

# PROYECTO FINAL CIBERSEGURIDAD



Institución Educativa: 4geeksAcademy

- Nombre del estudiante: Gedson Honorio da Silva
- Nombre del docente: Rubens Alonso Cebrián
- Fecha de entrega: 06-06-2025
- Curso: Ciberseguridad:

# FASE 1: DETECCIÓN Y CORRECCIÓN DE VULNERABILIDADES EXPLOTADAS



Este proyecto final explora los principios, técnicas y herramientas fundamentales en el ámbito de la ciberseguridad, abordando aspectos como la protección de datos, la gestión de riesgos, y el diseño de sistemas seguros para enfrentar los desafíos actuales de seguridad digital.

El proyecto será presentado en 3 fases:

**Fase 1 - Corrección de un hackeo** Durante la primera fase, debes realizar un análisis forense del incidente, identificar las vulnerabilidades explotadas por el atacante, y bloquear el exploit para evitar que el ataque escale.

**Fase 2 - Detección y corrección de una nueva vulnerabilidad** En la segunda fase, escanea el sistema en busca de una vulnerabilidad adicional, diferente a la explotada anteriormente. Tras detectarla, explota la vulnerabilidad de manera controlada para entender su impacto, escalar sus privilegios, corregirla, y crear un informe que explique todo el proceso.

**Fase 3 - Plan de respuesta a incidentes y certificación** La fase final consiste en diseñar un plan de respuesta a incidentes basado en las mejores prácticas de la industria, como las recomendaciones del NIST. Como parte de este ejercicio, debes desarrollar un Sistema de Gestión de Seguridad de la Información (SGSI) conforme a la norma ISO 27001, que incluya medidas para prevenir la fuga de datos mediante políticas de prevención de pérdida de datos (DLP).

## INDICE

1. INTRODUCCIÓN - página 5
  2. OBJETIVO Y ALCANCE- página 5
  3. HERRAMIENTAS UTILIZADAS. - página 6
  4. DETECCIÓN DE VULNERABILIDADES - página 6
    - 4.1 Análisis de logs -página 6
    - 4.2 Análisis en MySQL - página 8
    - 4.3 Sitio WordPress- página 8
  5. ESCANEEO DE ROOTKIT Y MALWARE - página 12
    - 5.1 Instalación de chkrootkit y rkhunter -página 12
  6. SERVICIO FTP - página 13
  7. CORRECCIÓN DE VULNERABILIDADES- página 14
    - 7.1 SSH-página14
    - 7.2 FTP-página 15
    - 7.3 MySql-página 15,16 y 17
    - 7.4 WordPress-página 18 y 19
  8. FASE 2: DETECCIÓN DE UNA NUEVA VULNERABILIDAD – pagina 22
    - 8.1 Introducción- página 22
    - 8.2 Objetivo y Alcance: - página 22
    - 8.3 Herramientas y Técnicas utilizadas – página 22
  9. DETECCIÓN DE VULNERABILIDAD CON NMAP – pagina 23
    - 9.1 Vulnerabilidad Encontradas- pagina 23 y 24
  10. VULNERABILIDAD DETECTADA- pagina 24 y 25
    - 10.1 Puerto 21/FTP- página 25
    - 10.2 EXPLOTACIÓN VULNERABILIDAD VSFTPD 3.0.3- pagina 25,26 y 27
    - 10.3 MITIGACIÓN FTP – pagina 28 y 29
    - 10.4 MITIGACIÓN SSH- página 29
- Fase 3: Plan de respuesta de incidentes-página 30
1. Identificación - pagina30
  2. Contención - página 30
  3. Erradicación - página 30
  4. Recuperación -página 31
  5. Lecciones aprendidas – página 31
  6. Sistema de Gestión de Seguridad de la Información (SGSI) – página 31
    1. Análisis de riesgos -página 31
    2. Definición de políticas de seguridad – página 31
    3. Protección de datos -página 32
    4. Planes de acción – página 32
    5. Conclusión -página 32

.

## 1. INTRODUCCIÓN

El informe generará análisis del servidor que ha sufrido modificaciones, en el informe debemos detallar las vulnerabilidades que el atacante utilizó para acceder al sistema.

## 2. OBJETIVO Y ALCANCE

El objetivo es identificar las vulnerabilidades y servicios comprometidos en la máquina hackeada Debian-

Este informe se realizó en entorno controlado, he utilizado VirtualBox, Kali y la máquina hackeada Debian.

## 3. HERRAMIENTAS Y TÉCNICAS UTILIZADAS

He utilizado para recopilación de informaciones NMAP, una herramienta de código abierto utilizada por profesionales de ciberseguridad y para auditorias, tiene como características escanear puertos, versiones y sistemas operativos.

Kali Linux, una distribución basada en Debian diseñada para auditorias de seguridad que incluye herramientas muy importantes y con un papel crucial para realizar análisis y pruebas de penetración, en este caso hemos utilizado comandos con scripts que facilitaron la detección de vulnerabilidades en la máquina Debian.

Wireshark es una herramienta clave en ciberseguridad, permite capturar y examinar los datos que circulan en una red en tiempo real, puede ser de gran utilidad para identificar patrones sospechosos o analizar las conexiones realizadas por el atacante.

## 4. IDENTIFICACIÓN DE VULNERABILIDADES

### 4.1 Análisis de logs

Para verificar los logs he utilizado el comando `cd /var/log/`

He utilizado el comando `ls` (listar)

Ahora he utilizado comando `cat README` para abrir el archivo y acceder a las informaciones

```
debian@debian:/var/log$ cat README
You are looking for the traditional text log files in /var/log, and they are gone?

Here's an explanation on what's going on:

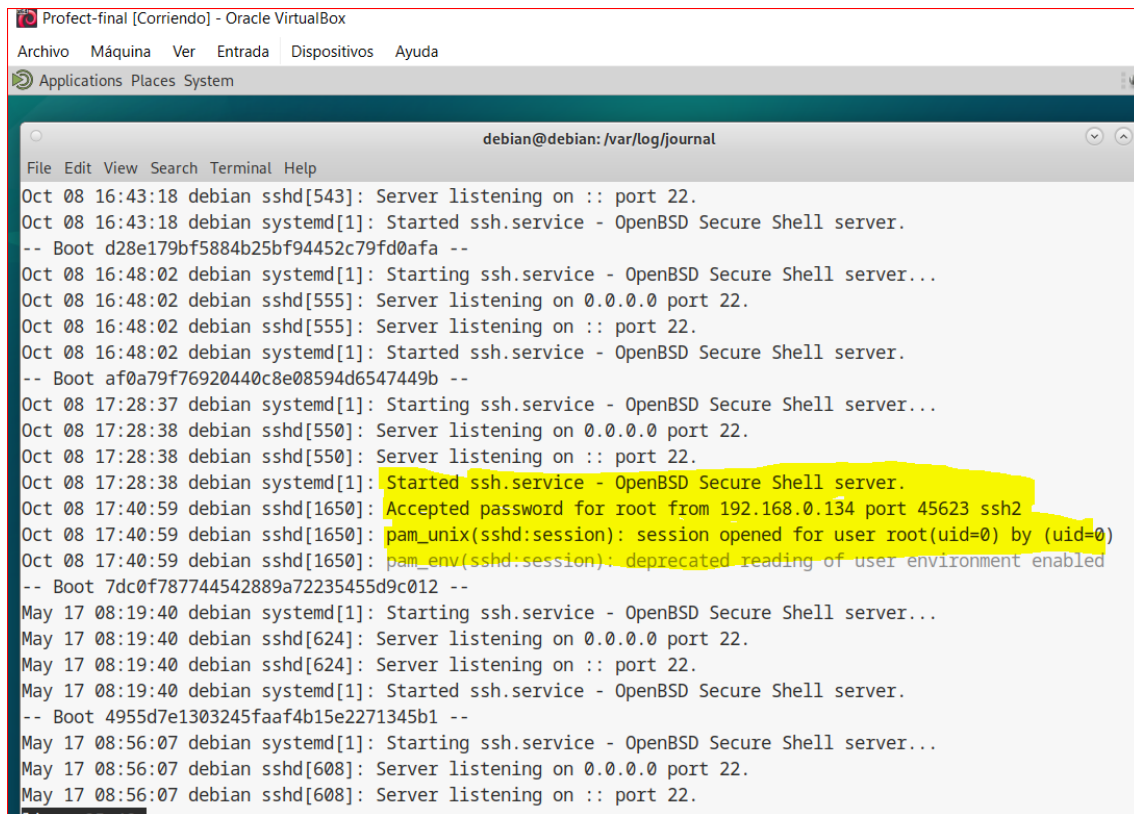
You are running a systemd-based OS where traditional syslog has been replaced with the Journal. The journal stores the same (and more) information as classic syslog. To make use of the journal and access the collected log data simply invoke "journalctl", which will output the logs in the identical text-based format the syslog files in /var/log used to be. For further details, please refer to journalctl(1).

Alternatively, consider installing one of the traditional syslog implementations available for your distribution, which will generate the classic log files for you. Syslog implementations such as syslog-ng or rsyslog may be installed side-by-side with the journal and will continue to function the way they always did.

Thank you!

Further reading:
    man:journalctl(1)
    man:systemd-journald.service(8)
    man:journald.conf(5)
```

Utilizando el comando **journalctl -u ssh** para filtrar un servicio específico.



```
debian@debian: /var/log/journal
File Edit View Search Terminal Help
Oct 08 16:43:18 debian sshd[543]: Server listening on :: port 22.
Oct 08 16:43:18 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
-- Boot d28e179bf5884b25bf94452c79fd0afa --
Oct 08 16:48:02 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Oct 08 16:48:02 debian sshd[555]: Server listening on 0.0.0.0 port 22.
Oct 08 16:48:02 debian sshd[555]: Server listening on :: port 22.
Oct 08 16:48:02 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
-- Boot af0a79f76920440c8e08594d6547449b --
Oct 08 17:28:37 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Oct 08 17:28:38 debian sshd[550]: Server listening on 0.0.0.0 port 22.
Oct 08 17:28:38 debian sshd[550]: Server listening on :: port 22.
Oct 08 17:28:38 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
Oct 08 17:40:59 debian sshd[1650]: Accepted password for root from 192.168.0.134 port 45623 ssh2
Oct 08 17:40:59 debian sshd[1650]: pam_unix(sshd:session): session opened for user root(uid=0) by (uid=0)
Oct 08 17:40:59 debian sshd[1650]: pam_env(sshd:session): deprecated reading of user environment enabled
-- Boot 7dc0f787744542889a72235455d9c012 --
May 17 08:19:40 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
May 17 08:19:40 debian sshd[624]: Server listening on 0.0.0.0 port 22.
May 17 08:19:40 debian sshd[624]: Server listening on :: port 22.
May 17 08:19:40 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
-- Boot 4955d7e1303245faaf4b15e2271345b1 --
May 17 08:56:07 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
May 17 08:56:07 debian sshd[608]: Server listening on 0.0.0.0 port 22.
May 17 08:56:07 debian sshd[608]: Server listening on :: port 22.
```

## Reporte del Acceso no Autorizado.

Fecha del incidente 08 de Octubre

Horario: 17:40:59s

IP de la máquina atacante: 192.168.0.134

Puerto 45623

Servicio SSH

Resumen:

El día 08 de Octubre, se detectó el acceso no autorizado al servidor, el atacante se ha aprovechado de la vulnerabilidad del puerto 45623/SSH y pudo acceder al sistema como usuario con privilegios de administrador utilizando las credenciales comprometidas.



## 4.2 Análisis en MySQL

Buscar dentro del sistema los rastros y posibles cambios que ha hecho el atacante, identificar es el primer paso para evitar fugas de datos, archivos, usuarios, etc.

He utilizado el comando **SELECT user, host from mysql.user** para listar usuarios de la base de datos,

```
MariaDB [(none)]> SELECT user, host from mysql.user;
+-----+-----+
| User          | Host          |
+-----+-----+
| mariadb.sys   | localhost     |
| mysql         | localhost     |
| root          | localhost     |
| user          | localhost     |
| wordpressuser | localhost     |
+-----+-----+
```

El comando **SHOW GRANTS FOR "wordpressuser" @"localhost"**; permite ver los permisos del usuario.

```
MariaDB [(none)]> SHOW GRANTS FOR "wordpressuser"@"localhost";
+-----+
---+
| Grants for wordpressuser@localhost
|
+-----+
---+
| GRANT USAGE ON *.* TO `wordpressuser`@`localhost` IDENTIFIED BY PASSWORD '*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9'
|
| GRANT ALL PRIVILEGES ON `wordpress`.* TO `wordpressuser`@`localhost`
|
+-----+
---+
2 rows in set (0.000 sec)
```

Plugin indica el método de autenticación utilizado, que pueden ser:

**mysql\_native\_password**: contraseñas cifradas con un hash basado en sha-1

**Authentication\_string**: contiene el hash de cifrado de la contraseña, si está vacío significa que el usuario no tiene contraseña configurada, un riesgo gravísimo de seguridad



## Reporte de Vulnerabilidad:

### Introducción:

Este privilegio **GRANT ALL PRIVILEGES**, puede otorgar sus propios privilegios a otros usuarios, y podría ser explotado para escalar privilegios.

### Impacto:

El usuario puede realizar cualquier operación en la base de datos, modificar, crear, cambiar crear usuarios con privilegios adicionales.

```
MariaDB [(none)]> SELECT user, host, plugin, authentication_string FROM mysql.user;
```

User	Host	plugin	authentication_string
mariadb.sys	localhost	mysql_native_password	
root	localhost	mysql_native_password	*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9
mysql	localhost	mysql_native_password	invalid
wordpressuser	localhost	mysql_native_password	*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9
user	localhost	mysql_native_password	*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19

5 rows in set (0.002 sec)

Authentication\_string almacena el hash de la contraseña del usuario, si está invalid es porque no tiene configurada la contraseña y puede permitir que el atacante entre con privilegios del usuario, normal o root, es un riesgo para seguridad.

### 4.3 Sitio WordPress.

He utilizado el comando `ls -l /var/www/html/`

Me di cuenta que los archivos tienen todos los permisos, lectura, escritura y ejecución (rwx), usuario, grupo y otros usuarios puede acceder, leer, escribir y modificar, es un riesgo de seguridad.

El archivo `wp-config.php` es un archivo que contiene las credenciales del sitio wordpress, y posee todos los permisos, genera un grave riesgo a seguridad.

```
debian@debian:~$ ls -l /var/www/html/wp-config.php
-rwxrwxrwx 1 www-data www-data 3083 May 17 10:05 /var/www/html/wp-config.php
```

```
debian@debian: ~  
File Edit View Search Terminal Help  
debian@debian:~$ ls -l /var/www/html/  
total 248  
-rwxrwxrwx 1 www-data www-data 10701 Sep 30 2024 index.html  
-rwxrwxrwx 1 www-data www-data 405 Feb 6 2020 index.php  
-rwxrwxrwx 1 www-data www-data 19903 May 17 09:39 license.txt  
-rwxrwxrwx 1 www-data www-data 7425 May 17 09:39 readme.html  
-rwxrwxrwx 1 www-data www-data 7387 Feb 13 2024 wp-activate.php  
drwxrwxrwx 9 www-data www-data 4096 Sep 10 2024 wp-admin  
-rwxrwxrwx 1 www-data www-data 351 Feb 6 2020 wp-blog-header.php  
-rwxrwxrwx 1 www-data www-data 2323 Jun 14 2023 wp-comments-post.php  
-rwxrwxrwx 1 www-data www-data 3083 May 17 10:05 wp-config.php  
-rwxrwxrwx 1 www-data www-data 3336 May 17 09:39 wp-config-sample.php  
drwxrwxrwx 6 www-data www-data 4096 May 17 11:52 wp-content  
-rwxrwxrwx 1 www-data www-data 5617 May 17 09:39 wp-cron.php  
drwxrwxrwx 30 www-data www-data 12288 May 17 09:39 wp-includes  
-rwxrwxrwx 1 www-data www-data 2502 Nov 26 2022 wp-links-opml.php  
-rwxrwxrwx 1 www-data www-data 3937 Mar 11 2024 wp-load.php  
-rwxrwxrwx 1 www-data www-data 51414 May 17 09:39 wp-login.php  
-rwxrwxrwx 1 www-data www-data 8727 May 17 09:39 wp-mail.php  
-rwxrwxrwx 1 www-data www-data 30081 May 17 09:39 wp-settings.php  
-rwxrwxrwx 1 www-data www-data 34516 May 17 09:39 wp-signup.php  
-rwxrwxrwx 1 www-data www-data 5102 May 17 09:39 wp-trackback.php  
-rwxrwxrwx 1 www-data www-data 3205 May 17 09:39 xmlrpc.php  
debian@debian:~$
```

El servidor apache lo consultamos con el comando `sudo systemctl status apache2`, está activo.

```

debian@debian:~$ sudo systemctl status apache2
[sudo] password for debian:
• apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-05-19 04:19:40 EDT; 9h ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 576 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 689 (apache2)
    Tasks: 6 (limit: 2284)
   Memory: 42.3M
      CPU: 289ms
   CGroup: /system.slice/apache2.service
           └─689 /usr/sbin/apache2 -k start
             └─709 /usr/sbin/apache2 -k start
               └─710 /usr/sbin/apache2 -k start
                 └─711 /usr/sbin/apache2 -k start
                   └─712 /usr/sbin/apache2 -k start
                     └─713 /usr/sbin/apache2 -k start

May 19 04:19:40 debian systemd[1]: Starting apache2.service - The Apache HTTP Server:
May 19 04:19:40 debian systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-10/10 (END)

```

Utilizando el comando `sudo /etc/apache2/apache.conf`

```

GNU nano 7.2 /etc/apache2/apache2.conf
# your system is serving content from a sub-directory in /srv you must allow
# access here, or in any related virtual host.
<Directory />
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory /usr/share>
    AllowOverride None
    Options Indexes FollowSymLinks
    Require all granted
</Directory>

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

```

^G Help    ^O Write Out    ^W Where Is    ^K Cut    ^T Execute    ^C Location  
 ^X Exit    ^R Read File    ^\ Replace    ^U Paste    ^J Justify    ^\_ Go To Line

La configuración permite el atacante listar los Directorios

## 5. ESCANEOS DE ROOTKIT Y MALWARE

Las herramientas chkrootkit y rkhunter pueden detectar anomalías en el sistema, revisan en busca de rootkits y posibles amenazas que puedan comprometer la integridad del servidor.

## 5.1 Instalación de herramientas chkrootkit y rkhunter

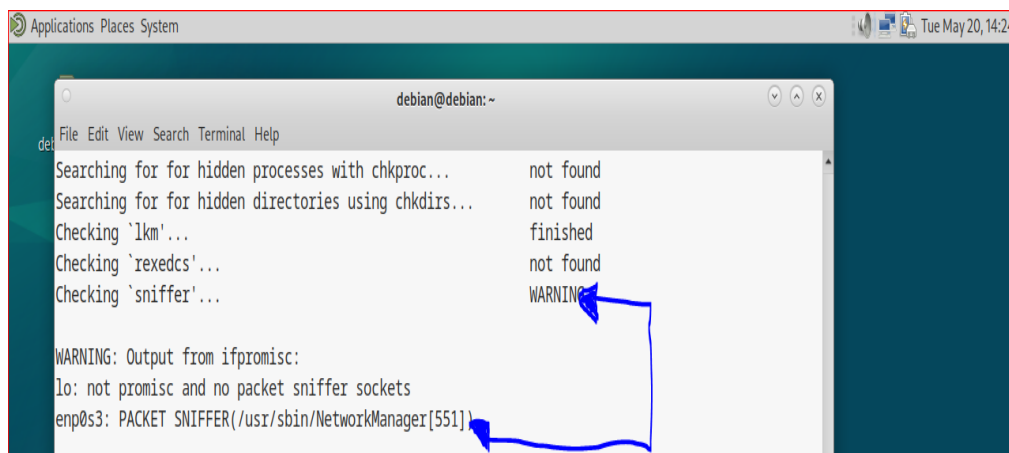
**sudo apt install chkrootkit**

**sudo apt install rkhunter**

Con la herramienta chrootkit (posee las mismas características de rkhunter)

```
debian@debian:~$ sudo chkrootkit
ROOTDIR is '/'
Checking `amd'... not found
Checking `basename'... not infected
Checking `biff'... not found
Checking `chfn'... not infected
Checking `chsh'... not infected
```

Después de escanear, aparece warning (advertencia)



Warning no suelen detener la ejecución de un programa o proceso, pero alertan los administradores sobre posibles problemas.

**lo** es la interfaz loopback (localhost) que permite un sistema comunicarse consigo mismo.

**Not promisc** indica que la interface **lo** no esta en modo promiscuo. El modo promiscuo permite que una interfaz de red capture todos los paquetes que pasa por ella, no solo aquellos destinados a su propia dirección MAC.

**Packet sniffer** indica que hay una aplicación configurada para capturar paquetes en esta interfaz **/usr/sbin/**.

**NetworkManager** es un gestor de redes para Linux que maneja conexiones de red y configura interfaces de red.

El mensaje dice que NetworkManager está funcionando correctamente, captura los paquetes en la interface de red enp0s3.

### Escaneo con rkhunter:

rkhunter Permite buscar rootkit, puertas traseras, exploits, etc.

```
/usr/bin/whereis [ OK ]
/usr/bin/which [ OK ]
/usr/bin/who [ OK ]
/usr/bin/whoami [ OK ]
/usr/bin/numfmt [ OK ]
/usr/bin/kmod [ OK ]
/usr/bin/systemd [ OK ]
/usr/bin/systemctl [ OK ]
/usr/bin/gawk [ OK ]
/usr/bin/lwp-request [ Warning ]
/usr/bin/mail.mailutils [ OK ]
/usr/bin/dash [ OK ]
/usr/bin/x86_64-linux-gnu-size [ OK ]
/usr/bin/x86_64-linux-gnu-strings [ OK ]
/usr/bin/inetutils-telnet [ OK ]
/usr/bin/which.debianutils [ OK ]
/usr/lib/systemd/systemd [ OK ]
```

El [warning](#) está relacionado con la herramienta (LWP- Library for WWW in Pearl) es un comando que hace parte de una biblioteca de perl para acceder a

recursos web y se utiliza para realizar solicitudes HTTP desde línea de comandos.

He utilizado `ls -l /usr/bin/lwp-request` para ver los permisos, y los permisos son los recomendados, no vi vulnerabilidad que pueda traer riesgos, debe estar actualizada.

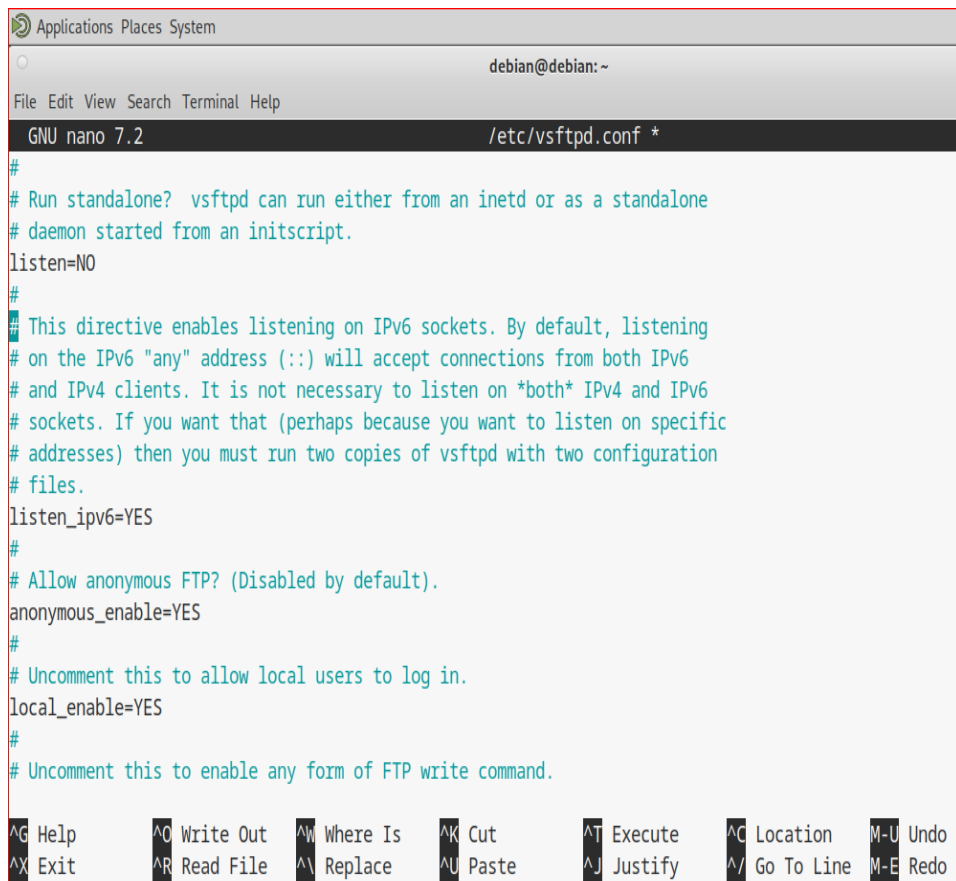
```
debian@debian:~$ ls -l /usr/bin/lwp-request  
-rwxr-xr-x 1 root root 16202 Mar  1 2023 /usr/bin/lwp-request
```

He utilizado el comando `lwp-request` para ver la versión, y la versión es la más reciente incluye varias mejoras y correcciones de errores, y no veo algo que pueda comprometer la seguridad.



## 6. SERVICIO FTP (File Transfer Protocol)

He utilizado el comando `sudo nano /etc/vsftpd.conf` para averiguar la configuración, y la opción Anonymous está habilitada, y permite otros usuarios acceder al sistema sin autenticación y contraseña.



```
Applications Places System
debian@debian: ~
File Edit View Search Terminal Help
GNU nano 7.2 /etc/vsftpd.conf *
#
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo
```

## 7. CORRECCIÓN DE VULNERABILIDADES

### 7.1 SSH

En el servicio SSH haré la corrección con medidas de seguridad para evitar pérdidas de datos y accesos no autorizados.

- `sudo apt install ufw` (instalación del firewall UFW)
- `sudo ufw status` (para ver si está activo)

```
debian@debian:~$ sudo systemctl status ufw
○ ufw.service - Uncomplicated firewall
   Loaded: loaded (/lib/systemd/system/ufw.service; enabled; preset: enabled)
   Active: inactive (dead)
     Docs: man:ufw(8)
```

- `sudo ufw deny from 192.168.1.134` (comando para bloquear la IP del atacante).
- `sudo ufw status` (para consulta de las reglas establecidas).

```
debian@debian:~$ sudo ufw deny from 192.168.1.134
Rule added
debian@debian:~$ sudo ufw status
Status: active

To Action From
--
Anywhere DENY 192.168.1.134
```

**PasswordAuthentication** acceder al sistema solamente con clave pública, evitando el uso de contraseñas que permiten ataques de fuerza bruta, phishing, etc.

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
```

**PermitRootLogin** deshabilitada no permite inicio de sesión como usuario root, primero accede como usuario normal y después utilizar `sudo` para obtener privilegios de root.

**MaxTries** 3 el usuario tiene 3 intentos para inicio de sesión, después se bloquea.

```
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
MaxAuthTries 3
```

## 7. Mitigación en el servicio FTP

- En el servicio ftp está habilitada Anonymous:
- Anonymous permite el acceso sin credenciales y contraseñas, debemos configurar como no y IPV6 si no lo estás usando también.

```
# files.
listen_ipv6=NO
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
```

- Intento de conexión después de los cambios de corrección, la conexión ha sido rechazada.

```
➔ ~ ftp 192.168.1.70
ftp: Can't connect to `192.168.1.70:21': Network is unreachable
ftp: Can't connect to `192.168.1.70:ftp'
ftp>
```

El protocolo IPV6 deshabilítalo, si no está en uso deberías desactivarlo y disminuir el área de ataque

## 7.2 Usuario mysql

El usuario mysql tiene los privilegios elevados, el usuario lo eliminaremos:

```
debian@debian:~$ sudo mysql -e "DROP USER 'mysql'@'localhost';"
debian@debian:~$ sudo mysql -e "SELECT user, host, password FROM mysql.user;"
```

User	Host	Password
mariadb.sys	localhost	
root	localhost	*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9
wordpressuser	localhost	*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9
user	localhost	*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19

Configurar el servidor mysql para evitar accesos remotos

Utilizamos el comando

`sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf`

```

Applications Places System
debian@debian: /etc/mysql/mariadb.conf.d
File Edit View Search Terminal Help
GNU nano 7.2 50-server.cnf
#user = mysql
pid-file = /run/mysqld/mysqld.pid
basedir = /usr
#datadir = /var/lib/mysql
#tmpdir = /tmp

# Broken reverse DNS slows down connections considerably and name resolve is
# safe to skip if there are no "host by domain name" access grants
#skip-name-resolve

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address = 127.0.0.1

#
# * Fine Tuning
#
#key_buffer_size = 128M
#max_allowed_packet = 1G

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C L
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justifv    ^/ G

```

bind-address = 127.0.0.1 hará con que mysql solamente escuche la interfaz de red local

**sudo service mysql restart** ( para reiniciar el servicio de mysql y los cambios surtan efecto.

Para comprobación he utilizado el comando

**sudo netstat -tlnp | grep mariadb**

```

debian@debian:~$ sudo netstat -tlnp | grep mariadb
tcp        0      0 127.0.0.1:3306        0.0.0.0:*        LISTEN      677/mariadb

```

A través del comando es posible ver que mariadb está escuchando la red local.

## Corrección de vulnerabilidad en el Servidor Apache2

En el servidor apache2 vamos a configurar para que el atacante no pueda acceder y listar los archivos y directorios.

```
GNU nano 7.2 /etc/apache2/apache2.conf
# your system is serving content from a sub-directory in /srv you must allow
# access here, or in any related virtual host.
<Directory />
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory /usr/share>
    AllowOverride None
    Options Indexes FollowSymLinks
    Require all granted
</Directory>

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

**sudo service apache2 restart** (para reiniciar el servidor y guardar los cambios)

### 7.3 Corrección de Vulnerabilidad WordPress

El archivo wp-config.php es un archivo que contiene datos importantes para el funcionamiento del sitio web, contiene las credenciales de la base de datos, y debemos utilizar el principio de menor privilegio, evitando acceso inapropiado y los riesgos para ciberseguridad.

con el comando ls -l vemos los permisos

sudo ls -l /var/www/html/wp-config.php

```

debian@debian: /var/www/html
File Edit View Search Terminal Help
-rwxrwxrwx 1 www-data www-data 3017 Sep 30 2024 wp-config.php
debian@debian:/var/www/html$ ls -l
total 244
-rwxrwxrwx 1 www-data www-data 10701 Sep 30 2024 index.html
-rwxrwxrwx 1 www-data www-data 405 Feb 6 2020 index.php
-rwxrwxrwx 1 www-data www-data 19915 Dec 31 2023 license.txt
-rwxrwxrwx 1 www-data www-data 7409 Jun 18 2024 readme.html
-rwxrwxrwx 1 www-data www-data 7387 Feb 13 2024 wp-activate.php
drwxrwxrwx 9 www-data www-data 4096 Sep 10 2024 wp-admin
-rwxrwxrwx 1 www-data www-data 351 Feb 6 2020 wp-blog-header.php
-rwxrwxrwx 1 www-data www-data 2323 Jun 14 2023 wp-comments-post.php
-rwxrwxrwx 1 www-data www-data 3017 Sep 30 2024 wp-config.php
drwxrwxrwx 5 www-data www-data 4096 Oct 8 2024 wp-content
-rwxrwxrwx 1 www-data www-data 5638 May 30 2023 wp-cron.php
drwxrwxrwx 30 www-data www-data 12288 Sep 10 2024 wp-includes
-rwxrwxrwx 1 www-data www-data 2502 Nov 26 2022 wp-links-opml.php
-rwxrwxrwx 1 www-data www-data 3937 Mar 11 2024 wp-load.php
-rwxrwxrwx 1 www-data www-data 51238 May 28 2024 wp-login.php
-rwxrwxrwx 1 www-data www-data 8525 Sep 16 2023 wp-mail.php
-rwxrwxrwx 1 www-data www-data 28774 Jul 9 2024 wp-settings.php
-rwxrwxrwx 1 www-data www-data 34385 Jun 19 2023 wp-signup.php
-rwxrwxrwx 1 www-data www-data 4885 Jun 22 2023 wp-trackback.php
-rwxrwxrwx 1 www-data www-data 3246 Mar 2 2024 xmlrpc.php
  
```

- Ahora utilizamos chmod para cambiar permisos
- con el comando chmod 600 wp-config.php cambiamos los permisos para que solamente el propietario tenga los permisos para leer, alterar y ejecutar.

```
debian@debian: /var/www/html
File Edit View Search Terminal Help
[sudo] password for debian:
debian@debian: /var/www/html$ ls -l
total 244
-rwxrwxrwx 1 www-data www-data 10701 Sep 30 2024 index.html
-rwxrwxrwx 1 www-data www-data 405 Feb 6 2020 index.php
-rwxrwxrwx 1 www-data www-data 19915 Dec 31 2023 license.txt
-rwxrwxrwx 1 www-data www-data 7409 Jun 18 2024 readme.html
-rwxrwxrwx 1 www-data www-data 7387 Feb 13 2024 wp-activate.php
drwxrwxrwx 9 www-data www-data 4096 Sep 10 2024 wp-admin
-rwxrwxrwx 1 www-data www-data 351 Feb 6 2020 wp-blog-header.php
-rwxrwxrwx 1 www-data www-data 2323 Jun 14 2023 wp-comments-post.php
-rw-r----- 1 www-data www-data 3017 Sep 30 2024 wp-config.php
drwxrwxrwx 5 www-data www-data 4096 Oct 8 2024 wp-content
-rwxrwxrwx 1 www-data www-data 5638 May 30 2023 wp-cron.php
drwxrwxrwx 30 www-data www-data 12288 Sep 10 2024 wp-includes
-rwxrwxrwx 1 www-data www-data 2502 Nov 26 2022 wp-links-opml.php
-rwxrwxrwx 1 www-data www-data 3937 Mar 11 2024 wp-load.php
-rwxrwxrwx 1 www-data www-data 51238 May 28 2024 wp-login.php
-rwxrwxrwx 1 www-data www-data 8525 Sep 16 2023 wp-mail.php
-rwxrwxrwx 1 www-data www-data 28774 Jul 9 2024 wp-settings.php
-rwxrwxrwx 1 www-data www-data 34385 Jun 19 2023 wp-signup.php
-rwxrwxrwx 1 www-data www-data 4885 Jun 22 2023 wp-trackback.php
-rwxrwxrwx 1 www-data www-data 3246 Mar 2 2024 xmlrpc.php
```



## 8. FASE 2: IDENTIFICACIÓN DE NUEVA VULNERABILIDAD

### 8.1 INTRODUCCIÓN

Realizar una búsqueda detallada de vulnerabilidades encontradas en la máquina hackeada, identificar y corregir, el ataque consiste atacar desde fuera explotando la vulnerabilidad en la máquina proporcionada Debian.

### 8.2 OBJETIVO Y ALCANCE:

El principal objetivo es detectar fallos de seguridad, tras detectar explotar de manera controlada y observar su impacto y generar un informe con las informaciones generadas, alcanzar los objetivos de explotación y corrección del sistema Debian.

### 8.3 HERRAMIENTAS Y TECNICAS UTILIZADAS

En la máquina Kali Linux, que es muy utilizada en ciberseguridad por tener herramientas de mucho valor para encontrar y corregir vulnerabilidades, vamos a utilizar NMAP que una herramienta de escaneo que permite ver los puertos, sistemas operativos, versiones y servicios, también es posible utilizar scripts para mejorar la calidad del escaneo.

Wireshark es una herramienta clave en ciberseguridad, permite capturar y examinar los datos que circulan en una red en tiempo real, puede ser de gran utilidad para identificar patrones sospechosos o analizar las conexiones realizadas por el atacante.

CVE- es un sistema internacional de identificación de vulnerabilidades en software y hardware, lo utilizaremos para consultas de las vulnerabilidades encontradas.

INCIBE- Instituto Nacional de Ciberseguridad en España, es una entidad publica española dedica a mejorar la seguridad de ciudadanos, empresas y administraciones, ofrece servicio de alerta, ofrece servicio de formación, alerta , respuestas a incidentes y concienciación.

## 9. DETECCIÓN DE VULNERABILIDAD CON NMAP

### 9.1 Vulnerabilidad Encontradas

El comando **sudo nmap -sV -O -A --script=vuln 192.168.1.51** genera informaciones de servicios, sistemas operativos, versiones y puertos.

The image shows two terminal windows. The left window displays the output of an Nmap scan on 192.168.1.51, identifying various CVEs and exploits. The right window shows the output of the 'ip a' command on a Debian system, displaying network interface details.

```

kali@kali:~$ sudo nmap -sV -O -A --script=vuln 192.168.1.51
Nmap done: 1 IP address (1 host up) scanned in 88.62 seconds
+ sudo nmap -sV -O -A --script=vuln 192.168.1.51
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-10 07:35 EDT
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 192.168.1.51
Host is up (0.00059s latency).
Not shown: 997 closed tcp ports (reset)
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
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
| vulners:
|   cpe:/a:openssh:openssh:9.2p1:
|     2C119FFA-EC68-5E14-AA44-354A2C38071A 10.0 https://vulners.com/githubexploit/2C119FFA-EC68-5E14-AA44-354A2C38071A *EXPLOIT*
|     CVE-2023-38408 9.8 https://vulners.com/cve/CVE-2023-38408
|     CVE-2023-28531 9.8 https://vulners.com/cve/CVE-2023-28531
|     8B190CDB-3E89-5631-9828-8064A1575823 9.8 https://vulners.com/githubexploit/8B190CDB-3E89-5631-9828-8064A1575823 *EXPLOIT*
|     BFC9C5AB-3968-5F3C-825E-E80B53794623 9.8 https://vulners.com/githubexploit/BFC9C5AB-3968-5F3C-825E-E80B53794623 *EXPLOIT*
|     BAD01159-548E-546E-AA87-20E89F3927EC 9.8 https://vulners.com/githubexploit/BAD01159-548E-546E-AA87-20E89F3927EC *EXPLOIT*
|     5E6968B4-08D6-57FA-BF6E-D9822190B27A 9.8 https://vulners.com/githubexploit/5E6968B4-08D6-57FA-BF6E-D9822190B27A *EXPLOIT*
|     33D623F7-98E8-5F75-80FA-B1AA66601340 9.8 https://vulners.com/githubexploit/33D623F7-98E8-5F75-80FA-B1AA66601340 *EXPLOIT*
|     22277290-6700-5C8F-8930-1EEAFD4B9FF0 9.8 https://vulners.com/githubexploit/22277290-6700-5C8F-8930-1EEAFD4B9FF0 *EXPLOIT*
|     0221525F-07F5-5790-912D-F4B9E201B587 9.8 https://vulners.com/githubexploit/0221525F-07F5-5790-912D-F4B9E201B587 *EXPLOIT*
|     PACKETSTORM:190587 8.1 https://vulners.com/packetstorm/PACKETSTORM:190587 *EXPLOIT*
|     PACKETSTORM:179290 8.1 https://vulners.com/packetstorm/PACKETSTORM:179290 *EXPLOIT*
|     FB2E9ED1-43D7-585C-A197-8D6628B20134 8.1 https://vulners.com/githubexploit/FB2E9ED1-43D7-585C-A197-8D6628B20134 *EXPLOIT*
|     FA3992CE-9C4C-5350-8134-177126E0B03F 8.1 https://vulners.com/githubexploit/FA3992CE-9C4C-5350-8134-177126E0B03F *EXPLOIT*
|     F8981437-1287-5869-93F1-6570F81DCE59 8.1 https://vulners.com/githubexploit/F8981437-1287-5869-93F1-6570F81DCE59 *EXPLOIT*

debian@debian:~$ 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 noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:29:87:0a brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.51/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
        valid_lft 43124sec preferred_lft 43124sec
    inet6 fe80::a00:27ff:fe29:870a/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
  
```

```

→ ~ sudo nmap -sV -A --script=vuln 192.168.1.51 -p 21
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-10 09:04 EDT
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 192.168.1.51
Host is up (0.00071s 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
MAC Address: 08:00:27:29:87:0A (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
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: OS: Unix

TRACEROUTE
HOP RTT      ADDRESS
1   0.71 ms  192.168.1.51

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 37.80 seconds

```

## 10. VULNERABILIDAD DETECTADA

### 10.1 Puerto 21/FTP

## Vulnerabilidad en VSFTPD (CVE-2021-30047)

Gravedad CVSS v3.1: ALTA

Tipo: No Disponible / Otro tipo

Fecha de publicación: 22/08/2023

Última modificación: 25/08/2023

## Descripción

VSFTPD 3.0.3 permite a los atacantes provocar una denegación de servicio debido al número limitado de conexiones permitidas.

## Impacto

Vector 3.x CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H

Puntuación base 3.x 7.50

Gravedad 3.x ALTA

La versión vsftpd está desactualizada, permite al atacante ataque de denegación de servicio (DDoS)

Consecuencia: permite al atacante atacar el servidor enviando peticiones en grande cantidad y el servidor no tendrá tiempo para responder, causando bloqueo del servidor o

## 10.2 EXPLOTACIÓN VULNERABILIDAD VSFTPD 3.0.3

Como vamos a realizar un ataque DDoS al servidor ftp, instalaremos en la máquina Debian la herramienta wireshark, esta herramienta nos permite visualizar los paquetes y el trafico de hacia la máquina objetivo.

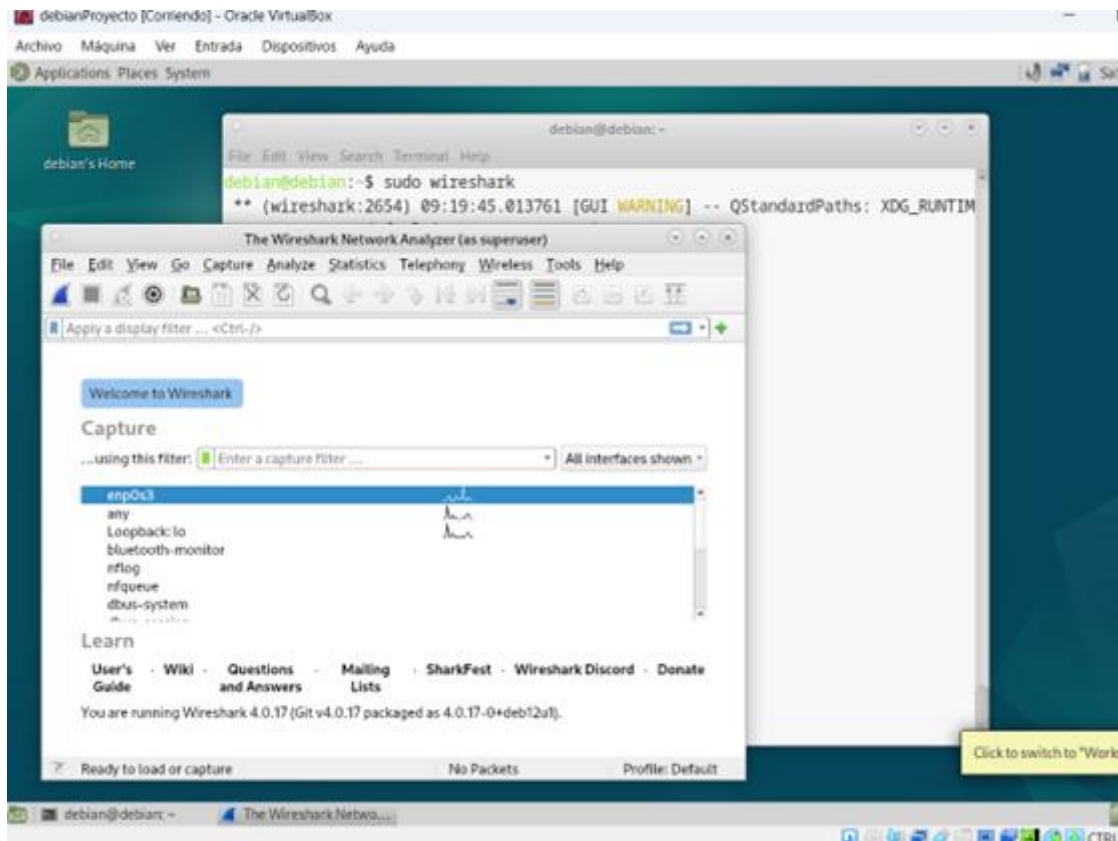
En Debian **sudo apt install wireshark**

```
debian@debian:~$ sudo apt install wireshark
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  linux-image-6.1.0-22-amd64
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libbcg729-0 libc-ares2 libdouble-conversion3 liblua5.2-0 libmd4c0
  libminizip1 libpcre2-16-0 libqt5core5a libqt5dbus5 libqt5gui5
  libqt5multimedia5 libqt5multimedia5-plugins libqt5multimediagsttools5
  libqt5multimediawidgets5 libqt5network5 libqt5printsupport5 libqt5qml5
```

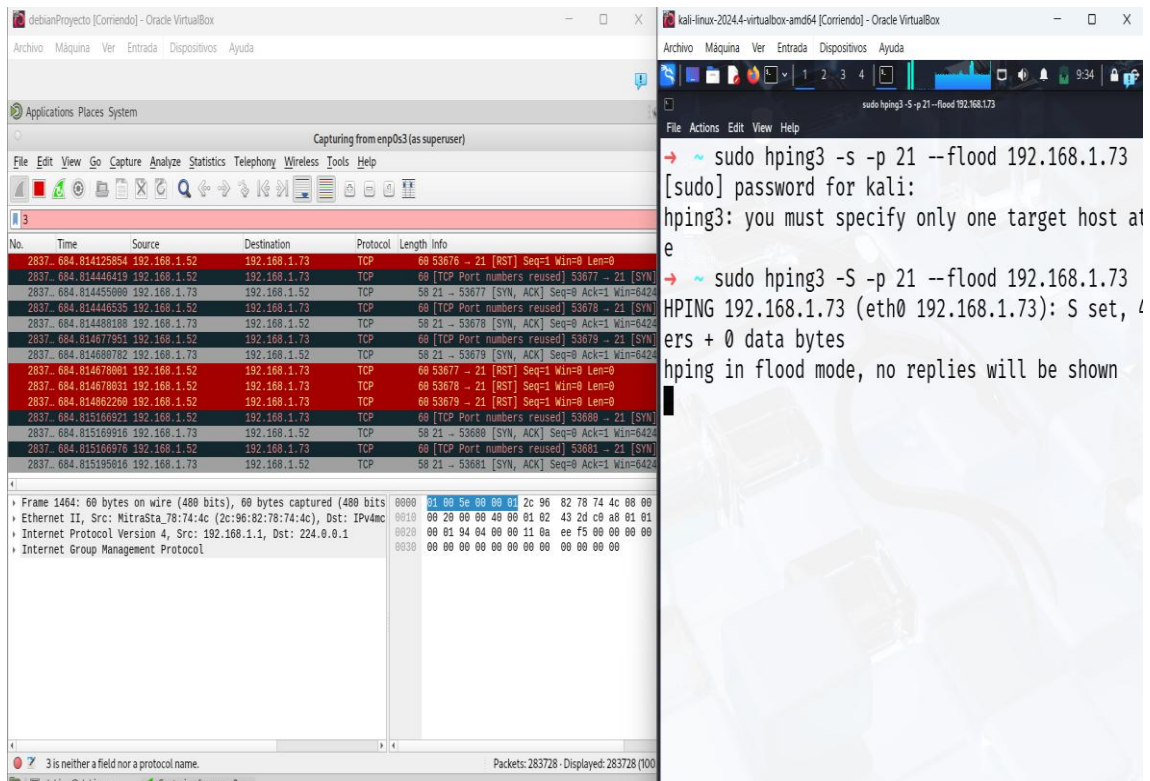
**sudo apt list wireshark** para verificar si está instalado

```
debian@debian:~$ sudo apt list wireshark
Listing... Done
wireshark/stable,now 4.0.17-0+deb12u1 amd64 [installed]
N: There is 1 additional version. Please use the '-a' switch to see it
```

sudo wireshark (para abrir wireshark)



El ataque se ha realizado con éxito, la maquina Debian se ha bloqueada después del ataque y el sistema ya no funciona correctamente.





Aquí podemos ver la cantidad de paquetes enviados a maquina atacada(Debian), causando el bloqueo del sistema.

```
→ ~ sudo hping3 -S -p 21 --flood 192.168.1.73
HPING 192.168.1.73 (eth0 192.168.1.73): S set,
ers + 0 data bytes
hping in flood mode, no replies will be shown
^C
— 192.168.1.73 hping statistic —
23607509 packets transmitted, 0 packets receive
packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

### 10.3 MITIGACIÓN FTP

He utilizado el firewall UFW para rechazar la IP utilizada para acceder a nuestro servidor FTP.

Sudo apt install UFW (instalación del firewall UFW)

```
debian@debian:~$ sudo ufw deny from 192.168.1.134
Rule added
debian@debian:~$ sudo ufw status
Status: active
```

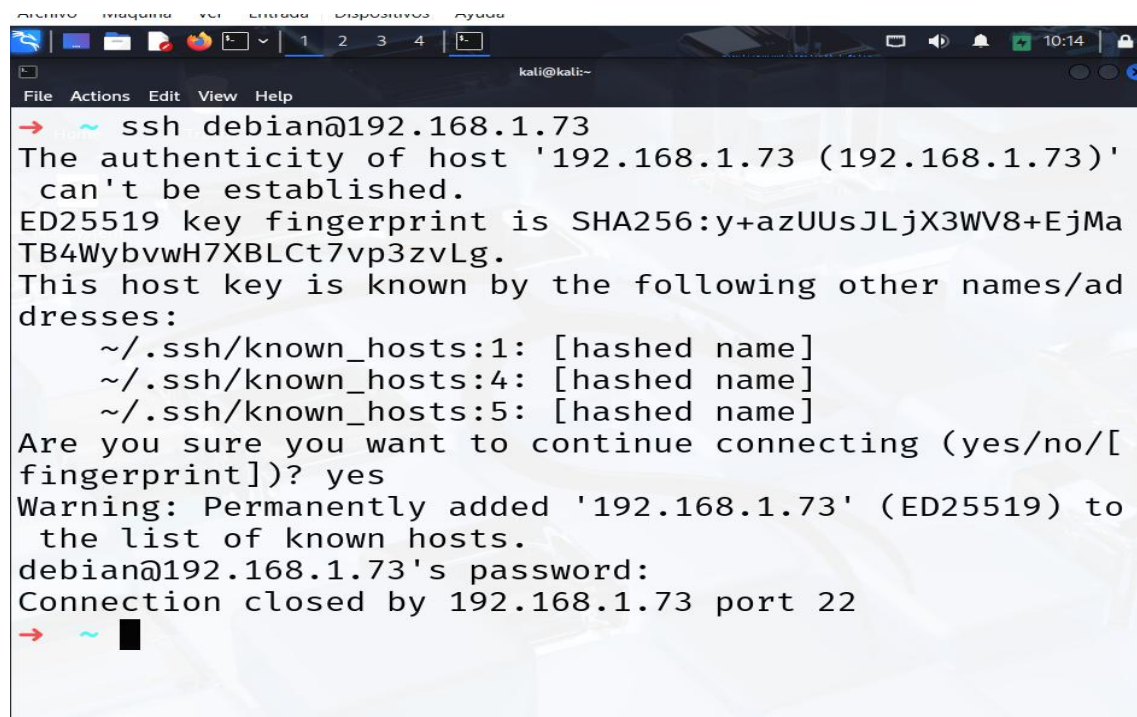
To	Action	From
--	-----	----
Anywhere	DENY	192.168.1.134

A través del firewall vamos limitar las conexiones por IP a 3 y también limitar la cantidad de conexiones a 6 cada 60 segundos

```
debian@debian:~$ sudo iptables -A INPUT -p tcp --dport 21 -m connlimit --connlimit-above 3 -j DROP
debian@debian:~$ sudo iptables -A INPUT -p tcp --syn --dport 21 -m recent --set --name FTP
debian@debian:~$ sudo iptables -A INPUT -p tcp --syn --dport 21 -m recent --update --seconds 60 --hitcount 5 --name FTP
-j DROP
```

## 10.4 MITIGACIÓN SSH

La versión SSH está corregida, solamente podrá acceder a través de claves de autenticación SSH



```
kali@kali:~$ ssh debian@192.168.1.73
The authenticity of host '192.168.1.73 (192.168.1.73)'
can't be established.
ED25519 key fingerprint is SHA256:y+azUUsJLjX3WV8+EjMa
TB4WybvW7XBLct7vp3zvLg.
This host key is known by the following other names/ad
dresses:
  ~/.ssh/known_hosts:1: [hashed name]
  ~/.ssh/known_hosts:4: [hashed name]
  ~/.ssh/known_hosts:5: [hashed name]
Are you sure you want to continue connecting (yes/no/[
fingerprint])? yes
Warning: Permanently added '192.168.1.73' (ED25519) to
the list of known hosts.
debian@192.168.1.73's password:
Connection closed by 192.168.1.73 port 22
```



# Fase 3: Plan de respuesta de incidentes

Objetivo: Diseñar un plan de respuesta a incidentes basado en las mejores prácticas y desarrollar un SGSI conforme a la norma ISO 27001

## Introducción

El manejo adecuado de los incidentes de seguridad es esencial para proteger los activos de información de una organización. Este documento presenta un plan de respuesta a incidentes basado en la guía del NIST SP 800-61, así como un Sistema de Gestión de Seguridad de la Información (SGSI) conforme a la norma ISO 27001 para garantizar la seguridad y continuidad de las operaciones.

## Plan de respuesta a incidentes

Basado en la guía NIST SP 800-61, el plan de respuesta a incidentes se divide en las siguientes fases: identificación, contención, erradicación, recuperación y lecciones aprendidas.

### 1. Identificación

La organización debe:

- Implementar sistemas de monitoreo para detectar actividades sospechosas y anomalías.
- Establecer un equipo de respuesta a incidentes (IRT) capacitado para analizar y clasificar los eventos.
- Utilizar herramientas como Wireshark para analizar tráfico de red y detectar patrones relacionados con ataques.
- Registrar y documentar cada evento para su análisis detallado.

### 2. Contención

Para evitar la propagación del ataque:

- Aplicar reglas de firewall (como UFW) para bloquear direcciones IP maliciosas.
- Reducir la cantidad de conexiones simultáneas a servicios vulnerables.
- Desactivar servicios afectados temporalmente.

### 3. Erradicación

Las medidas incluyen:

- Identificar y eliminar software o configuraciones comprometidas.
- Actualizar el sistema operativo y las aplicaciones afectadas.
- Corregir configuraciones inseguras, como la implementación de claves de autenticación seguras para SSH.

### 4. Recuperación

Para restablecer operaciones normales:

- Utilizar respaldos periódicos para restaurar sistemas comprometidos.
- Verificar la integridad del sistema antes de la reactivación.
- Monitorear de cerca los sistemas restaurados para detectar signos de reinfección.

### 5. Lecciones aprendidas

Posteriormente, es esencial:

- Realizar un análisis postmortem del incidente.
- Actualizar políticas y procedimientos de seguridad.
- Capacitar al personal con base en las vulnerabilidades detectadas.

## Sistema de Gestión de Seguridad de la Información (SGSI)

Un SGSI conforme a ISO 27001 proporciona un marco para gestionar la seguridad de la información. Los componentes clave incluyen:

### 1. Análisis de riesgos

- Identificar activos críticos de información.
- Evaluar riesgos potenciales y su impacto.
- Establecer controles personalizados para mitigar estos riesgos.

### 2. Definición de políticas de seguridad

- Crear políticas claras para el manejo de datos sensibles.
- Definir roles y responsabilidades dentro de la organización.
- Establecer procedimientos para el uso de cifrado y controles de acceso.

### 3. Protección de datos

- Implementar respaldos periódicos y almacenarlos en ubicaciones seguras.

- Usar cifrado avanzado para proteger datos sensibles en tránsito y en reposo.
- Limitar el acceso a la información crítica mediante autenticación multifactor.

#### 4. Planes de acción

- Desarrollar medidas de respuesta rápida ante incidentes.
- Mantener actualizados los sistemas y aplicaciones para prevenir vulnerabilidades.
- Establecer simulacros periódicos de respuesta a incidentes.

## Conclusión

La aplicación de este plan de respuesta a incidentes y el desarrollo de un SGSI conforme a ISO 27001 garantizarán la capacidad de la organización para prevenir, manejar y recuperarse de futuros ataques de seguridad. Al seguir las mejores prácticas, la organización puede proteger la información crítica y mantener la confianza de sus colaboradores y clientes.