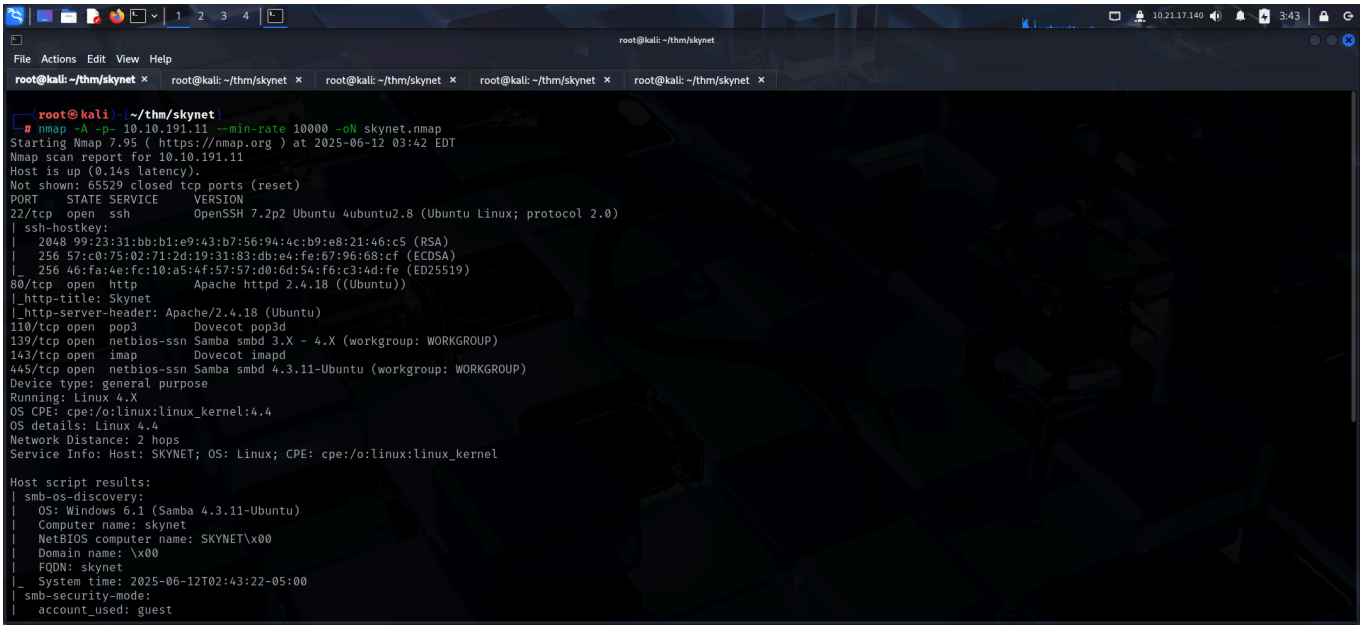
SKYNET

To access the machine, click on the link given below:

- https://tryhackme.com/room/skynet

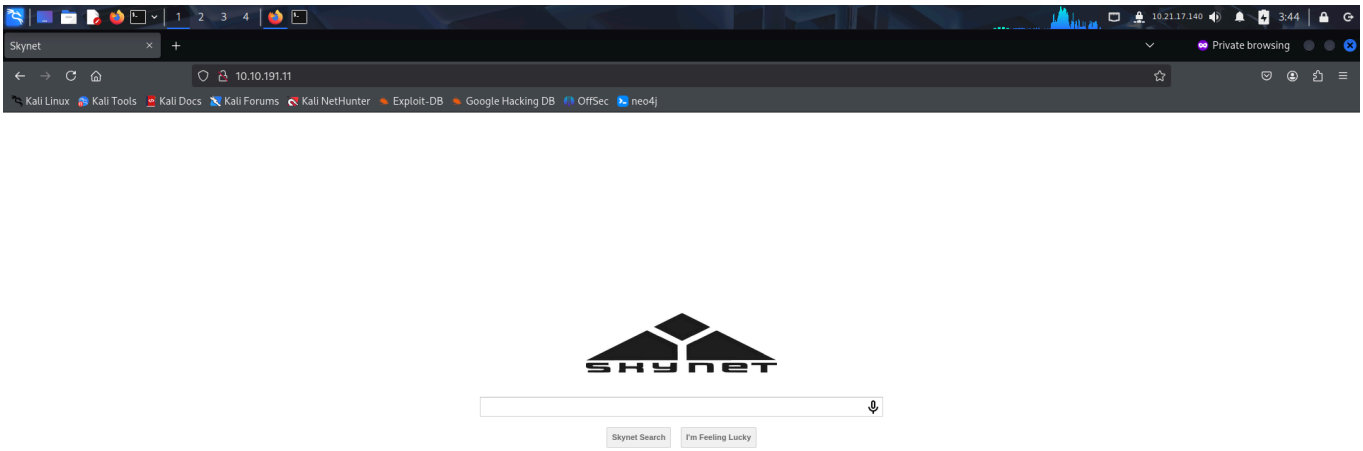
SCANNING

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

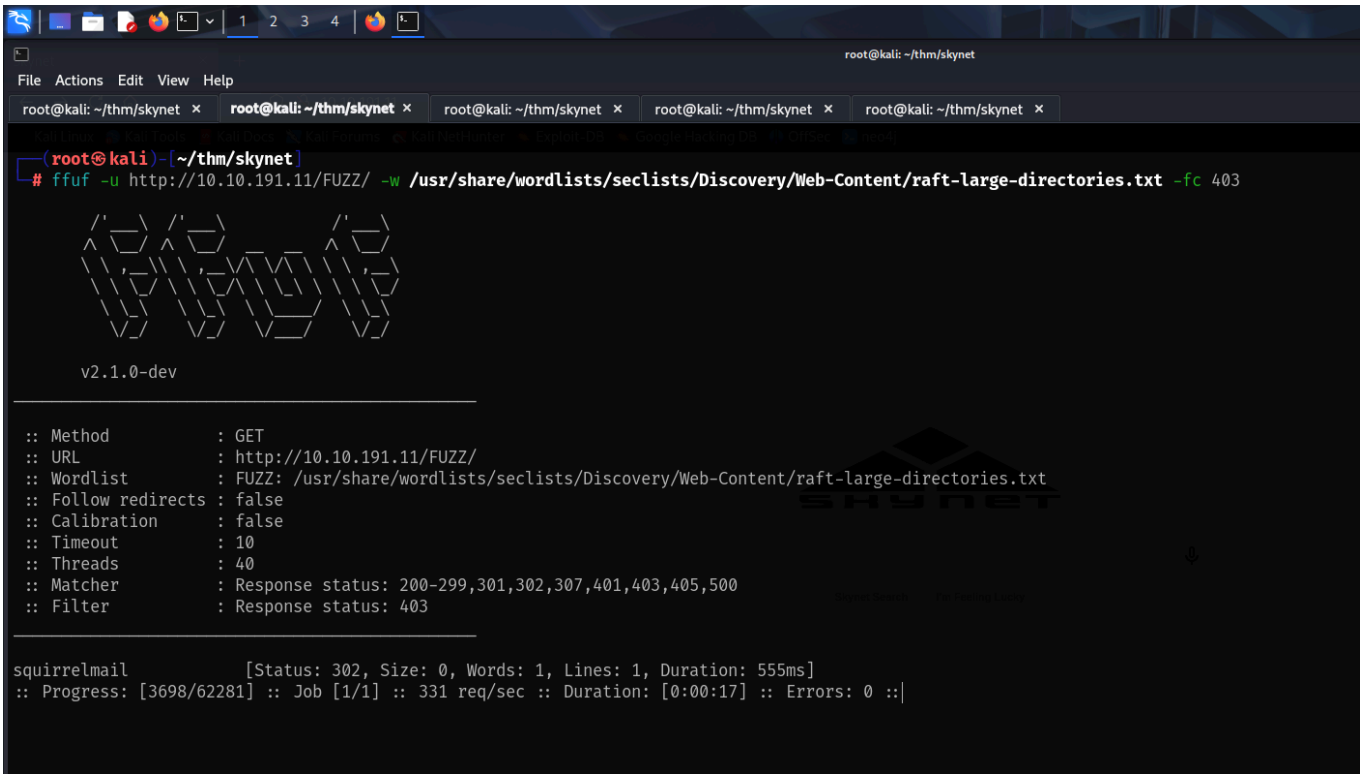


FOOTHOLD

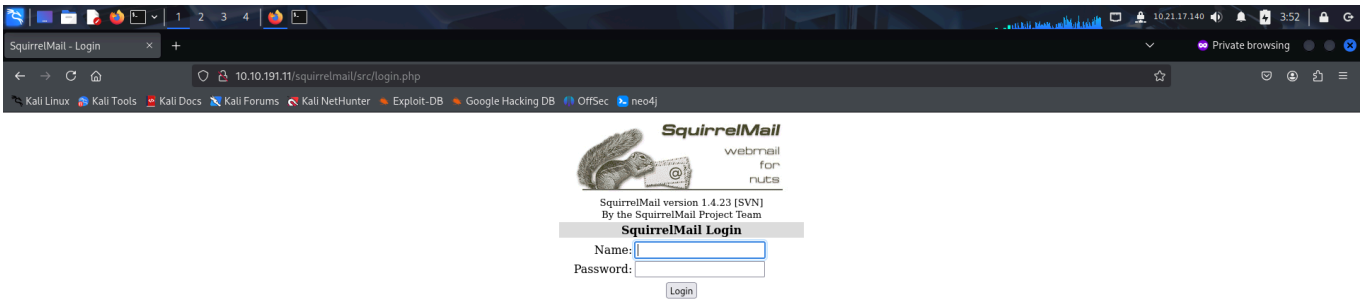
I visited the web application running on the target through my browser.



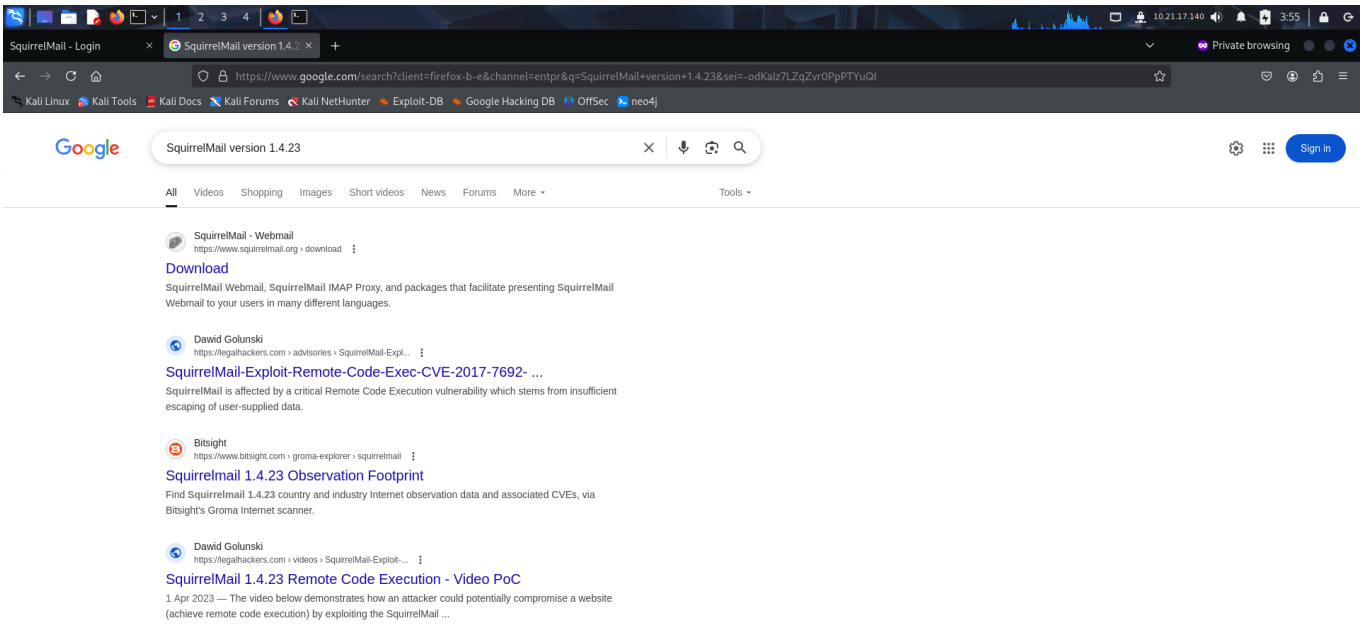
I then fuzzed for interesting directories using **ffuf** and found a directory called *squirrelmail*



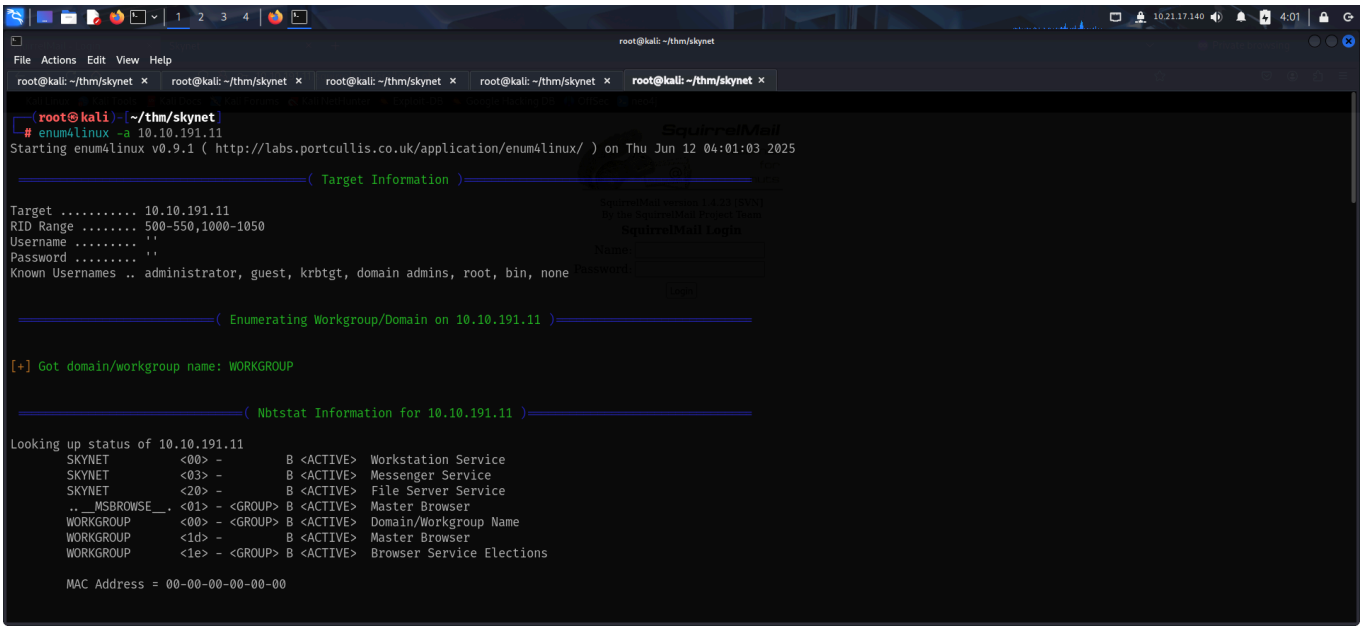
Visiting the endpoint revealed a login panel.

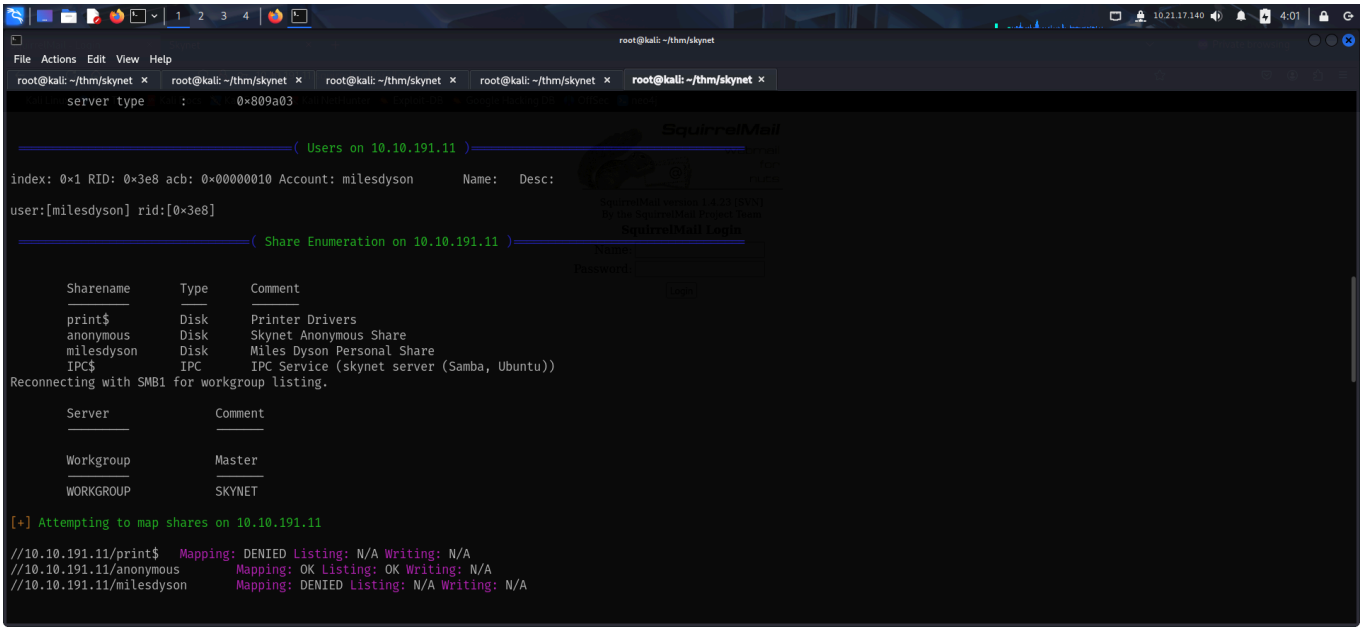


I did a google search regarding the version of *squirrelmail* and found some articles that hinted towards a remote code execution vulnerability.

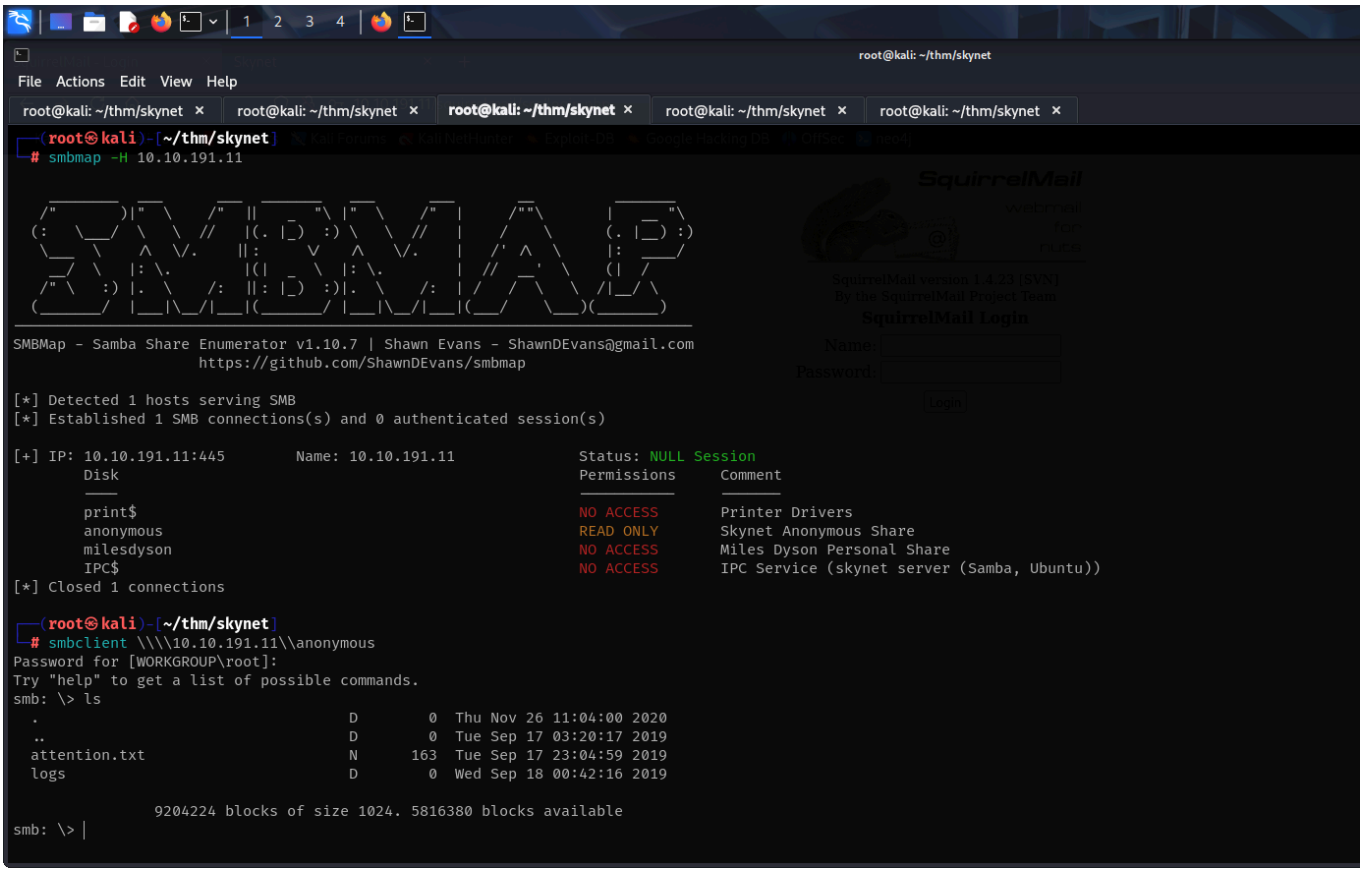


I then enumerated smb running on the target using **enum4linux** and found a username and interesting shares.

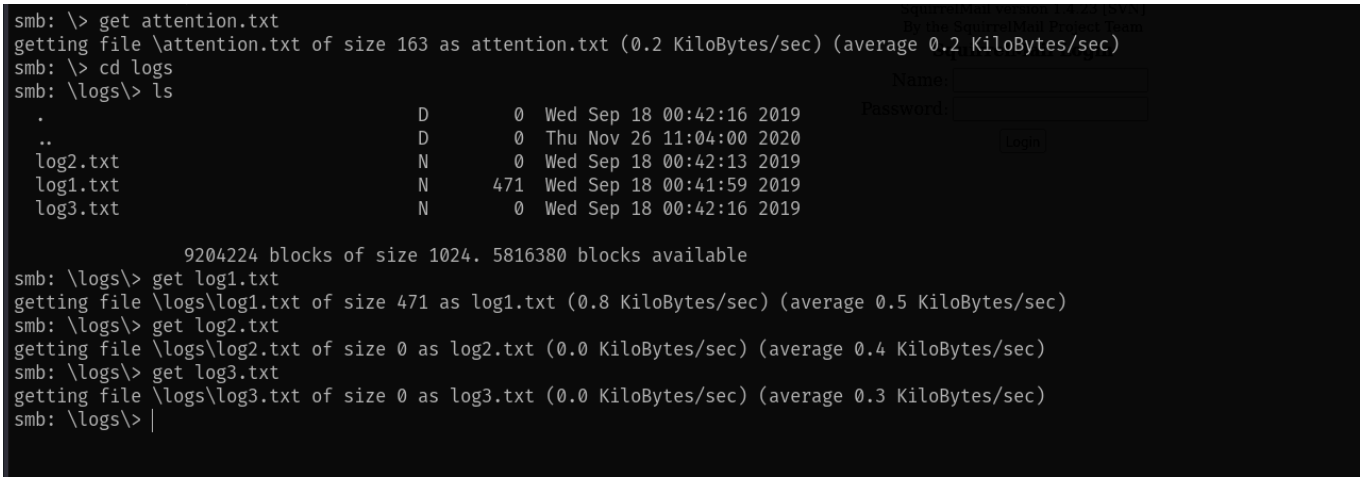




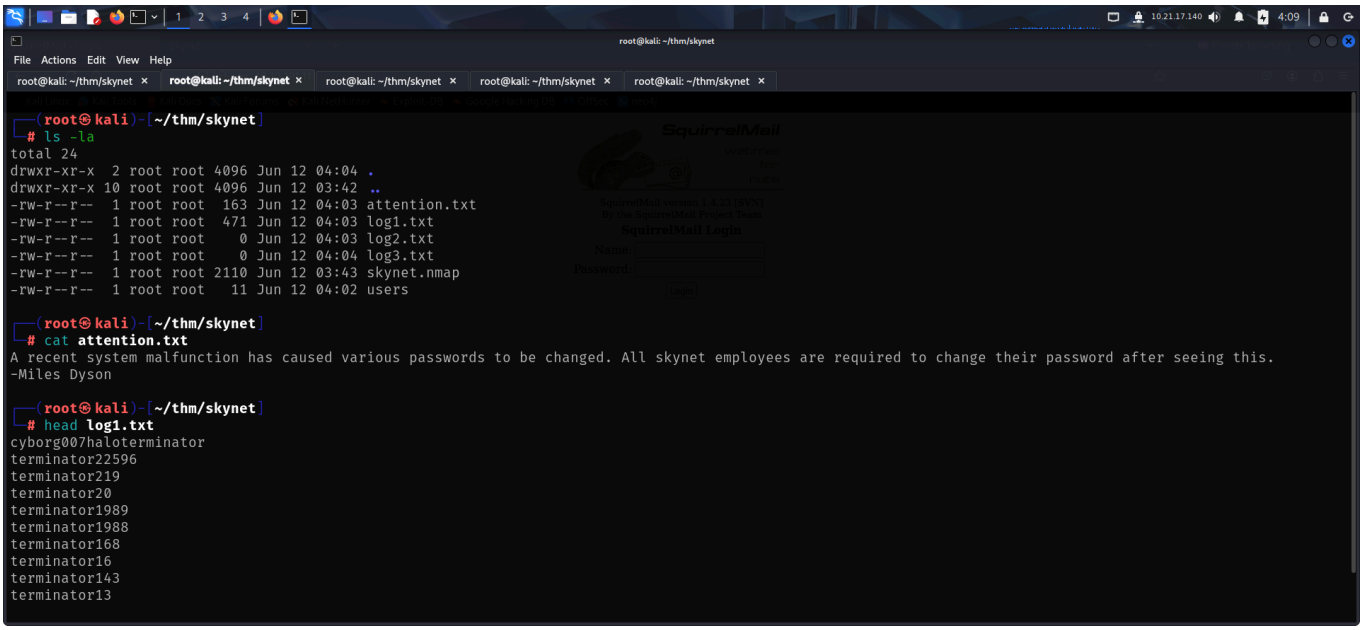
The *anonymous* share seemed interesting, so I connected to it and found some text files.



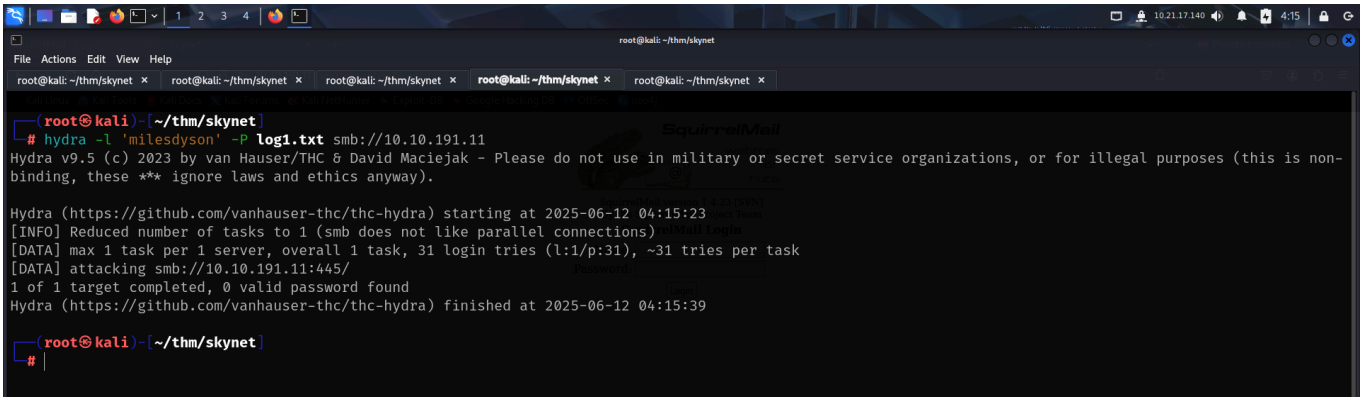
I downloaded these files onto my local system.



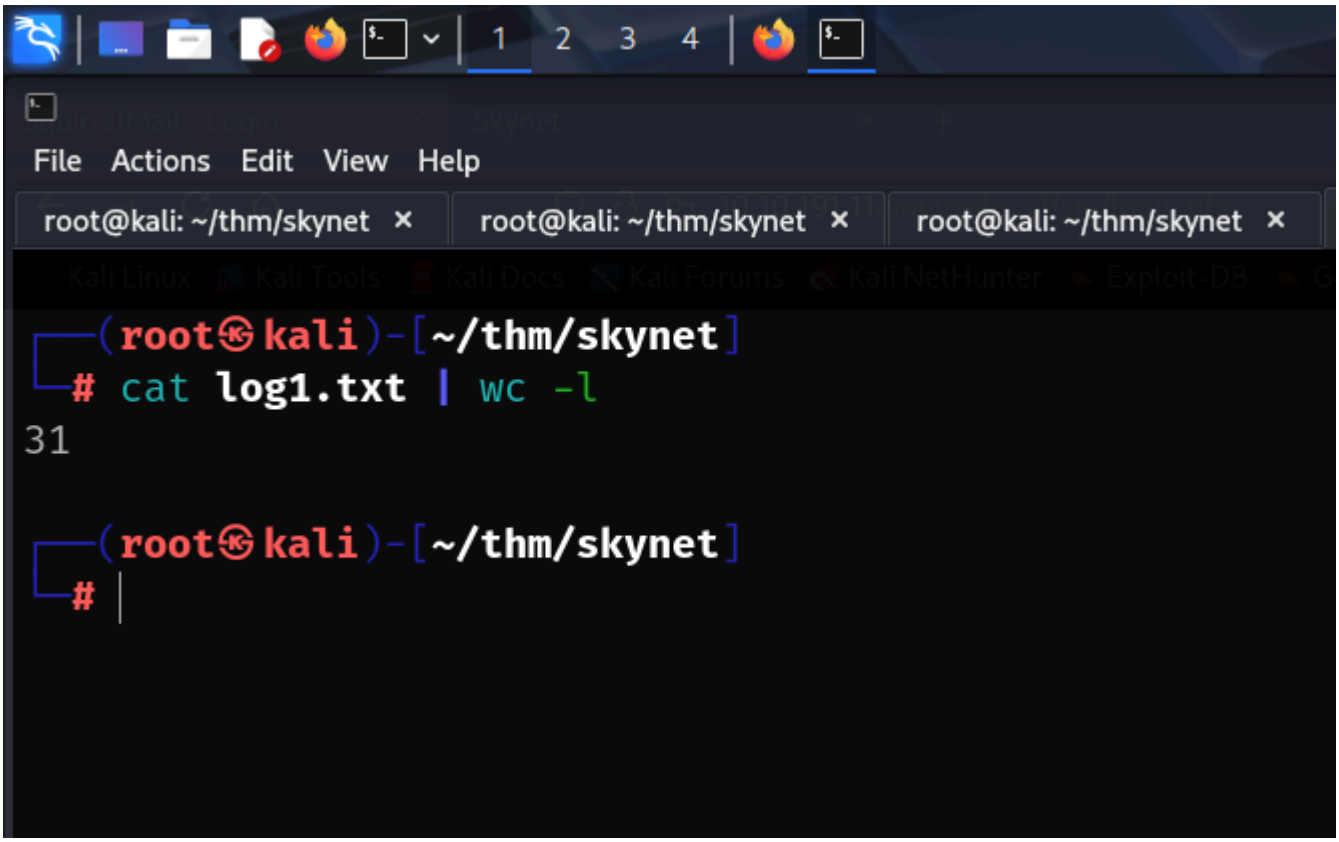
The *attention.txt* file confirmed the existence of a user called *milesdyson*. *log1.txt* contained a wordlist.

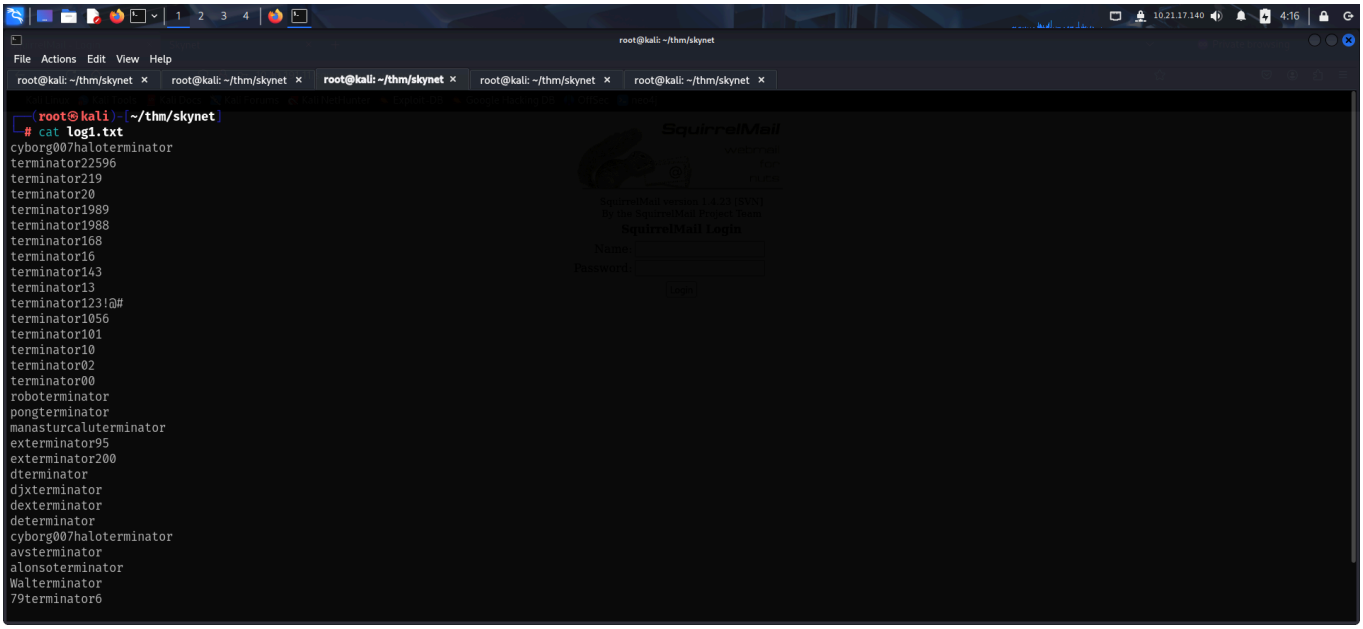


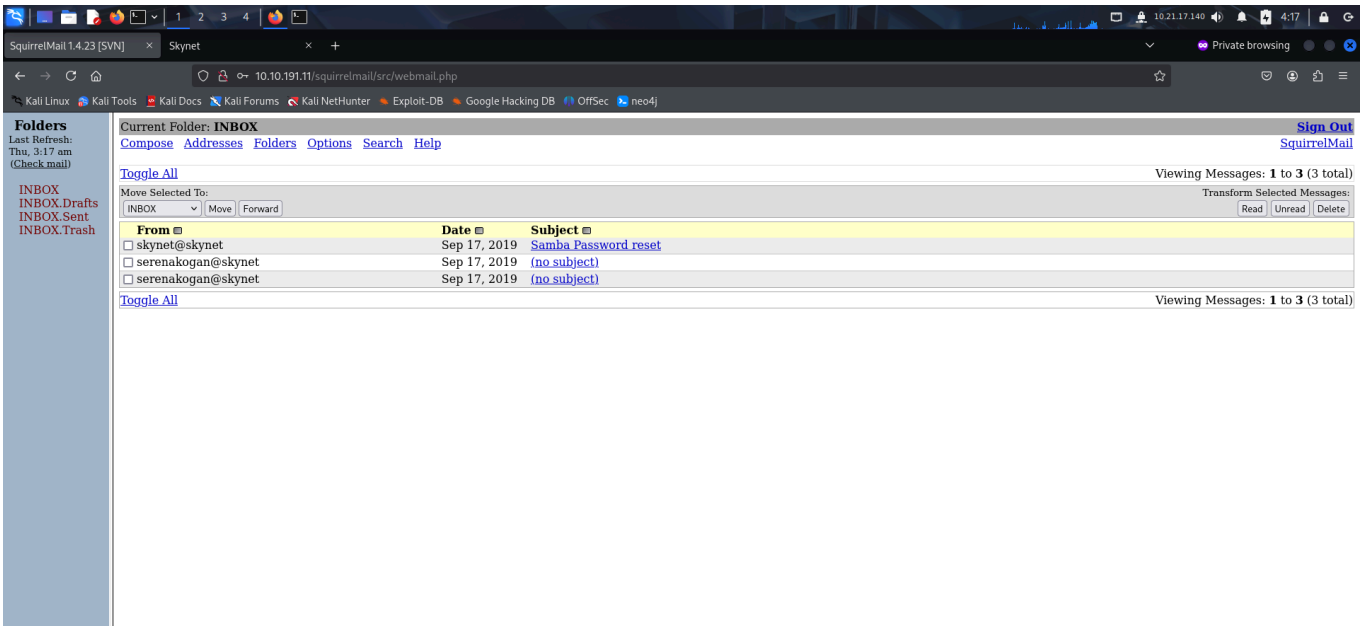
I tried brute forcing the password of *milesdyson* but failed.



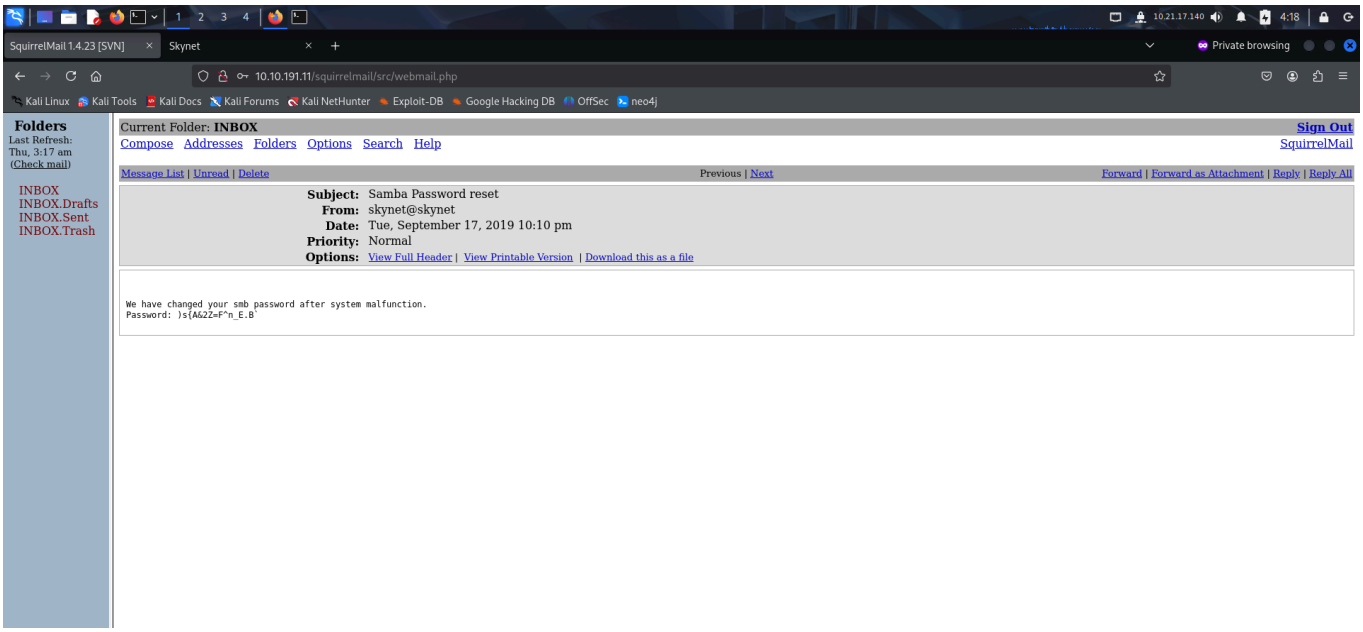
Since the wordlist only contained 31 words, I used them to try logging into *squirrelmail* as *milesdyson* and was successfully able to log in using the first password.

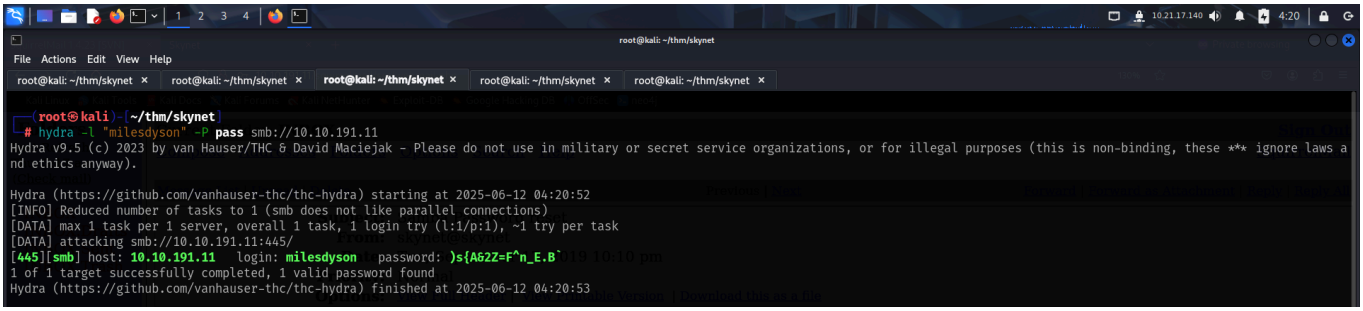

File Actions Edit View Help
root@kali: ~/thm/skynet
root@kali: ~/thm/skynet
root@kali: ~/thm/skynet
(root㉿kali)-[~/thm/skynet]
cat log1.txt | wc -l
31
(root㉿kali)-[~/thm/skynet]
#


File Actions Edit View Help
root@kali: ~/thm/skynet
(root㉿kali)-[~/thm/skynet]
cat log1.txt
cyborg007haloterminator
terminator22596
terminator219
terminator20
terminator1989
terminator1988
terminator168
terminator16
terminator143
terminator13
terminator123!@#
terminator1056
terminator101
terminator10
terminator02
terminator00
roboterminator
pongterminator
manasturcaluterminator
exterminator95
exterminator200
dterminator
djxterminator
dexterminator
determinator
cyborg007haloterminator
avsterminator
alonsoterminator
Walterminator
79terminator6

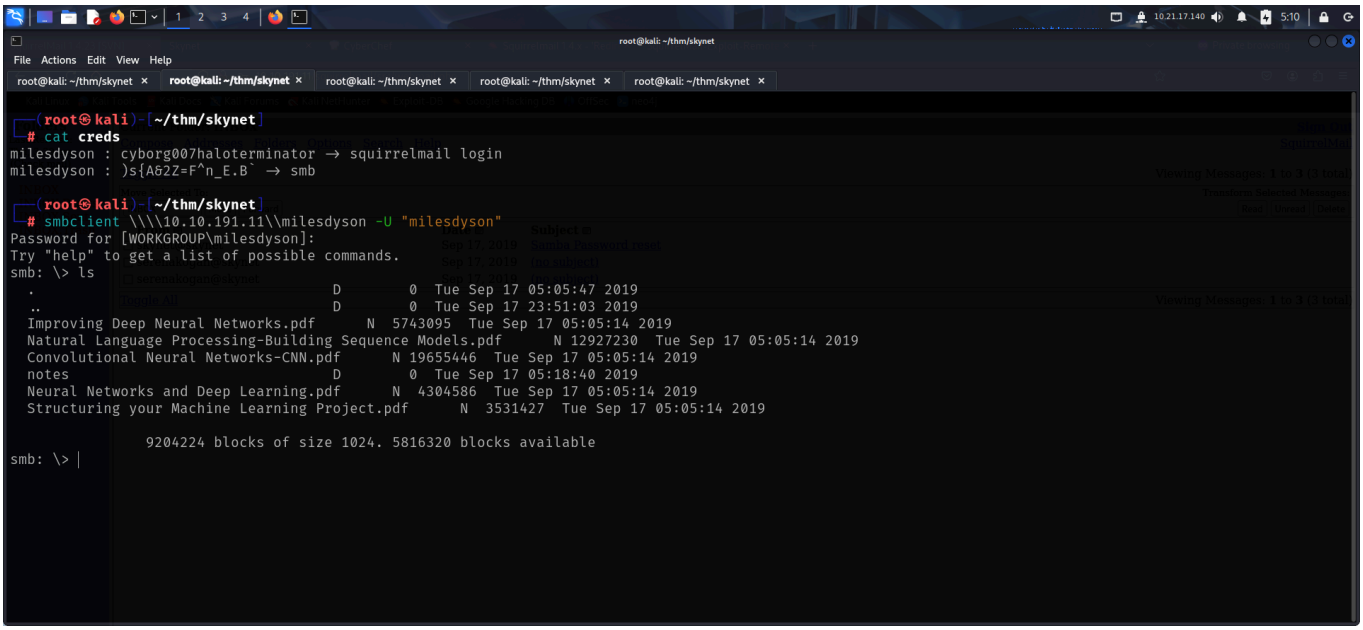


The email from *skynet* regarding SAMBA password reset seemed interesting so I opened it and found my SAMBA password.

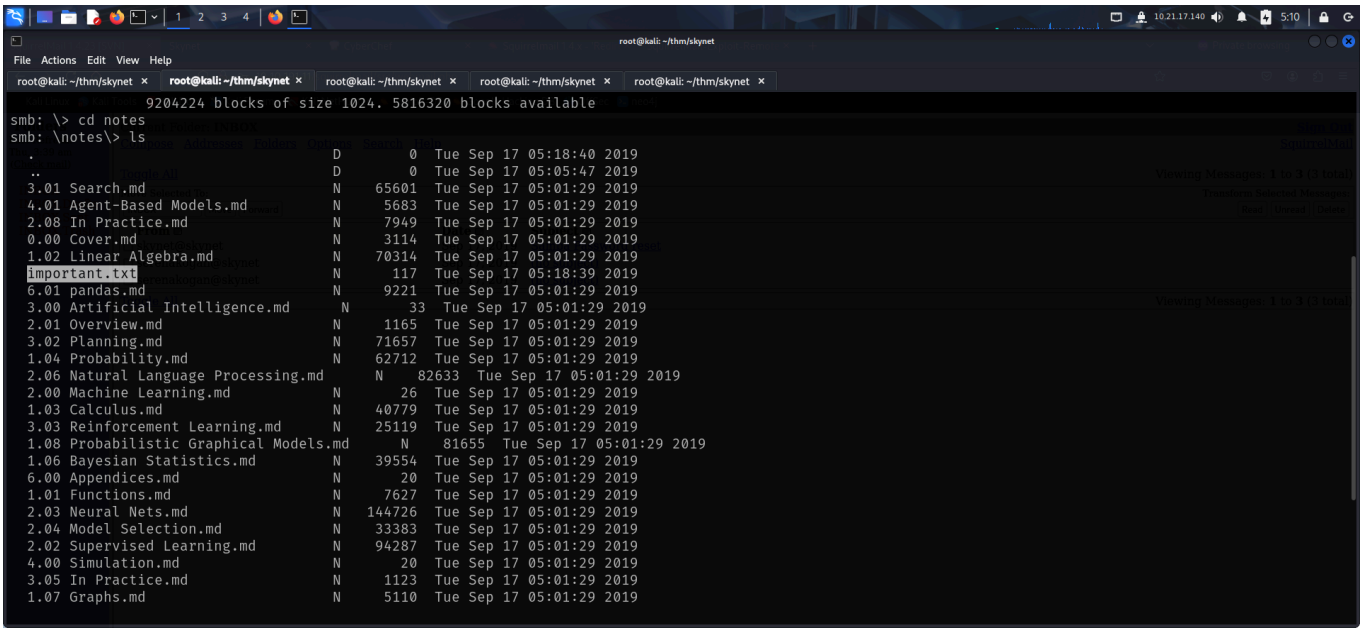


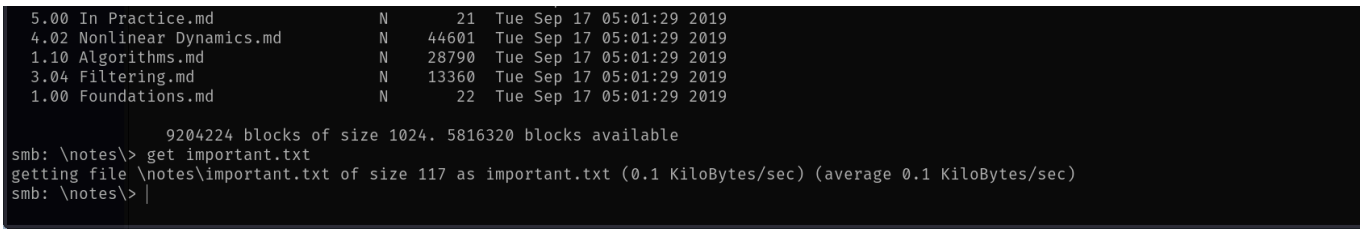


I then accessed my share using these credentials and found a *notes* directory.

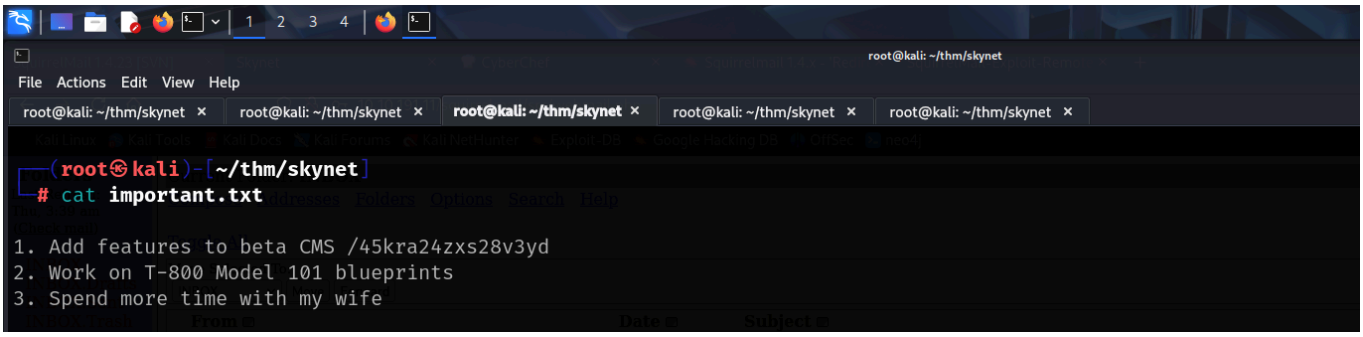


The *notes* directory contained a file called *important.txt* so I downloaded it.

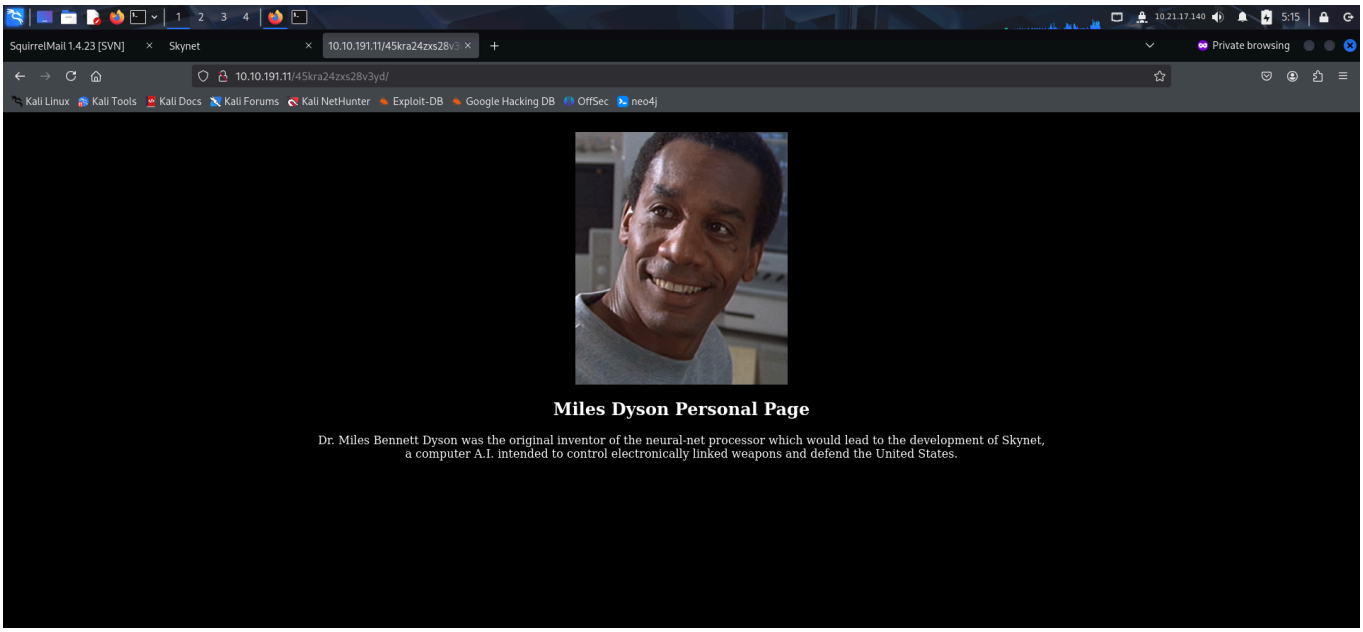




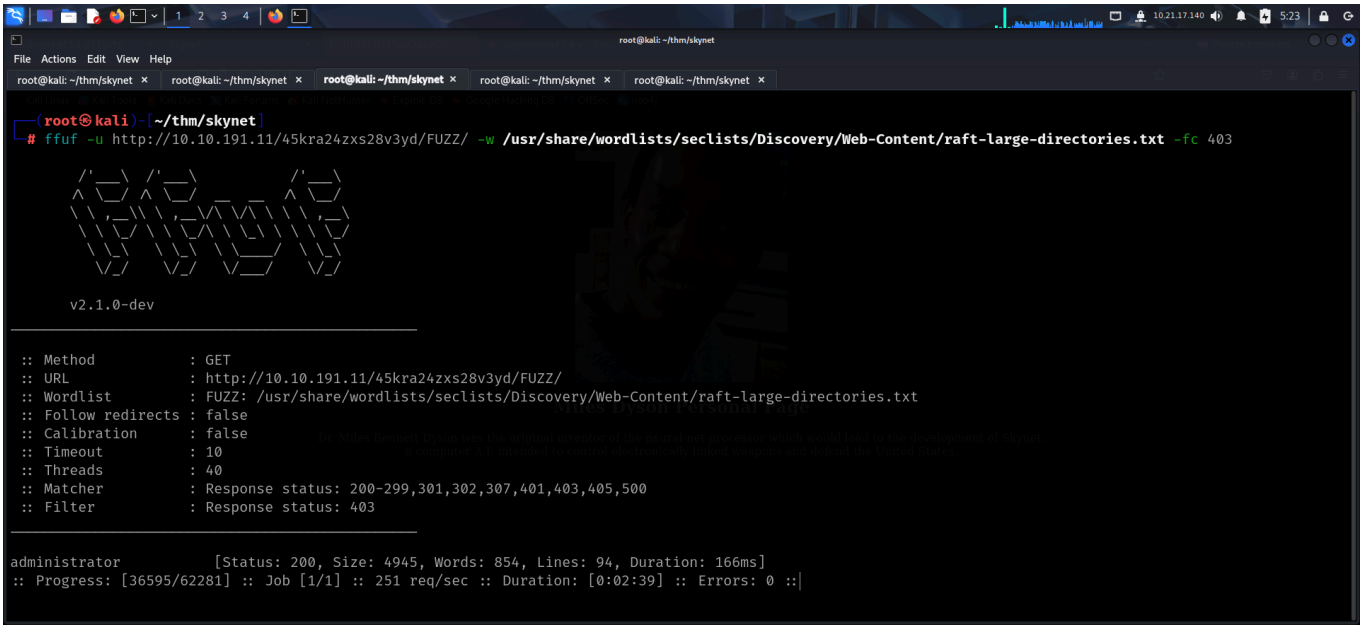
The text file revealed a new endpoint.



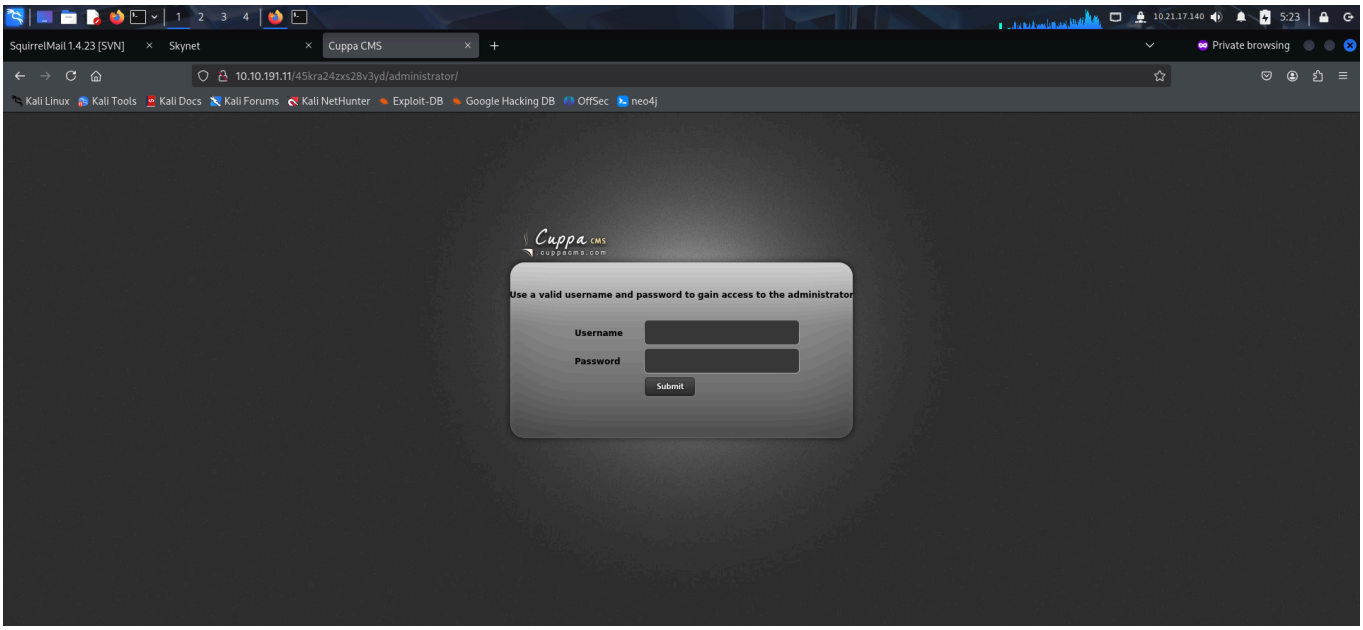
I accessed the endpoint but did not find anything useful at first.



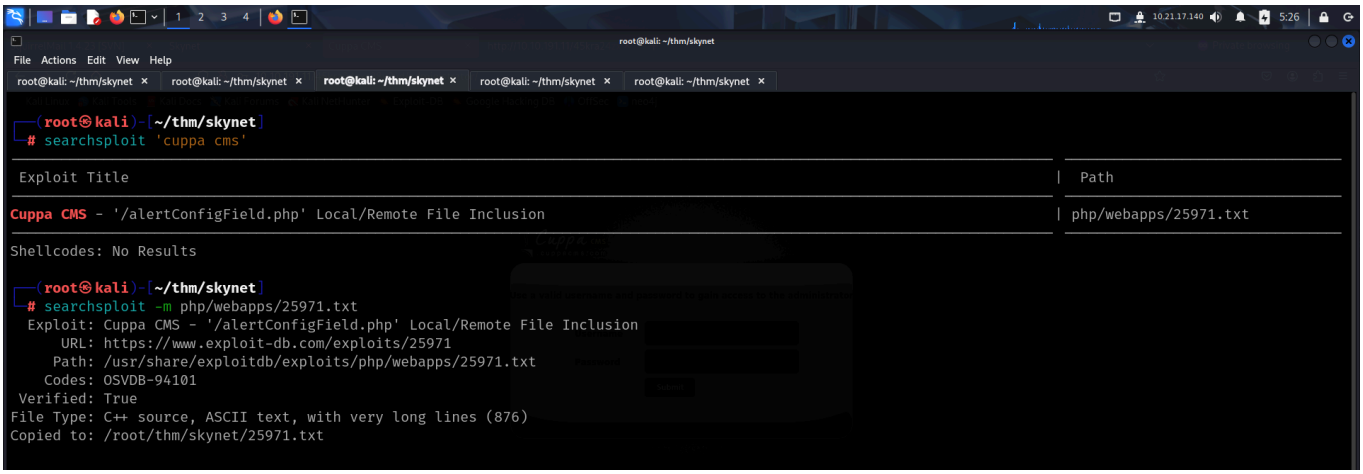
I then fuzzed for directories and found the *administrator* endpoint.



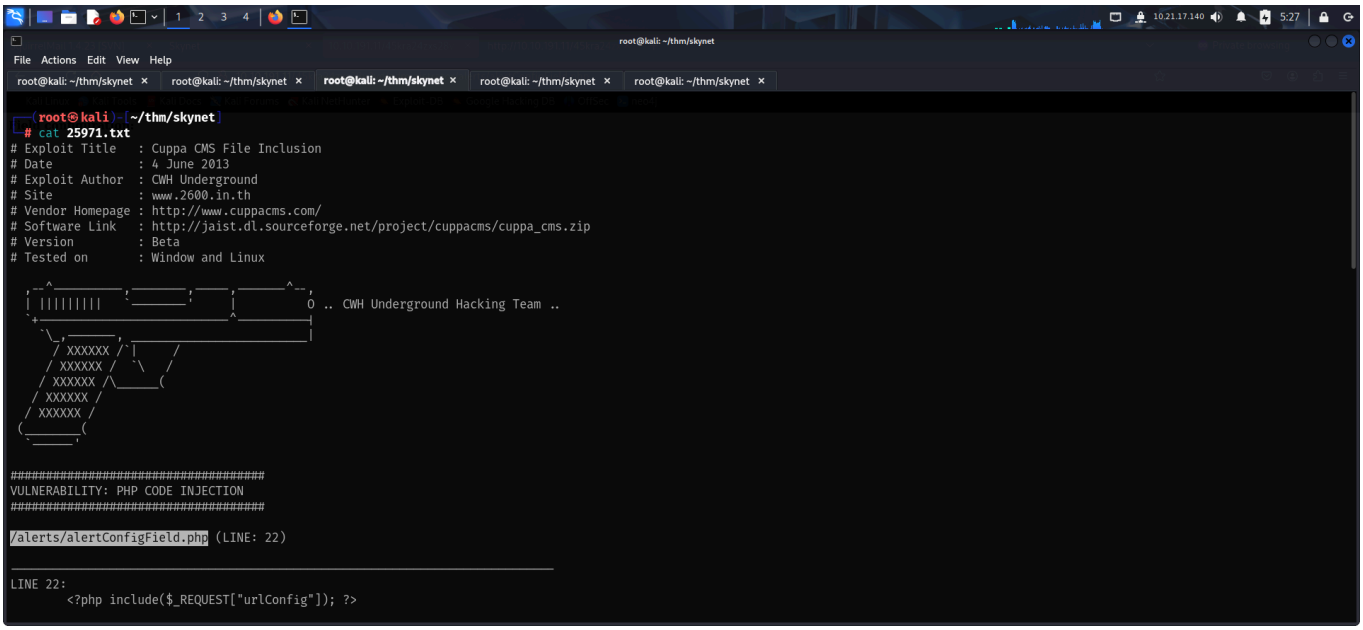
I accessed it and found a *Cuppa* CMS login panel.

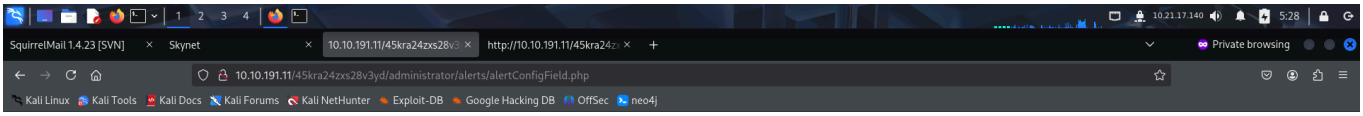


I searched for exploits related to it and found that it was vulnerable to file inclusion.



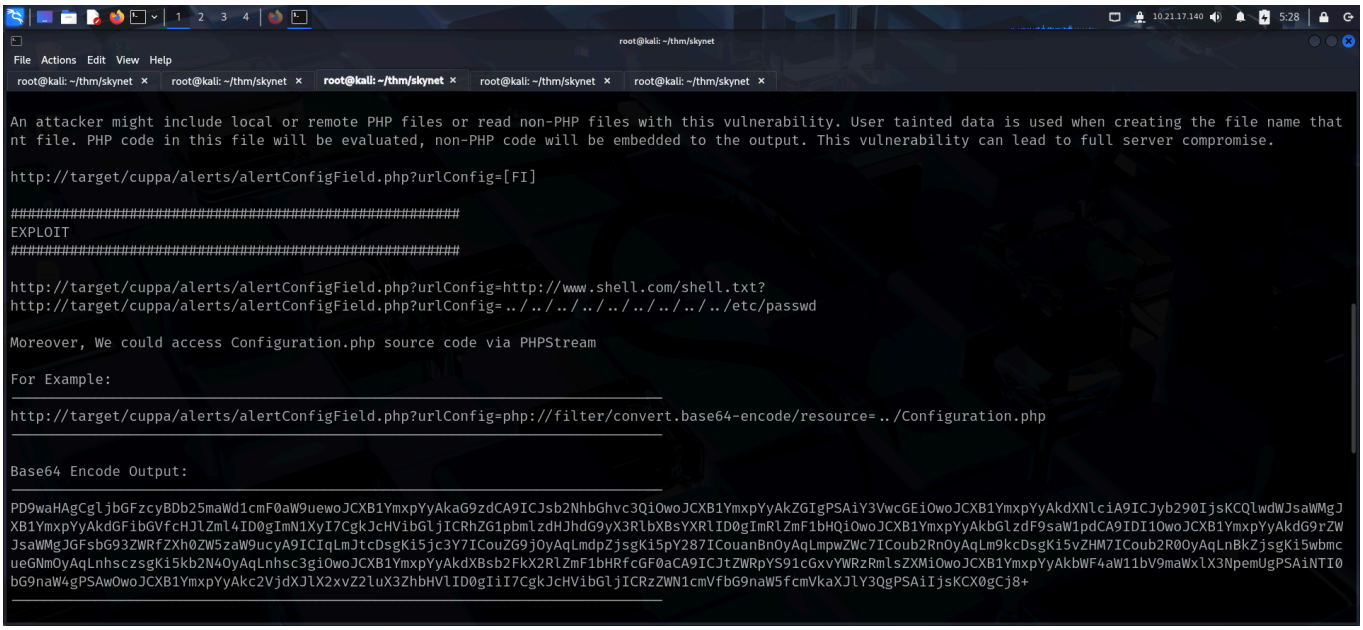
I read the file and looked for the endpoint that was vulnerable.



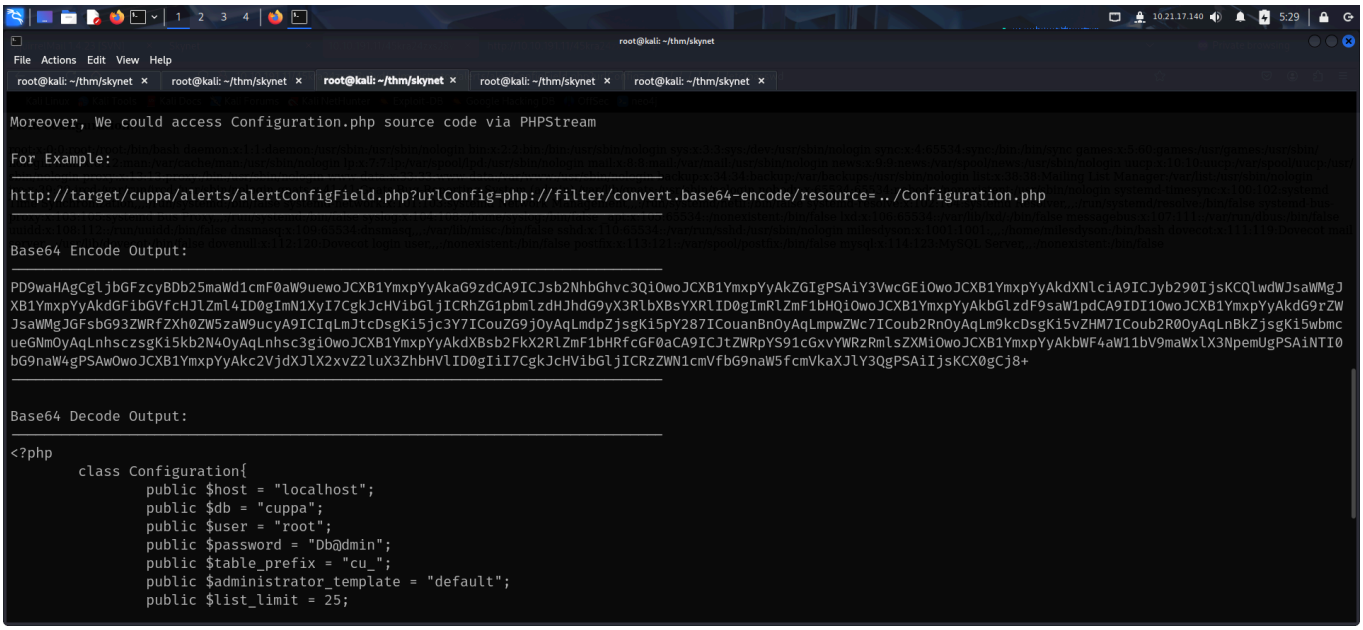


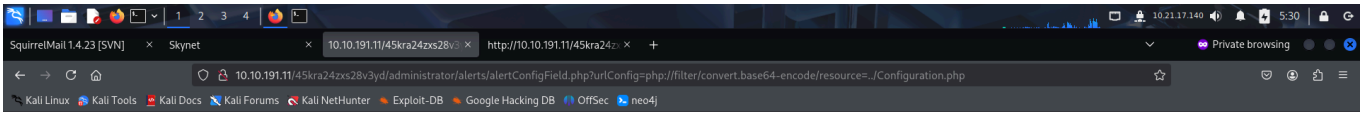
Field configuration:

After verifying that the endpoint existed, I tried the exploit and was successfully able to read the `/etc/passwd` file.



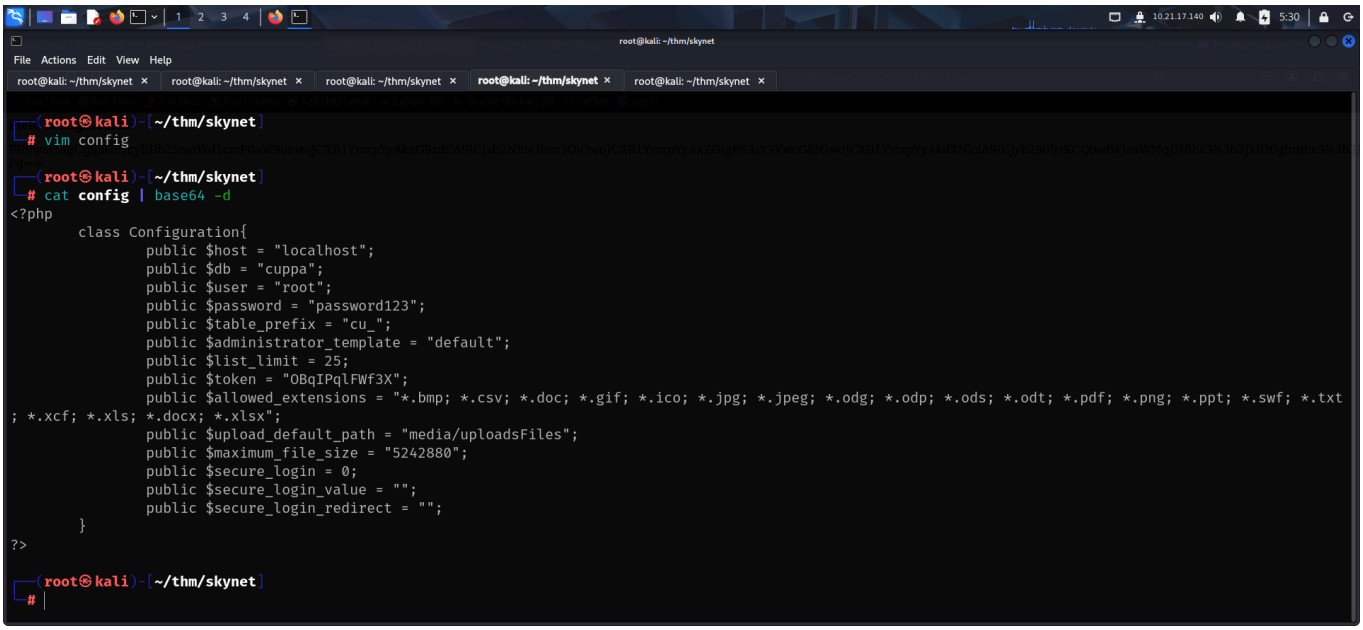
I also tried reading the configuration file and found the credentials for the database.





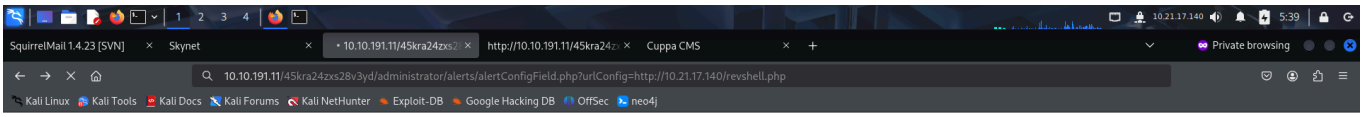
Field configuration:

PD9waHAgCgljbGFzcyBDb25maWd1cmF0aW9uewoJCXB1YmxpYyAkaG9zdCA9ICJsb2NhbGhvc3QiOwoJCXB1YmxpYyAkZGIgPSAiY3VwcGEiOwoJCXB1YmxpYyAkdXNlciA9ICJyb290IjsKCQlwdWJsaWMgJHBhc3N3b3JkID0gIkRiQGFkbWluIjsKCQlwdWJsaWMgJHRhYmxlX3ByZWZpeCA9ICJjdV8iOwoJCXB1YmxpYyAkYWRtaW5pc3RyYXRvcl90ZW1wbGF0ZSA9ICJkZWZhdWx0IjsKCQlwdWJsaWMgJGxpc3RfbGltaXQgPSAyNTsKCX0KPz4=



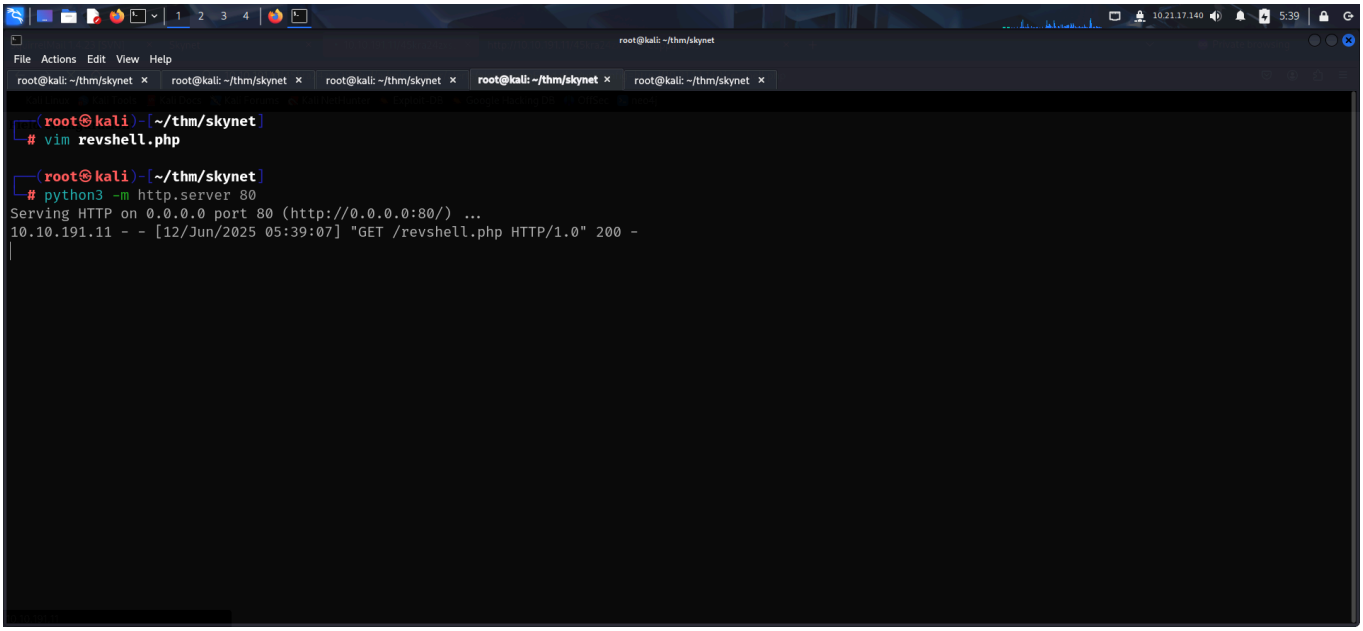
I then created a **php** reverse shell and hosted it on an **http** server locally.

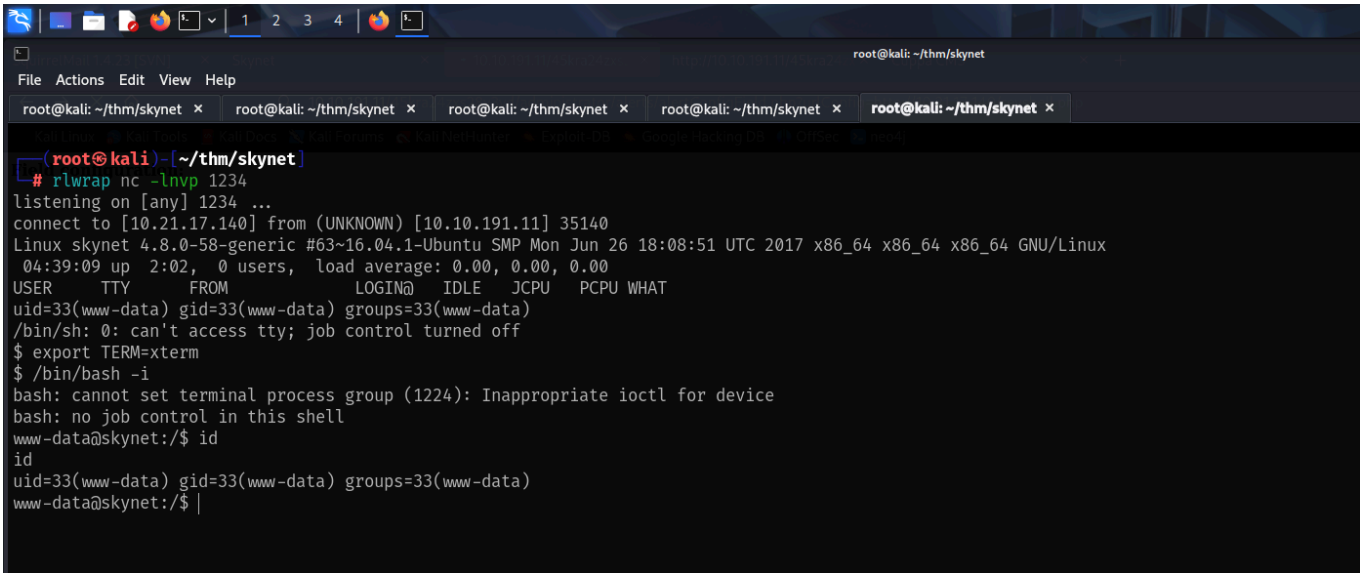
I exploited the file inclusion vulnerability to include the **php** payload to get a reverse shell on my **netcat** listener.



Field configuration:

```
10.10.191.11
```





I then captured the user flag from *mikedyson*'s home directory.

```
www-data@skynet:/$ cd home
cd home
www-data@skynet:/home$ ls
ls
milesdyson
www-data@skynet:/home$ cd milesdyson
cd milesdyson
www-data@skynet:/home/milesdyson$ ls -la
ls -la
total 36
drwxr-xr-x 5 milesdyson milesdyson 4096 Sep 17  2019 .
drwxr-xr-x 3 root       root       4096 Sep 17  2019 ..
lrwxrwxrwx 1 root       root          9 Sep 17  2019 .bash_history → /dev/null
-rw-r--r-- 1 milesdyson milesdyson  220 Sep 17  2019 .bash_logout
-rw-r--r-- 1 milesdyson milesdyson 3771 Sep 17  2019 .bashrc
-rw-r--r-- 1 milesdyson milesdyson  655 Sep 17  2019 .profile
drwxr-xr-x 2 root       root       4096 Sep 17  2019 backups
drwx------ 3 milesdyson milesdyson 4096 Sep 17  2019 mail
drwxr-xr-x 3 milesdyson milesdyson 4096 Sep 17  2019 share
-rw-r--r-- 1 milesdyson milesdyson   33 Sep 17  2019 user.txt
```

```
www-data@skynet:/home/milesdyson$ cat user.txt
cat user.txt
7ce[redacted]
www-data@skynet:/home/milesdyson$
```

PRIVILEGE ESCALATION

I examined my home directory and found an interesting directory called *backups*. Inside the directory, there was a **tar** archive and a bash script.

```
www-data@skynet:/home/milesdyson$ ls -la
ls -la
total 36
drwxr-xr-x 5 milesdyson milesdyson 4096 Sep 17  2019 .
drwxr-xr-x 3 root       root       4096 Sep 17  2019 ..
lrwxrwxrwx 1 root       root          9 Sep 17  2019 .bash_history → /dev/null
-rw-r--r-- 1 milesdyson milesdyson  220 Sep 17  2019 .bash_logout
-rw-r--r-- 1 milesdyson milesdyson 3771 Sep 17  2019 .bashrc
-rw-r--r-- 1 milesdyson milesdyson  655 Sep 17  2019 .profile
drwxr-xr-x 2 root       root       4096 Sep 17  2019 backups
drwx------ 3 milesdyson milesdyson 4096 Sep 17  2019 mail
drwxr-xr-x 3 milesdyson milesdyson 4096 Sep 17  2019 share
-rw-r--r-- 1 milesdyson milesdyson   33 Sep 17  2019 user.txt
www-data@skynet:/home/milesdyson$ cd backups
cd backups
www-data@skynet:/home/milesdyson/backups$ ls
ls
backup.sh  backup.tgz
www-data@skynet:/home/milesdyson/backups$ ls -la
ls -la
total 4584
drwxr-xr-x 2 root       root         4096 Sep 17  2019 .
drwxr-xr-x 5 milesdyson milesdyson    4096 Sep 17  2019 ..
-rwxr-xr-x 1 root       root           74 Sep 17  2019 backup.sh
-rw-r--r-- 1 root       root       4679680 Jun 12 04:41 backup.tgz
```

I viewed the script and found it was used to create a backup of contents inside `/var/www/html` and save it as an archive inside the *backup.tgz* file.

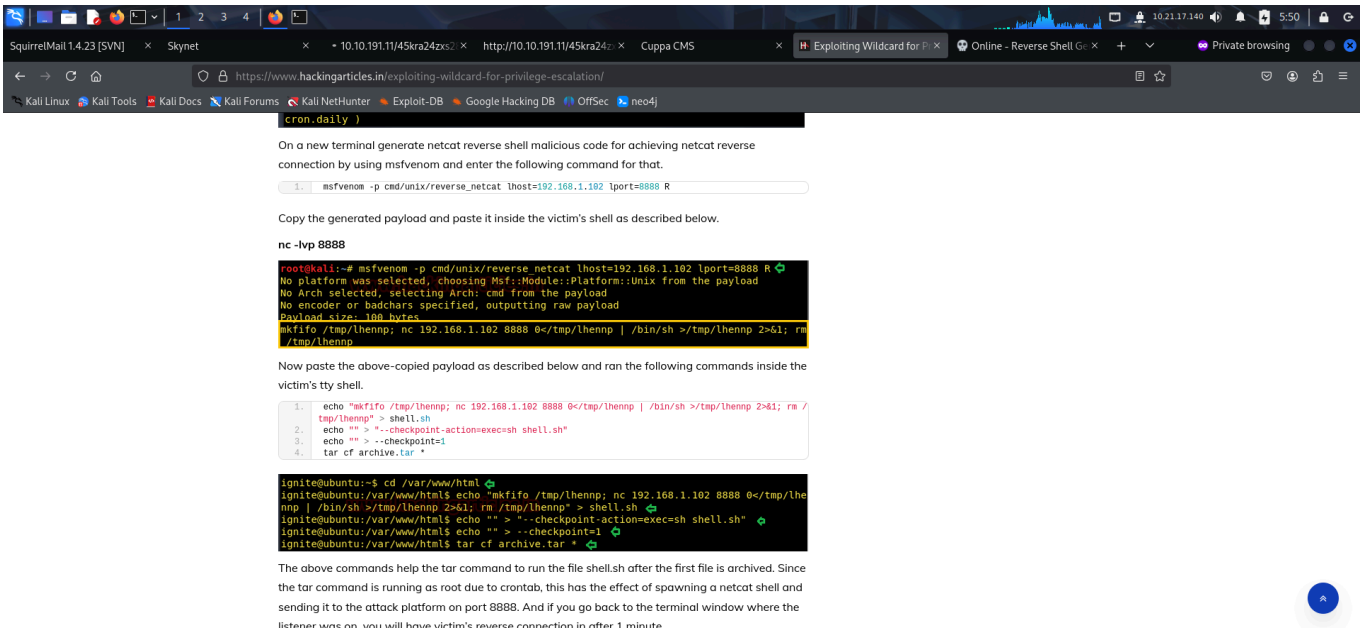
```
www-data@skynet:/home/milesdyson/backups$ cat backup.sh
cat backup.sh
#!/bin/bash
cd /var/www/html
tar cf /home/milesdyson/backups/backup.tgz *
www-data@skynet:/home/milesdyson/backups$
```

I could use the **wildcard** to escalate my privilege if it could be executed as root. I checked the `/etc/crontab` file and found that the *root* user executed the **bash** script to create the backup.

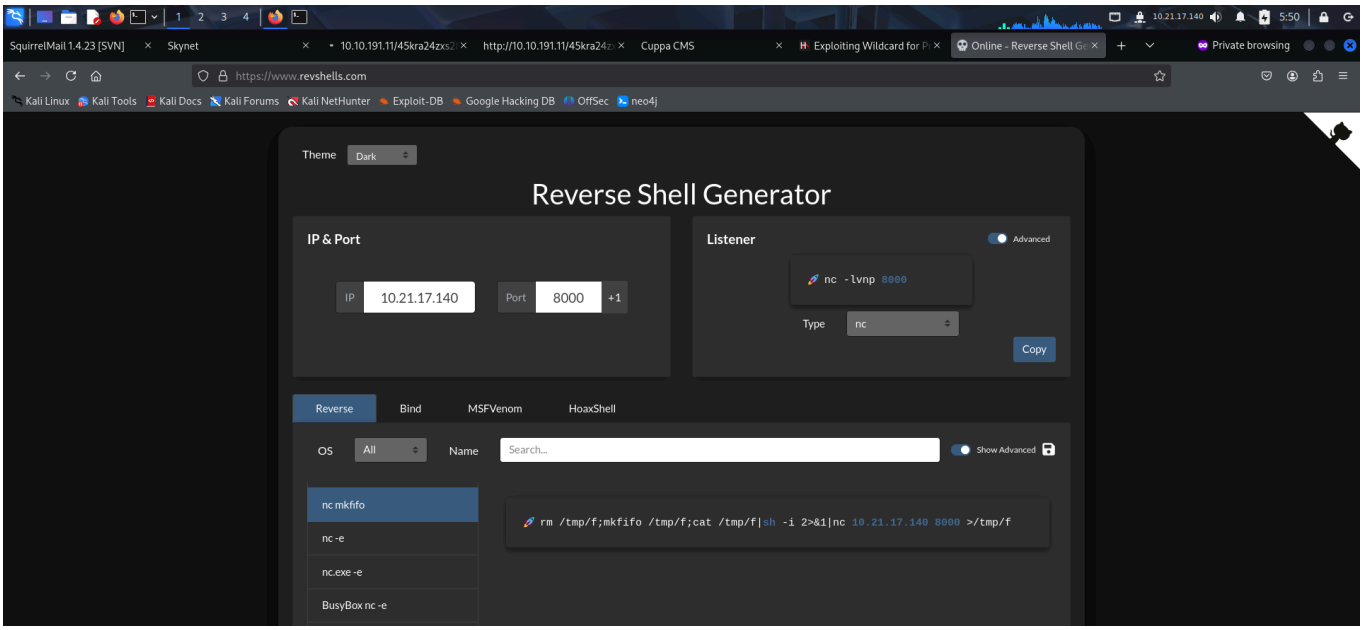
```
www-data@skynet:/home/milesdyson/backups$ cat /etc/crontab
cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
*/1 *   * * *   root    /home/milesdyson/backups/backup.sh
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
www-data@skynet:/home/milesdyson/backups$
```

I referred the the following article to escalate my privilege by exploiting wildcard:

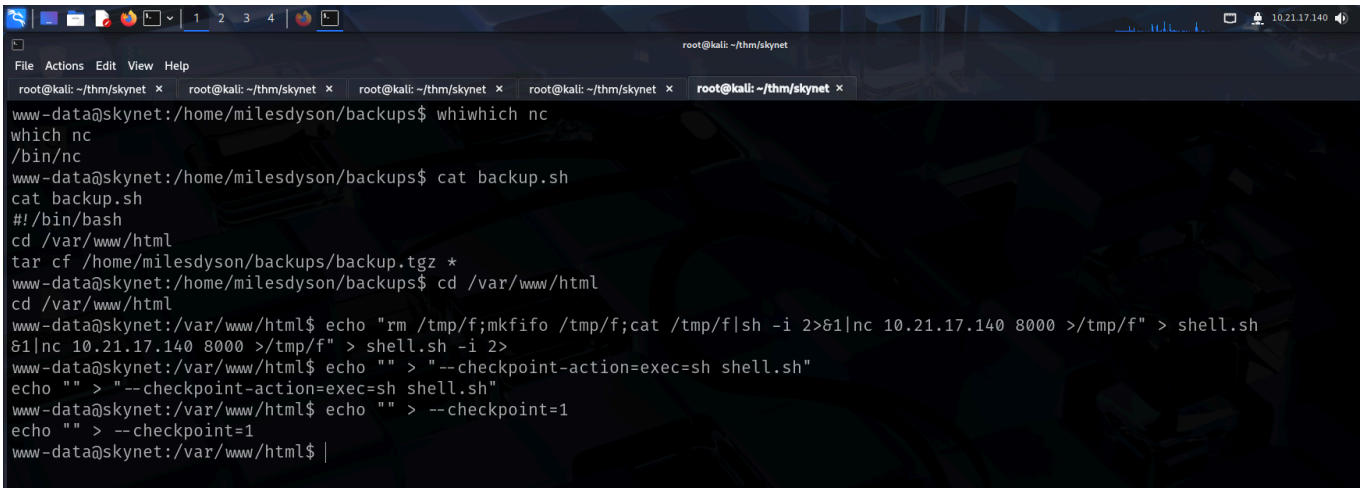
- https://www.hackingarticles.in/exploiting-wildcard-for-privilege-escalation/

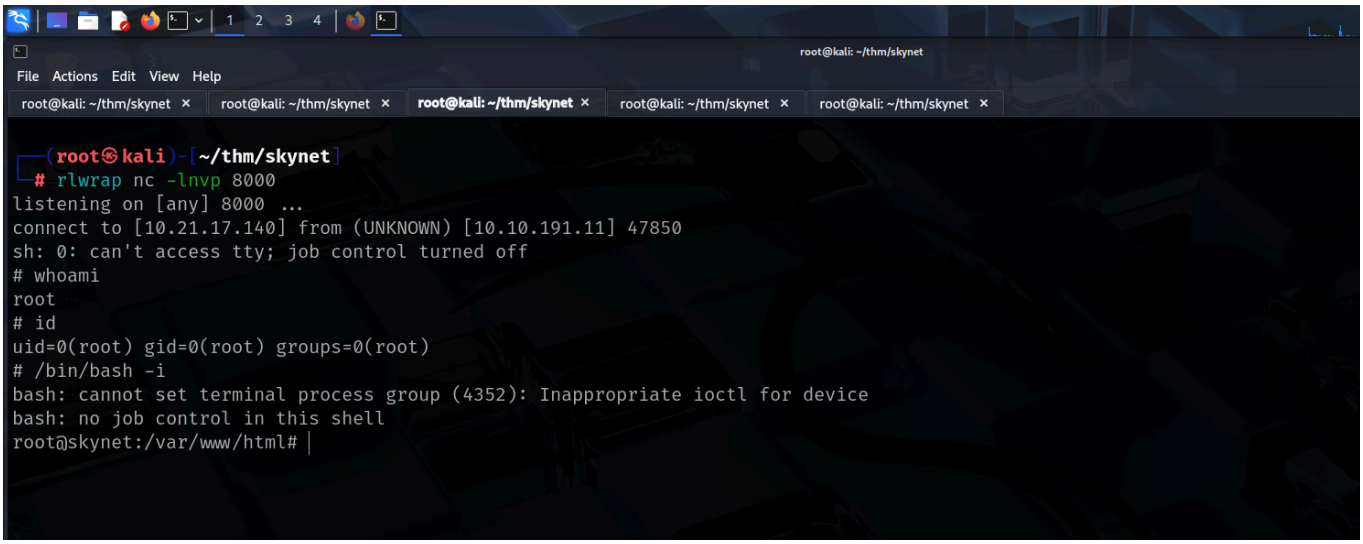


I copied a **netcat** reverse shell payload from **revhsells.com**.

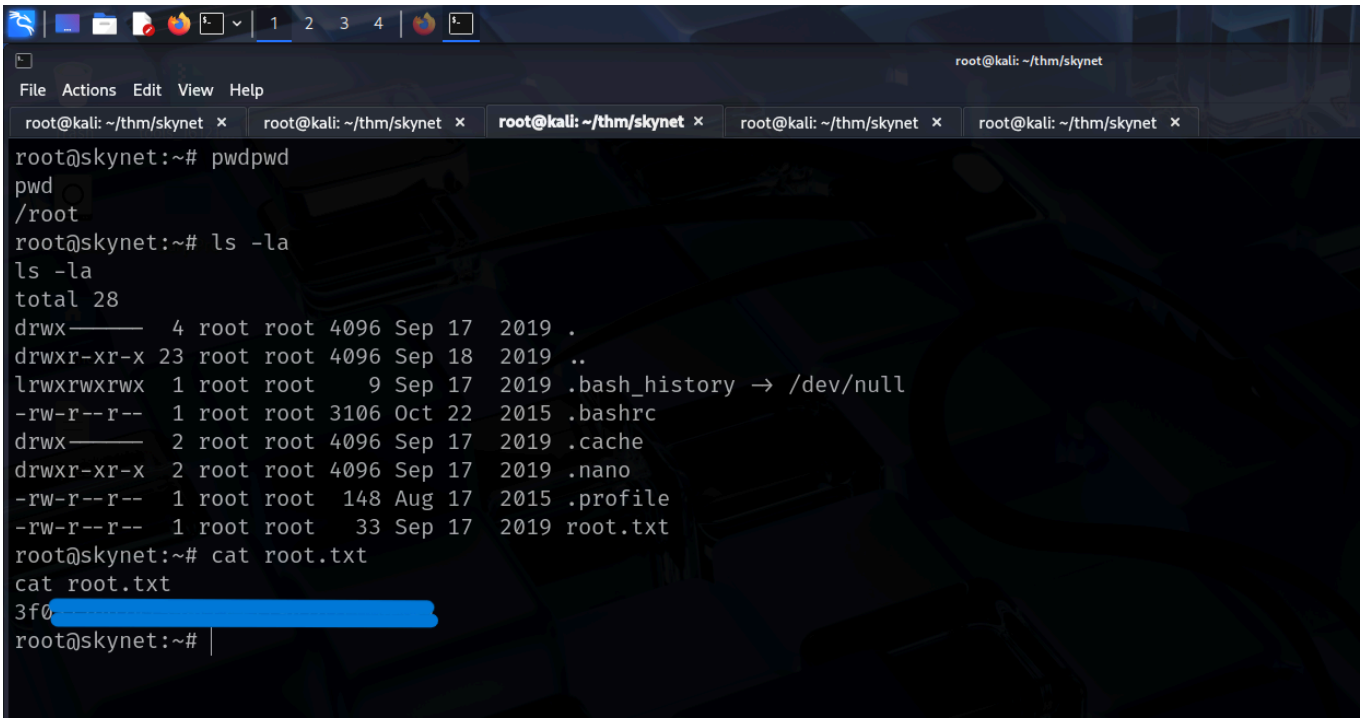


Finally, I followed the methods from the article and used the payload to get a reverse shell as *root* user on another **netcat** listener.





After gaining root access, I captured the final flag from `/root` directory.



That's it from my side! Until next time :)