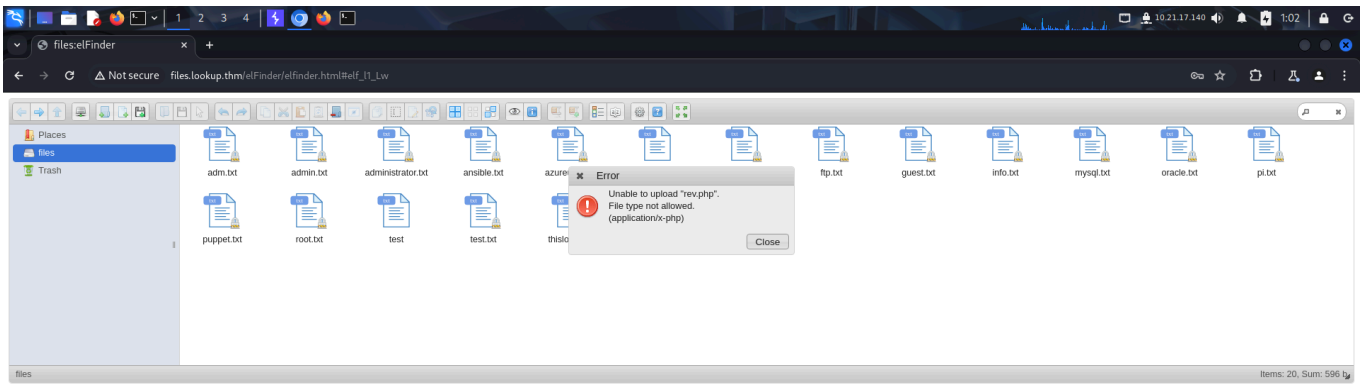
This seemed like a file system. I viewed each file but found nothing interesting.



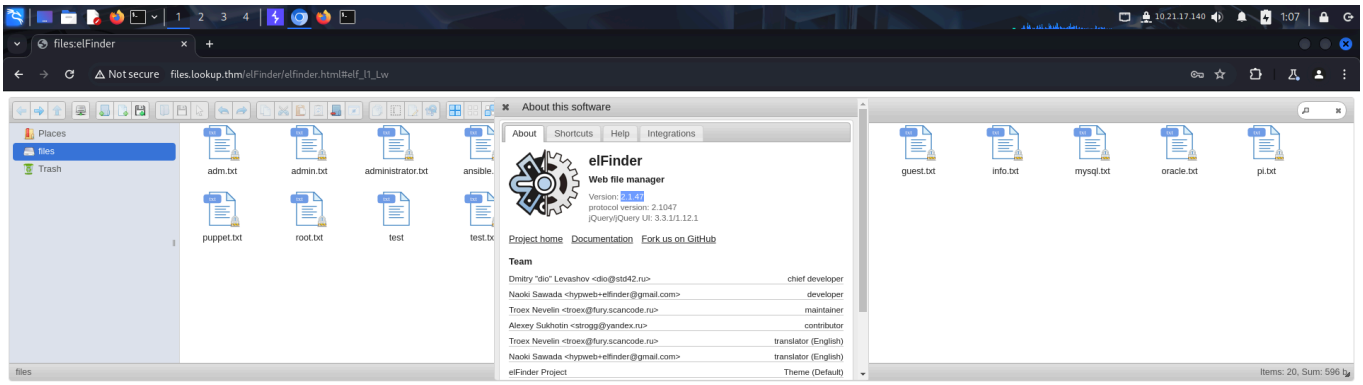
I tested the upload functionality by uploading a php script but failed.

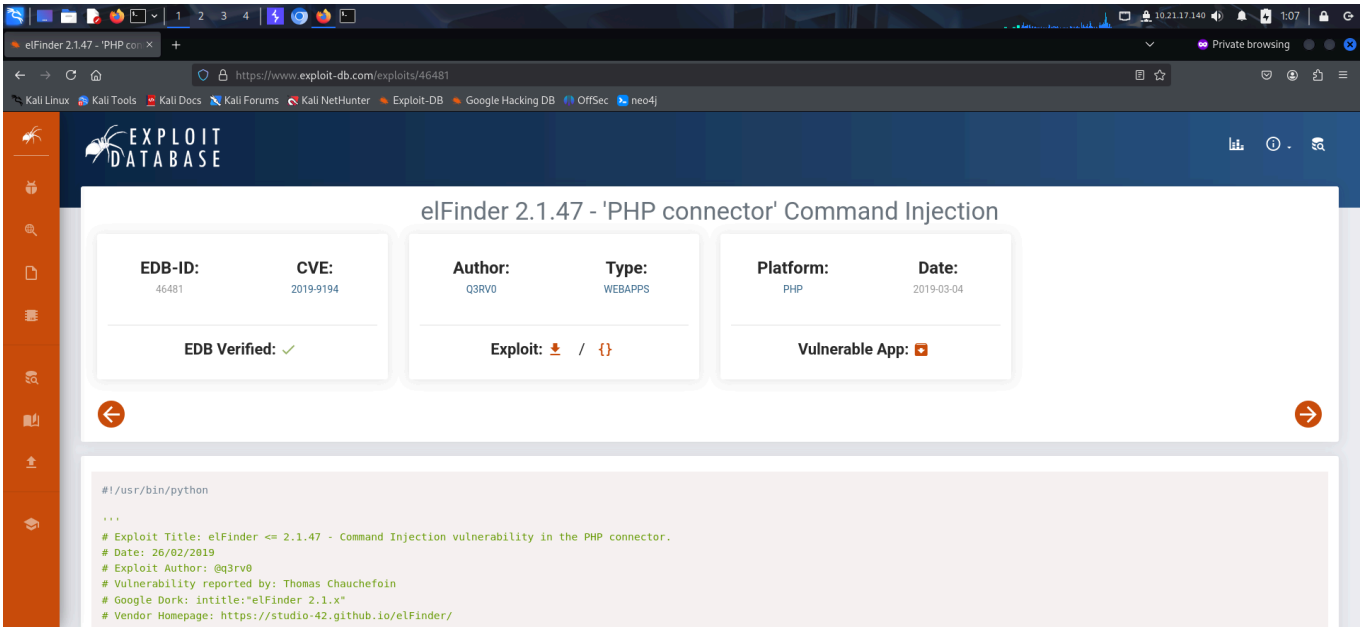


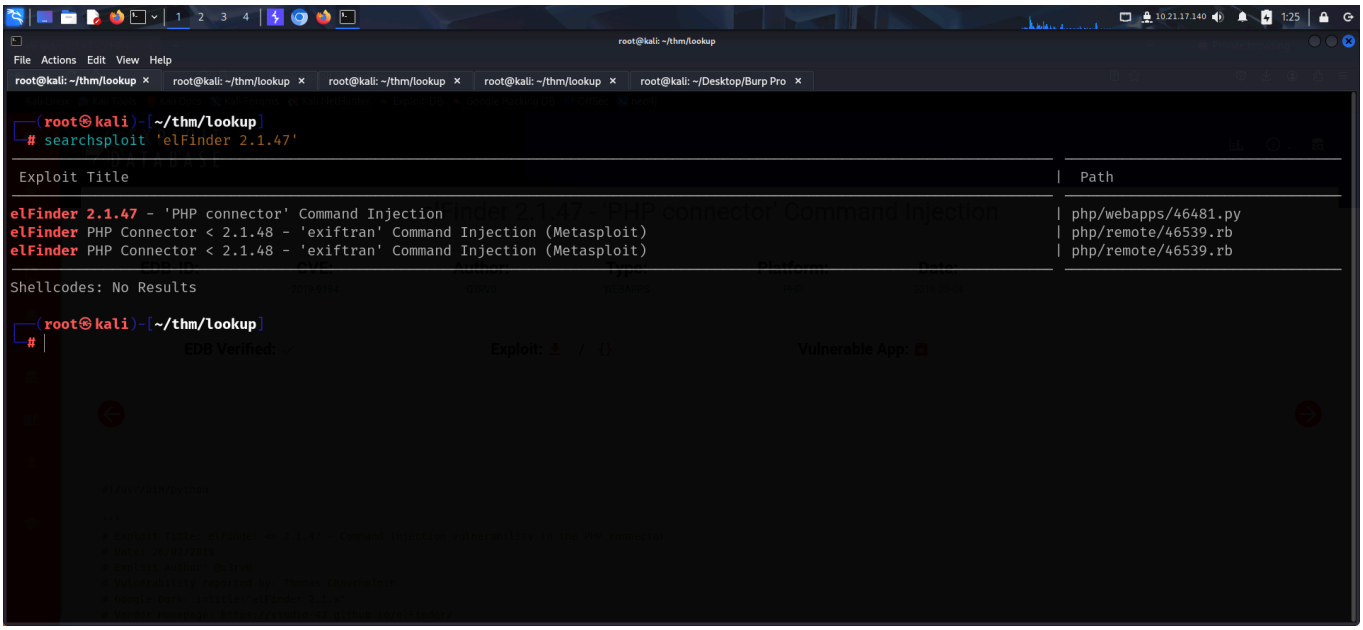




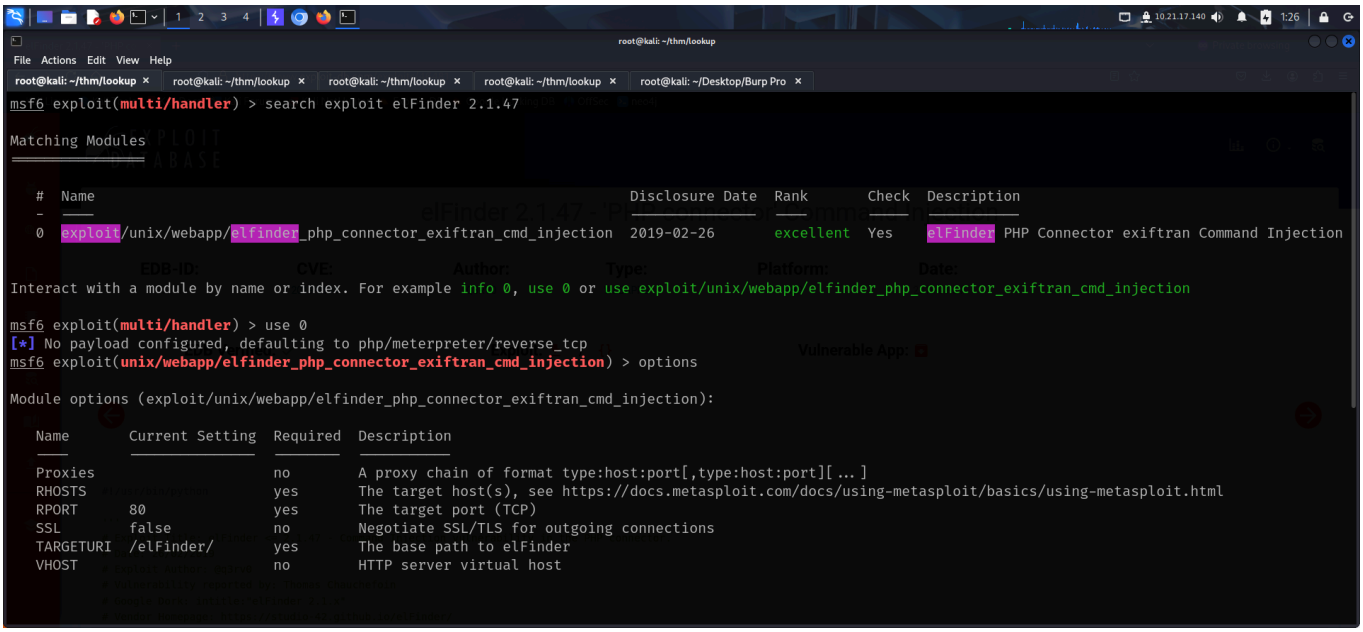
Then I found the version of the cms being used and looked for available exploits.


files:elFinder
Not secure files.lookup.thm/elFinder/elfinder.html#elf_l1_Lw
Places
files
Trash
adm.txt
admin.txt
administrator.txt
puppet.txt
root.txt
test
About this software
About
Shortcuts
Help
Integrations
elFinder
Web file manager
Version: 2.1.47
protocol version: 2.1047
jQuery/jQuery UI: 3.3.1/1.12.1
Project home
Documentation
Fork us on GitHub
Team
Dmitry "dio" Levashov <dio@std42.ru> chief developer
Naoki Sawada <hypweb+elfinder@gmail.com> developer
Troex Nevelin <troex@fury.scancode.ru> maintainer
Alexey Sukhotin <strogg@yandex.ru> contributor
Troex Nevelin <troex@fury.scancode.ru> translator (English)
Naoki Sawada <hypweb+elfinder@gmail.com> translator (English)
elFinder Project Theme (Default)
guest.txt
info.txt
mysql.txt
oracle.txt
pi.txt
Items: 20, Sum: 596

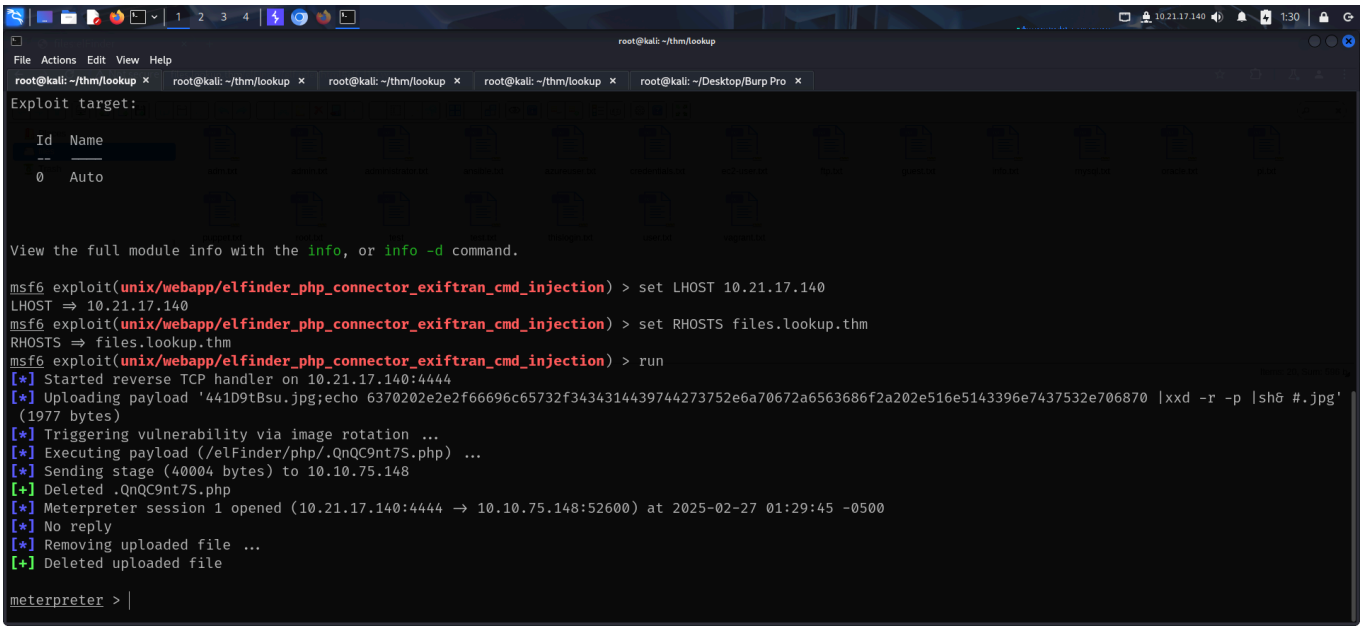

elFinder 2.1.47 - 'PHP con
Private browsing
https://www.exploit-db.com/exploits/46481
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
neo4j
EXPLOIT DATABASE
elFinder 2.1.47 - 'PHP connector' Command Injection
EDB-ID:
46481
CVE:
2019-9194
Author:
Q3RV0
Type:
WEBAPPS
Platform:
PHP
Date:
2019-03-04
EDB Verified:
Exploit:
Vulnerable App:
#!/usr/bin/python
'''
Exploit Title: elFinder <= 2.1.47 - Command Injection vulnerability in the PHP connector.
Date: 26/02/2019
Exploit Author: @q3rv0
Vulnerability reported by: Thomas Chauchefoin
Google Dork: intitle:"elFinder 2.1.x"
Vendor Homepage: https://studio-42.github.io/elFinder/



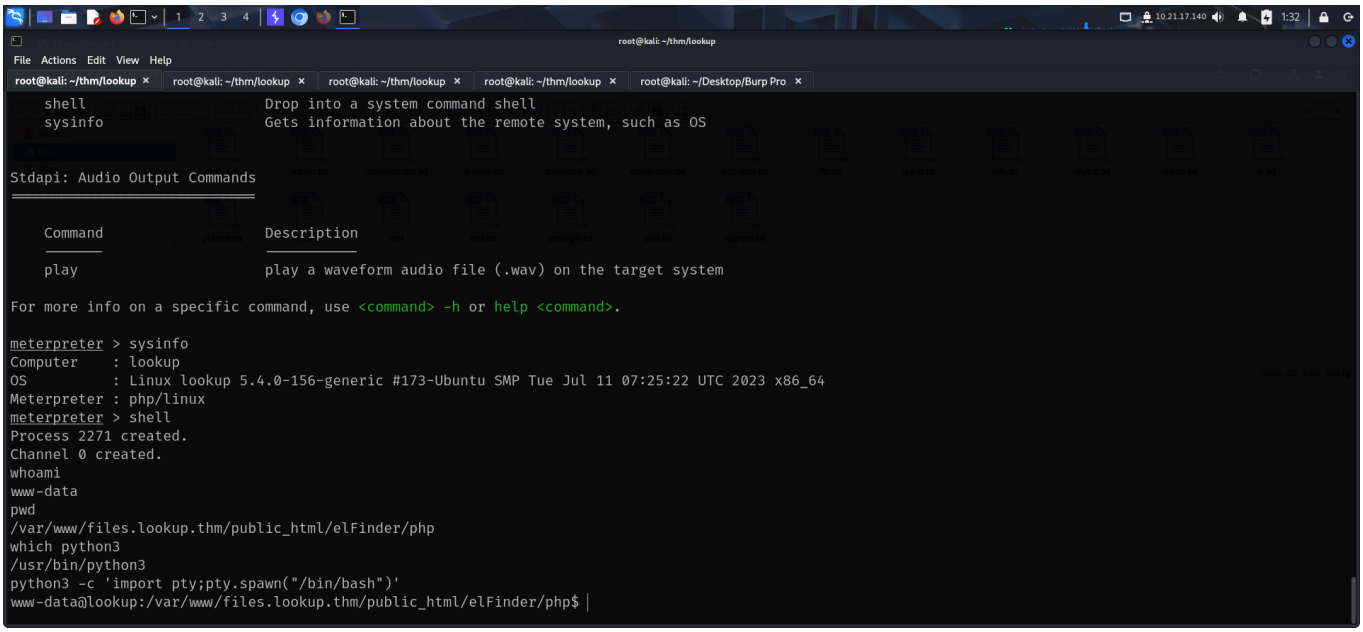
Since there was an exploit available on **Metasploit**, I booted the **metasploit framework** and selected the exploit.



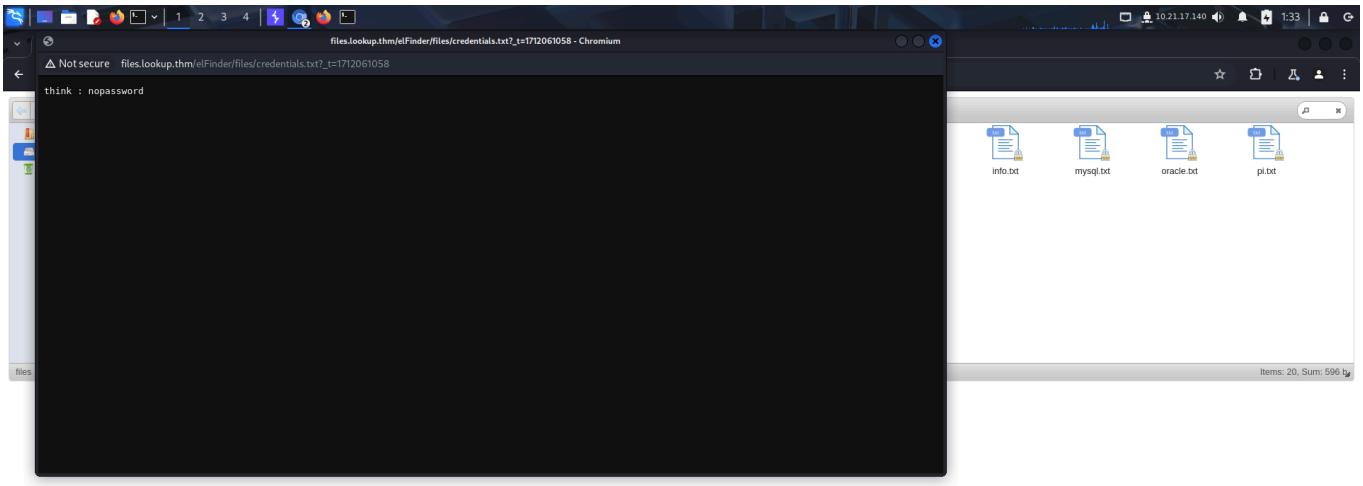
I added the appropriate values and ran the exploit to get a **meterpreter** shell.



I entered shell mode and spawned a tty shell.



I then tried accessing the user flag but failed due to lack of permission. The file system contained some credentials related to the user *think* so I tried switching user using those creds.

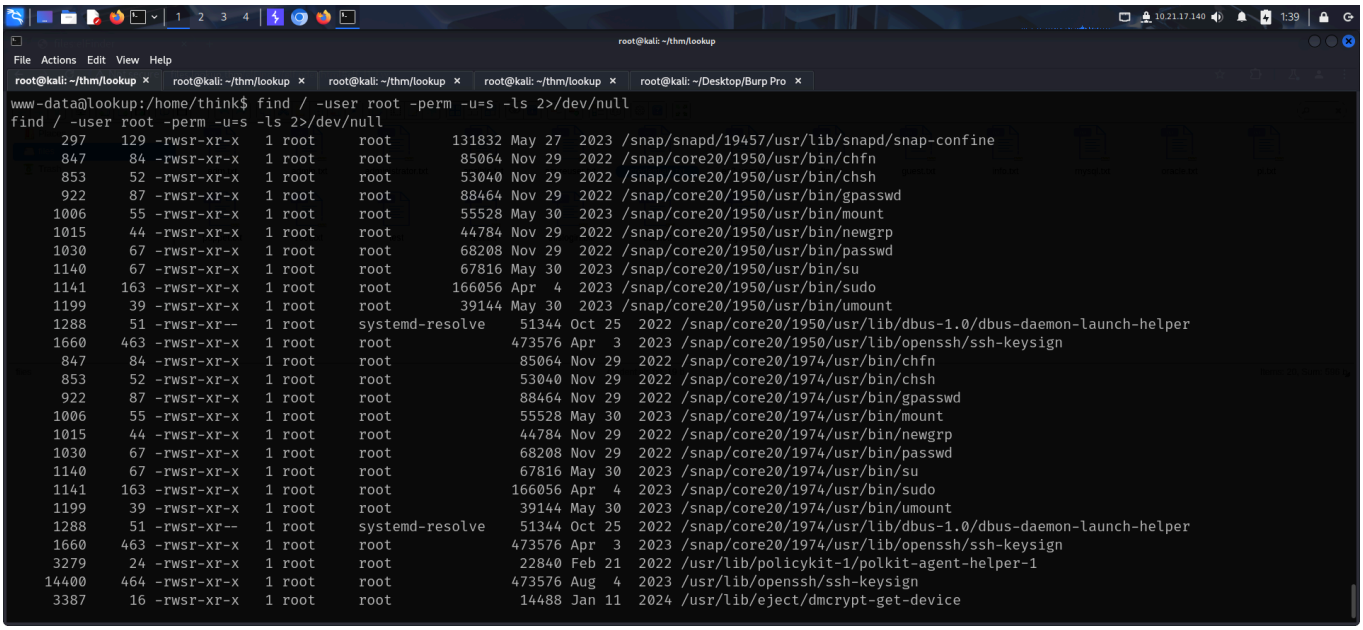


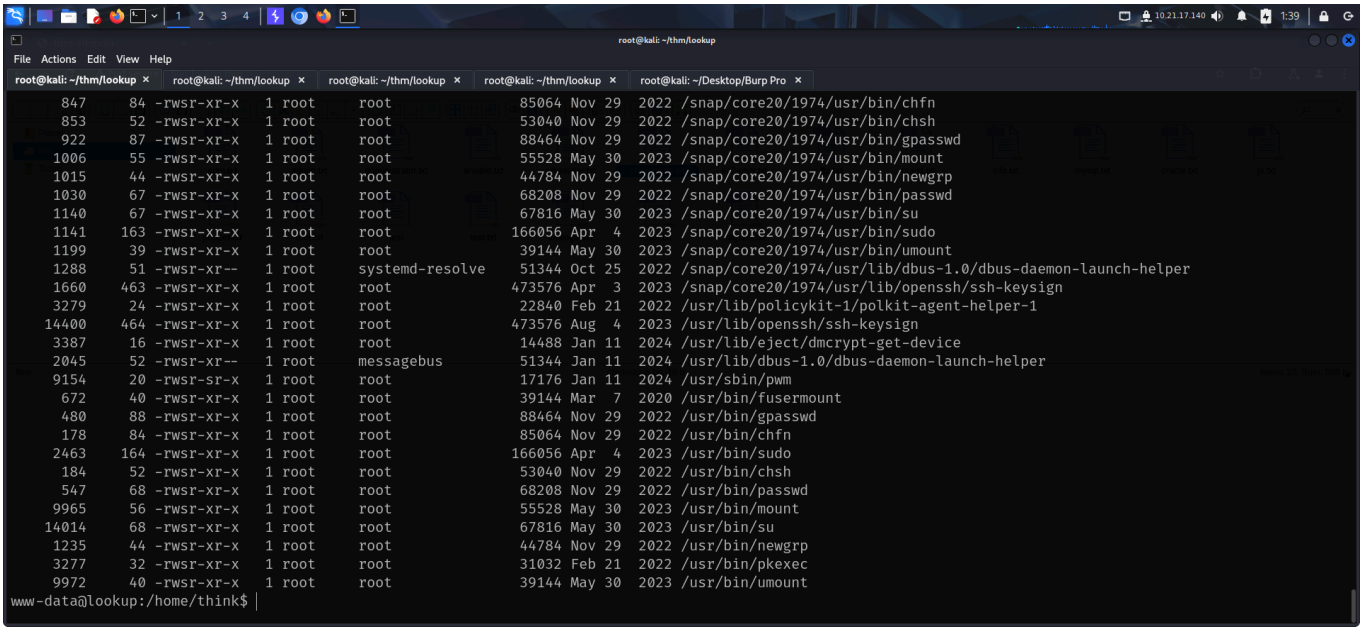

```
www-data@lookup:/home/think$ su -l think
su -l think
Password: nopassword

su: Authentication failure
```

However, I failed. I then looked for anything else that could be useful in *think*'s home directory and found a file called *.passwords*

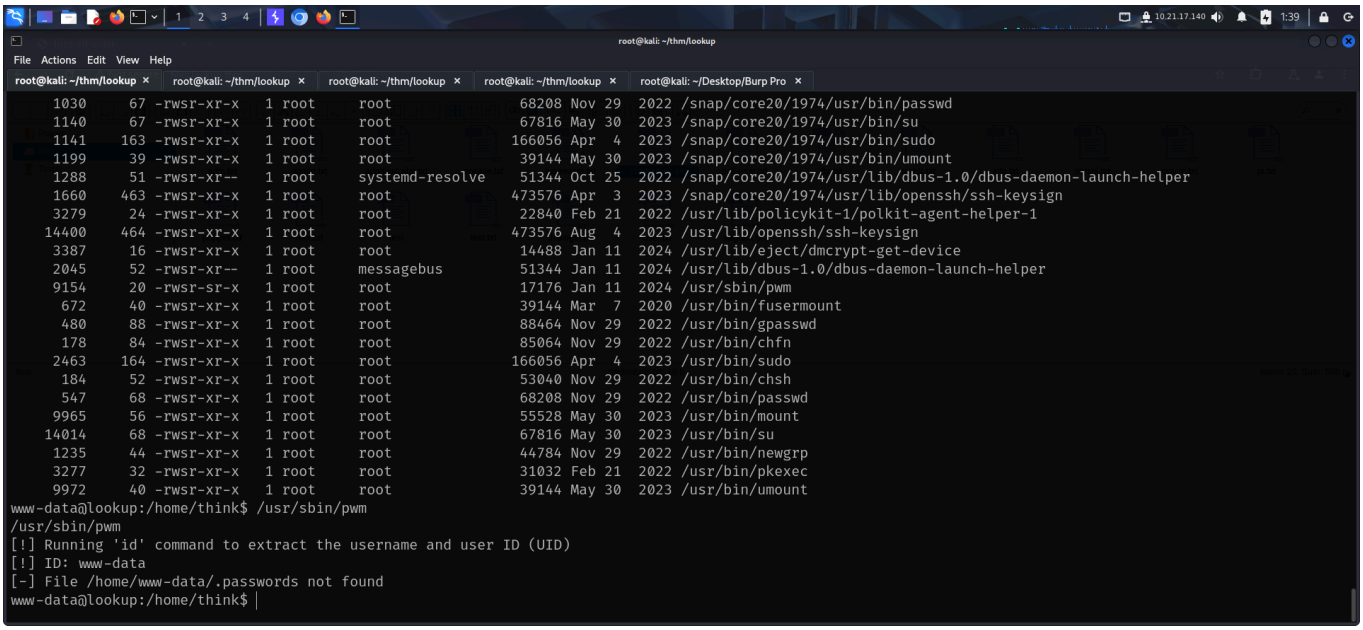
```
www-data@lookup:/home/think$ ls -la
ls -la
total 40
drwxr-xr-x 5 think think 4096 Jan 11  2024 .
drwxr-xr-x 3 root  root  4096 Jun  2  2023 ..
lrwxrwxrwx 1 root  root     9 Jun 21  2023 .bash_history → /dev/null
-rwxr-xr-x 1 think think  220 Jun  2  2023 .bash_logout
-rwxr-xr-x 1 think think 3771 Jun  2  2023 .bashrc
drwxr-xr-x 2 think think 4096 Jun 21  2023 .cache
drwx------ 3 think think 4096 Aug  9  2023 .gnupg
-rw-r----- 1 root  think  525 Jul 30  2023 .passwords
-rwxr-xr-x 1 think think  807 Jun  2  2023 .profile
drw-r----- 2 think think 4096 Jun 21  2023 .ssh
lrwxrwxrwx 1 root  root     9 Jun 21  2023 .viminfo → /dev/null
-rw-r----- 1 root  think   33 Jul 30  2023 user.txt
```

I then looked for binaries with suid bit set.



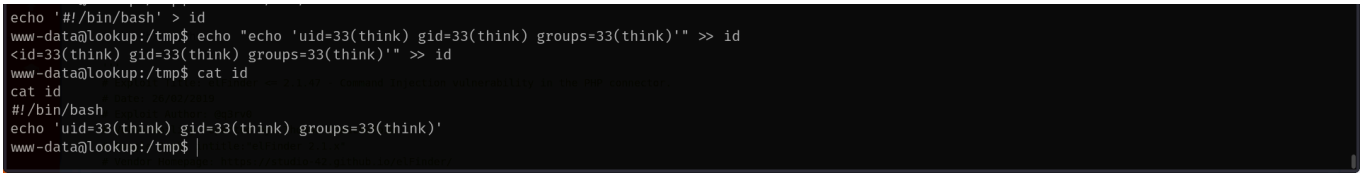


The **pwm** binary seemed interesting so I executed it to see what it does.

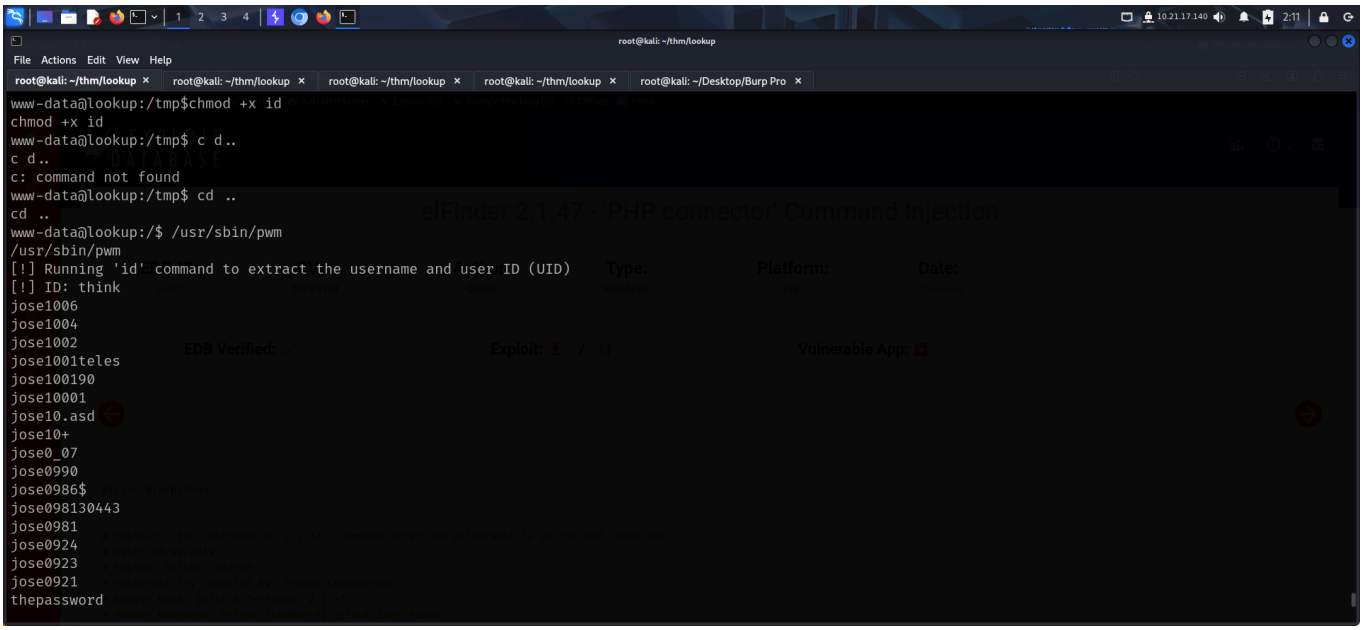


It ran the **id** command and tried extracting passwords from *.passwords* file. I assumed that the whole path of **id** was not being used and hence tried exploiting it.

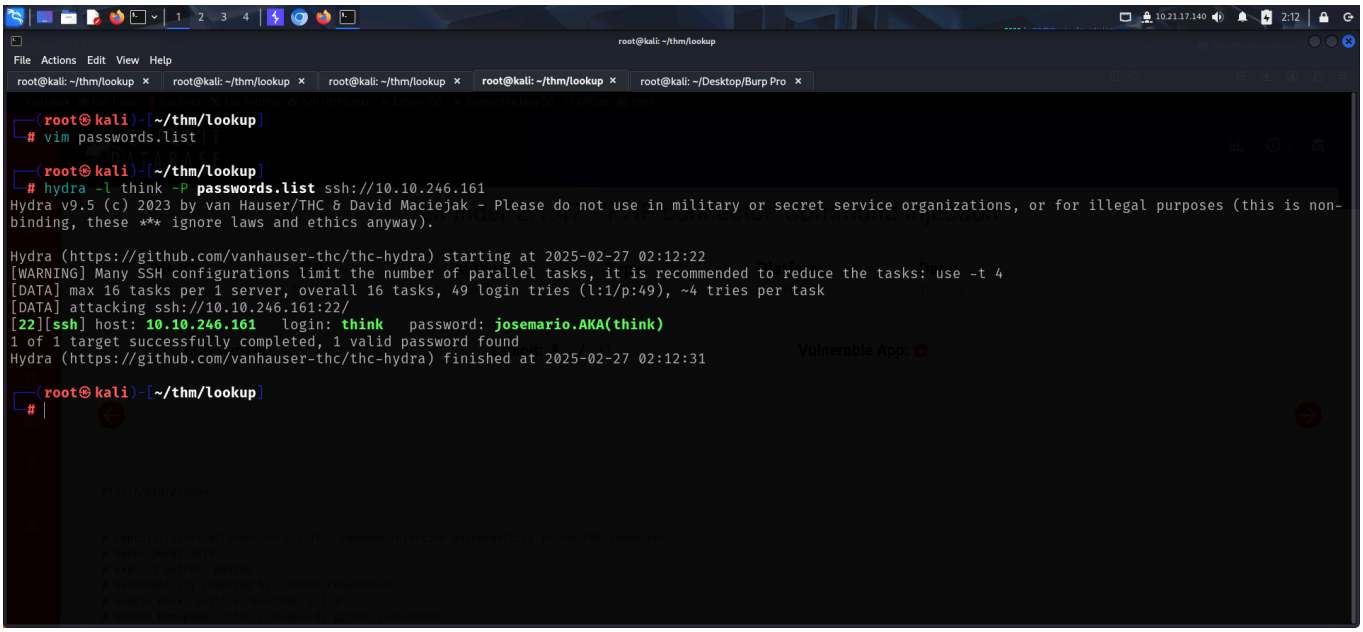
I created a **bash** script that echoed the output of **id** command. I used the id of my current user i.e *www-data* (I got it from */etc/passwd*) and replaced the username with *think*. I saved the script as **id**.



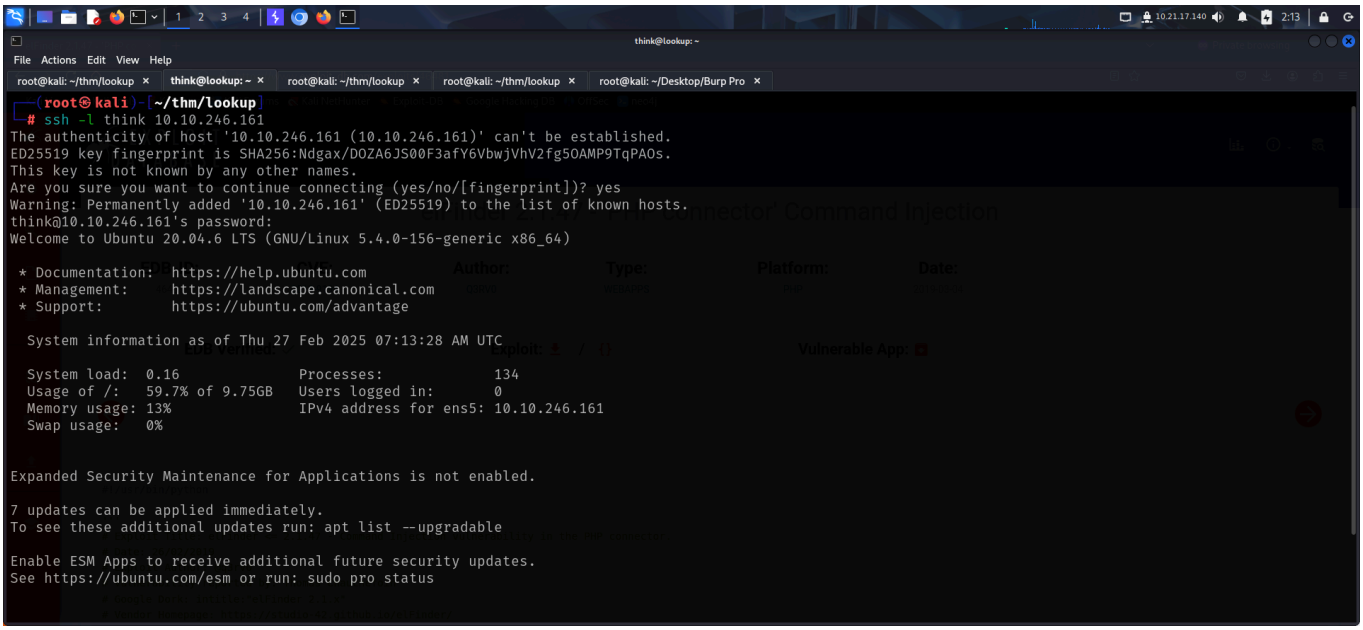
I then appended the */tmp* directory at the start of my path variable and ran the **pwm** command to get a list of passwords for the *think* user.

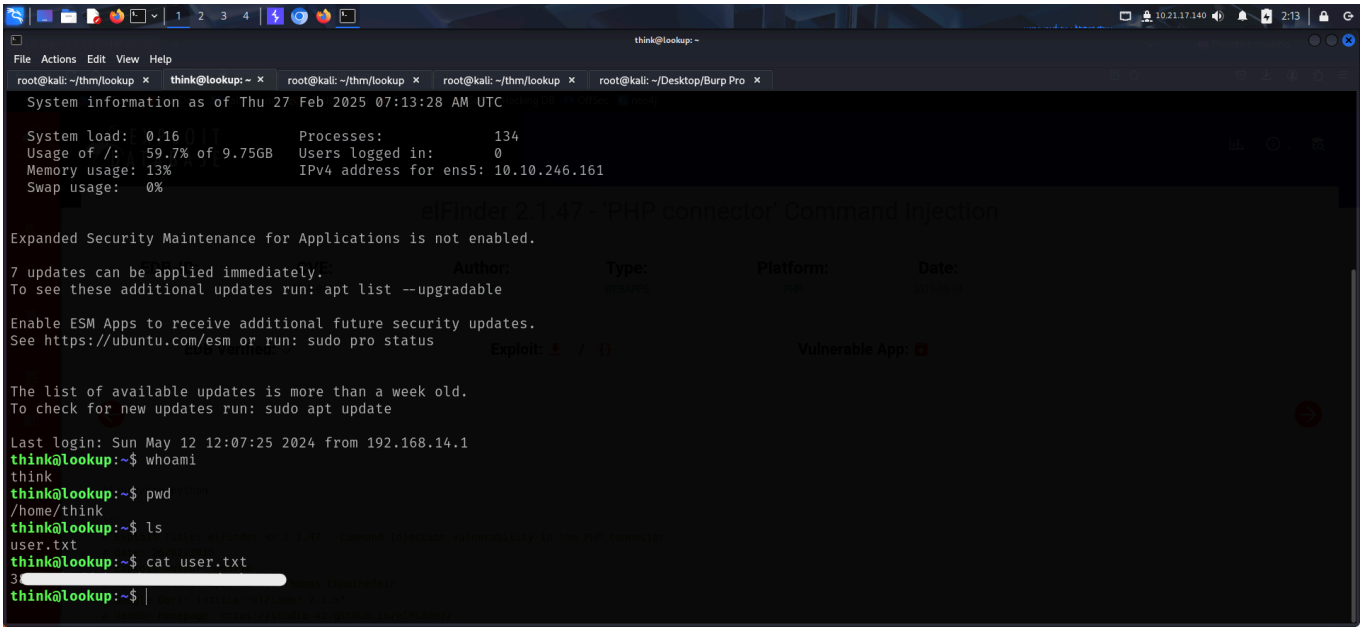


I created a wordlist using these passwords and bruteforced the correct login credentials using **hydra**.



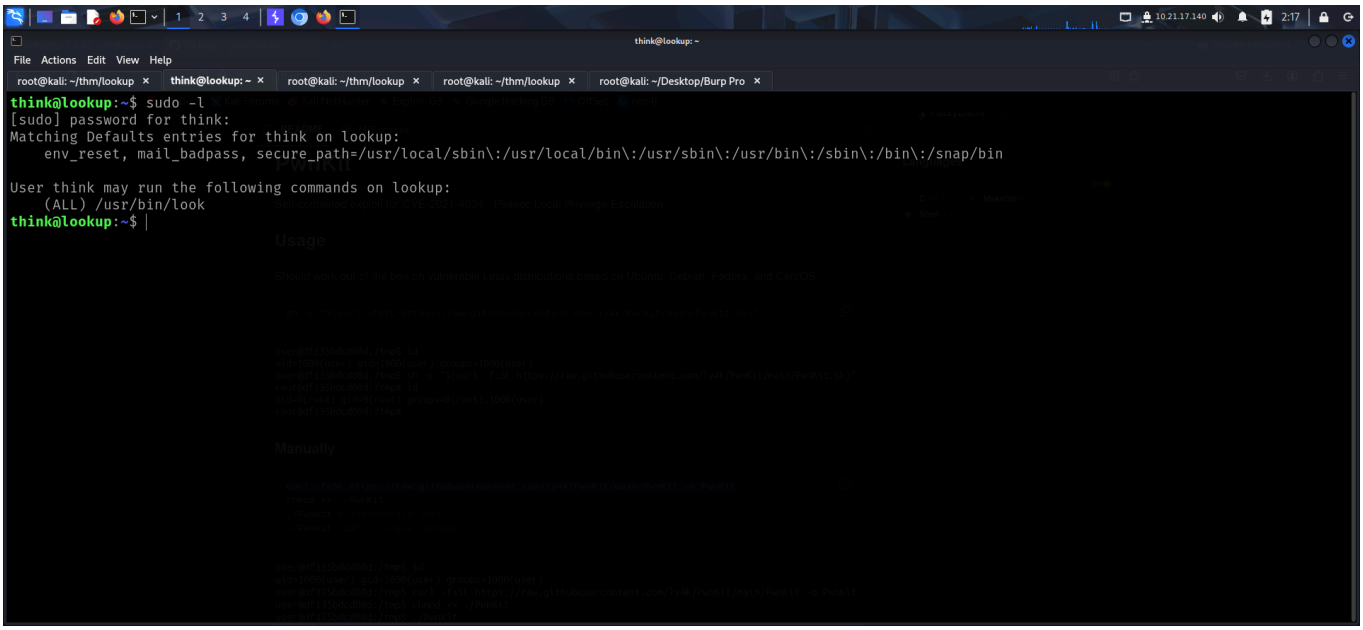
I logged in as *think* and got the user flag from the */home/think* directory.



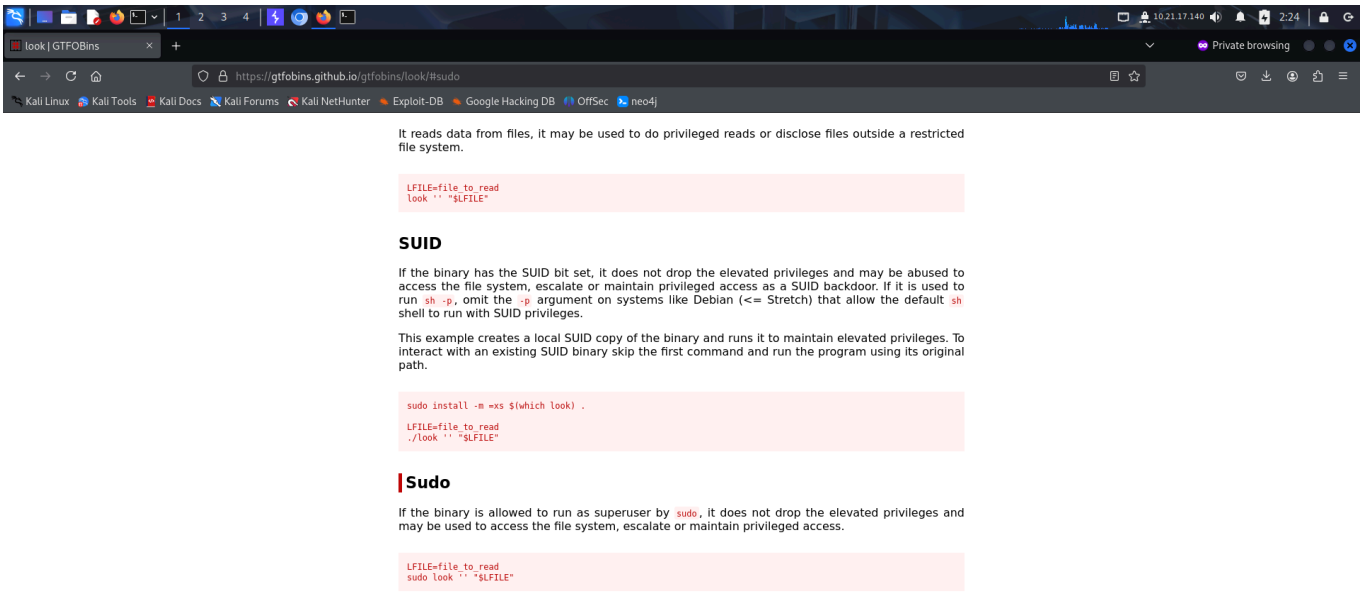


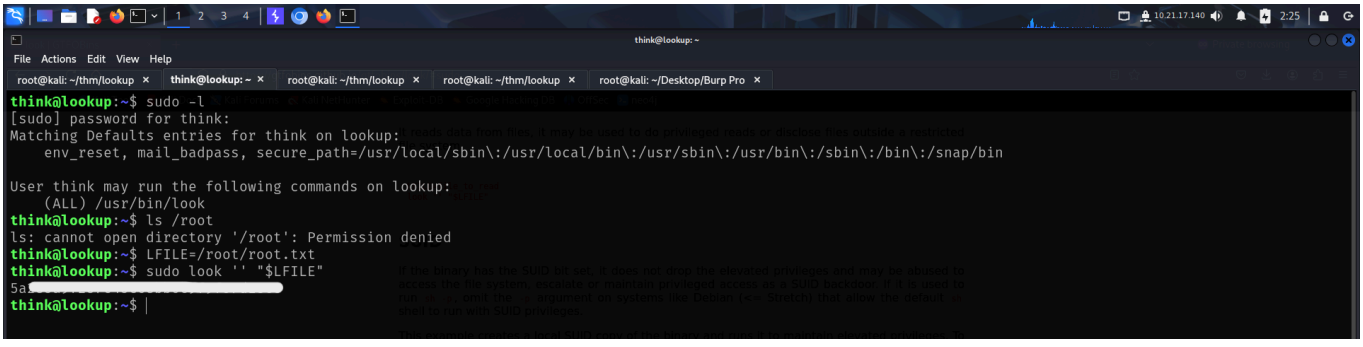
PRIVILEGE ESCALATION

I then looked for **sudo** permissions and found I was allowed to run the **look** command.



I visited **gtfobins** and found a way to use **look** to read the root flag.





CONCLUSION

Here's a short summary of how I pwned **LOOKUP**:

- I exploited the **cms** vulnerability to get reverse shell as *www-data*.
- I modified the path environment variable and exploited the **pwm** binary containing an suid bit to get passwords of *think* user.
- I captured the user flag from *think*'s home directory.
- I then exploited the **sudo** privileges to read the root flag present in */root* directory.

That's it from my side, until next time :)
