22. [Hacking Etico] Rompiendo WordPress para Aprender a Repararlo

PARTE 1

1) Requisitos previos (comprobación rápida)

1.1 Verificar Docker instalado

- docker --version

```
ub@forense-ai:~$ docker --version
Docker version 29.2.0, build 0b9d198
```

- sudo docker ps

```
ub@forense-ai:~$ sudo docker ps
[sudo] contraseña para ub:
CONTAINER ID   IMAGE              COMMAND                  CREATED        STATUS              PORTS
                                  NAMES
f78426341423   wordpress:latest   "docker-entrypoint.s…"   2 months ago   Up About a minute   0.0.0.
0:8000->80/tcp, [::]:8000->80/tcp   wp_site
5d771c5bb40a   mariadb:latest     "docker-entrypoint.s…"   2 months ago   Up About a minute   3306/t
cp                                  wp_mariadb
```

2) Crear la red para comunicar contenedores

- sudo docker network create wp-net

```
ub@forense-ai:~$ sudo docker network create wp-net
bee1eacc3ea2edbcdd9e03a70af035672d4ebb5288828e50f58e6c520d545eb1
```

- sudo docker network ls

```
ub@forense-ai:~$ sudo docker network ls
NETWORK ID     NAME                                              DRIVER    SCOPE
9b24f4037bd1   bridge                                            bridge    local
49a1fc55ab01   host                                              host      local
3ad05d5a7461   none                                              null      local
7145529f577b   red1_mynetwork                                    bridge    local
6268eb7532dd   starfleet-prueba-competencial-lixinyuan_default   bridge    local
bee1eacc3ea2   wp-net                                            bridge    local
```

3) Crear volúmenes para persistencia

- sudo docker volume create wp-db

```
ub@forense-ai:~$ sudo docker volume create wp-db
wp-db
```

- sudo docker volume create wp-html

```
ub@forense-ai:~$ sudo docker volume create wp-html
wp-html
```

- sudo docker volume ls

```
ub@forense-ai:~$ sudo docker volume ls
DRIVER    VOLUME NAME
local     1db859d2f1bed46433a3a9837e3e90991f0470dc35103fef79b59caf08165608
local     85e18fb0005d454dbab0030df54d4643f9b06be4038ab85fc947cc9b820ebd41
local     f50db82dd0c095486f67c528b236fd6bd360fc35c900cac4a53f44c5dc41187b
local     starfleet-prueba-competencial-lixinyuan_db_data
local     starfleet-prueba-competencial-lixinyuan_wordpress_data
local     wp-db
local     wp-html
```

4) Levantar la base de datos (MySQL)

4.1 Ejecutar MySQL con variables de entorno

Crea un contenedor llamado wp-mysql:

```
sudo docker run -d \
--name wp-mysql \
--network wp-net \
-v wp-db:/var/lib/mysql \
-e MYSQL_DATABASE=wordpress \
-e MYSQL_USER=wpuser \
-e MYSQL_PASSWORD=wp-pass-123 \
-e MYSQL_ROOT_PASSWORD=root-pass-123 \
mysql:8.0
```

```
ub@forense-ai:~$ sudo docker run -d \
--name wp-mysql \
--network wp-net \
-v wp-db:/var/lib/mysql \
-e MYSQL_DATABASE=wordpress \
-e MYSQL_USER=wpuser \
-e MYSQL_PASSWORD=wp-pass-123 \
-e MYSQL_ROOT_PASSWORD=root-pass-123 \
mysql:8.0
```

```
[sudo] contraseña para ub:
Unable to find image 'mysql:8.0' locally
8.0: Pulling from library/mysql
b4f9f94601d6: Pull complete
ce1944d72bca: Pull complete
c21bb7e51cd3: Pull complete
d9bba75f9c49: Pull complete
d7842cc74782: Pull complete
526dbe6f3591: Pull complete
c21ebd7bbded: Pull complete
142c5fc9ea8a: Pull complete
3145e04eb470: Pull complete
b05b17de9630: Pull complete
f79b7cf07833: Pull complete
db1d6a1e96dc: Download complete
16a042b28046: Download complete
Digest: sha256:9c7897818a32cb639b0404fadd828c7cbc522da90398107fbae55682aee577c9
Status: Downloaded newer image for mysql:8.0
5fc8a5674dcf9a6c69931dc0cd235641d15e801c599415e58adb964be427b5d7
```

4.2 Comprobar que está vivo

- docker ps docker logs wp-mysql --tail 30

```
ub@forense-ai:~$ sudo docker logs wp-mysql --tail 30
2026-02-05T11:11:41.553113Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Socket:
/var/run/mysqld/mysqlx.sock
2026-02-05T11:11:41.553191Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections.
 Version: '8.0.45'  socket: '/var/run/mysqld/mysqld.sock'  port: 0  MySQL Community Server - GPL.
2026-02-05 11:11:41+00:00 [Note] [Entrypoint]: Temporary server started.
```

5) Levantar WordPress

5.1 Ejecutar WordPress apuntando a la DB

Creamos el contenedor wp-web y lo publicamos en el puerto 8080:

```
sudo docker run -d \
--name wp-web \
--network wp-net \
-p 8080:80 \
-v wp-html:/var/www/html \
-e WORDPRESS_DB_HOST=wp-mysql:3306 \
-e WORDPRESS_DB_NAME=wordpress \
-e WORDPRESS_DB_USER=wpuser \
-e WORDPRESS_DB_PASSWORD=wp-pass-123 \
wordpress:latest
```

```
ub@forense-ai:~$ sudo docker run -d \
--name wp-web \
--network wp-net \
-p 8080:80 \
-v wp-html:/var/www/html \
-e WORDPRESS_DB_HOST=wp-mysql:3306 \
-e WORDPRESS_DB_NAME=wordpress \
-e WORDPRESS_DB_USER=wpuser \
-e WORDPRESS_DB_PASSWORD=wp-pass-123 \
wordpress:latest
108022afa3c5b853e98e7c6a3f488d9a5096fd7962f68937242b42dc8fb35f34
```

5.2 Verificar logs

- sudo docker logs wp-web --tail 30

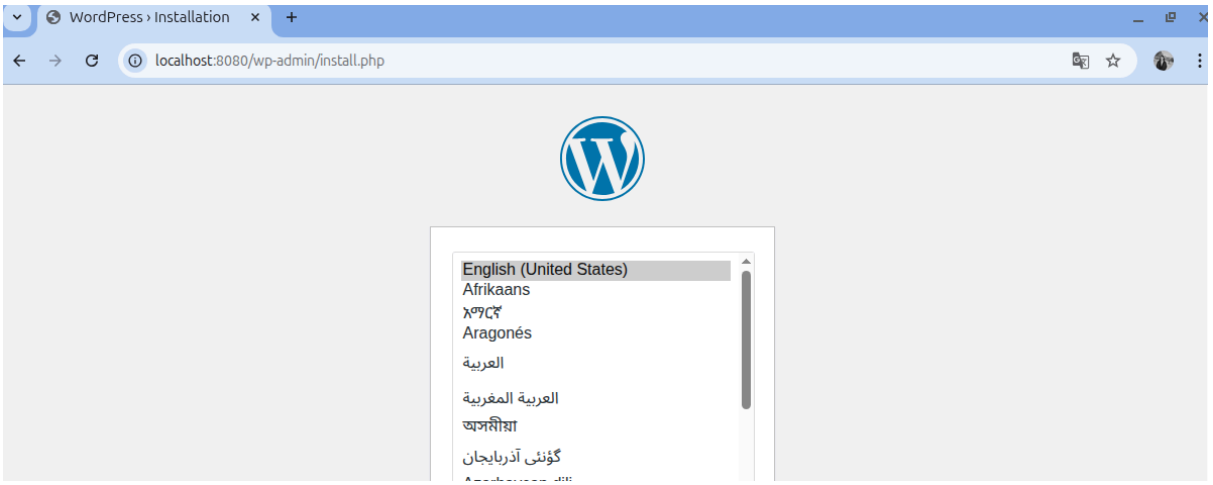
```
ub@forense-ai:~$ sudo docker logs wp-web --tail 30
WordPress not found in /var/www/html - copying now...
Complete! WordPress has been successfully copied to /var/www/html
No 'wp-config.php' found in /var/www/html, but 'WORDPRESS_...' variables supplied; copying 'wp-confi
g-docker.php' (WORDPRESS_DB_HOST WORDPRESS_DB_NAME WORDPRESS_DB_PASSWORD WORDPRESS_DB_USER)
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.1
9.0.3. Set the 'ServerName' directive globally to suppress this message
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.1
9.0.3. Set the 'ServerName' directive globally to suppress this message
[Thu Feb 05 11:20:12.851987 2026] [mpm_prefork:notice] [pid 1:tid 1] AH00163: Apache/2.4.65 (Debian)
 PHP/8.3.27 configured -- resuming normal operations
[Thu Feb 05 11:20:12.852013 2026] [core:notice] [pid 1:tid 1] AH00094: Command line: 'apache2 -D FOR
EGROUND'
```

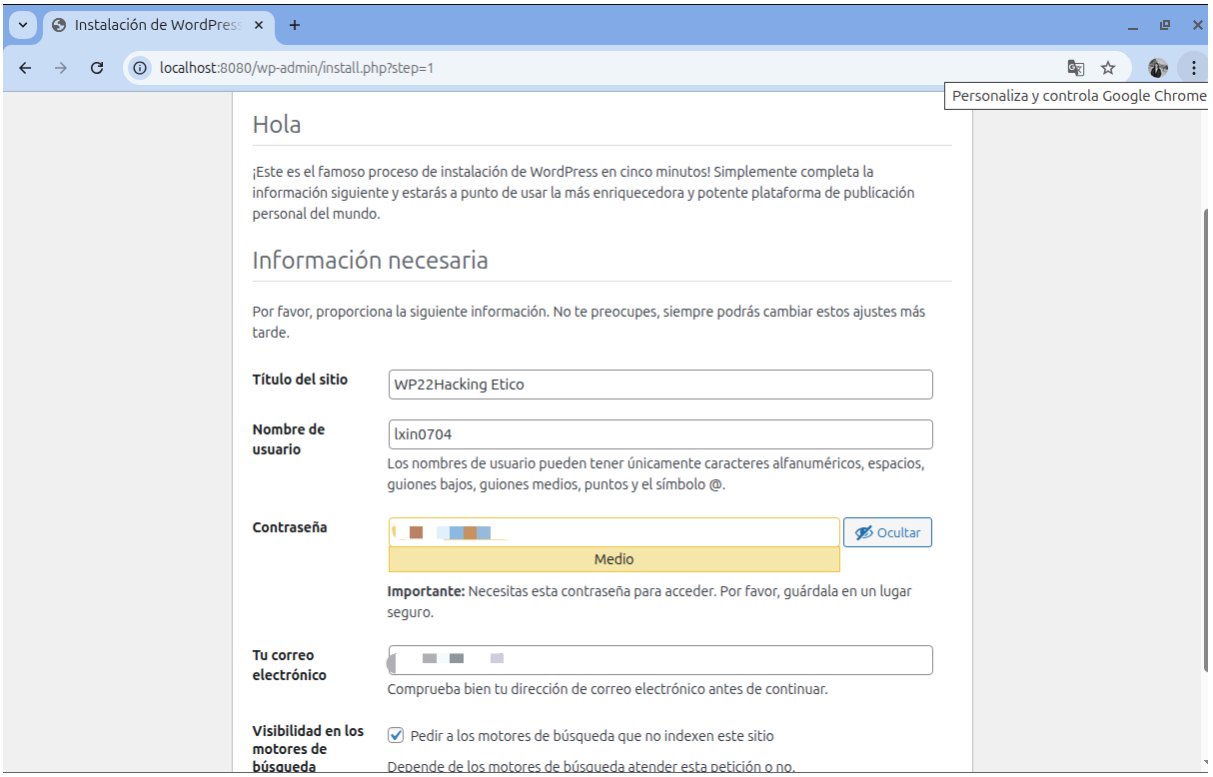
6) Acceso por navegador y asistente de instalación

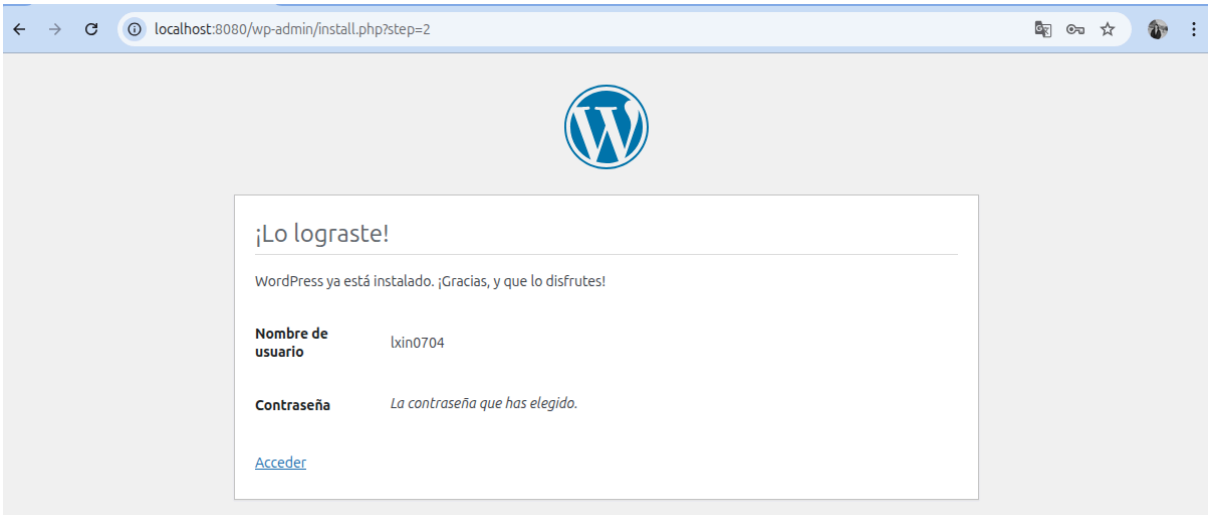
Abre:

- http://localhost:8080

Completa el instalador: ● Idioma ● Título del sitio ● Usuario admin ● Password ● Email







7) Verificación técnica (para alumnos curiosos)

7.1 Ver la red y quién está conectado

- sudo docker network inspect wp-net

```
ub@forense-ai:~$ sudo docker network inspect wp-net
[
    {
        "Name": "wp-net",
        "Id": "bee1eacc3ea2edbcdd9e03a70af035672d4ebb5288828e50f58e6c520d545eb1",
        "Created": "2026-02-05T11:44:10.702583332+01:00",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv4": true,
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.19.0.0/16",
                    "Gateway": "172.19.0.1"
```

7.2 Ver volúmenes y dónde se usan

- sudo docker inspect wp-mysql | grep -A 10 Mounts

```
ub@forense-ai:~$ sudo docker inspect wp-mysql | grep -A 10 Mounts
        "Mounts": [
            {
                "Type": "volume",
                "Name": "wp-db",
                "Source": "/var/lib/docker/volumes/wp-db/_data",
                "Destination": "/var/lib/mysql",
                "Driver": "local",
                "Mode": "z",
                "RW": true,
                "Propagation": ""
            }
```

- sudo docker inspect wp-web | grep -A 10 Mounts

```
ub@forense-ai:~$ sudo docker inspect wp-web | grep -A 10 Mounts
        "Mounts": [
            {
                "Type": "volume",
                "Name": "wp-html",
                "Source": "/var/lib/docker/volumes/wp-html/_data",
                "Destination": "/var/www/html",
                "Driver": "local",
                "Mode": "z",
                "RW": true,
                "Propagation": ""
            }
```

8) Parar, arrancar y reiniciar (operaciones típicas)

8.1 Parar todo

- sudo docker stop wp-web wp-mysql

```
ub@forense-ai:~$ sudo docker stop wp-web wp-mysql
wp-web
wp-mysql
```

```
ub@forense-ai:~$ sudo docker ps -a
CONTAINER ID   IMAGE              COMMAND                  CREATED        STATUS                     PORTS                                      NAMES
108022afa3c5   wordpress:latest   "docker-entrypoint.s…"   4 hours ago    Exited (0) 38 seconds ago                                             wp-web
5fc8a5674dcf   mysql:8.0          "docker-entrypoint.s…"   4 hours ago    Exited (0) 38 seconds ago                                             wp-mysql
1966731bd0a9   red1-web1          "httpd-foreground"       4 weeks ago    Exited (0) 4 weeks ago                                                red1-web1-1
94f1fc73bb80   red1-web2          "httpd-foreground"       4 weeks ago    Exited (0) 4 weeks ago                                                red1-web2-1
640fa97ff40f   red1-parrot        "bash"                   4 weeks ago    Exited (137) 4 weeks ago                                              red1-parrot-1
bc229cdadbbe   red1-firewall      "/usr/local/bin/fire…"   4 weeks ago    Exited (137) 4 weeks ago                                              red1-firewall-1
f78426341423   wordpress:latest   "docker-entrypoint.s…"   2 months ago   Up 19 minutes              0.0.0.0:8000->80/tcp, [::]:8000->80/tcp    wp_site
5d771c5bb40a   mariadb:latest     "docker-entrypoint.s…"   2 months ago   Up 19 minutes              3306/tcp                                   wp_mariadb
```

8.2 Arrancar todo

- sudo docker start wp-mysql wp-web

```
ub@forense-ai:~$ sudo docker start wp-web wp-mysql
wp-web
wp-mysql
ub@forense-ai:~$ sudo docker ps -a
CONTAINER ID   IMAGE              COMMAND                  CREATED        STATUS                     PORTS                                      NAMES
108022afa3c5   wordpress:latest   "docker-entrypoint.s…"   4 hours ago    Up 3 seconds               0.0.0.0:8080->80/tcp, [::]:8080->80/tcp    wp-web
5fc8a5674dcf   mysql:8.0          "docker-entrypoint.s…"   4 hours ago    Up 3 seconds               3306/tcp, 33060/tcp                        wp-mysql
1966731bd0a9   red1-web1          "httpd-foreground"       4 weeks ago    Exited (0) 4 weeks ago                                                red1-web1-1
94f1fc73bb80   red1-web2          "httpd-foreground"       4 weeks ago    Exited (0) 4 weeks ago                                                red1-web2-1
640fa97ff40f   red1-parrot        "bash"                   4 weeks ago    Exited (137) 4 weeks ago                                              red1-parrot-1
bc229cdadbbe   red1-firewall      "/usr/local/bin/fire…"   4 weeks ago    Exited (137) 4 weeks ago                                              red1-firewall-1
f78426341423   wordpress:latest   "docker-entrypoint.s…"   2 months ago   Up 20 minutes              0.0.0.0:8000->80/tcp, [::]:8000->80/tcp    wp_site
5d771c5bb40a   mariadb:latest     "docker-entrypoint.s…"   2 months ago   Up 20 minutes              3306/tcp                                   wp_mariadb
ub@forense-ai:~$
```

8.3 Reiniciar WordPress

- sudo docker restart wp-web

```
ub@forense-ai:~$ sudo docker restart wp-web
wp-web
```

PARTE 2

1. Contexto: qué es WPScan (y qué NO es)

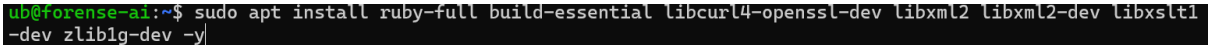
WPScan es un escáner de vulnerabilidades específico para WordPress.
No ataca servidores “a lo loco”: analiza temas, plugins, usuarios, versiones y configuraciones inseguras contrastándolas con una base de datos real de vulnerabilidades conocidas.
No explota por defecto. Observa, enumera y reporta. La maldad viene después… o no.

2. Entorno típico de laboratorio

- WordPress en:
  - Docker
  - VM local
  - Hosting de pruebas
- Kali / Parrot / Ubuntu atacante
- WPScan instalado
- Dominio de pruebas o IP local

Instalar WPScan

- sudo apt install ruby-full build-essential libcurl4-openssl-dev libxml2 libxml2-dev libxslt1-dev zlib1g-dev -y



- sudo gem install wpscan

- wpscan --version

3. Caso de uso 1 — Identificación básica del objetivo

Objetivo

Confirmar que el sitio usa WordPress y obtener información general sin ser agresivos.

- Detección de WordPress
  - wpscan --url http://localhost:8080

```
ub@forense-ai:~$ wpscan --url http://localhost:8080
```

- Versión del core

```
[+] WordPress version 6.8.3 identified (Outdated, released on 2025-09-30).
 | Found By: Rss Generator (Passive Detection)
 |  - http://localhost:8080/?feed=rss2, <generator>https://wordpress.org/?v=6.8.3</generator>
 |  - http://localhost:8080/?feed=comments-rss2, <generator>https://wordpress.org/?v=6.8.3</generator>
```

- Archivos expuestos (readme.html, xmlrpc.php)
  - xmlrpc.php

```
[+] XML-RPC seems to be enabled: http://localhost:8080/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/
```

  - readme.html

```
[+] WordPress readme found: http://localhost:8080/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
```

- Nivel de información filtrada públicamente
  1. **El servidor y el entorno de ejecución:**
     En las entradas de Headers, se revelan los siguientes detalles: Servidor: Apache/2.4.65 (Debian), X-Powered-By: PHP/8.3.27
     - Nivel de confianza del 100 %

  2. **El programa principal:**
     versión principal de WordPress 6.8.3 divulgada
     - Nivel de confianza del 100 %

  3. **El tema del sitio:**
     El tema actualmente activo es twentytwentyfive, que revela detalles fundamentales completos, incluyendo la versión 1.3 del tema, la ruta de almacenamiento, la fecha de actualización, el autor y los enlaces a las hojas de estilo.

- Nivel de confianza del 80 %

4. **Funciones o archivos sensibles:**
   Fuga de xmlrpc.php y readme.html expuestos, con la funcionalidad de tareas programadas externas habilitada en wp-cron.php.
   - Nivel de confianza: 60 %.

5. **Sin fuga de información de alta sensibilidad:**
   Tras la detección pasiva y activa, no se identificaron fugas de información de alta sensibilidad, como plugins del sitio, archivos de copia de seguridad de la configuración o similares.

4. Caso de uso 2 — Enumeración de usuarios

Objetivo

Descubrir usuarios válidos del sistema.

Comando

wpscan --url https://victima.local --enumerate u

```
ub@forense-ai:~$ wpscan --url http://localhost:8080 --enumerate u
```

Resultado típico

Enumeración de nombre de usuario: lxin0704.

```
[+] lxin0704
 | Found By: Rss Generator (Passive Detection)
 | Confirmed By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
```

XML-RPC habilitado (XML-RPC seems to be enabled)
Se puede utilizar System.multicall para intentar cientos de contraseñas simultáneamente.

```
[+] XML-RPC seems to be enabled: http://localhost:8080/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
```

Empleo de ataques de diccionario: Uso de archivos como rockyou.txt.

Acceso al backend: Control del sitio web.

5. Caso de uso 3 — Enumeración de plugins vulnerables

Objetivo

Detectar plugins instalados y si tienen vulnerabilidades conocidas.

Comando

- wpscan --url https://victima.local --enumerate p

```
ub@forense-ai:~$ wpscan --url http://localhost:8080 --enumerate p
```

Como el complemento aún no se ha instalado, no se ha detectado durante el análisis. Por lo tanto, no es posible determinar la versión del complemento ni otra información relevante.

```
[+] Enumerating Most Popular Plugins (via Passive Methods)

[i] No plugins Found.
```

6. Caso de uso 4 — Enumeración de temas

Objetivo

Detectar temas inseguros o abandonados.

Comando

- wpscan --url https://victima.local --enumerate t

```
ub@forense-ai:~$ wpscan --url http://localhost:8080 --enumerate p
```

- Temas antiguos = código muerto
    - El tema actualmente utiliza una versión obsoleta, la 1.3, y no se ha actualizado a la 1.4. Esto implica que la versión 1.3 contiene código que no ha sido parcheado oficialmente, junto con lógica funcional obsoleta (código muerto). Dicho código muerto puede albergar vulnerabilidades sin parchear y problemas de compatibilidad con el entorno, lo que constituye una debilidad directa para la seguridad del sitio.
- Temas premium mal mantenidos
    - Dado que se está utilizando la versión gratuita, no se han detectado temas premium/de pago
- Información de rutas y ficheros
    - Los resultados del análisis exponen directamente las rutas y archivos confidenciales relacionados con el sujeto a los que se puede acceder directamente

1. **Directorio raíz del tema:** http://localhost:8080/wp-content/themes/twentytwentyfive/
2. **Archivo de descripción del tema:** http://localhost:8080/wp-content/themes/twentytwentyfive/readme.txt
3. **Hoja de estilo principal:** http://localhost:8080/wp-content/themes/twentytwentyfive/style.css?ver=1.3

```
[+] WordPress theme in use: twentytwentyfive
 | Location: http://localhost:8080/wp-content/themes/twentytwentyfive/
 | Last Updated: 2025-12-03T00:00:00.000Z
 | Readme: http://localhost:8080/wp-content/themes/twentytwentyfive/readme.txt
 | [!] The version is out of date, the latest version is 1.4
 | Style URL: http://localhost:8080/wp-content/themes/twentytwentyfive/style.css?ver=1.3
 | Style Name: Twenty Twenty-Five
 | Style URI: https://wordpress.org/themes/twentytwentyfive/
 | Description: Twenty Twenty-Five emphasizes simplicity and adaptability. It offers flexible design options, suppor...
 | Author: the WordPress team
 | Author URI: https://wordpress.org
 |
 | Found By: Css Style In Homepage (Passive Detection)
 |
 | Version: 1.3 (80% confidence)
 | Found By: Style (Passive Detection)
 |  - http://localhost:8080/wp-content/themes/twentytwentyfive/style.css?ver=1.3, Match: 'Version: 1.3'
```

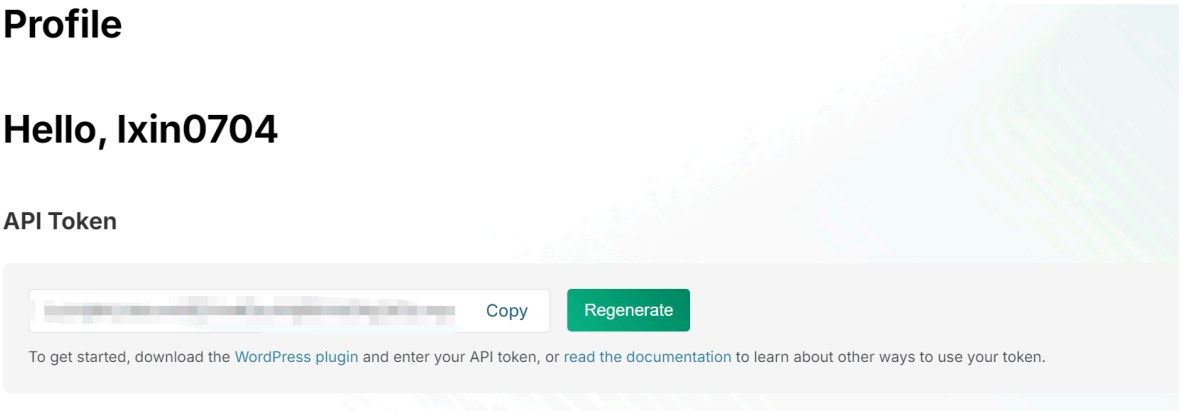
7. Caso de uso 5 — Uso de la API de WPScan

Objetivo

Obtener información completa y actualizada de vulnerabilidades reales.

Preparación

Registrar API token en wpscan.com



Comando

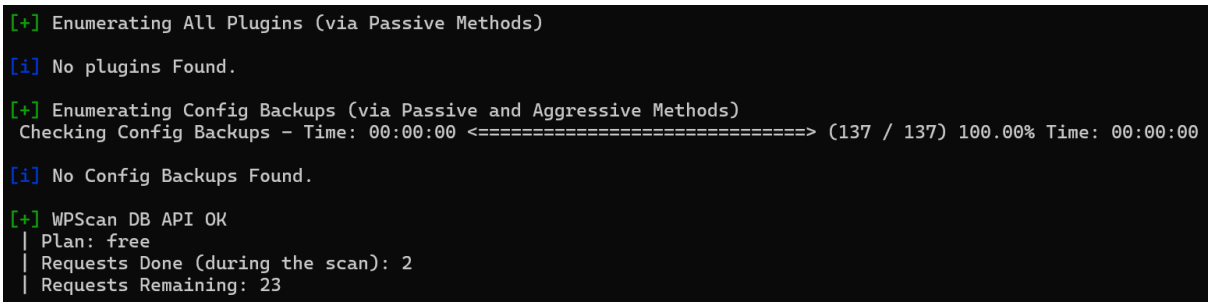
- wpscan --url https://victima.local --api-token TU_TOKEN

```
ub@forense-ai:~$ wpscan --url http://localhost:8080 --api-token Lz
```

WPScan DB API OK: Nos hemos conectado correctamente a la base de datos oficial de vulnerabilidades

Plan: free: Actualmente funcionando en modo de auditoría gratuita
Requests Done: Indica que se han enviado consultas a la base de datos de vulnerabilidades
Requests Remaining: La API está procesando.

Sin embargo, como el complemento sigue sin estar instalado, seguimos sin poder ver las vulnerabilidades reales y otra información.



8. Caso de uso 6 — Detección de configuraciones inseguras

Objetivo

Identificar malas prácticas sin atacar directamente

XML-RPC activo

1. La funcionalidad XML-RPC permanece activa, lo que supone un riesgo de ataques de fuerza bruta y DDoS. La funcionalidad externa WP-Cron está habilitada, lo que supone un riesgo de activación maliciosa que puede provocar el agotamiento de los recursos.

Directorios listables

2. Se puede acceder directamente al archivo confidencial readme.html, lo que revela información fundamental del sitio.

Versiones expuestas

3. La versión 6.8.3 del núcleo de WordPress está expuesta públicamente y ha llegado al final de su vida útil.

Backups accesibles

4. No se detectaron problemas relacionados con el listado de directorios ni con la accesibilidad de los archivos de copia de seguridad durante el análisis.
Sin embargo, estos dos elementos deben incorporarse a las medidas rutinarias de refuerzo de la seguridad y a las inspecciones periódicas, ya que representan riesgos de configuración operativa de alta frecuencia que se encuentran habitualmente en los sitios de WordPress. Estas vulnerabilidades son muy susceptibles de reaparecer debido a cambios en la configuración o actualizaciones de versión