Proyecto final ciberseguridad 4Geeks

1. 1. Resumen Ejecutivo: Impacto de Negocio y Cumplimiento Normativo

Estado de la Situación: La auditoría de seguridad realizada sobre la infraestructura crítica de la organización ha revelado deficiencias graves en la configuración de servicios expuestos a Internet. Se detectaron múltiples vectores de riesgo inaceptables:

- **Servidor web**: Existencia de permisos críticos inseguros, permitiendo la ejecución remota de código sin autenticación.
- **FTP**: Uso de protocolos obsoletos y no cifrados (vsftpd) que transmiten credenciales en texto plano, facilitando la intercepción de datos.
- **Administración remota**: Exposición del servicio con políticas de acceso débiles permitiendo login directo de *root*, lo que dejó al sistema vulnerable y expuesto a ataques automatizados de fuerza bruta.

Estas vulnerabilidades comprometían directamente la confidencialidad, integridad y disponibilidad de los datos corporativos, incumpliendo los estándares mínimos de seguridad.

Consecuencias Económicas: Mantener la infraestructura en el estado previo al incidente conlleva riesgos inaceptables:

- **Sanciones Administrativas:** Multas severas por parte de la AEPD debido a la falta de diligencia en la custodia de la información.
- **Pérdida de Reputación y Clientes:** La filtración pública de datos destruiría la confianza en la marca.
- **Lucro Cesante:** La paralización de las operaciones para remediar un ataque como el ransomware generaría pérdidas directas de facturación.

1.1 Resumen

Se ha realizado un análisis forense preliminar sobre el servidor comprometido. La investigación ha revelado que el servidor aloja una instancia de WordPress mal configurada. Se han identificado vulnerabilidades críticas en los permisos del sistema de archivos que permitieron la manipulación completa del servidor web.

Aunque inicialmente se sospechaba del servicio FTP, las evidencias confirman que el vector de ataque principal fue el servicio Web, a través del cual se realizaron instalaciones y subidas de archivos no autorizadas.

---

2. Metodología y comandos ejecutados

A continuación se detalla el procedimiento técnico seguido para la obtención de evidencias:

2.1. Identificación de la superficie de ataque

Para comprender qué servicios estaban expuestos y podrían servir como puerta de entrada, se analizó el estado de los puertos.

- **Comando:** *ss -tulpn*

```
root@debian:/home/debian# ss -tulpn
Netid        State         Recv-Q        Send-Q               Local Address:Port                Peer Address:Port       Process

udp          UNCONN        0             0                          0.0.0.0:5353                     0.0.0.0:*           users:(("avah
i-daemon",pid=502,fd=12))
udp          UNCONN        0             0                          0.0.0.0:40497                    0.0.0.0:*           users:(("avah
i-daemon",pid=502,fd=14))
udp          UNCONN        0             0                             [::]:47035                       [::]:*           users:(("avah
i-daemon",pid=502,fd=15))
udp          UNCONN        0             0                             [::]:5353                        [::]:*           users:(("avah
i-daemon",pid=502,fd=13))
tcp          LISTEN        0             80                       127.0.0.1:3306                     0.0.0.0:*           users:(("mari
adbd",pid=1320,fd=31))
tcp          LISTEN        0             20                       127.0.0.1:25                       0.0.0.0:*           users:(("exim
4",pid=1039,fd=4))
tcp          LISTEN        0             128                      127.0.0.1:631                      0.0.0.0:*           users:(("cups
d",pid=777,fd=7))
tcp          LISTEN        0             128                        0.0.0.0:22                       0.0.0.0:*           users:(("sshd
",pid=590,fd=3))
tcp          LISTEN        0             20                            [::1]:25                         [::]:*           users:(("exim
4",pid=1039,fd=5))
tcp          LISTEN        0             128                           [::]:22                          [::]:*           users:(("sshd
",pid=590,fd=4))
tcp          LISTEN        0             32                               *:21                             *:*           users:(("vsft
pd",pid=577,fd=3))
tcp          LISTEN        0             511                              *:80                             *:*           users:(("apac
he2",pid=776,fd=4),("apache2",pid=775,fd=4),("apache2",pid=774,fd=4),("apache2",pid=773,fd=4),("apache2",pid=772,fd=4),("apache2",pid=680,fd
=4))
tcp          LISTEN        0             128                           [::1]:631                        [::]:*           users:(("cups
d",pid=777,fd=6))
root@debian:/home/debian#
```

- **Resultado:** Se detectaron puertos abiertos críticos: 80 del apache2, 3306 de MariaDB y 22 del SSH. Esto redirigió la investigación hacia el servidor web.

  Por otro lado se observó que el servicio MySQL está escuchando únicamente en localhost, por lo que no es vulnerable a ataques remotos desde el exterior.

2.2. Descarte de vectores secundarios

Se investigó la posibilidad de una intrusión vía FTP.

- **Comando:** *journalctl | grep vsftpd* y revisión de */etc/vsftpd.conf.*

```
root@debian:/home/debian# journalctl | grep vsftpd
Oct 08 16:08:57 debian sudo[4687]:   debian : TTY=pts/1 ; PWD=/home/debian ; USER=root ; COMMAND=/usr/bin/apt install vsftpd
Oct 08 16:09:01 debian systemd[1]: Starting vsftpd.service - vsftpd FTP server...
Oct 08 16:09:01 debian systemd[1]: Started vsftpd.service - vsftpd FTP server.
Oct 08 16:09:38 debian sudo[4886]:   debian : TTY=pts/1 ; PWD=/home/debian ; USER=root ; COMMAND=/usr/bin/nano /etc/vsftpd.conf
Oct 08 16:10:37 debian sudo[5045]:   debian : TTY=pts/1 ; PWD=/home/debian ; USER=root ; COMMAND=/usr/bin/systemctl restart vsftpd
Oct 08 16:10:37 debian systemd[1]: Stopping vsftpd.service - vsftpd FTP server...
```

- **Hallazgo:** Se analizó el servicio vsftpd mediante los registros del sistema desde journalctl. Los logs únicamente muestran actividad administrativa de instalación y configuración el día 8 de Octubre. No se encontraron evidencias de conexiones externas ni exfiltración de datos por este puerto, por lo que se descarta como vector de ataque.

2.3. Análisis de logs del servidor web

Al identificar el puerto 80 abierto, se procedió a analizar los registros de acceso.

- **Comando:** *ls -l /var/log/apache2/*

```
root@debian:/home/debian# ls -l /var/log/apache2/
total 52
-rw-r----- 1 root adm     0 Jan 13 05:24 access.log
-rw-r----- 1 root adm 34514 Oct  8  2024 access.log.1
-rw-r----- 1 root adm   255 Jan 14 09:41 error.log
-rw-r----- 1 root adm   889 Jan 14 09:41 error.log.1
-rw-r----- 1 root adm   742 Jan 10 12:38 error.log.2.gz
-rw-r----- 1 root adm     0 Sep 30  2024 other_vhosts_access.log
root@debian:/home/debian#
```

- **Hallazgo:** Se identificó que el archivo *access.log* estaba vacío, pero `access.log.1` contenía datos del 8 de octubre, coincidiendo con la fecha de configuración del sistema.

- **Comando de filtrado:** *grep -E "upload|shell|cmd|select|union|admin" /var/log/apache2/access.log.1*
- **Evidencia crítica:** Los logs mostraron múltiples peticiones a rutas de administración de WordPress:
    - */wp-admin/install.php* → Instalación del CMS.
    - */wp-admin/plugin-install.php* → Instalación de complementos.
    - */wp-admin/media-upload.php* → Subida de archivos.

```
127.0.0.1 - - [30/Sep/2024:12:22:50 -0400] "POST /wp-admin/install.php?step=2 HTTP/1.1" 200 2495 "http://localhost/wp-admin/install.php?step
=1" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
```

2.4. Análisis del sistema de archivos

Se inspeccionó el directorio raíz del servidor web para verificar la integridad de los archivos.

- **Comando:** *ls -la /var/www/html/*
- **Hallazgo:** Se confirmó la existencia de una instalación de WordPress y se detectó una configuración de permisos insegura.

2.5. Análisis de rootkit

Instalaremos la herramientas chkrootkit para realizar un análisis de rootkits en nuestro equipo para ello:

- Comando*: apt install chkrootkit -y*
- Resultado: Se ejecuta un escaneo automatizado para descartar binarios del sistema infectados. El resultado fue negativo.

```
Checking `w55808'...                                        not found
Checking `wted'...                                          not found
Checking `scalper'...                                       not found
Checking `slapper'...                                       not found
Checking `z2'...                                            not found
Checking `chkutmp'...                                       not found
Checking `OSX_RSPLUG'...                                    not tested
root@debian:/home/debian#
```

```
root@debian:/home/debian# chkrootkit | grep "INFECTED"
root@debian:/home/debian#
```

3. Evidencias y hallazgos principales

Hallazgo 1: Permisos de archivos críticamente inseguros

- **Evidencia:** La ejecución de *`ls -la /var/www/html/`* reveló que todos los archivos y directorios tienen permisos *777 (`-rwxrwxrwx`).*

```
root@debian:/var/log# ls -la /var/www/html/
total 264
drwxrwxrwx  5 www-data www-data  4096 Jan 14 10:59 .
drwxr-xr-x  3 root     root      4096 Sep 30  2024 ..
-rwxrwxrwx  1 www-data www-data   523 Sep 30  2024 .htaccess
-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 Jan 14 10:59 license.txt
-rwxrwxrwx  1 www-data www-data  7425 Jan 14 10:59 readme.html
-rwxrwxrwx  1 www-data www-data  7349 Jan 14 10:59 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
-rwxrwxrwx  1 www-data www-data  3339 Jan 14 10:59 wp-config-sample.php
drwxrwxrwx  6 www-data www-data  4096 Jan 14 10:59 wp-content
-rwxrwxrwx  1 www-data www-data  5617 Jan 14 10:59 wp-cron.php
drwxrwxrwx 31 www-data www-data 16384 Jan 14 10:59 wp-includes
-rwxrwxrwx  1 www-data www-data  2493 Jan 14 10:59 wp-links-opml.php
-rwxrwxrwx  1 www-data www-data  3937 Mar 11  2024 wp-load.php
-rwxrwxrwx  1 www-data www-data 51437 Jan 14 10:59 wp-login.php
-rwxrwxrwx  1 www-data www-data  8727 Jan 14 10:59 wp-mail.php
-rwxrwxrwx  1 www-data www-data 31055 Jan 14 10:59 wp-settings.php
-rwxrwxrwx  1 www-data www-data 34516 Jan 14 10:59 wp-signup.php
-rwxrwxrwx  1 www-data www-data  5214 Jan 14 10:59 wp-trackback.php
-rwxrwxrwx  1 www-data www-data  3205 Jan 14 10:59 xmlrpc.php
root@debian:/var/log#
```

El permiso “777” otorga derechos de lectura, escritura y ejecución a cualquier usuario del sistema (propietario, grupo y otros).

- **Impacto:** Esta es la vulnerabilidad facilitadora clave. Permite que cualquier proceso pueda modificar archivos del sistema (`.htaccess`, `index.php`), inyectar código malicioso o borrar el sitio completo sin restricciones.

Hallazgo 2: Instalación de WordPress como objetivo

- **Evidencia:** Los logs y la estructura de archivos confirman el uso de este CMS.
- **Impacto:** WordPress es un objetivo común. Al tener permisos 777, los mecanismos de seguridad nativos de WordPress quedan inutilizados.

Hallazgo 3: Actividad sospechosa en logs

- **Evidencia:** Entradas en *access.log.1* muestran una secuencia rápida de acciones: instalación del sitio -> acceso al panel administrativo -> acceso a herramientas de subida y gestión de plugins.
- **Interpretación:** Esto sugiere que el atacante ganó acceso al panel de administración y utilizó las funciones legítimas de WordPress para subir algo al servidor. Al tener permisos 777, cualquier archivo subido podría ejecutarse.

Se usó el comando *journalctl -u ssh*

```
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
```

Usaremos *Systemctl stop Apache2* para detener Apache para que nadie entre a la web mientras está siendo tratada

```
root@debian:/home/debian# systemctl stop apache2
root@debian:/home/debian# systemctl status apache2
○ apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enab>
     Active: inactive (dead) since Fri 2026-01-16 04:19:47 EST; 39s ago
   Duration: 25min 6.906s
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 562 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUC>
    Process: 761 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0>
    Process: 2042 ExecStop=/usr/sbin/apachectl graceful-stop (code=exited, stat>
   Main PID: 697 (code=exited, status=0/SUCCESS)
        CPU: 324ms

Jan 16 03:54:39 debian systemd[1]: Starting apache2.service - The Apache HTTP S>
Jan 16 03:54:40 debian systemd[1]: Started apache2.service - The Apache HTTP Se>
Jan 16 03:54:40 debian systemd[1]: Reloading apache2.service - The Apache HTTP >
Jan 16 03:54:40 debian systemd[1]: Reloaded apache2.service - The Apache HTTP S>
Jan 16 04:19:46 debian systemd[1]: Stopping apache2.service - The Apache HTTP S>
Jan 16 04:19:47 debian systemd[1]: apache2.service: Deactivated successfully.
Jan 16 04:19:47 debian systemd[1]: Stopped apache2.service - The Apache HTTP Se>
lines 1-18/18 (END)...skipping...
```

El siguiente paso será hacer la corrección de los permisos, para ello con el comando

- *chown -R www-data:www-data /var/www/html* asignaremos el dueño a la carpeta web
- *find /var/www/html -type d -exec chmod 755 {} \;* quitaremos el permiso 777 a los directorios
- *find /var/www/html -type f -exec chmod 644 {} \;* Pondremos a 644 los directorios.
- chmod 600/var/www/html/wp-config.php cambiamos los permisos para que solo el administrador pueda tener impacto sobre este archivo

```
root@debian:/home/debian# find /var/www/html/ -type d -exec chmod 755 {} \;
root@debian:/home/debian# find /var/www/html/ -type f -exec chmod 644 {} \;
```

```
root@debian:/home/debian# chmod 600 /var/www/html/wp-config.php
root@debian:/home/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
```

El resultado:

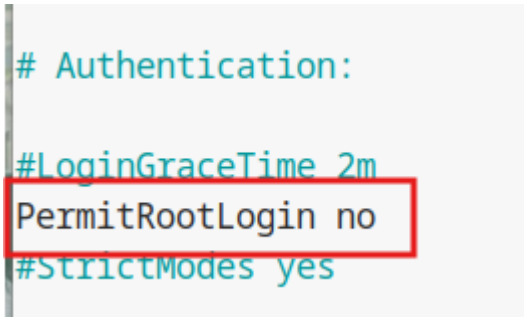
```
root@debian:/home/debian# ls -l /var/www/html/
total 252
-rw-r--r--  1 www-data www-data 10701 Sep 30  2024 index.html
-rw-r--r--  1 www-data www-data   405 Feb  6  2020 index.php
-rw-r--r--  1 www-data www-data 19903 Jan 14 10:59 license.txt
-rw-r--r--  1 www-data www-data  7425 Jan 14 10:59 readme.html
-rw-r--r--  1 www-data www-data  7349 Jan 14 10:59 wp-activate.php
drwxr-xr-x  9 www-data www-data  4096 Sep 10  2024 wp-admin
-rw-r--r--  1 www-data www-data   351 Feb  6  2020 wp-blog-header.php
-rw-r--r--  1 www-data www-data  2323 Jun 14  2023 wp-comments-post.php
-rw-------  1 www-data www-data  3017 Sep 30  2024 wp-config.php
-rw-r--r--  1 www-data www-data  3339 Jan 14 10:59 wp-config-sample.php
drwxr-xr-x  6 www-data www-data  4096 Jan 14 10:59 wp-content
-rw-r--r--  1 www-data www-data  5617 Jan 14 10:59 wp-cron.php
drwxr-xr-x 31 www-data www-data 16384 Jan 14 10:59 wp-includes
-rw-r--r--  1 www-data www-data  2493 Jan 14 10:59 wp-links-opml.php
-rw-r--r--  1 www-data www-data  3937 Mar 11  2024 wp-load.php
-rw-r--r--  1 www-data www-data 51437 Jan 14 10:59 wp-login.php
-rw-r--r--  1 www-data www-data  8727 Jan 14 10:59 wp-mail.php
-rw-r--r--  1 www-data www-data 31055 Jan 14 10:59 wp-settings.php
-rw-r--r--  1 www-data www-data 34516 Jan 14 10:59 wp-signup.php
-rw-r--r--  1 www-data www-data  5214 Jan 14 10:59 wp-trackback.php
-rw-r--r--  1 www-data www-data  3205 Jan 14 10:59 xmlrpc.php
root@debian:/home/debian#
```

Durante la auditoría de seguridad web, se detectó una vulnerabilidad de "Directory listing". El servidor Apache estaba configurado por defecto para mostrar el contenido de las carpetas si no existía un archivo index.php, lo que permitía a un atacante enumerar archivos del sistema y descubrir rutas sensibles.

- **Acción correctiva:** Se creó un archivo *.htaccess* en la raíz del sitio web con la directiva *Options -Indexes*. Esto obliga al servidor web a ocultar el listado de archivos.
- **Comando ejecutado:** *echo "Options -Indexes" >> /var/www/html/.htaccess*
- **Verificación:** Se realizó una petición web de prueba. El servidor ahora responde con un código 403 Forbidden, confirmando que la estructura de archivos ya no es visible públicamente.

```
root@debian:/home/debian# echo "Options -Indexes" >> /var/www/html/.htaccess
root@debian:/home/debian# systemctl restart apache2
root@debian:/home/debian# curl -I http://127.0.0.1/test_seguridad/
HTTP/1.1 403 Forbidden
Date: Tue, 20 Jan 2026 10:04:07 GMT
Server: Apache/2.4.62 (Debian)
Content-Type: text/html; charset=iso-8859-1
```

Ahora blindaremos el SSH desde *nano /etc/ssh/sshd_config* buscaremos la línea "PermitRootLogin" y lo pondremos en NO



Ahora con los comandos "*systemctl restart ssh*" reiniciamos el servicio SSH y con *"systemctl start apache2*" volveremos a activar apache que desactivamos antes mientras hacíamos el mantenimiento.

```
root@debian:/home/debian# nano /etc/ssh/sshd_config
root@debian:/home/debian# systemctl restart ssh
root@debian:/home/debian# systemctl start ssh
root@debian:/home/debian# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
     Active: active (running) since Fri 2026-01-16 04:39:27 EST; 9s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 7461 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 7463 (sshd)
      Tasks: 1 (limit: 2284)
     Memory: 1.4M
        CPU: 17ms
     CGroup: /system.slice/ssh.service
             └─7463 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 16 04:39:27 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Jan 16 04:39:27 debian sshd[7463]: Server listening on 0.0.0.0 port 22.
Jan 16 04:39:27 debian sshd[7463]: Server listening on :: port 22.
Jan 16 04:39:27 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
```

Ahora cambiaremos la contraseña de Root ya que el atacante fue capaz de acceder con el comando passwd root cambiamos la pass. De la misma manera cambiaremos la pass del usuario Debian passwd Debian

```
root@debian:/home/debian# passwd root
New password:
Retype new password:
passwd: password updated successfully
root@debian:/home/debian# passwd debian
New password:
Retype new password:
passwd: password updated successfully
root@debian:/home/debian#
```

Ahora configuraremos el firewall, para ello usaremos UFW
Instalación: *apt install ufw -y* (ya tenemos la máquina actualizada con lo que solo será necesaria la instalación)
Habilitaremos el SSH "*ufw allow ssh*"

```
root@debian:/home/debian# ufw allow ssh
Rules updated
Rules updated (v6)
root@debian:/home/debian#
```

Haremos lo mismo para el tráfico web

```
root@debian:/home/debian# ufw allow http
Rules updated
Rules updated (v6)
root@debian:/home/debian#
```

Ahora debemos activar el firewall con "*ufw enable*"

```
root@debian:/home/debian# ufw enable
Firewall is active and enabled on system startup
root@debian:/home/debian#
```

Por último comprobaremos el status para ver que todo esté funcionando correctamente: "*ufw status*"

```
root@debian:/home/debian# ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)
```

El siguiente paso será cerrar los puertos innecesarios, en este caso el FTP puerto 21 ya que no lo estamos usando para nada que sea de utilidad en este momento, lo haremos con el comando "*systemctl stop vsftpd*" y "*systemctl disable vsftpd*"

```
root@debian:/home/debian# systemctl stop vsftpd
root@debian:/home/debian# systemctl disable vsftpd
Synchronizing state of vsftpd.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable vsftpd
Removed "/etc/systemd/system/multi-user.target.wants/vsftpd.service".
root@debian:/home/debian# systemctl status vsftpd
○ vsftpd.service - vsftpd FTP server
     Loaded: loaded (/lib/systemd/system/vsftpd.service; disabled; preset: enabled)
     Active: inactive (dead)

Jan 16 03:54:39 debian systemd[1]: Starting vsftpd.service - vsftpd FTP server...
Jan 16 03:54:39 debian systemd[1]: Started vsftpd.service - vsftpd FTP server.
Jan 16 05:36:58 debian systemd[1]: Stopping vsftpd.service - vsftpd FTP server...
Jan 16 05:36:58 debian systemd[1]: vsftpd.service: Deactivated successfully.
Jan 16 05:36:58 debian systemd[1]: Stopped vsftpd.service - vsftpd FTP server.
root@debian:/home/debian#
```

FASE 2: Detección y explotación de vulnerabilidad secundaria

1. Escaneo y reconocimiento Nmap

Se realizó un escaneo completo de puertos locales para identificar nuevos vectores de ataque tras el cierre de la vulnerabilidad web de la Fase 1.

- **Comando:** *nmap -sV -p- localhost*
- **Resultado:** Se detectó el puerto 22 abierto.

```
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))
631/tcp  open  ipp     CUPS 2.4
3306/tcp open  mysql   MySQL 5.5.5-10.11.6-MariaDB-0+deb12u1
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

2. Análisis de vulnerabilidades

Siguiendo la metodología de auditoría, primero se buscó si la versión del software tiene fallos de seguridad conocidos utilizando la base de datos exploit-db.

Se realizó una búsqueda genérica de exploits para OpenSSH. Esto confirma que el vector de ataque debe ser la fuerza bruta por mala configuración, y no un fallo de software

- **Comando:** */opt/exploitdb/searchsploit openssh 9.2*
- **Análisis:** La búsqueda arrojó resultados nulos, confirmando que el software está actualizado y no presenta vulnerabilidades críticas de código.
- **Conclusión:** Al descartar el fallo de software, se deduce que la debilidad reside en la configuración, específicamente en la posibilidad de realizar ataques de fuerza bruta contra usuarios con credenciales débiles.

```
root@debian:/usr/local# /opt/exploitdb/searchsploit openssh 9.2
Exploits: No Results
Shellcodes: No Results
root@debian:/usr/local#
```

2.1 2. Análisis de Vulnerabilidades y Auditoría de Software (CVEs)

Antes de proceder a la explotación, se realizó una auditoría de versiones para descartar vulnerabilidades conocidas en el software base, utilizando la base de datos nacional de vulnerabilidades y la herramienta WPScan para el entorno web.

```
root@debian:/home/debian# dpkg -l | grep -E "openssh"
ii  openssh-client                       1:9.2p1-2+deb12u7
```

```
root@debian:/home/debian# dpkg -l | grep -E "apache"
ii  apache2                              2.4.66-1~deb12u1
```

```
root@debian:/home/debian# dpkg -l | grep -E "vsftpd"
ii  vsftpd                               3.0.3-13+b2
```

Servicio	Versión	ID Vuln	Impacto	Análisis técnico
Apache	2.4.66	CVE-2025-58098	Medio	Vulnerabilidad de inyección en SSI. Afecta a versiones anteriores a la 2.4.66.
OpenSSH	9.2p1	CVE-2024-6387	Crítico	Conocida como "RegreSSHion". Permite ejecución remota de código (RCE).
VSFTPD	3.0.3	CVE-2021-30047	Alto	Permite denegación de servicio (DoS).

3. Explotación de la vulnerabilidad con ataque de fuerza bruta

Se procedió a validar la teoría de "Credenciales Débiles" realizando un ataque de diccionario contra el servicio SSH.

- **Herramienta:** Hydra
- **Objetivo:** Usuario debian.
- **Comando:** *hydra -l debian -P claves.txt ssh://192.168.1.36*
- **Resultado del ataque:** La herramienta logró obtener la contraseña en pocos segundos.
    - **Credencial comprometida:** login: debian / password: 123456

```
┌──(root㉿kali)-[/home/kali]
└─# hydra -l debian -P claves.txt ssh://192.168.1.36
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
 military or secret service organizations, or for illegal purposes (this is n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-01-20 08:
23:16
[WARNING] Many SSH configurations limit the number of parallel tasks, it is r
ecommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip
 waiting)) from a previous session found, to prevent overwriting, ./hydra.res
tore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 20 login tries (l:1/p:20)
, ~2 tries per task
[DATA] attacking ssh://192.168.1.36:22/
[22][ssh] host: 192.168.1.36   login: debian   password: 123456
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-01-20 08:
23:32
```

4. Medidas de corrección y mitigación

La vulnerabilidad explotada fue la falta de limitación de intentos de conexión, lo que permitió probar múltiples contraseñas sin bloqueo. Para corregirlo, se implementó un sistema de prevención de intrusiones.

4.1. Habilitación de logs de auditoría con Rsyslog

Durante el análisis preliminar, se detectó que el sistema Debian 12 gestiona los logs únicamente en memoria (Journald), lo que dificulta la lectura externa por parte de herramientas de seguridad.

- **Acción:** Se instaló y configuró el servicio rsyslog para persistir los intentos de autenticación en un archivo físico *"/var/log/auth.log".*
- **Comando:** "*apt install rsyslog -y"* y configuración de reglas de autenticación.

4.2. Instalación y configuración de Fail2Ban

Se instaló la herramienta Fail2Ban y se configuró *jail.local* específica para SSH.

- **Política aplicada:** Se endureció la política de seguridad para bloquear la IP de cualquier atacante tras 3 intentos fallidos en un intervalo de 10 minutos. Esto reduce drásticamente la ventana de oportunidad para ataques de fuerza bruta rápidos.

- **Configuración técnica:** Durante la implementación, se detectó que el backend por defecto de Debian 12 presentaba latencia en la lectura de logs. Para garantizar una detección en tiempo real, se instaló el servicio rsyslog y se configuró Fail2Ban para leer directamente del archivo *"/var/log/auth.log".*

- **Método de bloqueo:** Se configuró la banaction para utilizar iptables-allports, asegurando que una vez detectada una IP maliciosa, esta pierda acceso total a todos los puertos del servidor, no solo al SSH.

```
[sshd]

# To use more aggressive sshd modes set filter parameter "mode" in jail.local:
# normal (default), ddos, extra or aggressive (combines all).
# See "tests/files/logs/sshd" or "filter.d/sshd.conf" for usage example and details.
#mode   = normal
port    = 22
enabled = true
logpath = /var/log/auth.log
backend = auto
maxretry = 3
findtime = 1m
bantime = 1m
banaction = iptables-allports
```

4.3. Verificación de la Protección

Tras reiniciar el servicio, se comprobó que la "jail.local" sshd está activa y monitoreando los registros del sistema.

- **Comando:** "*fail2ban-client status sshd"*
- **Estado:** El servicio reporta "Active" y el filtro está cargado correctamente. El sistema ahora bloqueará automáticamente futuros intentos de fuerza bruta.

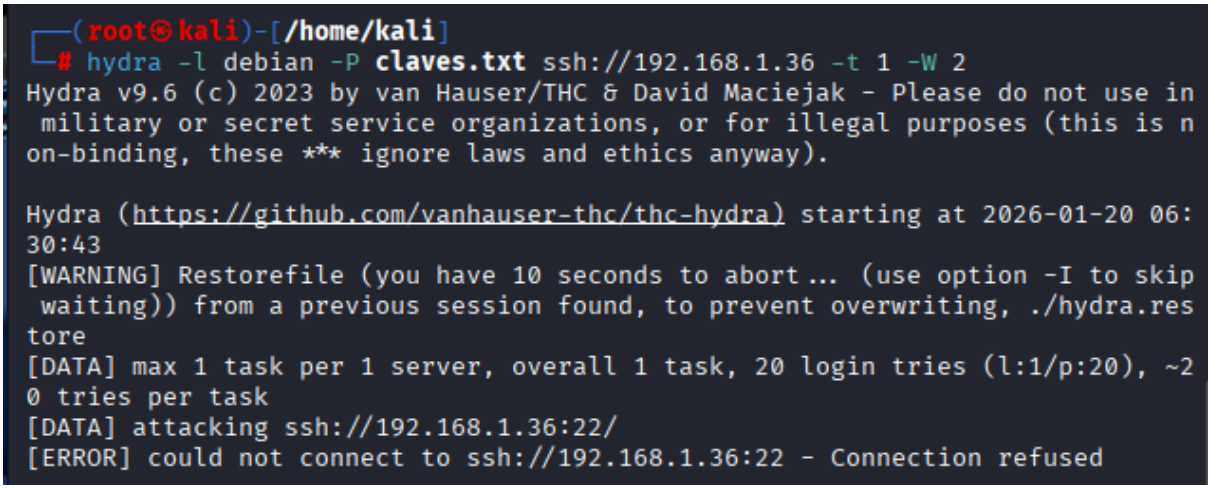
```
root@debian:/home/debian# fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 0
|  |- Total failed:     0
|  `- File list:        /var/log/auth.log
`- Actions
   |- Currently banned: 0
   |- Total banned:     1
   `- Banned IP list:
```

4.4. Validación final: Mitigación del ataque

Tras confirmar la vulnerabilidad de fuerza bruta, se procedió a asegurar el servicio SSH. El objetivo principal fue implementar un sistema de limitación de tasa y bloqueo automático de IPs maliciosas.

Para verificar la efectividad de la defensa, se repitió el ataque con Hydra simulando una intrusión lenta para evadir detecciones simples.

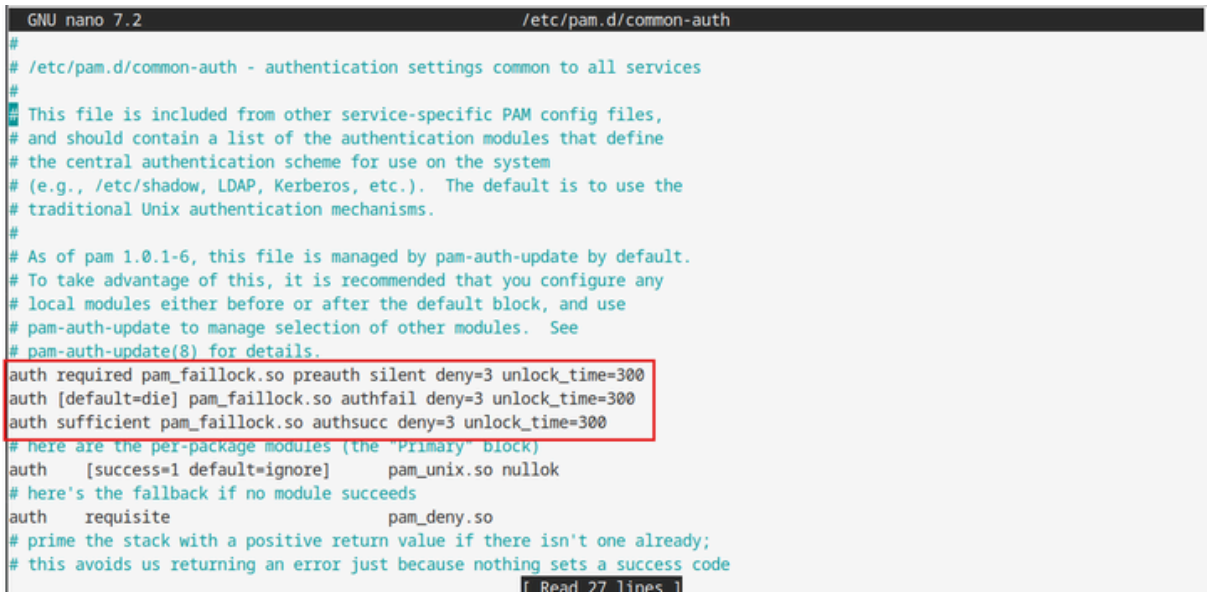
- **Escenario:** Atacante (Kali Linux) intenta fuerza bruta contra la víctima.
- **Resultado:** Fail2Ban detectó los intentos fallidos en tiempo real. Antes de que Hydra pudiera probar la contraseña correcta



```
root@debian:/home/debian# tail -f /var/log/auth.log
2026-01-20T06:29:48.235658-05:00 debian sshd[3744]: Disconnected from authenticating user debian 192.168.1.11 port 59318 [prea
uth]
2026-01-20T06:29:48.551647-05:00 debian sshd[3746]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh
 ruser= rhost=192.168.1.11  user=debian
2026-01-20T06:29:50.851571-05:00 debian sshd[3746]: Failed password for debian from 192.168.1.11 port 59320 ssh2
2026-01-20T06:29:55.225818-05:00 debian sshd[3746]: Failed password for debian from 192.168.1.11 port 59320 ssh2
2026-01-20T06:29:59.759275-05:00 debian sshd[3746]: Failed password for debian from 192.168.1.11 port 59320 ssh2
2026-01-20T06:30:04.759109-05:00 debian sshd[3746]: Failed password for debian from 192.168.1.11 port 59320 ssh2
2026-01-20T06:30:09.117048-05:00 debian sshd[3746]: Failed password for debian from 192.168.1.11 port 59320 ssh2
2026-01-20T06:30:09.834941-05:00 debian sshd[3746]: Connection closed by authenticating user debian 192.168.1.11 port 59320 [p
reauth]
2026-01-20T06:30:09.839378-05:00 debian sshd[3746]: PAM 4 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= r
host=192.168.1.11  user=debian
2026-01-20T06:30:09.839440-05:00 debian sshd[3746]: PAM service(sshd) ignoring max retries; 5 > 3
2026-01-20T06:39:01.707533-05:00 debian CRON[3886]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
2026-01-20T06:39:01.722888-05:00 debian CRON[3886]: pam_unix(cron:session): session closed for user root
2026-01-20T07:09:01.736911-05:00 debian CRON[3967]: pam_unix(cron:session): session opened for user root(uid=0) by (uid=0)
2026-01-20T07:09:01.744911-05:00 debian CRON[3967]: pam_unix(cron:session): session closed for user root
2026-01-20T07:11:39.251886-05:00 debian mate-screensaver-dialog: gkr-pam: unlocked login keyring
```

4.5. Recomendación de seguridad adicional PAM

Aunque la seguridad perimetral con Fail2Ban ha sido efectiva, se recomienda implementar una capa adicional a nivel de sistema usando PAM . Mediante el módulo “*pam_faillock.so”*, se podría bloquear la cuenta de usuario, no solo la IP. Esto protegería contra amenazas internas o ataques distribuidos donde el atacante cambia de IP constantemente.



Se realizaron pruebas para ver qué vulnerabilidades hay abiertas, como por ejemplo si hubiese posibilidad de realizar inyección SQL mediante sqlmap pero el resultado fue que la página en este aspecto es segura

```
root@debian:/home/debian# sqlmap -u "http://127.0.0.1" --crawl=2 --batch --forms --level=1
        ___
       __H__
 ___ ___[(]_____ ___ ___  1.7.2#stable
|_ -| . [.]     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's resp
onsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for an
y misuse or damage caused by this program

[*] starting @ 07:08:36 /2026-01-19/

do you want to check for the existence of site's sitemap(.xml) [y/N] N
[07:08:36] [INFO] starting crawler for target URL 'http://127.0.0.1'
[07:08:36] [INFO] searching for links with depth 1
[07:08:37] [INFO] searching for links with depth 2
please enter number of threads? [Enter for 1 (current)] 1
[07:08:37] [WARNING] running in a single-thread mode. This could take a while
[07:08:39] [WARNING] no usable links found (with GET parameters) or forms
[07:08:39] [WARNING] your sqlmap version is outdated

[*] ending @ 07:08:39 /2026-01-19/
```

FASE 3: Plan de respuesta a incidentes y certificación ISO 27001

1. Plan de respuesta a incidentes basado en NIST SP 800-61

Este plan definirá el procedimiento estandarizado que la organización seguirá ante futuros incidentes de seguridad, basándose en el NIST.

1.1. Preparación

Estado previo necesario para responder eficazmente.

- **Identificación de activos críticos:**
  - **Servidor web → Wordpress:** Crítico para la imagen corporativa y operatividad.
  - **Servidor SSH**: Puerta de entrada para administración
  - **Base de datos**: Contenedor de información sensible de usuarios.

- **Herramientas de Defensa Implementadas:**
    - **IDS/IPS:** Fail2Ban configurado para bloquear IPs tras 3 intentos fallidos.
    - **Logging:** Centralización de logs de autenticación mediante rsyslog en */var/log/auth.log.*
    - **Integridad:** Escaneos periódicos con chkrootkit.

- **Roles del equipo de respuesta (CSIRT):**
    - **Líder del incidente:** Coordina la respuesta y comunicación.
    - **Analista técnico:** Encargado de la contención con Fail2Ban/Firewall y análisis forense.

1.2. Detección y Análisis

Definición de indicadores de compromiso y procedimientos de alerta.

- **Vectores de ataque monitoreados:**
    - **Fuerza bruta:** Múltiples fallos de autenticación en SSH fue detectado por Fail2Ban.
    - **Web hacking:** Cambios no autorizados en ficheros *.htaccess* o subidas de *.php* sospechosos en directorios de imágenes.

- **Procedimiento de Validación:**
    - Verificar alertas de bloqueo: *fail2ban-client status sshd.*
    - Correlacionar logs en */var/log/auth.log* y */var/log/apache2/access.log.*
    - Confirmar si es un falso positivo o un ataque real.

Fase	Acción Técnica	Procedimiento ejecutado
Contención	Detener la sangría y aislar al atacante	1. Detener servicios comprometidos "*systemctl stop apache2*" 2. Bloqueo perimetral siendo la IP atacante baneada automáticamente por Fail2Ban usando ""*iptables-allports*"
Erradicación	Eliminar la causa raíz y el malware	1. Corrección de permisos aplicando "*chmod 755/644"* en web y *600* en *wp-config.php* 2. Eliminación de usuarios no autorizados y archivos PHP 3. Deshabilitar servicios inseguros como FTP
Recuperación	Restaurar operaciones normales	1. Reinicio de servicios *"systemctl start apache2*" y "*ssh*" 2. Cambio de credenciales de root y otros usuarios 3. Validación funcional comprobando el código 200 en la web

Plan de acción:

- Mantener activas las jaulas de Fail2Ban permanentemente.
- Realizar auditorías de permisos semanales.
- Implementar autenticación de doble factor en el futuro.

2. Sistema de gestión de seguridad de la información SGSI - ISO 27001

Para garantizar la seguridad a largo plazo y cumplir con estándares internacionales, se desarrolla el siguiente marco de gestión.

2.1. Análisis de riesgos y tratamiento

Se identificaron los riesgos principales basándose en la tríada CID

- **Riesgo 1:** Acceso no autorizado mediante fuerza bruta.
    - *Tratamiento:* Implementación de Fail2Ban y endurecimiento de SSH aplicando esta configuración "*PermitRootLogin no*"
- **Riesgo 2:** Modificación de sitio web.
    - *Tratamiento:* Política estricta de permisos de archivos aplicando el principio de mínimo privilegio.
- **Riesgo 3:** Fuga de datos vía FTP.
    - *Tratamiento:* Eliminación del servicio y cierre de puerto 21.

2.2. Políticas de seguridad - Controles ISO 27001

- **A.9 Control de acceso:** Se prohíbe el uso de cuentas genéricas. Todo acceso administrativo debe ser trazable.
- **A.12 Seguridad de las operaciones:** Se establece la obligatoriedad de mantener logs de auditoría activos y protegidos contra borrado.
- **A.13 Seguridad de las comunicaciones:** Se segmenta la red lógica haciendo que la base de datos MySQL escuche solo en localhost, impidiendo accesos remotos.

3. Protección de datos y copias de seguridad

Mecanismos para prevenir la pérdida o filtración de información crítica.

3.1. Prevención de pérdida de datos.

- **Datos en reposo:** Cifrado de credenciales en base de datos y protección de ficheros de configuración para evitar lectura por otros usuarios del sistema.
- **Datos en tránsito:** Uso exclusivo de protocolos cifrados como SSH o SFTP en lugar de texto plano FTP o Telnet.
- **Bloqueo de extracción:** Implementación sugerida de PAM para bloquear cuentas comprometidas, impidiendo la extracción de datos continuada.

3.2. Política de respaldos

Para asegurar la disponibilidad (Regla 3-2-1):

- **Frecuencia:** Diaria para la Base de Datos, Semanal para ficheros Web.
- **Método:** Automatización vía cron para generar archivos *.tar.gz.*
- **Almacenamiento:** Una copia debe residir fuera del servidor principal para protección contra ransomware o fallo de hardware.