Babatunde Ojo

Masterschool Cybersecurity Capture the Flag 4

https://tryhackme.com/room/masterschoolctf5yfr

July 8, 2023

Instructor: Mr James Key

Table of Contents

1. **Introduction**....4

2. **Linux Basics: User and File Management**....4

1.1 User Creation....4

1.2 User switch....4

1.3 Folder and File Creation....6

1.4 Switch Back to Original User....6

3. **File System Flags**....7

3.1 First flag {h4ck3r5_r_us}....7

3.2 Second flag {H1d3_1n_pl41n_s1gh7}....7

3.3 Third flag {Y0u_G0T_1t}....7

3.4 Fourth flag {St0ry_Fl4g}....8

4. **Webpage Flags**....9

4.1 First flag {STUDENT_CTF_Web}....9

4.2 Second flag {Another_Web_Flag}....9

4.3 Third flag {Another_Web_Flag}....9

4.4 Fourth flag {Another_Web_Flag}....9

4.5 Fifth flag {Another_Web_Flag}....9

4.6 Sixth flag {Another_Web_Flag}....9

5. **Hidden Flags Challenge**....10

5.1 First flag {ftp_server_4_lyfe}....10

5.2 Second Flag {CTF_Time}....11

5.3 Third Flag {F1nd_Fl4g_Fun}....12

5.4 Fourth Flag {S3cr3t_Fl4g}....12

5.5 Fifth Flag {Robots_Flag}....12

5.6 Sixth Flag {Fl4g_fl4g_fl4g}....12

6. **Hash Cracking**....13

6.1 First Flag {C0d3_0b5cur3r_Flag}....14

6.2 Second Flag {C0d3_5l4y3r_Flag}....14

6.3 Third Flag {H4ck3r_Flag}....15

6.4 Fourth Flag {L0ck_Flag}..15
6.5 Fifth Flag {S3cur1ty_Flag}..15
7. **NMAP Scan Report**..17
7.1 Host information..17
7.2 Open Ports and Services..17
7.3 Potential Vulnerabilities and Solution –...18

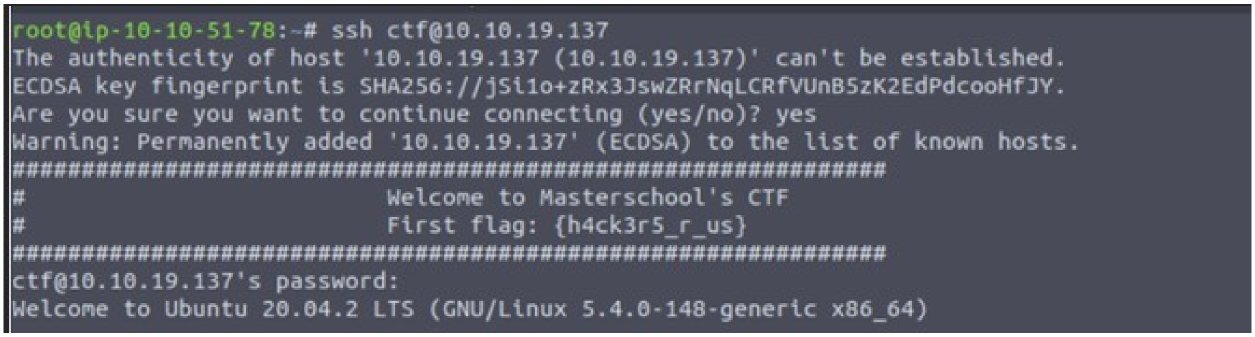
1. Introduction

This report provides an overview and analysis of the Cybersecurity Capture The Flag (CTF) project. The project focuses on testing and enhancing cybersecurity skills through a series of challenges and scenarios. Participants are tasked with identifying vulnerabilities, exploiting systems, and uncovering hidden flags within a controlled environment.

The objective of the CTF project is to simulate real-world cyber threats and provide hands-on experience in securing systems and networks.

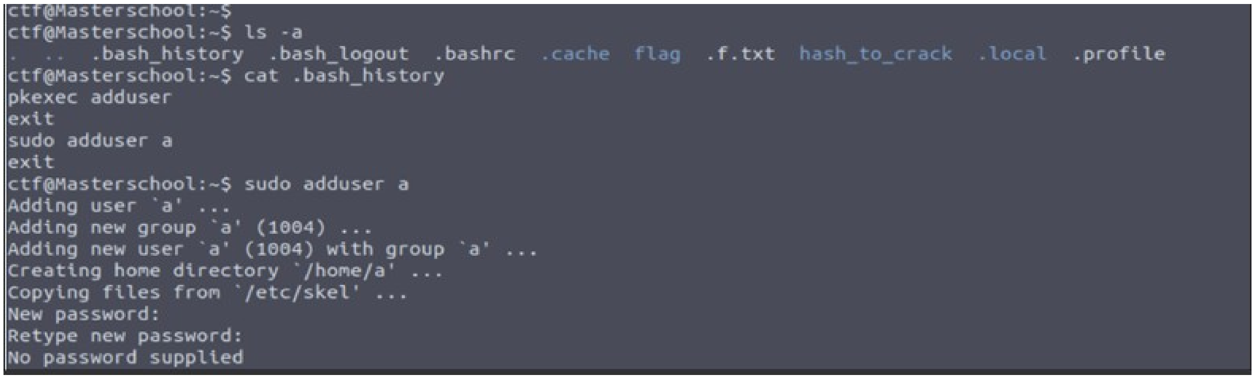
2. Linux Basics: User and File Management

2.1 User Creation: In my attack machine, I used SSH to get in the Masterschool CTF machine. The CTF login and username were given, command on attack box to get in ctf machine is ***ssh ctf@<machine_IP>***



```
root@ip-10-10-51-78:~# ssh ctf@10.10.19.137
The authenticity of host '10.10.19.137 (10.10.19.137)' can't be established.
ECDSA key fingerprint is SHA256://jSi1o+zRx3JswZRrNqLCRfVUnB5zK2EdPdcooHfJY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.19.137' (ECDSA) to the list of known hosts.
#################################################################
#                           Welcome to Masterschool's CTF
#                           First flag: {h4ck3r5_r_us}
#################################################################
ctf@10.10.19.137's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-148-generic x86_64)
```

User ctf is not a root, therefore he is unable to create the user. Checked *.bash_history* and found user executed user add. The action is still recorded under sudo, so we can add this user with *sudo adduser a*. Gave the new user password.

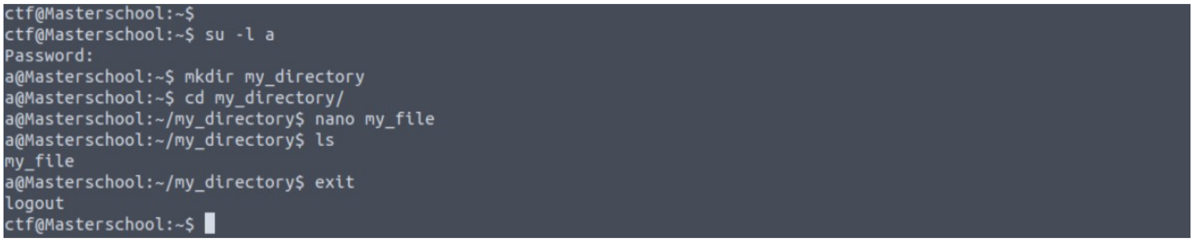


```
ctf@Masterschool:~$
ctf@Masterschool:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .cache  flag  .f.txt  hash_to_crack  .local  .profile
ctf@Masterschool:~$ cat .bash_history
pkexec adduser
exit
sudo adduser a
exit
ctf@Masterschool:~$ sudo adduser a
Adding user `a' ...
Adding new group `a' (1004) ...
Adding new user `a' (1004) with group `a' ...
Creating home directory `/home/a' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
No password supplied
```

2.2 User switch: I used command *su-l <username>* (Username in this case is “a”) to login with shell so we can get full access to the environment.

2.3 **Folder and File Creation:** I used *mkdir <directory_name>* to make directory, *cd <directory name>* to move into the directory just created, and *nano <filename>* to make file inside the directory. I wrote in nano mode "*Hello from a*" and used *CTRL+S* to save the nano and *CTRL+X* to exit nano mode

2.4 **Switch Back to Original User:** I used command *exit* to switch back to user ctf (original user)

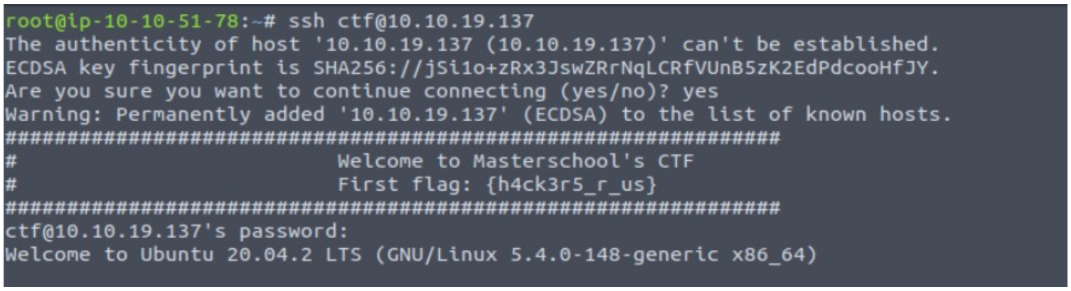


```
ctf@Masterschool:~$
ctf@Masterschool:~$ su -l a
Password:
a@Masterschool:~$ mkdir my_directory
a@Masterschool:~$ cd my_directory/
a@Masterschool:~/my_directory$ nano my_file
a@Masterschool:~/my_directory$ ls
my_file
a@Masterschool:~/my_directory$ exit
logout
ctf@Masterschool:~$ 
```

3 File System Flags

3.2 First flag: {h4ck3r5_r_us}

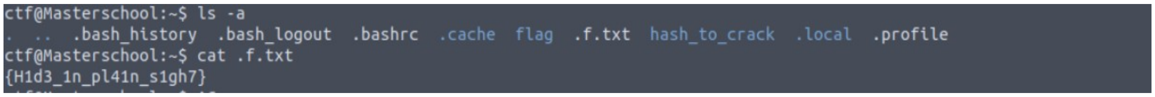
Found flag when you first ssh into ctf account on machine.



```
root@ip-10-10-51-78:~# ssh ctf@10.10.19.137
The authenticity of host '10.10.19.137 (10.10.19.137)' can't be established.
ECDSA key fingerprint is SHA256://jSi1o+zRx3JswZRrNqLCRfVUnB5zK2EdPdcooHfJY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.19.137' (ECDSA) to the list of known hosts.
###############################################################
#                          Welcome to Masterschool's CTF
#                          First flag: {h4ck3r5_r_us}
###############################################################
ctf@10.10.19.137's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-148-generic x86_64)
```

3.3 Second flag: {H1d3_1n_pl41n_s1gh7}

Found flag inside hidden file *.f.txt* on ctf account. I used *ls -a* command to show hidden files.

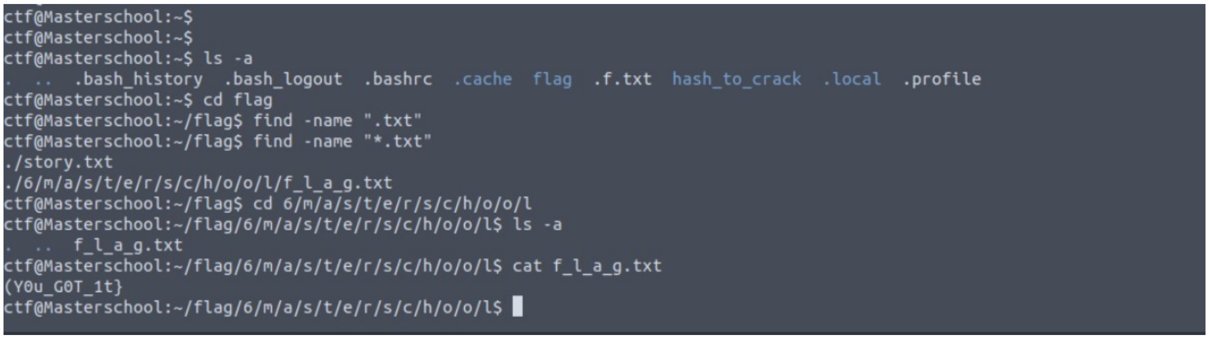


```
ctf@Masterschool:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .cache  flag  .f.txt  hash_to_crack  .local  .profile
ctf@Masterschool:~$ cat .f.txt
{H1d3_1n_pl41n_s1gh7}
```

3.4 Third flag: {Y0u_G0T_1t}

Used deductive reasoning. The flags are usually hidden in a *.txt* file. I used *cd flag* to get in flag directory. Then ran find command on any file that is a text file: find -name ".*.txt" I got two output. I found a directory inside multiple directories that leads to *f_l_a_g.txt*

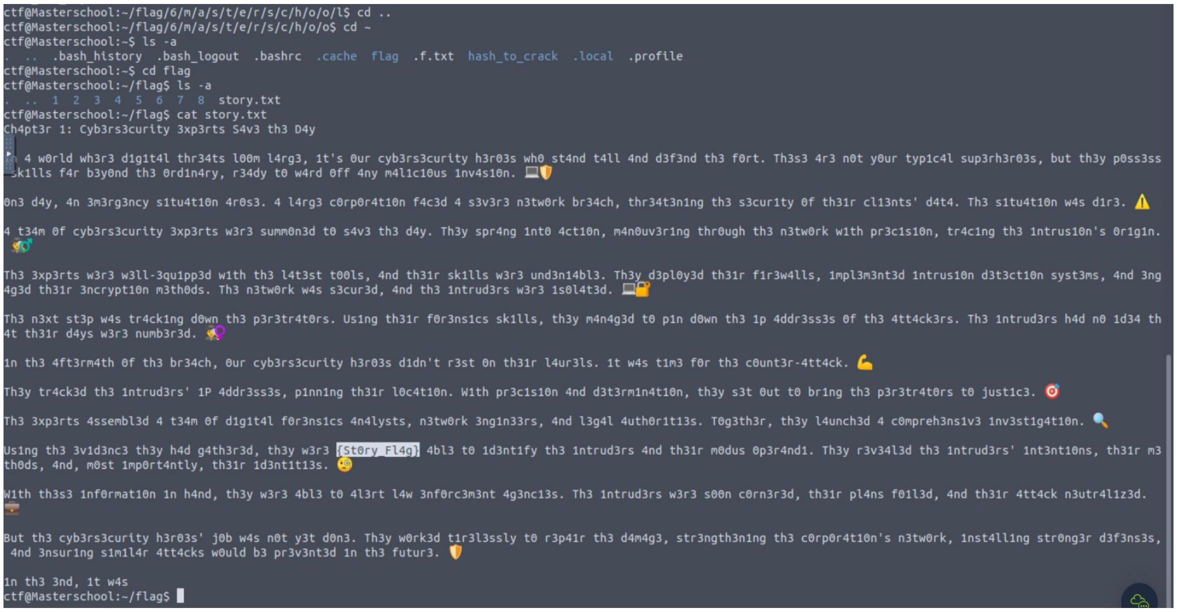
I went into the multiple directories using *cd 6/m/a/s/t/e/r/s/c/h/o/o/l*



```
ctf@Masterschool:~$
ctf@Masterschool:~$
ctf@Masterschool:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .cache  flag  .f.txt  hash_to_crack  .local  .profile
ctf@Masterschool:~$ cd flag
ctf@Masterschool:~/flag$ find -name ".txt"
ctf@Masterschool:~/flag$ find -name "*.txt"
./story.txt
./6/m/a/s/t/e/r/s/c/h/o/o/l/f_l_a_g.txt
ctf@Masterschool:~/flag$ cd 6/m/a/s/t/e/r/s/c/h/o/o/l
ctf@Masterschool:~/flag/6/m/a/s/t/e/r/s/c/h/o/o/l$ ls -a
.  ..  f_l_a_g.txt
ctf@Masterschool:~/flag/6/m/a/s/t/e/r/s/c/h/o/o/l$ cat f_l_a_g.txt
{Y0u_G0T_1t}
ctf@Masterschool:~/flag/6/m/a/s/t/e/r/s/c/h/o/o/l$ █
```

3.5 Fourth flag: {St0ry_Fl4g}

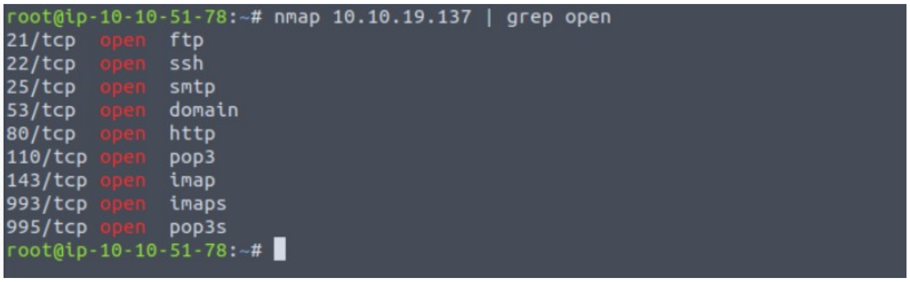
Remember that I got two text file when we did the find. Now I go inside the *./story/txt* I read the story, and inside the story, the flag was hidden in it.



```
ctf@Masterschool:~/flag/6/m/a/s/t/e/r/s/c/h/o/o/l$ cd ..
ctf@Masterschool:~/flag/6/m/a/s/t/e/r/s/c/h/o/o$ cd ~
ctf@Masterschool:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .cache  flag  .f.txt  hash_to_crack  .local  .profile
ctf@Masterschool:~$ cd flag
ctf@Masterschool:~/flag$ ls -a
.  ..  1  2  3  4  5  6  7  8  story.txt
ctf@Masterschool:~/flag$ cat story.txt
Ch4pt3r 1: Cyb3rs3curity 3xp3rts S4v3 th3 D4y

1n 4 w0rld wh3r3 d1g1t4l thr34ts l00m l4rg3, 1t's 0ur cyb3rs3curity h3r03s wh0 st4nd t4ll 4nd d3f3nd th3 f0rt. Th3s3 4r3 n0t y0ur typ1c4l sup3rh3r03s, but th3y p0ss3ss
sk1lls f4r b3y0nd th3 0rd1n4ry, r34dy t0 w4rd 0ff 4ny m4l1c10us 1nv4s10n. 💻🛡️

0n3 d4y, 4n 3m3rg3ncy s1tu4t10n 4r0s3. 4 l4rg3 c0rp0r4t10n f4c3d 4 s3v3r3 n3tw0rk br34ch, thr34t3n1ng th3 s3cur1ty 0f th31r cl13nts' d4t4. Th3 s1tu4t10n w4s d1r3. ⚠️

4 t34m 0f cyb3rs3curity 3xp3rts w3r3 summ0n3d t0 s4v3 th3 d4y. Th3y spr4ng 1nt0 4ct10n, m4n0uv3r1ng thr0ugh th3 n3tw0rk w1th pr3c1s10n, tr4c1ng th3 1ntrus10n's 0r1g1n.
 🕵️‍♂️

Th3 3xp3rts w3r3 w3ll-3qu1pp3d w1th th3 l4t3st t00ls, 4nd th31r sk1lls w3r3 und3n14bl3. Th3y d3pl0y3d th31r f1r3w4lls, 1mpl3m3nt3d 1ntrus10n d3t3ct10n syst3ms, 4nd 3ng
4g3d th31r 3ncrypt10n m3th0ds. Th3 n3tw0rk w4s s3cur3d, 4nd th3 1ntrud3rs w3r3 1s0l4t3d. 💻🔐

Th3 n3xt st3p w4s tr4ck1ng d0wn th3 p3r3tr4t0rs. Us1ng th31r f0r3ns1cs sk1lls, th3y m4n4g3d t0 p1n d0wn th3 1p 4ddr3ss3s 0f th3 4tt4ck3rs. Th3 1ntrud3rs h4d n0 1d34 th
4t th31r d4ys w3r3 numb3r3d. 🕵️‍♀️

1n th3 4ft3rm4th 0f th3 br34ch, 0ur cyb3rs3curity h3r03s d1dn't r3st 0n th31r l4ur3ls. 1t w4s t1m3 f0r th3 c0unt3r-4tt4ck. 💪

Th3y tr4ck3d th3 1ntrud3rs' 1P 4ddr3ss3s, p1nn1ng th31r l0c4t10n. W1th pr3c1s10n 4nd d3t3rm1n4t10n, th3y s3t 0ut t0 br1ng th3 p3r3tr4t0rs t0 just1c3. 🎯

Th3 3xp3rts 4ssembl3d 4 t34m 0f d1g1t4l f0r3ns1cs 4n4lysts, n3tw0rk 3ng1n33rs, 4nd l3g4l 4uth0r1t13s. T0g3th3r, th3y l4unch3d 4 c0mpreh3ns1v3 1nv3st1g4t10n. 🔍

Us1ng th3 3v1d3nc3 th3y h4d g4th3r3d, th3y w3r3 {St0ry_Fl4g} 4bl3 t0 1d3nt1fy th3 1ntrud3rs 4nd th31r m0dus 0p3r4nd1. Th3y r3v34l3d th3 1ntrud3rs' 1nt3nt10ns, th31r m3
th0ds, 4nd, m0st 1mp0rt4ntly, th31r 1d3nt1t13s. 🧐

W1th th3s3 1nf0rmat10n 1n h4nd, th3y w3r3 4bl3 t0 4l3rt l4w 3nf0rc3m3nt 4g3nc13s. Th3 1ntrud3rs w3r3 s00n c0rn3r3d, th31r pl4ns f01l3d, 4nd th31r 4tt4ck n3utr4l1z3d.
💼

But th3 cyb3rs3curity h3r03s' j0b w4s n0t y3t d0n3. Th3y w0rk3d t1r3l3ssly t0 r3p41r th3 d4m4g3, str3ngth3n1ng th3 c0rp0r4t10n's n3tw0rk, 1nst4ll1ng str0ng3r d3f3ns3s,
 4nd 3nsur1ng s1m1l4r 4tt4cks w0uld b3 pr3v3nt3d 1n th3 futur3. 🛡️

1n th3 3nd, 1t w4s
ctf@Masterschool:~/flag$ █
```

4 Webpage Flags

I hopped back on my attack machine and ran an nmap scan to see which web port was open using *nmap <ctf_machineIP> | grep open*



```
root@ip-10-10-51-78:~# nmap 10.10.19.137 | grep open
21/tcp  open  ftp
22/tcp  open  ssh
25/tcp  open  smtp
53/tcp  open  domain
80/tcp  open  http
110/tcp open  pop3
143/tcp open  imap
993/tcp open  imaps
995/tcp open  pop3s
root@ip-10-10-51-78:~# 
```

Now that we see that http port 80 is open, I open a web page using *http://<ctf_machineIP>*

4.2 First flag: {STUDENT_CTF_Web}

This was the welcome page when you go on the website.

4.3 Second flag: {Another_Web_Flag}

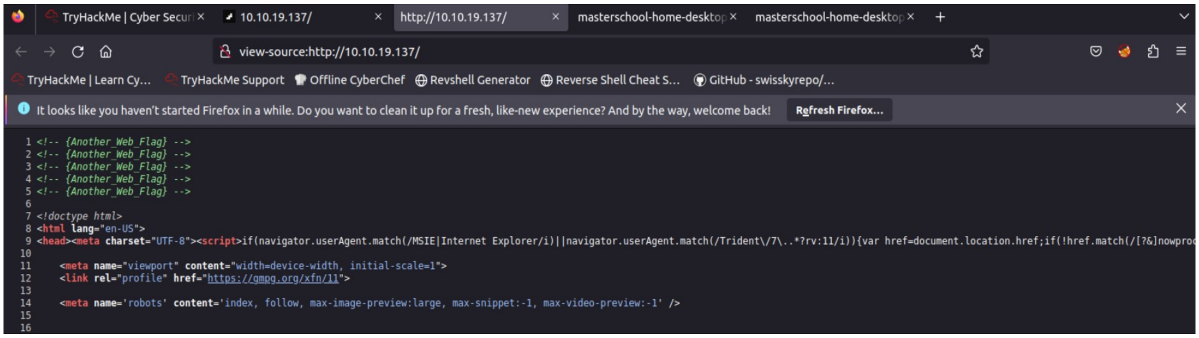
4.4 Third flag: {Another_Web_Flag}

4.5 Fourth flag: {Another_Web_Flag}

4.6 Fifth flag: {Another_Web_Flag}

4.7 Sixth flag: {Another_Web_Flag}

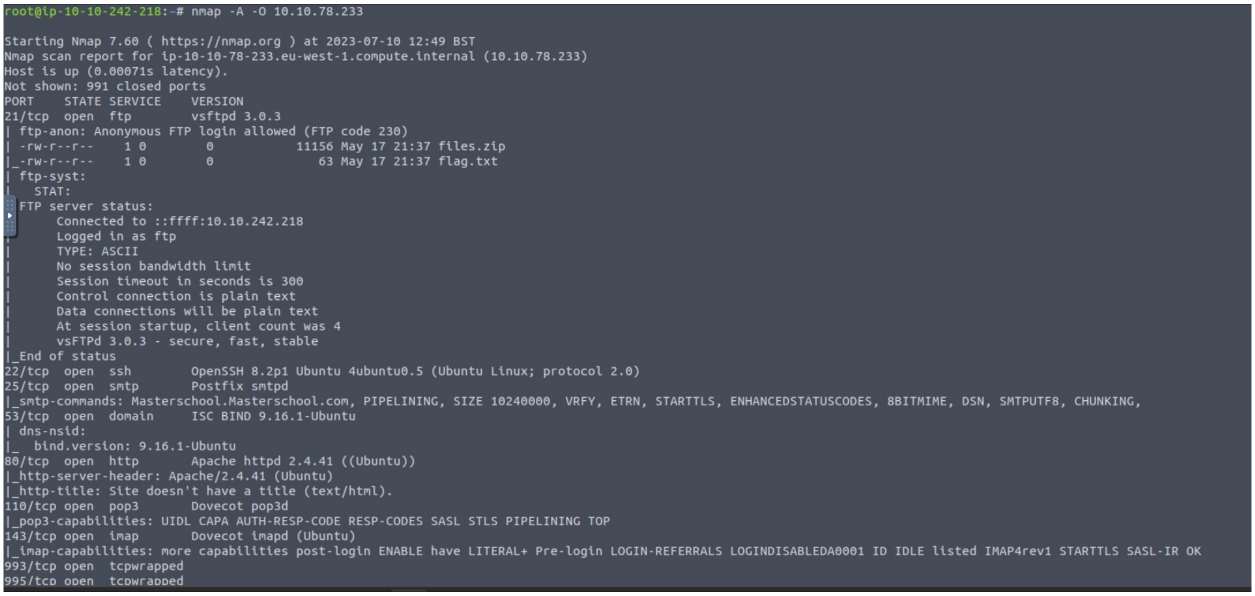
Found some of the flags on the webpage, I right clicked and chose “View page source”


TryHackMe | Cyber Securi
10.10.19.137/
http://10.10.19.137/
masterschool-home-desktop
masterschool-home-desktop
view-source:http://10.10.19.137/
TryHackMe | Learn Cy...
TryHackMe Support
Offline CyberChef
Revshell Generator
Reverse Shell Cheat S...
GitHub - swisskyrepo/...
It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!
Refresh Firefox...

```
<!-- {Another_Web_Flag} -->
<!-- {Another_Web_Flag} -->
<!-- {Another_Web_Flag} -->
<!-- {Another_Web_Flag} -->
<!-- {Another_Web_Flag} -->

<!doctype html>
<html lang="en-US">
<head><meta charset="UTF-8"><script>if(navigator.userAgent.match(/MSIE|Internet Explorer/i)||navigator.userAgent.match(/Trident\/7\..*?rv:11/i)){var href=document.location.href;if(!href.match(/[?&]nowproc

    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="profile" href="https://gmpg.org/xfn/11">

    <meta name='robots' content='index, follow, max-image-preview:large, max-snippet:-1, max-video-preview:-1' />


```

5 Hidden Flags Challenge

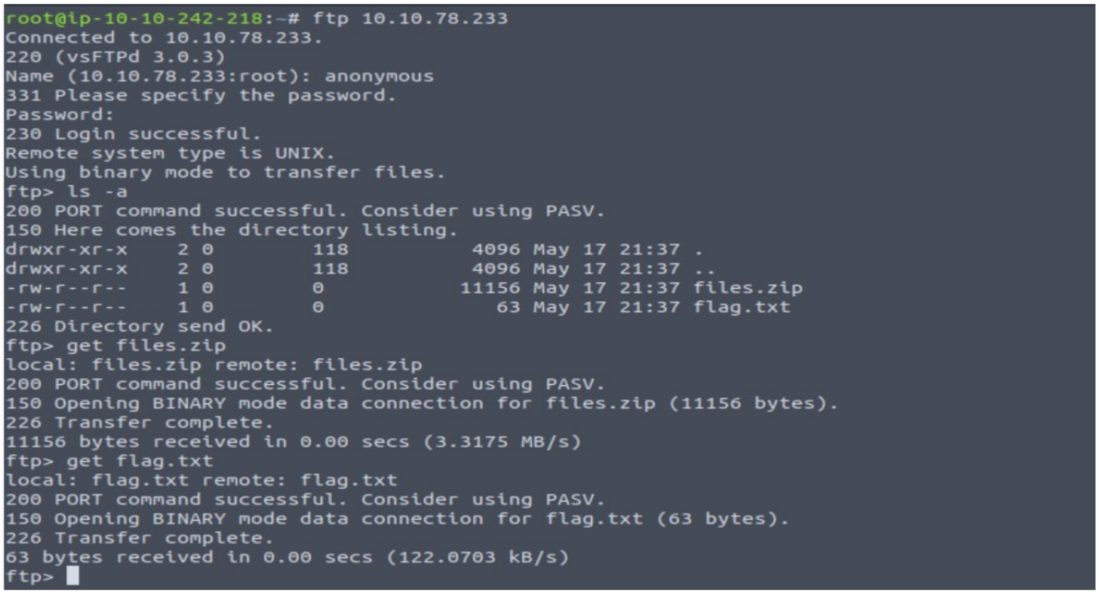
Since I already ransacked the whole file, it is not bound to be in files anymore. Now I am looking into vulnerabilities and other logs.
I ran a detailed nmap scan on my attack machine to find vulnerabilities/open ports on the ctf machine using *nmap -A -O <ctfmachine>*



```
root@ip-10-10-242-218:~# nmap -A -O 10.10.78.233

Starting Nmap 7.60 ( https://nmap.org ) at 2023-07-10 12:49 BST
Nmap scan report for ip-10-10-78-233.eu-west-1.compute.internal (10.10.78.233)
Host is up (0.00071s latency).
Not shown: 991 closed ports
PORT    STATE SERVICE    VERSION
21/tcp  open  ftp        vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| -rw-r--r--    1 0        0           11156 May 17 21:37 files.zip
|_-rw-r--r--    1 0        0              63 May 17 21:37 flag.txt
| ftp-syst:
|   STAT:
 FTP server status:
      Connected to ::ffff:10.10.242.218
      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 3.0.3 - secure, fast, stable
|_End of status
22/tcp  open  ssh        OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
25/tcp  open  smtp       Postfix smtpd
|_smtp-commands: Masterschool.Masterschool.com, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN, SMTPUTF8, CHUNKING,
53/tcp  open  domain     ISC BIND 9.16.1-Ubuntu
| dns-nsid:
|_  bind.version: 9.16.1-Ubuntu
80/tcp  open  http       Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
110/tcp open  pop3       Dovecot pop3d
|_pop3-capabilities: UIDL CAPA AUTH-RESP-CODE RESP-CODES SASL STLS PIPELINING TOP
143/tcp open  imap       Dovecot imapd (Ubuntu)
|_imap-capabilities: more capabilities post-login ENABLE have LITERAL+ Pre-login LOGIN-REFERRALS LOGINDISABLEDA0001 ID IDLE listed IMAP4rev1 STARTTLS SASL-IR OK
993/tcp open  tcpwrapped
995/tcp open  tcpwrapped
```

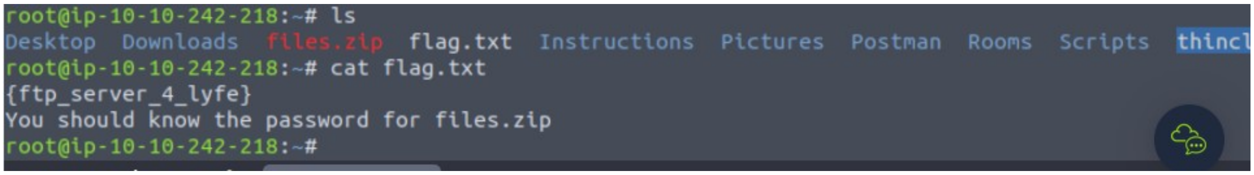
5.2 First flag: {ftp_server_4_lyfe}

This was found from ftp login. With the scan above, I saw that ftp allows anonymous login. On attack machine, I logged in with command *ftp <ctfmachine>*
Checked for files on it using ls -a and found two named "flag.txt" and "files.zip"



```
root@ip-10-10-242-218:~# ftp 10.10.78.233
Connected to 10.10.78.233.
220 (vsFTPd 3.0.3)
Name (10.10.78.233:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -a
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 0        118          4096 May 17 21:37 .
drwxr-xr-x    2 0        118          4096 May 17 21:37 ..
-rw-r--r--    1 0        0           11156 May 17 21:37 files.zip
-rw-r--r--    1 0        0              63 May 17 21:37 flag.txt
226 Directory send OK.
ftp> get files.zip
local: files.zip remote: files.zip
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for files.zip (11156 bytes).
226 Transfer complete.
11156 bytes received in 0.00 secs (3.3175 MB/s)
ftp> get flag.txt
local: flag.txt remote: flag.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for flag.txt (63 bytes).
226 Transfer complete.
63 bytes received in 0.00 secs (122.0703 kB/s)
ftp> 
```

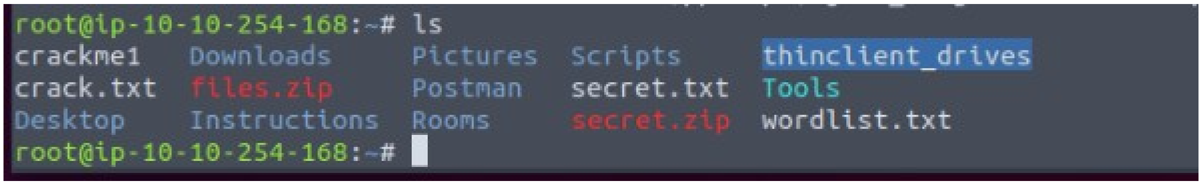
I used get command to download text file and zip file to my attack box. On my attack box, I opened up the flag.txt and got the first hidden flag.



```
root@ip-10-10-242-218:~# ls
Desktop  Downloads  files.zip  flag.txt  Instructions  Pictures  Postman  Rooms  Scripts  thincl
root@ip-10-10-242-218:~# cat flag.txt
{ftp_server_4_lyfe}
You should know the password for files.zip
root@ip-10-10-242-218:~#
```

5.3 Second Flag: {CTF_Time}

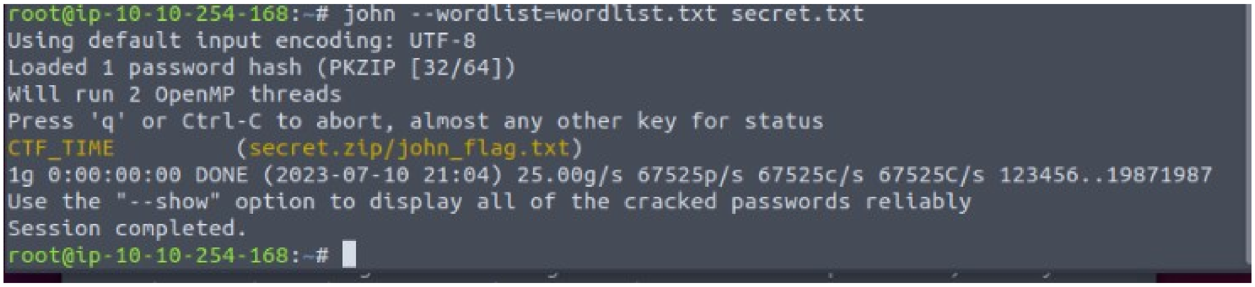
Next is to open the zip file. The zip file is password protected but we got a hint from flag.txt that we should know the password. After multiple combinations, "Masterschool" password worked. I opened the *files.zip* and found another zip file in it. This time, we have a wordlist with it.
I used zip2john to convert the *secret.zip* to *secret.txt*



```
root@ip-10-10-254-168:~# ls
crackme1   Downloads     Pictures  Scripts     thinclient_drives
crack.txt  files.zip     Postman   secret.txt  Tools
Desktop    Instructions  Rooms     secret.zip  wordlist.txt
root@ip-10-10-254-168:~# 
```

zip2john files.zip > secret.txt

I ran john against the hash I got with the downloaded wordlist.



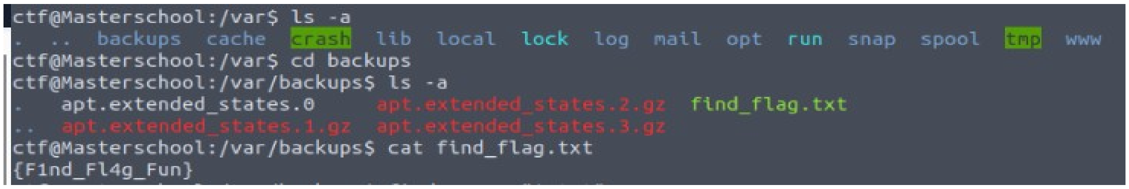
```
root@ip-10-10-254-168:~# john --wordlist=wordlist.txt secret.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
CTF_TIME         (secret.zip/john_flag.txt)
1g 0:00:00:00 DONE (2023-07-10 21:04) 25.00g/s 67525p/s 67525c/s 67525C/s 123456..19871987
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
root@ip-10-10-254-168:~# 
```

Now I will begin to look inside of important directories such as /etc, /var, /usr, /proc, /bin, /sbin, /tmp, /mnt, /lib, /home, /root, /srv

In Var directory, I used find -name "*.txt" and got a list of text files in the directory.

5.4 Third Flag: {F1nd_Fl4g_Fun}

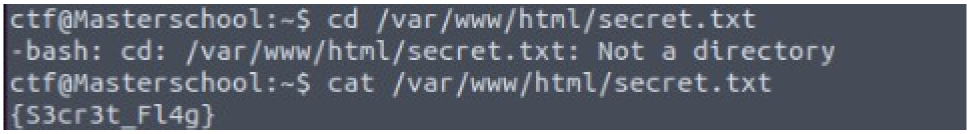
Found in /var/backups/find_flag.txt



```
ctf@Masterschool:/var$ ls -a
.  ..  backups  cache  crash  lib  local  lock  log  mail  opt  run  snap  spool  tmp  www
ctf@Masterschool:/var$ cd backups
ctf@Masterschool:/var/backups$ ls -a
.   apt.extended_states.0     apt.extended_states.2.gz  find_flag.txt
..  apt.extended_states.1.gz  apt.extended_states.3.gz
ctf@Masterschool:/var/backups$ cat find_flag.txt
{F1nd_Fl4g_Fun}
```

5.5 Fourth Flag: {S3cr3t_Fl4g}

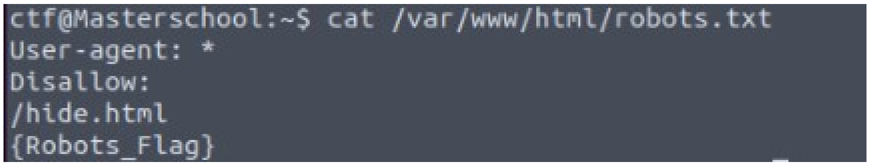
Found in /var/www/html/secret.txt



```
ctf@Masterschool:~$ cd /var/www/html/secret.txt
-bash: cd: /var/www/html/secret.txt: Not a directory
ctf@Masterschool:~$ cat /var/www/html/secret.txt
{S3cr3t_Fl4g}
```

5.6 Fifth Flag: {Robots_Flag}

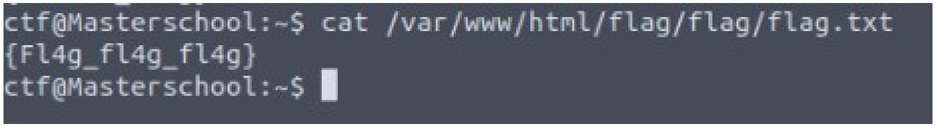
Found in /var/www/html/robots.txt



```
ctf@Masterschool:~$ cat /var/www/html/robots.txt
User-agent: *
Disallow:
/hide.html
{Robots_Flag}
```

5.7 Sixth Flag: {Fl4g_fl4g_fl4g}

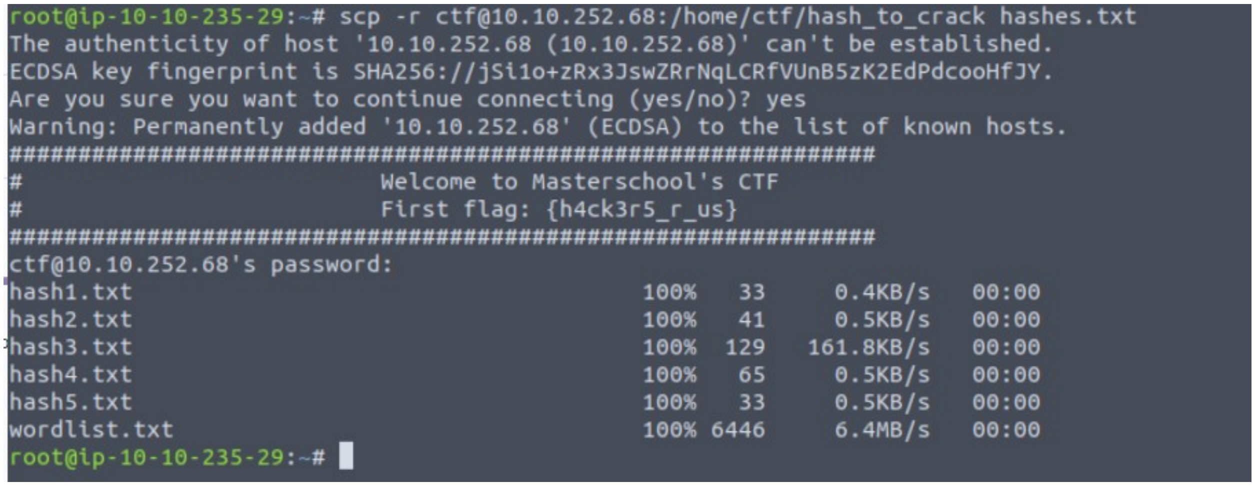
Found in /var/www/html/flag/flag/flag.txt



```
ctf@Masterschool:~$ cat /var/www/html/flag/flag/flag.txt
{Fl4g_fl4g_fl4g}
ctf@Masterschool:~$ █
```

6 Hash Cracking

In the ctf machine, from previous exploitations that I did, I know that the hashes are inside a directory called "*hash_to_crack*" In order to be able to crack the hash, I need to get it inside my attack box by using source and destination command *scp -r <username>@<ctf_machineip>:/home/ctf/hash_to_crack hashes.txt*

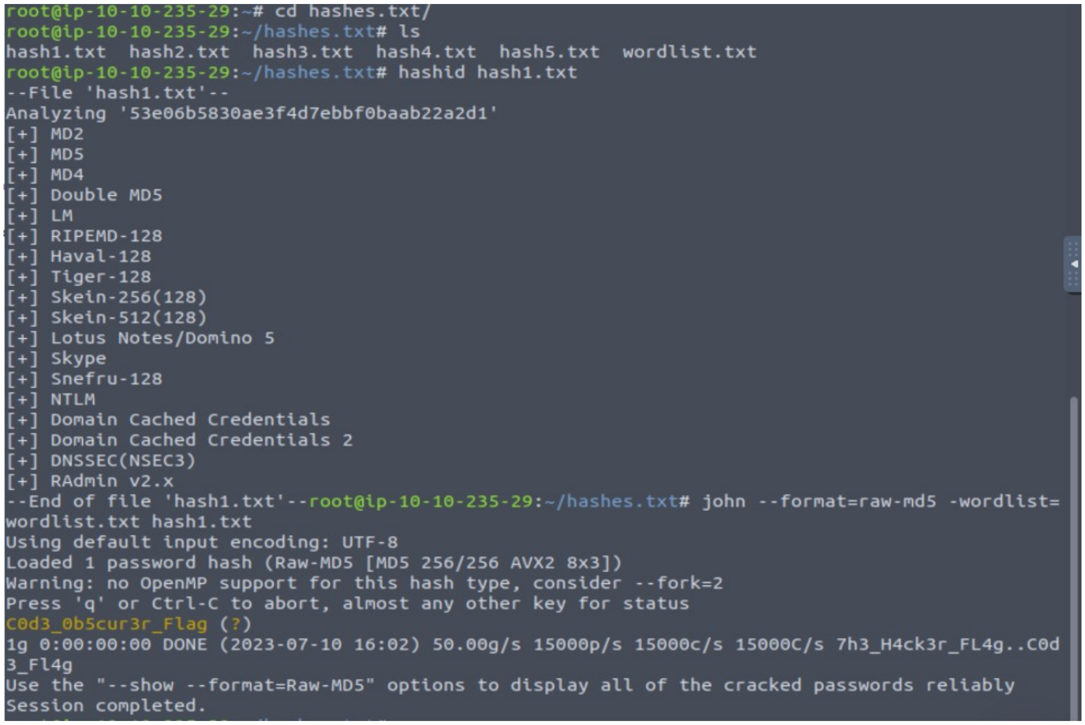


```
root@ip-10-10-235-29:~# scp -r ctf@10.10.252.68:/home/ctf/hash_to_crack hashes.txt
The authenticity of host '10.10.252.68 (10.10.252.68)' can't be established.
ECDSA key fingerprint is SHA256://jSi1o+zRx3JswZRrNqLCRfVUnB5zK2EdPdcooHfJY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.252.68' (ECDSA) to the list of known hosts.
###############################################################
#                          Welcome to Masterschool's CTF
#                          First flag: {h4ck3r5_r_us}
###############################################################
ctf@10.10.252.68's password:
hash1.txt                                     100%   33     0.4KB/s   00:00
hash2.txt                                     100%   41     0.5KB/s   00:00
hash3.txt                                     100%  129   161.8KB/s   00:00
hash4.txt                                     100%   65     0.5KB/s   00:00
hash5.txt                                     100%   33     0.5KB/s   00:00
wordlist.txt                                  100% 6446     6.4MB/s   00:00
root@ip-10-10-235-29:~# 
```

I then installed hashid using *sudo apt-get install hashid -y*

cd into "hashes.txt" and ran *hashid hash1.txt* to get the hash format for john.

Ran *john --format=raw-md5 -wordlist=wordlist.txt hash1.txt* to crack the first hash.

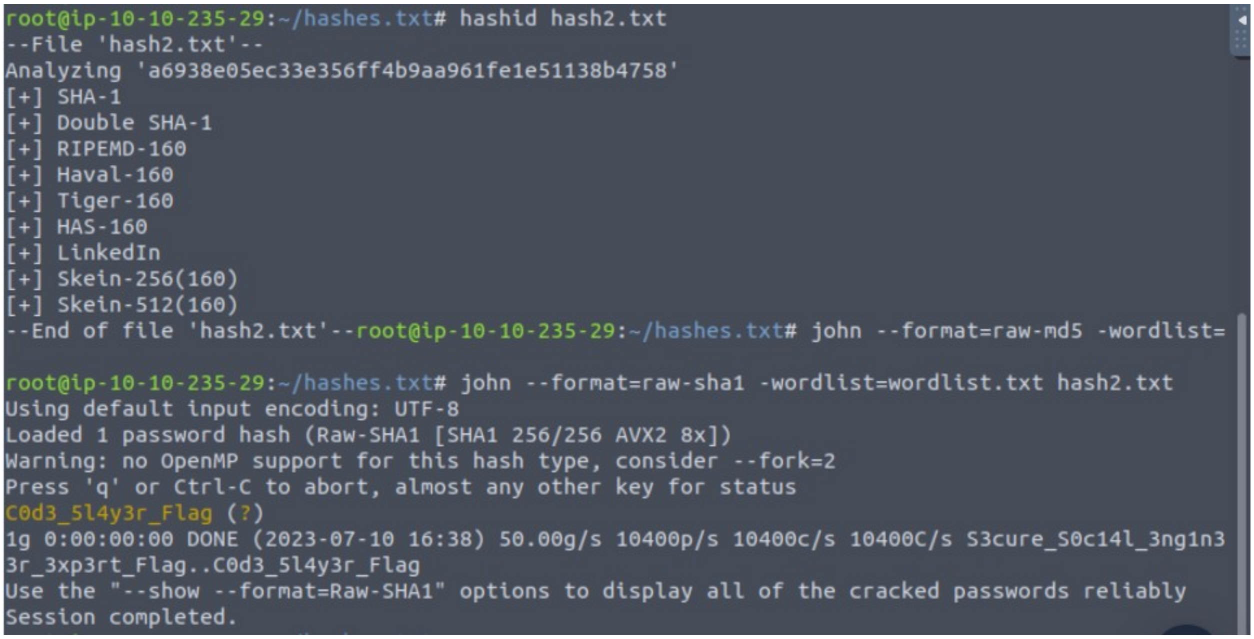


```
root@ip-10-10-235-29:~# cd hashes.txt/
root@ip-10-10-235-29:~/hashes.txt# ls
hash1.txt  hash2.txt  hash3.txt  hash4.txt  hash5.txt  wordlist.txt
root@ip-10-10-235-29:~/hashes.txt# hashid hash1.txt
--File 'hash1.txt'--
Analyzing '53e06b5830ae3f4d7ebbf0baab22a2d1'
[+] MD2
[+] MD5
[+] MD4
[+] Double MD5
[+] LM
[+] RIPEMD-128
[+] Haval-128
[+] Tiger-128
[+] Skein-256(128)
[+] Skein-512(128)
[+] Lotus Notes/Domino 5
[+] Skype
[+] Snefru-128
[+] NTLM
[+] Domain Cached Credentials
[+] Domain Cached Credentials 2
[+] DNSSEC(NSEC3)
[+] RAdmin v2.x
--End of file 'hash1.txt'--root@ip-10-10-235-29:~/hashes.txt# john --format=raw-md5 -wordlist=
wordlist.txt hash1.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
C0d3_0b5cur3r_Flag (?)
1g 0:00:00:00 DONE (2023-07-10 16:02) 50.00g/s 15000p/s 15000c/s 15000C/s 7h3_H4ck3r_FL4g..C0d
3_Fl4g
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

6.2 First Flag: {C0d3_0b5cur3r_Flag}

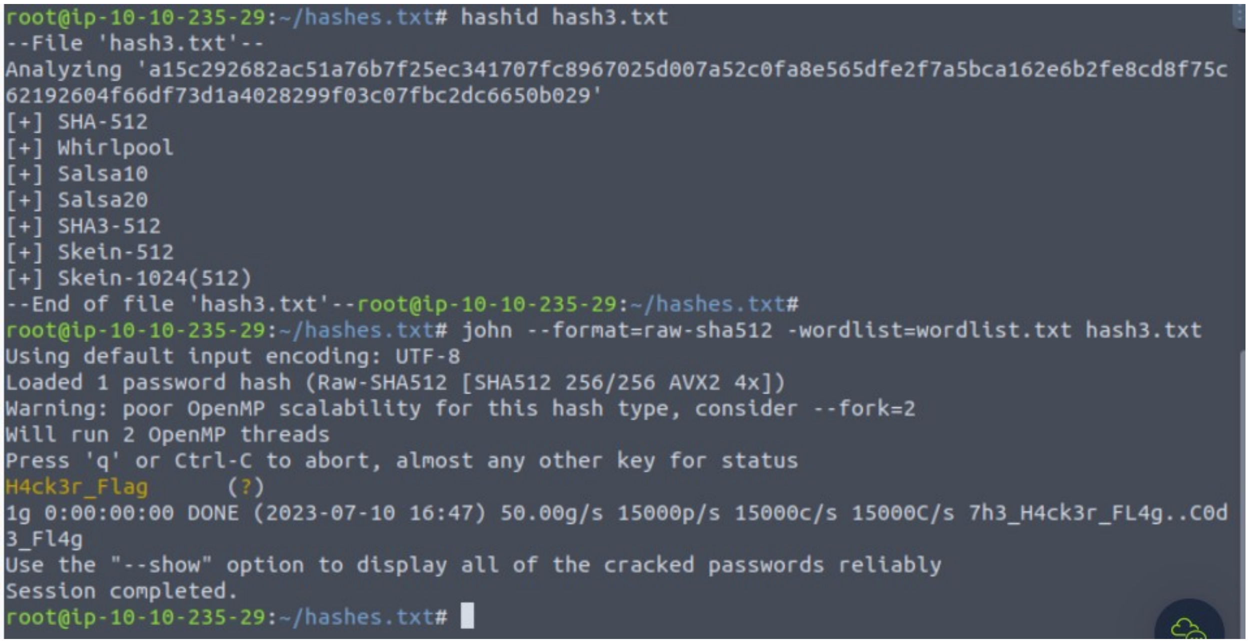
6.3 Second Flag: {C0d3_5l4y3r_Flag}

Found by running hash id to find the hash format and then john against the second hash. Follow the same process to crack hashe3, 4, and 5.



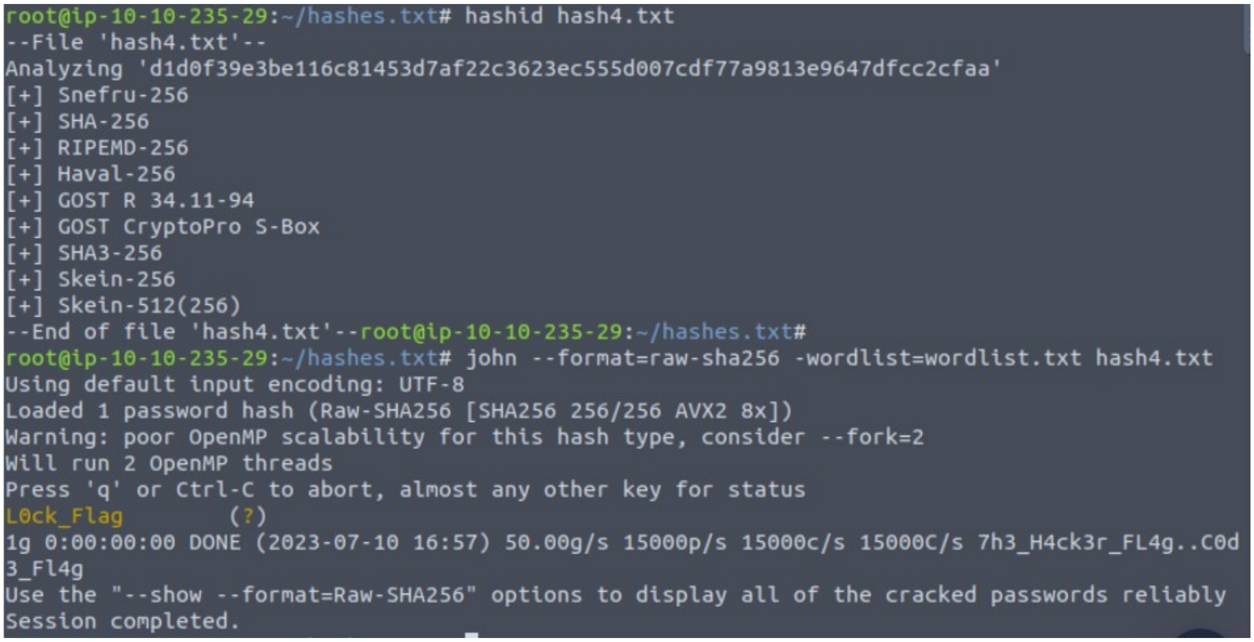
```
root@ip-10-10-235-29:~/hashes.txt# hashid hash2.txt
--File 'hash2.txt'--
Analyzing 'a6938e05ec33e356ff4b9aa961fe1e51138b4758'
[+] SHA-1
[+] Double SHA-1
[+] RIPEMD-160
[+] Haval-160
[+] Tiger-160
[+] HAS-160
[+] LinkedIn
[+] Skein-256(160)
[+] Skein-512(160)
--End of file 'hash2.txt'--root@ip-10-10-235-29:~/hashes.txt# john --format=raw-md5 -wordlist=

root@ip-10-10-235-29:~/hashes.txt# john --format=raw-sha1 -wordlist=wordlist.txt hash2.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 256/256 AVX2 8x])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
C0d3_5l4y3r_Flag (?)
1g 0:00:00:00 DONE (2023-07-10 16:38) 50.00g/s 10400p/s 10400c/s 10400C/s S3cure_S0c14l_3ng1n3
3r_3xp3rt_Flag..C0d3_5l4y3r_Flag
Use the "--show --format=Raw-SHA1" options to display all of the cracked passwords reliably
Session completed.
```

6.4 Third Flag: {H4ck3r_Flag}



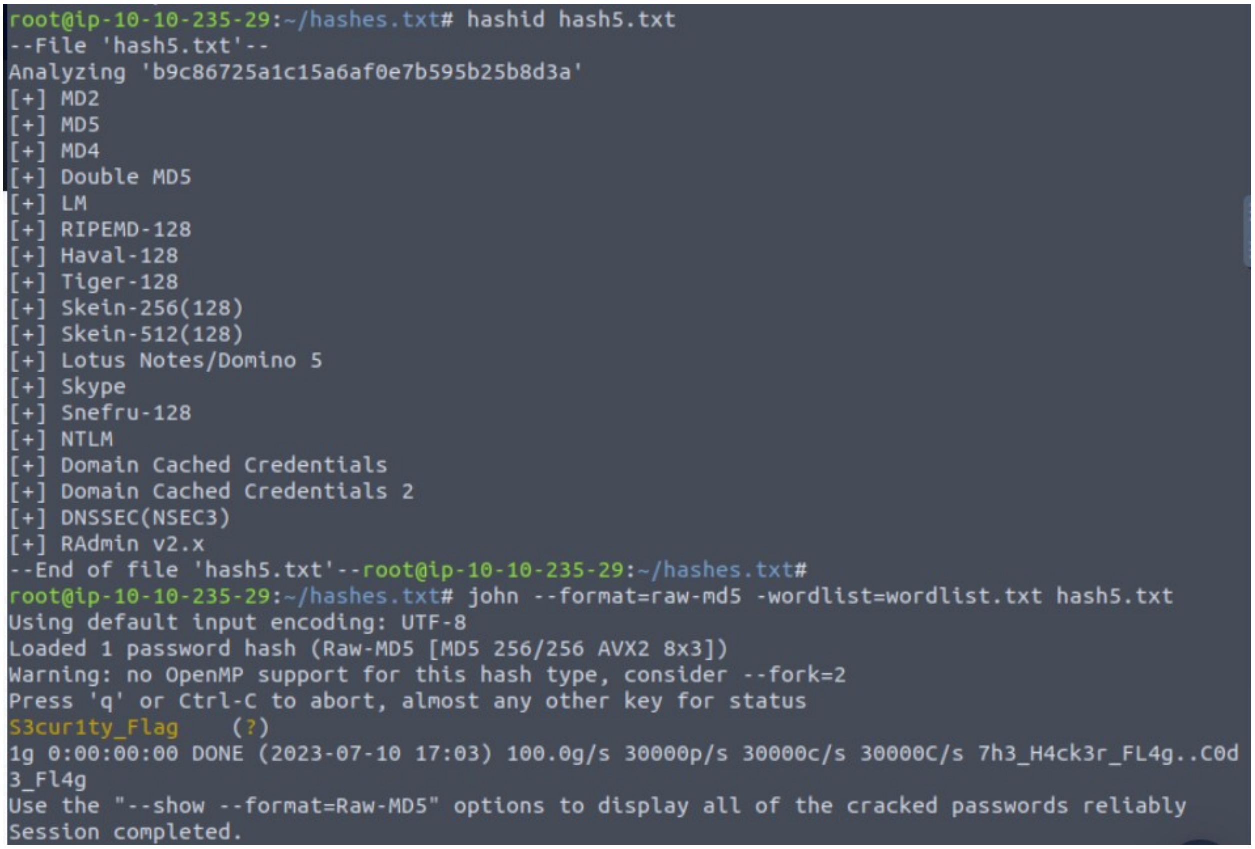
```
root@ip-10-10-235-29:~/hashes.txt# hashid hash3.txt
--File 'hash3.txt'--
Analyzing 'a15c292682ac51a76b7f25ec341707fc8967025d007a52c0fa8e565dfe2f7a5bca162e6b2fe8cd8f75c
62192604f66df73d1a4028299f03c07fbc2dc6650b029'
[+] SHA-512
[+] Whirlpool
[+] Salsa10
[+] Salsa20
[+] SHA3-512
[+] Skein-512
[+] Skein-1024(512)
--End of file 'hash3.txt'--root@ip-10-10-235-29:~/hashes.txt#
root@ip-10-10-235-29:~/hashes.txt# john --format=raw-sha512 -wordlist=wordlist.txt hash3.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA512 [SHA512 256/256 AVX2 4x])
Warning: poor OpenMP scalability for this hash type, consider --fork=2
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
H4ck3r_Flag      (?)
1g 0:00:00:00 DONE (2023-07-10 16:47) 50.00g/s 15000p/s 15000c/s 15000C/s 7h3_H4ck3r_FL4g..C0d
3_Fl4g
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
root@ip-10-10-235-29:~/hashes.txt#
```

6.5 Fourth Flag: {L0ck_Flag}



```
root@ip-10-10-235-29:~/hashes.txt# hashid hash4.txt
--File 'hash4.txt'--
Analyzing 'd1d0f39e3be116c81453d7af22c3623ec555d007cdf77a9813e9647dfcc2cfaa'
[+] Snefru-256
[+] SHA-256
[+] RIPEMD-256
[+] Haval-256
[+] GOST R 34.11-94
[+] GOST CryptoPro S-Box
[+] SHA3-256
[+] Skein-256
[+] Skein-512(256)
--End of file 'hash4.txt'--root@ip-10-10-235-29:~/hashes.txt#
root@ip-10-10-235-29:~/hashes.txt# john --format=raw-sha256 -wordlist=wordlist.txt hash4.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 256/256 AVX2 8x])
Warning: poor OpenMP scalability for this hash type, consider --fork=2
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
L0ck_Flag        (?)
1g 0:00:00:00 DONE (2023-07-10 16:57) 50.00g/s 15000p/s 15000c/s 15000C/s 7h3_H4ck3r_FL4g..C0d
3_Fl4g
Use the "--show --format=Raw-SHA256" options to display all of the cracked passwords reliably
Session completed.
```

6.6 Fifth Flag: {S3cur1ty_Flag}

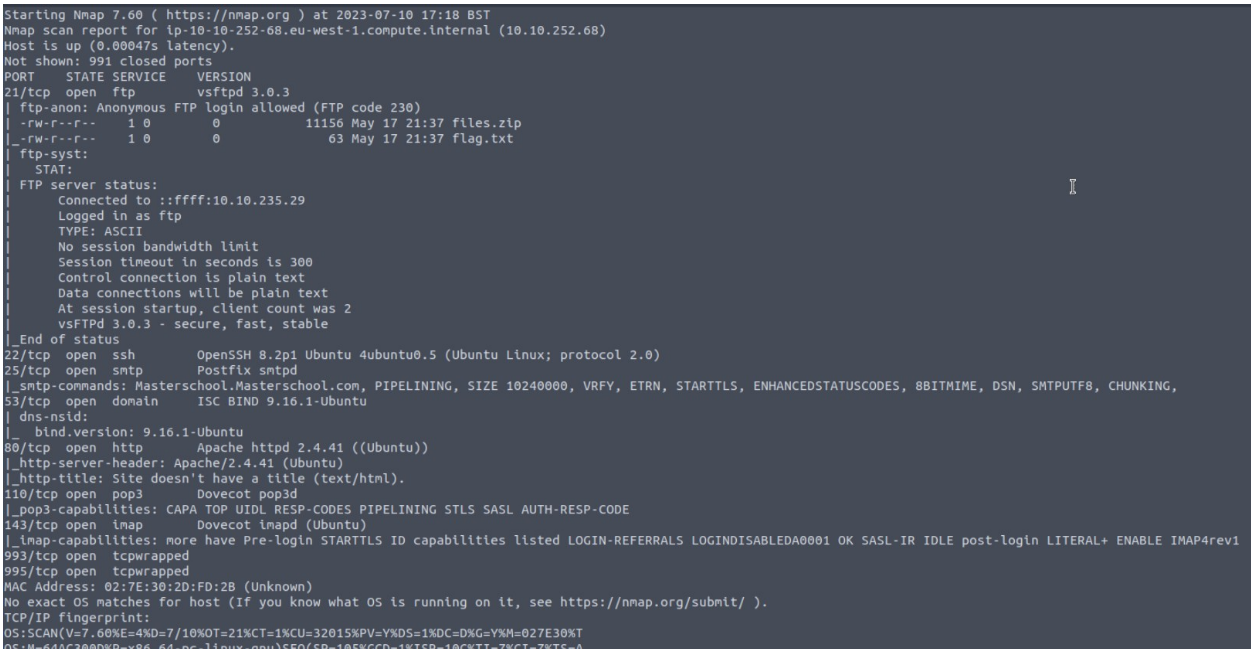


```
root@ip-10-10-235-29:~/hashes.txt# hashid hash5.txt
--File 'hash5.txt'--
Analyzing 'b9c86725a1c15a6af0e7b595b25b8d3a'
[+] MD2
[+] MD5
[+] MD4
[+] Double MD5
[+] LM
[+] RIPEMD-128
[+] Haval-128
[+] Tiger-128
[+] Skein-256(128)
[+] Skein-512(128)
[+] Lotus Notes/Domino 5
[+] Skype
[+] Snefru-128
[+] NTLM
[+] Domain Cached Credentials
[+] Domain Cached Credentials 2
[+] DNSSEC(NSEC3)
[+] RAdmin v2.x
--End of file 'hash5.txt'--root@ip-10-10-235-29:~/hashes.txt#
root@ip-10-10-235-29:~/hashes.txt# john --format=raw-md5 -wordlist=wordlist.txt hash5.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
S3cur1ty_Flag    (?)
1g 0:00:00:00 DONE (2023-07-10 17:03) 100.0g/s 30000p/s 30000c/s 30000C/s 7h3_H4ck3r_FL4g..C0d
3_Fl4g
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

7 NMAP Scan Report

I ran a detailed nmap scan with *nmap -A -O <ctf_machineip>*

Nmap performs a scan with aggressive options such as OS detection, against the ctf machine. It gathered information about the target's open ports, services, and operating system.



```
Starting Nmap 7.60 ( https://nmap.org ) at 2023-07-10 17:18 BST
Nmap scan report for ip-10-10-252-68.eu-west-1.compute.internal (10.10.252.68)
Host is up (0.00047s latency).
Not shown: 991 closed ports
PORT    STATE SERVICE    VERSION
21/tcp  open  ftp        vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| -rw-r--r--    1 0        0           11156 May 17 21:37 files.zip
|_-rw-r--r--    1 0        0              63 May 17 21:37 flag.txt
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.10.235.29
|      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 2
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp  open  ssh        OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
25/tcp  open  smtp       Postfix smtpd
|_smtp-commands: Masterschool.Masterschool.com, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN, SMTPUTF8, CHUNKING,
53/tcp  open  domain     ISC BIND 9.16.1-Ubuntu
| dns-nsid:
|_  bind.version: 9.16.1-Ubuntu
80/tcp  open  http       Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
110/tcp open  pop3       Dovecot pop3d
|_pop3-capabilities: CAPA TOP UIDL RESP-CODES PIPELINING STLS SASL AUTH-RESP-CODE
143/tcp open  imap       Dovecot imapd (Ubuntu)
|_imap-capabilities: more have Pre-login STARTTLS ID capabilities listed LOGIN-REFERRALS LOGINDISABLEDA0001 OK SASL-IR IDLE post-login LITERAL+ ENABLE IMAP4rev1
993/tcp open  tcpwrapped
995/tcp open  tcpwrapped
MAC Address: 02:7E:30:2D:FD:2B (Unknown)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.60%E=4%D=7/10%OT=21%CT=1%CU=32015%PV=Y%DS=1%DC=D%G=Y%M=027E30%T
OS:M=64AC300D%P=x86_64-pc-linux-gnu)SEQ(SP=105%GCD=1%ISR=10C%TI=Z%CI=Z%TS=A
```

7.2 Host information:

IP Address: 10.10.252.68
Hostname: ip-10-10-252-68.eu-west-1.compute.internal
MAC Address: 02:7E:30:2D:FD:2B
Host: Masterschool.Masterschool.com

7.3 Open Ports and Services:

- Port 21/tcp: Open FTP port running vsftpd 3.0.3
    - Anonymous FTP login is allowed, indicating potential data exposure

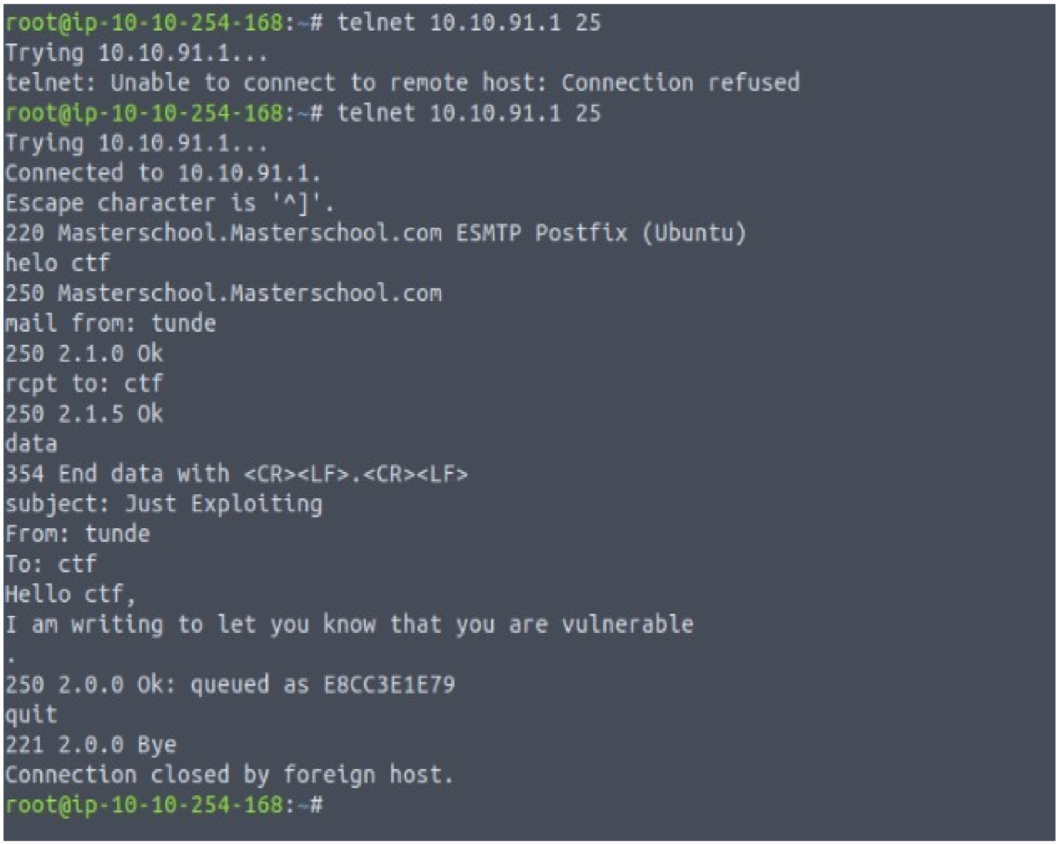
    - Files identified on the FTP server: "files.zip" (size: 11,156 bytes) and "flag.txt" (size: 63 bytes)
- Port 22/tcp: Open SSH port running OpenSSH 8.2p1 Ubuntu 4ubuntu0.5
- Port 25/tcp: Open SMTP port running Postfix smtpd
    - Supports various SMTP commands including PIPELINING, STARTTLS, and CHUNKING
- Port 53/tcp: Open DNS port running ISC BIND 9.16.1-Ubuntu
    - DNS server version identified as 9.16.1-Ubuntu
- Port 80/tcp: Open HTTP port running Apache httpd 2.4.41 (Ubuntu)
- Port 110/tcp: Open POP3 port running Dovecot pop3d
    - Supports various POP3 capabilities including TOP, UIDL, and SASL.
- Port 143/tcp: Open IMAP port running Dovecot imapd (Ubuntu).
    - Supports various IMAP capabilities including STARTTLS and IDLE.
- Port 993/tcp: Open port, possibly running a service that is wrapped in a secure layer (e.g., SSL/TLS).
- Port 995/tcp: Open port, possibly running a service that is wrapped in a secure layer (e.g., SSL/TLS).

7.3 **Potential Vulnerabilities and Solution –**

- Anonymous FTP Access (Port 21/tcp): The FTP server allows anonymous logins, which could potentially lead to unauthorized access or data leakage. The files "files.zip" and "flag.txt" are accessible, indicating the need for securing FTP access and evaluating the contents of these files for sensitive information. FTP sends data in plaintext; it is best not to be used.

- OpenSSH (Port 22/tcp): The OpenSSH version 8.2p1 Ubuntu 4ubuntu0.5 is running. Ensure that the SSH service is properly configured with strong authentication and encryption settings to prevent unauthorized access.

- Postfix SMTP Service (Port 25/tcp): The Postfix SMTP service is running, supporting several SMTP commands and extensions. Regularly apply security updates and follow best practices to protect the SMTP service from potential vulnerabilities and abuse.
    - I was able to get on SMTP and send out email. Check screenshot below:

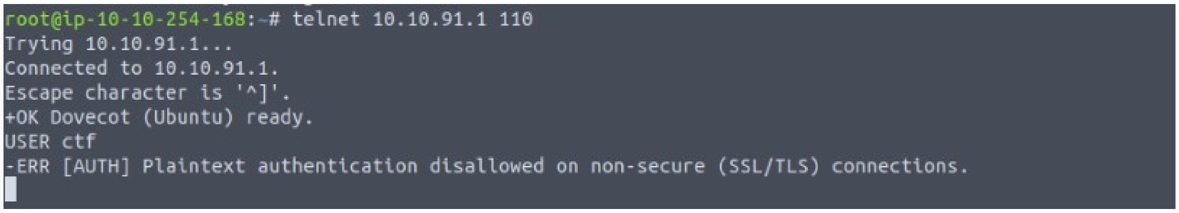


```
root@ip-10-10-254-168:~# telnet 10.10.91.1 25
Trying 10.10.91.1...
telnet: Unable to connect to remote host: Connection refused
root@ip-10-10-254-168:~# telnet 10.10.91.1 25
Trying 10.10.91.1...
Connected to 10.10.91.1.
Escape character is '^]'.
220 Masterschool.Masterschool.com ESMTP Postfix (Ubuntu)
helo ctf
250 Masterschool.Masterschool.com
mail from: tunde
250 2.1.0 Ok
rcpt to: ctf
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
subject: Just Exploiting
From: tunde
To: ctf
Hello ctf,
I am writing to let you know that you are vulnerable
.
250 2.0.0 Ok: queued as E8CC3E1E79
quit
221 2.0.0 Bye
Connection closed by foreign host.
root@ip-10-10-254-168:~#
```

- ISC BIND DNS Server (Port 53/tcp): Best security practice on this is to keep DNS software up to date and follow best practices.

- Apache HTTP Server (Port 80/tcp): Regularly apply security patches, It is best to stay up to date and use HTTPS as HTTP is more susceptible to attacks such as response splitting, and injection attacks

- Dovecot POP3 (Port 110/tcp) and IMAP (Port 143/tcp) Services: The Dovecot POP3 and IMAP services are open. Ensure that proper authentication mechanisms are in place and SSL/TLS is correctly configured to secure email communications. I am unable to exploit these services because plaintext authentication is disallowed. This is good practice.



```
root@ip-10-10-254-168:~# telnet 10.10.91.1 110
Trying 10.10.91.1...
Connected to 10.10.91.1.
Escape character is '^]'.
+OK Dovecot (Ubuntu) ready.
USER ctf
-ERR [AUTH] Plaintext authentication disallowed on non-secure (SSL/TLS) connections.
```