# Detecta y corrige una vulnerabilidad diferente

Tras reconocer y recolectar las evidencias pasadas, el objetivo que nos proponemos ahora es escanear, detectar y explotar una vulnerabilidad diferente a la explotada anteriormente y documentar todo el proceso además de configurar el sistema para asegurar que esa misma vulnerabilidad no vuelva a pasar.

## 1. Proceso y vulnerabilidades encontradas

Se utilizan herramientas para la ayuda en el proceso de explotación y averiguar más fácilmente las vulnerabilidades encontradas por cada herramienta que se va a utilizar.

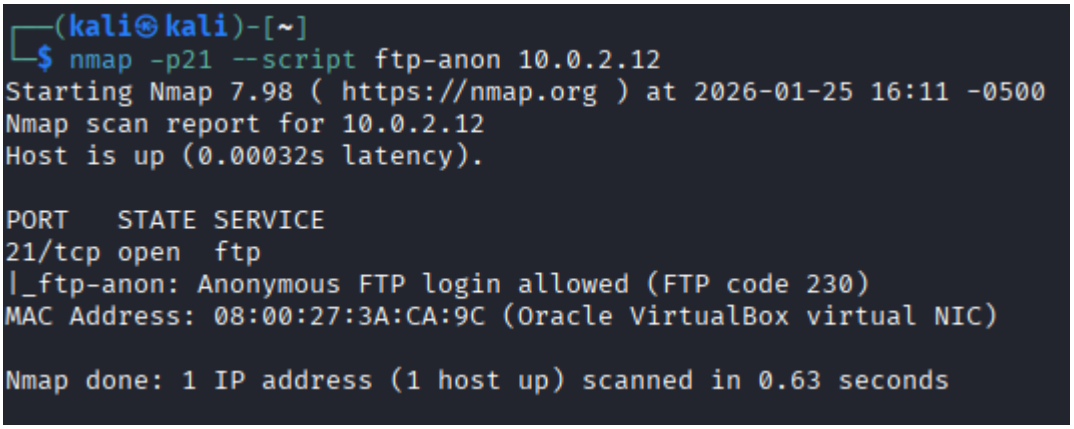
### 1. Nmap:

Es una herramienta para explorar y auditar la seguridad de las redes, escaneando puertos, detectando las versiones de los servicios e identificando los sistemas operativos. Con el comando <sudo nmap -sV -p- “IP de la Debian”>, se analiza todos los puertos activos.

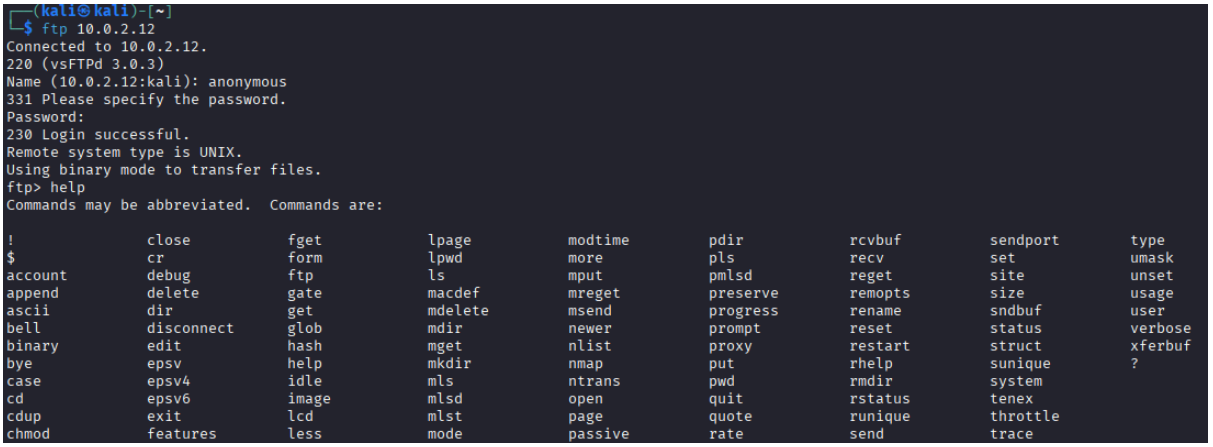
```
debian@debian:~$ sudo nmap -sV -p- 10.0.2.12
Starting Nmap 7.93 ( https://nmap.org ) at 2026-01-24 12:39 EST
Nmap scan report for 10.0.2.12
Host is up (0.0000010s latency).
Not shown: 65532 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
80/tcp open  http    Apache httpd 2.4.62 ((Debian))
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.99 seconds
```

Con el puerto de “vsftpd” abierto, se puede intentar entrar en este puerto, por ello primero se debe investigar si hay una intrusión de una mala configuración de este puerto con un acceso anónimo.

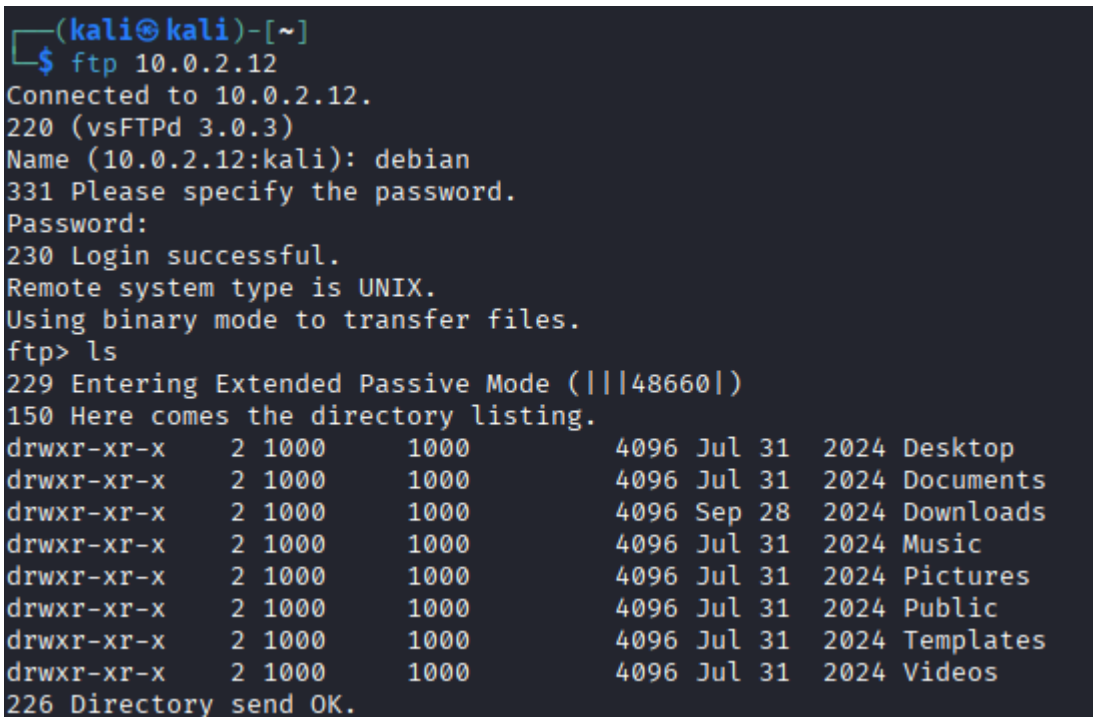
Con ello se utiliza el comando “nmap -p21 --script ftp-anon <IP de la debian>” con esto se escanea el puerto y se utiliza un “script” para filtrar si el acceso anónimo está accesible.




Como se observa en la imagen, está accesible y con ello sólo accediendo desde “ftp <IP de la debian>” se puede entrar con el acceso “nombre=anonymous y la contraseña=anonymous”.




Con ello se contempla que se ha entrado desde el puerto 21, además con una consola con comandos y con los comandos ejecutables con “help”, incluso si se adivinase el nombre y contraseña con una fuerza bruta, con el acceso de: “nombre = debian y la contraseña=123456”, se puede obtener una consola como si fuésemos administradores.




Con ello se ha encontrado una vulnerabilidad desde el puerto 21 el cual es una “bind shell” muy crítica hacia cualquier sistema o máquina.

## 2. Nikto:

Nikto es una herramienta que escanea servidores webs de una “IP” en concreto o de una página web, con ello se puede comprobar vulnerabilidades, software desactualizado y malas configuraciones. Para hacer el pentesting, se usa una máquina atacante en este caso la “kali”, el cual nos ayudará en la búsqueda de vulnerabilidades, además teniendo todo actualizado, podrá ser más efectivo haciéndolo así.


Tras el nikto, se comprueba que la página de Apache tiene muchas vulnerabilidades disponibles, los cuales casi todos son archivos sensibles con mucha información de por medio, la cual los atacantes podría aprovecharse como:

1. **Falta de header X-Frame-Option:** Algún atacante podría cargar en la web un iframe el cual engañaría a los usuarios que cliquen en botones invisibles por el cual se podría borrar cuentas o incluso cambiar la contraseña.
2. **Falta del header X-Content-Type-Options:** Los atacantes podrían subir archivos inofensivos (tienen códigos ocultos dentro) y podrían ejecutar estos códigos o “scripts” maliciosos comprometiendo así la máquina.
3. **Divulgación de información:** El propio nikto nos avisa que tenemos mucha información interna expuesta a cualquiera que intente entrar a nuestra máquina, como información interna, fechas y tamaños de los ficheros o actualizaciones, nodos del sistema, o incluso está revelando dos cosas, el uso de “Wordpress” y “/wp-content/uploads/”.

4. **Wordpress:** Revelando que el servidor en uso es “Wordpress”, facilita bastante ataques específicos y permite buscar “exploits” hacia este sitio web.
5. **/wp-content/uploads/:** Con el listado de de directorios, se puede comprobar todos los archivos subidos, pueden incluso descargar copias de seguridad hechas anteriormente y localizar documentos sensibles por el cual pueden usar scripts para amenazarlos.
6. **Robots.txt:** No es tan relevante como las anteriores pero pueden revelar rutas ocultas, paneles internos o directorios administrativos para descubrir amenazas.
7. **Archivos logs públicos:** Con los “logs” públicos, tienen una alta criticidad por el cual puede contener errores, usuarios e incluso información del sistema por el cual se podría filtrar información crítica y ayudar directamente en la explotación de esta.

Como se observa, el propio Apache sin medidas de seguridad, es vulnerable e incluso se encuentran información crítica que podrían robar sin que nos diésemos cuenta.

# 2. Medidas de corrección

Las medidas de corrección se aplican de uno en uno a cada una de las vulnerabilidades anteriormente vistas. Por ello se clasificaron en los siguientes parámetros:

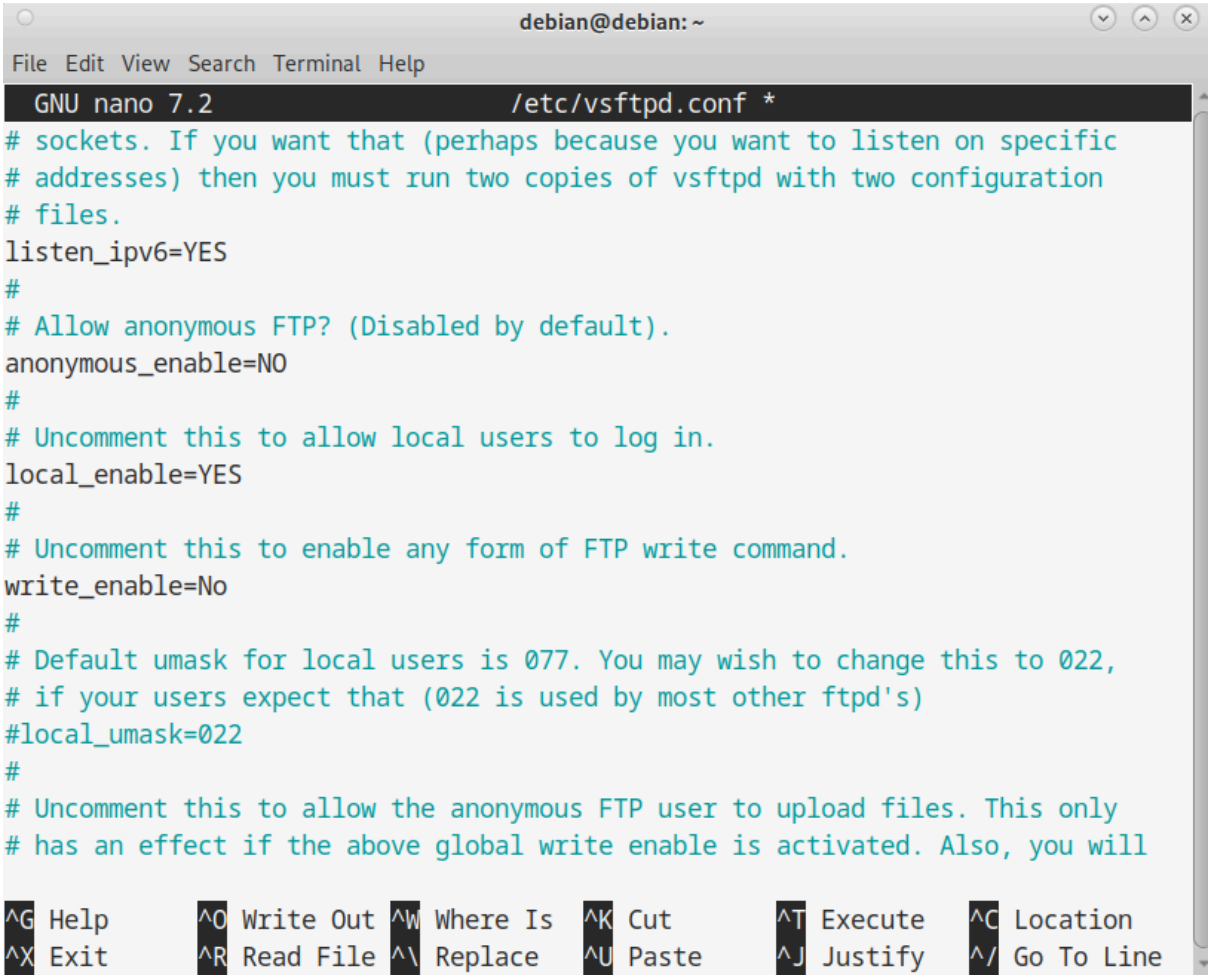
## 1. Vsftpd

Para esta vulnerabilidad la mejor recomendación es directamente apagar este puerto, el hecho es que el puerto del SSH es mucho mejor que el puerto vsftpd, porque tiene cifrado de datos tanto en reposo como en tránsito, si el puerto se necesita estar abierto por alguna razón, se debe seguir las siguientes recomendaciones.

1. **Deshabilitar acceso anónimo en FTP:** Desde la máquina debian, abrimos la configuración de vsftpd con “sudo nano /etc/vsftpd.conf”

```
debian@debian:~$ sudo nano /etc/vsftpd.conf
```

Con ello cambiaremos la configuración de “anonymous_enable= Yes” a un “No”.




- Además si incluimos “write_enable=Yes” a un “No”, evitará la modificación o subida de archivos desde este puerto. Se aplica estos cambios con un “sudo systemctl restart vsftpd”.

```
debian@debian:~$ sudo systemctl restart vsftpd
```

2. **Aplicar un firewall:** Si se tiene que tener este puerto encendido, se necesita una gran protección contra cualquiera que pueda estar a la escucha o entrada a este. Por ello se aplicará un buen sistema de firewall para la ayuda contra estas vulnerabilidades. Se usa el firewall “ufw” con los comandos siguientes:

“sudo ufw allow from 192.168.1.50 to any port 21” y “sudo ufw deny 21”

```
debian@debian:~$ sudo ufw allow from 192.168.1.50 to any port 21
Rule added
```

```
debian@debian:~$ sudo ufw deny 21
Rule added
Rule added (v6)
```

- Con ello el puerto ya no es accesible públicamente y sólo para máquinas autorizadas, esto reduce directamente los ataques hacia esta máquina desde sistemas desconocidos.

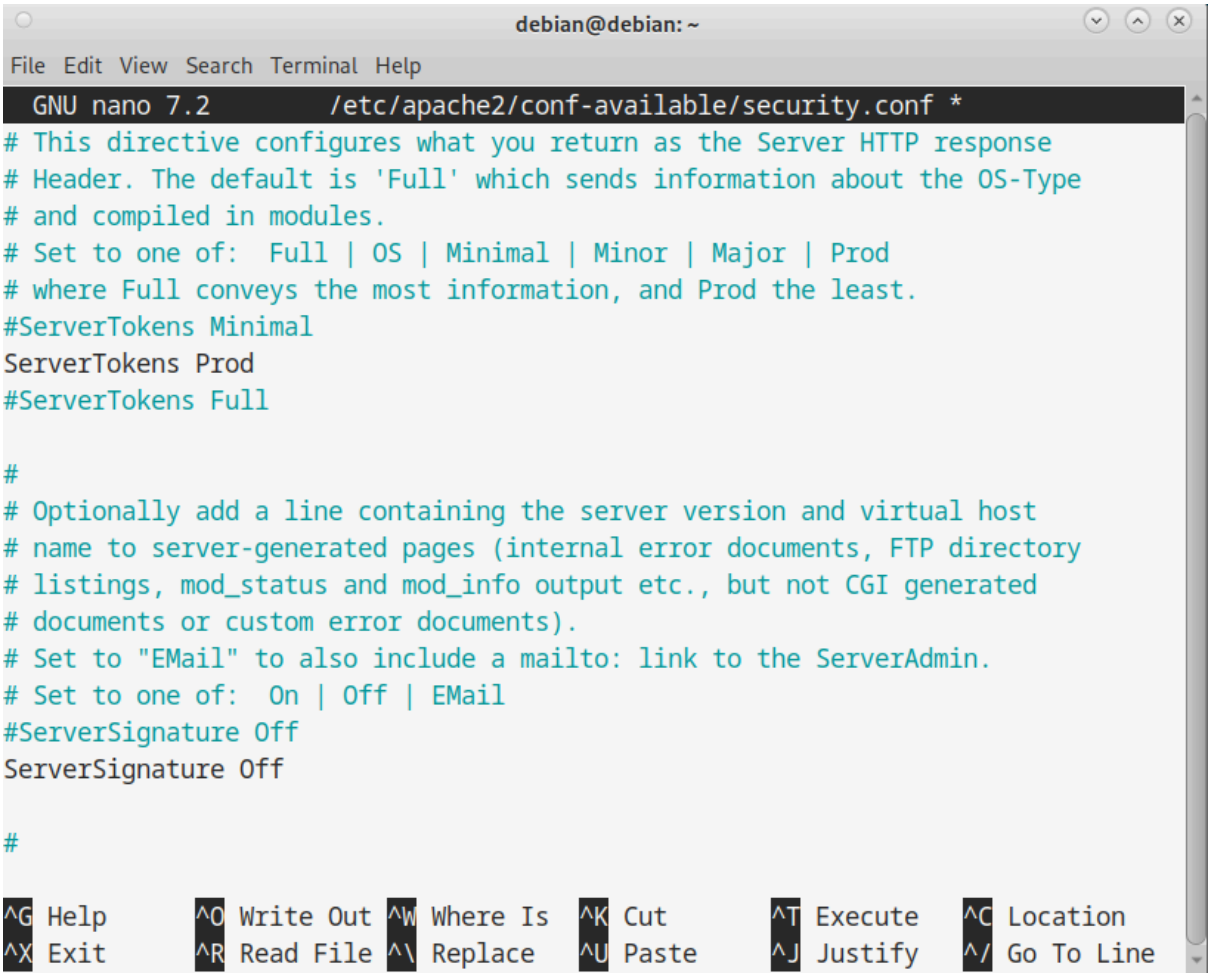
## 2. Apache

Para la configuración de Apache, se necesitan bastantes cambios, no sólo porque hay muchos datos críticos, sino también es necesario para tener un servidor web encendido.

1. **Ocultar información de Apache:** En el puerto de apache, se ha observado, que se encuentra la versión exacta de la página, el sistema operativo y varios detalles internos, por ello se hace una configuración para ayudar a reforzar el servidor de Apache.

Con el comando “sudo nano /etc/apache2/conf-available/ security.conf”, se buscará “ServerTokens” y “ServerSignature”, el primero se usará “Prod”; esto hace que se oculte la versión exacta, el sistema operativo y los módulos instalados, y se aplicará en “ServerSignature Off” para evitar en páginas de error aparezca la versión de apache, sistemas operativo y puertos.

```
debian@debian:~$ sudo nano /etc/apache2/conf-available/security.conf
```




2. **Desactivar listado de directorios:** Para evitar el listado de los directorios y se puedan aprovechar de esta vulnerabilidad para el ataque hacia el puerto, se necesita:

El comando “sudo nano /etc/apache2/apache2.conf” y cambiar “Option -Indexes”.

```
debian@debian:~$ sudo nano /etc/apache2/apache2.conf
```

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

3. **Añadir headers:** En el ataque realizado con “Nikto”, se puede comprobar la falta de “headers”, por ello se tendrá que incluir en nuestra página para evitar estos tipos de vulnerabilidades.

Se usa “sudo nano /etc/apache2/conf-available/security.conf” y en el apartado de “Headers” se agrega en el final:

```
debian@debian:~$ sudo nano /etc/apache2/conf-available/security.conf
```

```
Header always set X-Frame-Options "SAMEORIGIN"
Header always set X-Content-Type-Options "nosniff"
Header always set X-XSS-Protection "1; mode=block"
```

Con esta configuración se protege la web contra ataques de “clickjacking”, evita que el navegador interprete archivos de forma equivocada (que una imagen, se vuelva un ejecutable) y se activa un filtro para bloquear inyección de “scripts” maliciosos a la web.

4. **Proteger archivos sensibles:** Para asegurar que los archivos sensibles sean amenazados, se entrará en la configuración de Apache.

```
debian@debian:~$ sudo nano /etc/apache2/apache2.conf
```

```
<FilesMatch "^\.">
        Require all denied
</FilesMatch>
```

Con esto se bloquea el acceso a archivos críticos y se evita las filtraciones de estos.

5. **Restringir sitios innecesarios:** Algunos vectores de ataques vienen de sitios innecesarios por el cual no se ha pensado de estos, por ello se bloqueará estos sitios para evitar la escalación de esta vulnerabilidad.

Desde el apartado de “sudo nano /etc/apache2/sites-available/ 000-default.conf”

```
debian@debian:~$ sudo nano /etc/apache2/sites-available/000-default.conf
```

```
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html
         <Directory /var/www/html>
                <LimitExcept GET POST>
                        Require all denied
                </LimitExcept>
        </Directory>
```

Se aplica la siguiente configuración, con ello se evita la escalación de cualquier ataque y se reducirá el riesgo a otras vulnerabilidades.

6. **Reiniciar para aplicar cambios:** Con todos los cambios realizados, para tenerlos aplicados, se necesita reiniciar el puerto de Apache, con el uso de “sudo systemctl restart apache2” se añadirá todos los cambios anteriores.

```
debian@debian:~$ sudo systemctl restart apache2
```

Con esto se ha aplicado correctamente todos los cambios anteriores, así estará mejor configurada el servidor de Apache.