1. Introducción

El presente informe de Pentesting tiene como objetivo identificar vulnerabilidades de seguridad en un servidor Debian GNU/Linux que aloja distintos servicios, entre ellos un servidor web Apache con WordPress, servicio SSH y base de datos MariaDB.

El análisis se realizó en un entorno controlado, siguiendo un enfoque estructurado de identificación, validación y corrección de vulnerabilidades, con el fin de mejorar la postura de seguridad del sistema.

2. Alcance y metodología

El análisis incluyó:

- Escaneo de puertos y servicios expuestos.
- Identificación de configuraciones inseguras.
- Verificación de permisos y accesos.
- Pruebas de endurecimiento (hardening).
- Escaneo de malware y rootkits.
- Aplicación de medidas correctivas.

Las pruebas se realizaron directamente sobre el servidor objetivo, utilizando herramientas nativas del sistema operativo y utilidades de seguridad estándar.

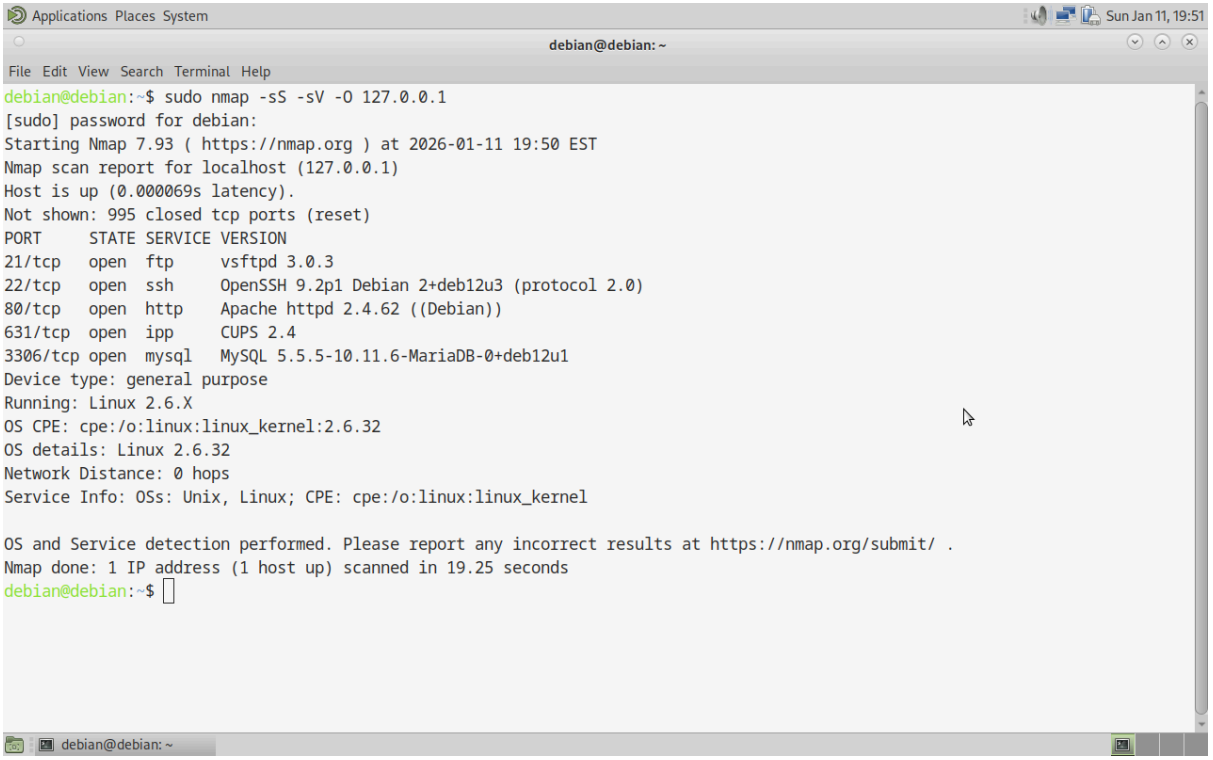
3. Identificación de vulnerabilidades

1. Puertos y servicios expuestos

Se realizó un análisis de puertos para identificar los servicios activos en el sistema. Se detectaron servicios como:

- SSH (puerto 22)
- Apache (puerto 80)
- MariaDB (puerto 3306, restringido a localhost)
- FTP (vsftpd, puerto 21)

Se hizo un escaneó con nmap y este fue el resultado:



2 . Servicio FTP inseguro

Se identificó que el servicio FTP (vsftpd) estaba activo y configurado con opciones inseguras:

- Acceso anónimo habilitado.
- Permisos de escritura activos.

Esto representa un riesgo, ya que un atacante podría subir o modificar archivos sin autenticación.

```
debian@debian:~$ ls -l /var/www/html/wp-config.php
-rwxrwxrwx 1 www-data www-data 3017 Sep 30  2024 /var/www/html/wp-config.php
debian@debian:~$
```

3 . Permisos inseguros en wp-config.php

Se detectó que el archivo wp-config.php, que contiene credenciales sensibles de la base de datos de WordPress, tenía permisos demasiado permisivos.

4. Directorio web con listado habilitado

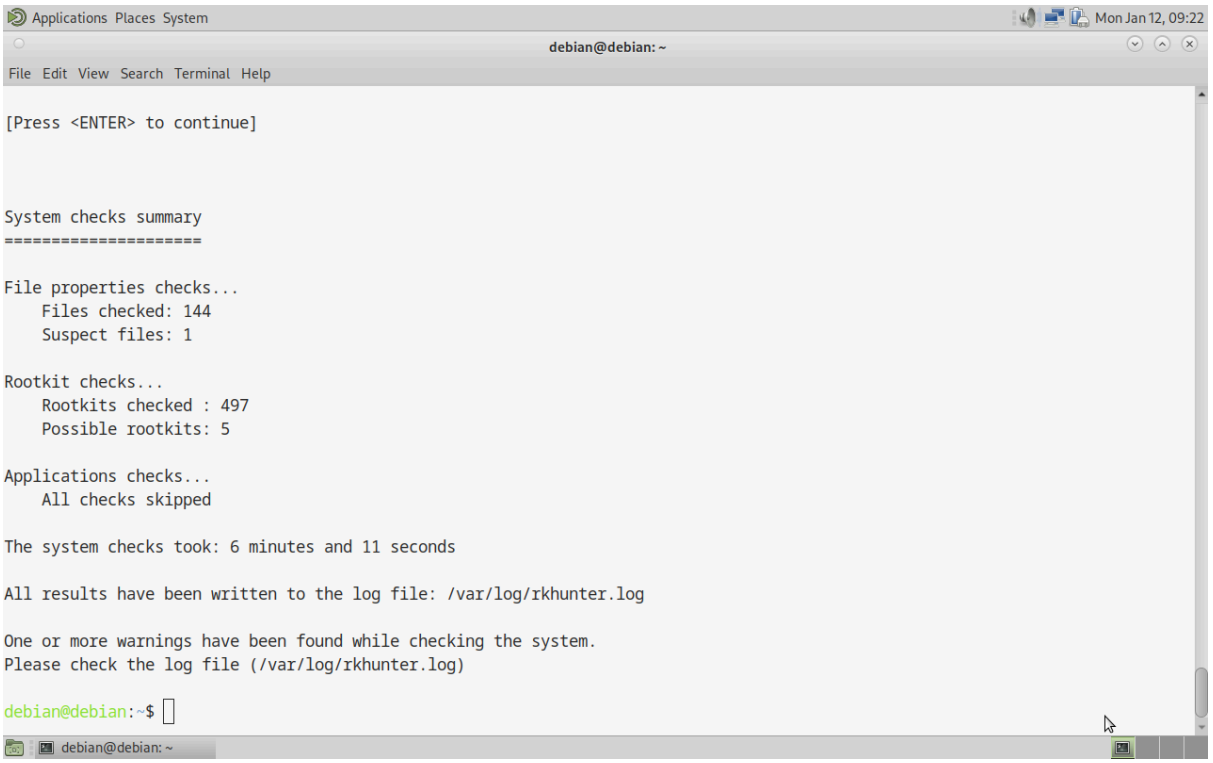
Se verificó la configuración del servidor Apache y se identificó que el listado de directorios podía estar habilitado, lo que permitiría a un atacante visualizar archivos internos del servidor web.

Todos los resultado del escaneo se encuentran en este link
https://docs.google.com/document/d/1PPO6vm6vT8oC2re6SM1s5UlJnKF_DEaXXV0Aduo0Fhk/edit?usp=sharing

5. Escaneo de malware y rootkits

Se realizó un análisis completo del sistema utilizando **rkhunter** para detectar posibles rootkits, backdoors o malware activo.

El escaneo no detectó rootkits ni malware activo. Únicamente se generaron advertencias menores.



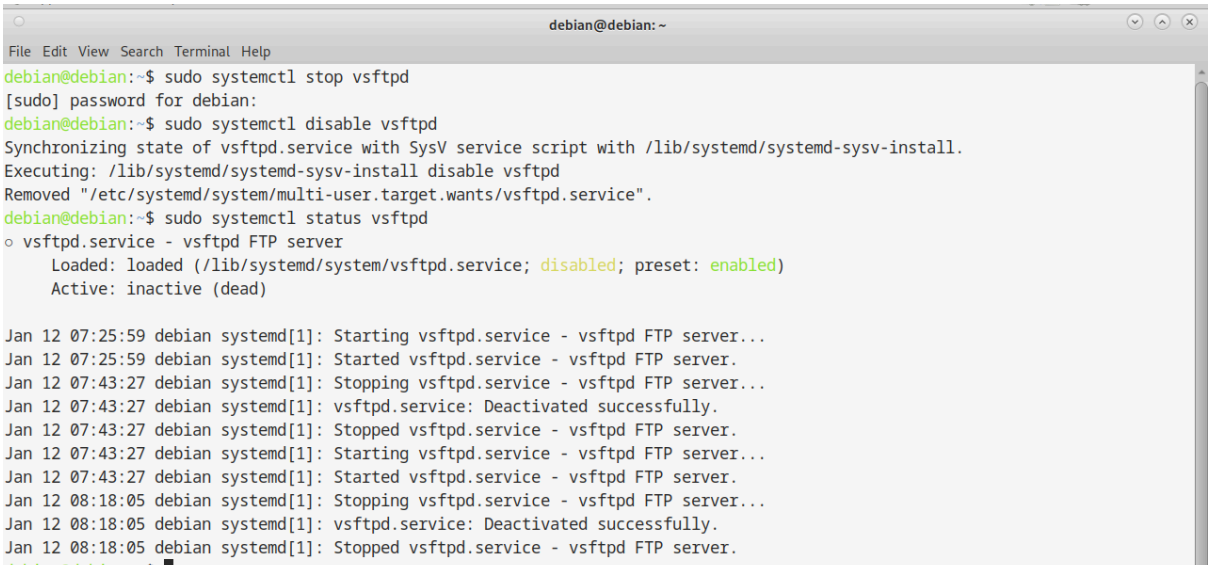
Resultados completos:

https://docs.google.com/document/d/1KZprZDIL5t62XrWnVXJNAHRxXiQ7xViCQ36cWVAVGEs/edit?usp=sharing

4. Medidas correctivas aplicadas

1. Deshabilitación del servicio FTP

Dado que el servicio FTP no era necesario para el funcionamiento del servidor, se procedió a detenerlo y deshabilitarlo permanentemente.



2. Permisos inseguros en `wp-config.php`

Se identificó que el archivo `wp-config.php`, que contiene credenciales sensibles, tenía permisos excesivos.

Esto permitía acceso no autorizado al archivo.

3. Corrección de permisos en `wp-config.php`

Se aplicó el principio de mínimo privilegio, restringiendo el acceso al archivo.

```
debian@debian:~$ chmod 600 /var/www/html/wp-config.php
chmod: changing permissions of '/var/www/html/wp-config.php': Operation not permitted
debian@debian:~$ sudo chmod 600 /var/www/html/wp-config.php
[sudo] password for debian:
debian@debian:~$ ls -l /var/www/html/wp-config.php
-rw------- 1 www-data www-data 3017 Sep 30  2024 /var/www/html/wp-config.php
debian@debian:~$ sudo chown www-data:www-data /var/www/html/wp-config.php
debian@debian:~$ ls -l /var/www/html/wp-config.php
-rw------- 1 www-data www-data 3017 Sep 30  2024 /var/www/html/wp-config.php
```

4. Directorio web con listado habilitado

Se revisó la configuración del servidor Apache y se detectó que el listado de directorios podía estar habilitado. En el comando sudo, se aplicó este cambio

<Directory /var/www/>

Options -Indexes +FollowSymLinks

AllowOverride None

Require all granted

</Directory>

```
debian@debian: ~
File  Edit  View  Search  Terminal  Help
debian@debian:~$ sudo nano /etc/apache2/apache2.conf
[sudo] password for debian:
Sorry, try again.
[sudo] password for debian:
debian@debian:~$ sudo systemctl reload apache2
debian@debian:~$
```

5. Escaneo de malware y rootkits

Se realizó un análisis del sistema para detectar posibles rootkits o malware utilizando la herramienta **rkhunter**.

Resultados:

- No se detectaron rootkits.
- No se detectó malware activo.
- Se generaron advertencias menores habituales, sin impacto real.

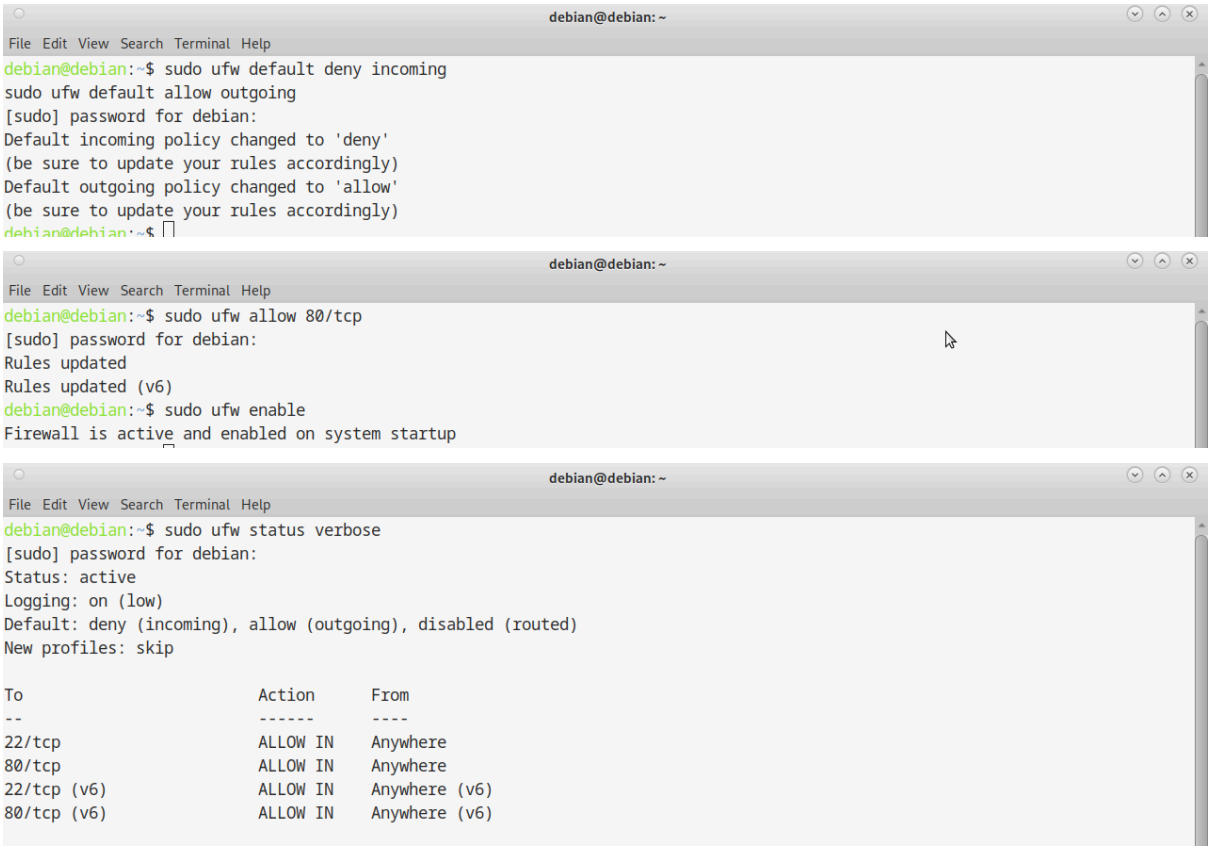
https://docs.google.com/document/d/1KZprZDIL5t62XrWnVXJNAHRxXiQ7xViCQ36cWVAVGEs/edit?usp=sharing

6. Configuración del firewall (UFW)

Se configuró un firewall básico para limitar el tráfico entrante.

Comandos utilizados:

- sudo ufw default deny incoming
- sudo ufw default allow outgoing
- sudo ufw allow 22/tcp
- sudo ufw allow 80/tcp
- sudo ufw enable
- sudo ufw status verbose



5. Conclusión

Tras la realización del análisis de seguridad y las pruebas de pentesting, se logró identificar varias debilidades relacionadas principalmente con configuraciones inseguras, servicios innecesarios activos y permisos mal asignados. Estas vulnerabilidades podrían haber sido aprovechadas por un atacante para obtener acceso no autorizado o comprometer la información del sistema.

Las acciones correctivas aplicadas permitieron reducir significativamente la superficie de ataque del servidor, asegurando únicamente los servicios necesarios y reforzando la configuración del sistema. Además, el escaneo de malware y rootkits confirmó que el servidor no presenta signos de compromiso activo, lo que indica que el sistema se encuentra en un estado estable y controlado.

En conclusión, el servidor quedó correctamente endurecido y alineado con buenas prácticas de seguridad, quedando mejor preparado para prevenir futuros incidentes similares.

6. Recomendaciones finales

- Mantener el sistema actualizado.
- Revisar periódicamente los servicios activos.
- Aplicar escaneos de seguridad regulares.
- Mantener políticas estrictas de firewall.
- Monitorizar los logs del sistema.

7. Comandos utilizados en el pentesting

Enumeración de puertos y servicios

sudo ss -tulnp

sudo ss -tulnp | grep :21

sudo ss -tulnp | grep :80

Acceso y verificación de MySQL / MariaDB

sudo mysql

SELECT user, host FROM mysql.user;

SELECT user, host, plugin FROM mysql.user;

Gestión y análisis del servicio FTP (vsftpd)

sudo grep -E "^(anonymous_enable|local_enable|write_enable)" /etc/vsftpd.conf

sudo nano /etc/vsftpd.conf

sudo systemctl restart vsftpd

sudo systemctl stop vsftpd

sudo systemctl disable vsftpd

sudo systemctl status vsftpd

Verificación y corrección de permisos en WordPress

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

sudo chmod 600 /var/www/html/wp-config.php

sudo chown www-data:www-data /var/www/html/wp-config.php

Configuración del servidor Apache

sudo nano /etc/apache2/apache2.conf

sudo systemctl reload apache2

sudo systemctl status apache2

Escaneo de rootkits y malware

sudo rkhunter --update

sudo rkhunter --check

sudo less /var/log/rkhunter.log

Configuración del firewall (UFW)

sudo ufw default deny incoming

sudo ufw default allow outgoing

sudo ufw allow 22/tcp

sudo ufw allow 80/tcp

sudo ufw enable

sudo ufw status verbose