Proyecto Final de Ciberseguridad

Carlos Vicent Arnau Chuquispuma

4 Geeks Academy

spain-cs-pt-11

Índice

1. Introducción

1.1. Objetivo

El objetivo de este proyecto es actuar como un analista de ciberseguridad para restaurar, asegurar y normalizar un servidor crítico de la organización 4 Geeks Academy que se ha visto comprometido. El proyecto abarca desde la respuesta inmediata al incidente, hasta la identificación proactiva de nuevas vulnerabilidades, finalizando con la creación de un marco de gobernanza basado en el ISO 27001 y planes de respuesta según estándares del NIST.

1.2. Contexto

Se presenta un escenario crítico, donde un servidor Debian que aloja servicios críticos, se ha visto comprometido. La Organización 4 Geeks Academy requiere recuperar la operatividad, y obtener evidencias digitales de como ha ocurrido el incidente. El entorno simula una infraestructura empresarial donde la continuidad del negocio y la protección de los datos sensibles son la máxima prioridad tras detectar una intrusión activa.

1.3. Alcance del análisis

El análisis se divide en cuatro fases críticas que delimitan el campo de actuación:

- Análisis forense
  El análisis forense incluye la investigación de logs, detección de backdoors y procesos maliciosos.
- Búsqueda de vulnerabilidad
  Escaneo integral de la superficie de ataque para detectar vulnerabilidades no explotadas.
- Mitigación de las vulnerabilidades
  Solucionar las vulnerabilidades reportadas.
- Creación de un plan de respuesta ante incidentes
  Diseño de un sistema de gestión de seguridad de la información y políticas de prevención de perdidas de datos.

1.4. Enfoque Profesional

Para cumplir con los objetivos, se adoptará un enfoque multidisciplinar:

- Analista forense
  Para la recolección de evidencias y auditoria de las acciones ocurridas.
- Pentester
  Para auditar y explotar vulnerabilidades de forma controlada.
- Consultor de ciberseguridad
  Para realizar soluciones a los problemas de seguridad detectada y alinearlo con los estándares ISO 27001 y NIST.

1.5. Herramientas utilizadas

El ecosistema de herramientas seleccionado permite cubrir todas las fases del ciclo de vida del incidente, desde la adquisición de la evidencia hasta el endurecimiento del sistema:

- Análisis Forense
  - **Qemu-img**
    Utilizada para la conversión y obtención de una imagen en formato bruto.
  - **FTK Imager**
    Empleada para el montaje de imágenes forenses, para garantizar la integridad de los datos.
  - **Autopsy**
    Herramienta principal de análisis forense digital para la recuperación de archivos eliminado, análisis de logs y registros del sistema.
  - Grep Journalctl
- Fase de Reconocimiento y Enumeración
  - **Nmap**
    Escaneo de puertos y detección de servicios/versiones (FTP, SSH, HTTP).
  - **Gobuster**
    Enumeración de directorios y archivos ocultos en el servidor web mediante técnicas de fuzzing.
  - **WPScan**
    Escaneo especializado en WordPress para identificar la versión del CMS, usuarios legítimos y vulnerabilidades en plugins o archivos como xmlrpc.php.
  - **Nikto**
    Escáner de vulnerabilidades web de código abierto que identifica configuraciones inseguras, archivos predeterminados peligrosos y versiones de software desactualizadas en el servidor Apache.
- Fase de Explotación y Acceso
  - **Hydra**
    Ejecución de ataques de fuerza bruta/diccionario contra el servicio SSH, logrando comprometer tanto al usuario debian como al superusuario root.
  - **SQLMap**
    Auditoría de la base de datos para detectar posibles inyecciones SQL y exfiltrar información.
  - **MySQL/MariaDB Client**
    Utilizado para la conexión directa a la base de datos una vez obtenidas las credenciales (123456) del archivo de configuración.
  - **SSH (Client)**
    Para la conexión remota y administración del servidor una vez obtenidas las credenciales.
  - **cURL**
    Herramienta de línea de comandos utilizada para validar manualmente los

hallazgos de los escáneres, permitiendo la inspección de cabeceras HTTP (Banner Grabbing), la verificación de métodos permitidos (como TRACE) y la interacción directa con la API del servidor.

Análisis Forense

2. Análisis forense

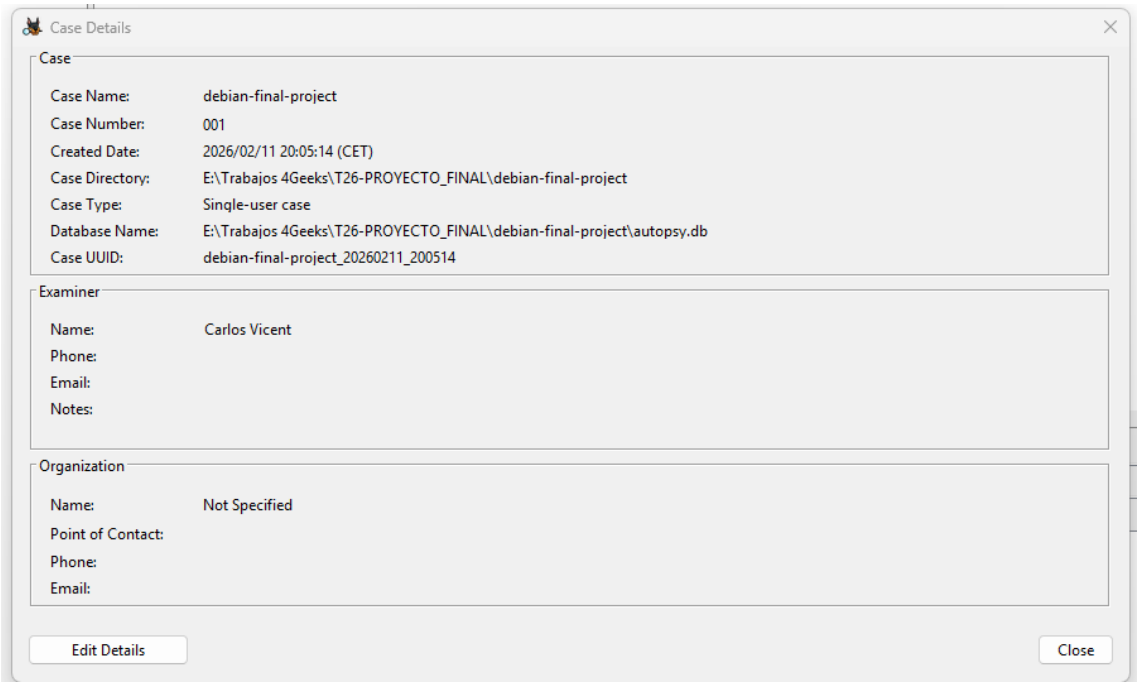
2.1. Información

Para preservar la integridad de la evidencia, se ha optado por una metodología de Dead acquisition (Adquisición en frío). Debido a que el sistema original se encontraba en un entorno virtualizado, se ha procedido a la conversión del disco duro virtual debian-disk001.vdi a un formato de imagen RAW mediante la herramienta qemu-img. Esta conversión permite tratar la evidencia con herramientas forenses sin alterar el estado del disco original.

Posteriormente, se ha utilizado la herramienta FTK Imager, junto con la metodología Dead acquisition (Adquisición en frío) para obtener la imagen forense definitiva. Durante este proceso se configuró una segmentación de archivo de 1500 MB, y un nivel de compresión 6 para optimizar el almacenamiento. Para asegurar la cadena de custodia y la inalterabilidad de los datos, se realizó el cálculo de las firmas digitales MD5 y SHA1

El análisis posterior de la imagen obtenida se ha llevado a cabo en un entorno controlado utilizando la herramienta Autopsy, manteniendo de esta forma el sistema original intacto y garantizando que la evidencia no ha sufrido modificaciones desde su adquisición.

Número de Caso	Debian-final-project
Número de Caso	001
Fuente original	Debian-disk001.vdi
Examinador	Carlos Vicent
Herramienta de Adquisición	FTK Imager
Fecha inicio	09/02/2026 20:27:28
Tipo de Imagen	RAW
Hash MD5	96e7972c41e2a1fcf8a8f408cbfe92e3
Hash SHA1	5a7c196152e66d6ddfad566c6dd8154fc35cca66



2.2. Análisis y reconstrucción de la intrusión

Tras el montaje de la imagen en Autopsy, se procedió a la auditoria de los registros de eventos del sistema para identificar el vector de entrada y las acciones realizadas por el atacante.

En esta distribución de Debian, se ha identificado que la gestión de registros se realiza mediante Systemd Journal, almacenándolos en formato binario dentro de /var/log/journal/. Para su análisis, se han exportado y examinado los registros de autenticación y seguridad.

2.2.1. Análisis de conexiones y logs de accesos

- Fallo de seguridad en Base de Datos 8 de octubre a las 17:28:39

El script de inicio de MariaDB está verificando cuentas de root inseguras.

```
Oct 08 17:28:39 debian systemd[1]: Started mariadb.service - MariaDB 10.11.6 database server.
Oct 08 17:28:39 debian /etc/mysql/debian-start[672]: Upgrading MySQL tables if necessary.
Oct 08 17:28:39 debian /etc/mysql/debian-start[685]: Checking for insecure root accounts.
Oct 08 17:28:39 debian /etc/mysql/debian-start[689]: Triggering myisam-recover for all MyISAM tables and aria-recover for all Aria tables
Oct 08 17:28:39 debian lightdm[718]: pam_unix(lightdm-greeter:session): session opened for user lightdm(uid=108) by (uid=0)
```

- Denegación de AppArmor

El sistema de control de acceso AppArmor bloqueo al servicio de impresión CUPS, cuando este intentaba obtener privilegios de administración de red.

```
Oct 08 17:28:37 debian systemd[1]: Started ModemManager.service - Modem Manager.
Oct 08 17:28:37 debian systemd[1]: Started udisks2.service - Disk Manager.
Oct 08 17:28:38 debian audit[522]: AVC apparmor="DENIED" operation="capable" profile="/usr/sbin/cupsd" pid=522 comm="cupsd" capability=12  capname="net_admin"
Oct 08 17:28:38 debian dbus-daemon[485]: [system] Successfully activated service 'org.freedesktop.hostname1'
Oct 08 17:28:38 debian NetworkManager[498]: <info>  [1728422918.0320] hostname: hostname: using hostnamed
```

- **Conexión remota exitosa SSH 8 de octubre a las 17:40:59**

Se detectó una conexión remota exitosa a través de SSH el 8 de octubre a las 17:40:59

No se han detectado intentos fallidos de inicio de sesión debido a que la credencial utilizada para el usuario root es tan vulnerable que se encuentra en la primera posición de la lista rockyou, lo que la hace extremadamente sensible.

Debido a que se ha realizar una conexión directa a través del SSH al usuario root, y que la conexión se ha realizado fuera del horario laboral, nos confirman las sospechas de que la actividad detectada no es legítima.

Se confirma que el servicio de ssh permite el login directo de root, lo cual es una vulnerabilidad grave. La IP del atacante queda identificada como la 192.168.0.134.

```
Oct 08 17:40:56 debian dbus-daemon[945]: [session uid=1000 pid=945] Activating service name='org.freedesktop.Notificat
Oct 08 17:40:56 debian dbus-daemon[945]: [session uid=1000 pid=945] Successfully activated service 'org.freedesktop.No
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 systemd[1]: Created slice user-0.slice - User Slice of UID 0.
```

- **Tareas de limpieza 8 de octubre a las 17:43:59**

  Aunque es un servicio automático, su ejecución tan cercana a la intrusión es sospechosa. El atacante pudo haber depositado scripts en /tmp o /var/tmp y forzado o esperado a que el sistema los limpiara para eliminar pruebas físicas del malware utilizado.

```
Oct 08 17:40:59 debian systemd[1653]: Startup finished in 132ms.
Oct 08 17:40:59 debian sshd[1650]: pam_env(sshd:session): deprecated reading of user environment enabled
Oct 08 17:43:59 debian systemd[1]: Starting systemd-tmpfiles-clean.service - Cleanup of Temporary Directories...
Oct 08 17:44:00 debian systemd[1]: systemd-tmpfiles-clean.service: Deactivated successfully.
Oct 08 17:44:00 debian systemd[1]: Finished systemd-tmpfiles-clean.service - Cleanup of Temporary Directories.
```

- **Modificación configuración de red 8 de octubre a las 17:58:38**

La interfaz de red obtiene una nueva IP. Esto puede indicar que el atacante está manipulando la configuración de red o que el sistema ha sido reiniciado/reconectado para intentar evadir bloqueos de firewall basados en IP o para establecer modificaciones en dichos servicios.

```
Oct 08 17:44:00 debian systemd[1]: run-credentials-systemd\x2dtmpfiles\x2dclean.service.mount: Deactivated successfully.
Oct 08 17:57:51 debian NetworkManager[498]: <info>  [1728424671.0885] dhcp6 (enp0s3): state changed new lease, address=2800:810:458:173:8879:c1b2:f8ba:ceb5
Oct 08 17:58:38 debian NetworkManager[498]: <info>  [1728424718.3542] dhcp4 (enp0s3): state changed new lease, address=192.168.0.137
Oct 08 17:59:59 debian systemd[1]: Starting fstrim.service - Discard unused blocks on filesystems from /etc/fstab...
Oct 08 18:00:00 debian systemd[1]: fstrim.service: Deactivated successfully.
```

- **Manipulación de bloques de disco 8 de octubre a las 18:00:00**

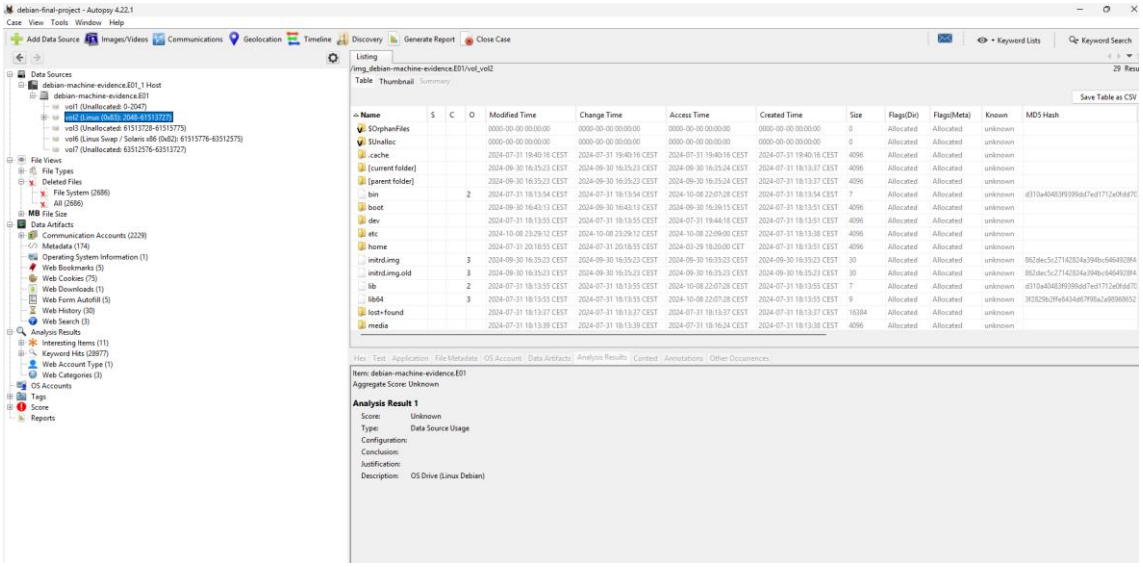
El comando fstrim descarta los bloques no utilizados en el sistema de archivos. Lo que hace es borrar físicamente los datos de los bloques libres, haciendo casi imposible recuperar archivos que el atacante haya borrado mediante técnicas de carving.

```
13686  Oct 08 17:57:51 debian NetworkManager[498]: <info>  [1728424671.0885] dhcp6 (enp0s3): state changed new lease, address=2800:810:458:173:8879:c1b2:f8ba:ceb5
13687  Oct 08 17:58:38 debian NetworkManager[498]: <info>  [1728424718.3542] dhcp4 (enp0s3): state changed new lease, address=192.168.0.137
13688  Oct 08 17:59:59 debian systemd[1]: Starting fstrim.service - Discard unused blocks on filesystems from /etc/fstab...
13689  Oct 08 18:00:00 debian systemd[1]: fstrim.service: Deactivated successfully.
13690  Oct 08 18:00:00 debian systemd[1]: Finished fstrim.service - Discard unused blocks on filesystems from /etc/fstab.
13691  Oct 08 18:03:07 debian kernel: usb 2-1: USB disconnect, device number 2
13692  Oct 08 18:03:07 debian kernel: usb 2-1: new full-speed USB device number 3 using ohci-pci
```

2.2.2. Identificación de usuarios y persistencia

Persistencia mediante Sockets de Agente: Se activaron sockets de gpg-agent-ssh.socket y gnome-keyring-daemon.socket para el usuario root. Esto permite al atacante reutilizar credenciales de sesión en memoria sin tener que volver a escribir la contraseña.

```
Oct 08 17:40:59 debian systemd[1653]: Listening on gnome-keyring-daemon.socket - GNOME Keyring daemon.
Oct 08 17:40:59 debian systemd[1653]: Listening on gpg-agent-browser.socket - GnuPG cryptographic agent and passphrase cache (access for web browsers).
Oct 08 17:40:59 debian systemd[1653]: Listening on gpg-agent-extra.socket - GnuPG cryptographic agent and passphrase cache (restricted).
Oct 08 17:40:59 debian systemd[1653]: Listening on gpg-agent-ssh.socket - GnuPG cryptographic agent (ssh-agent emulation).
Oct 08 17:40:59 debian systemd[1653]: Listening on gpg-agent.socket - GnuPG cryptographic agent and passphrase cache.
Oct 08 17:40:59 debian systemd[1653]: pulseaudio.socket - Sound System was skipped because of an unmet condition check (ConditionUser=!root).
```



2.2.3. Análisis de procesos y Malware

En los logs se detectan comportamientos automáticos y manuales que sugieren la preparación del sistema para una actividad maliciosa persistente.

- **Abuso de servicios de sistema (Anacron y Cron)**
  Inmediatamente antes y después de la intrusión, se activan tareas de CRON y anacron. Específicamente, se registra la ejecución de phpsessionclean.service. Los atacantes suelen esconder scripts de "backdoor" en directorios de limpieza de sesiones (como los de PHP o temporales) para que se ejecuten automáticamente con privilegios.
- **Manipulación de la interfaz de red enp0s3**
  Se registra que el NetworkManager cambió el estado de la conexión a CONNECTED_GLOBAL y renovó la IP por DHCP (192.168.0.137). Esto indica que el atacante aseguró la salida a internet del sistema para, posiblemente, descargar herramientas adicionales o establecer una reverse shell.
- **Ejecución de procesos de mantenimiento (Anti-forense)**
  La ejecución de systemd-tmpfiles-clean.service es altamente sospechosa en este contexto. Al limpiar archivos temporales, el sistema elimina rastro de binarios que el atacante pudo haber descargado mediante wget o curl en /tmp.

2.3. Inferencia

Tras la correlación de los eventos registrados en el sistema, se determinan las siguientes conclusiones sobre la intrusión:

- **Confirmación del vector de entrada:**
  El compromiso se originó mediante un ataque de fuerza bruta o diccionario sobre el servicio SSH. La evidencia clave se localiza a las 17:40:59, donde se registra una conexión aceptada para el usuario root desde la IP 192.168.0.134
- **Vulnerabilidad explotada:**
  - Configuración insegura del servicio SSH, la cual permitía acceso directo al superusuario sin pasar por una cuenta de usuario intermedio.
  - El acceso inmediato sin registros de bloqueos previos sugiere que la credencial del root era trivial, identificada como una de las primeras del diccionario rockyou.txt
- **Impacto y control del sistema:**
  Al obtener acceso directo como usuario root, el atacante alcanzó el máximo nivel de privilegios de forma instantánea. Esto le permitió manipular servicios y exfiltrar datos sin ningún impedimento.

Búsqueda de vulnerabilidades (Red Team)

3. Búsqueda de vulnerabilidades

3.1. Objetivo

Se ha realizado una auditoria sobre el servidor Debian proporcionado por la organización 4Geeks con el din de identificar y documentar las vulnerabilidades en el entorno.

La auditoría tiene como objetivo encontrar y vulnerar la máquina objetivo para poder proporcionar medidas correctivas y volver a dejar el servicio en correcto funcionamiento.

3.2. Alcance de la auditoria

- **Tipo de prueba:**
  Pentesting de caja gris. Se dispone de credenciales básicas de acceso para evaluar el impacto de un usuario interno o un atacante que ha logrado persistencia inicial.
- **Activos incluidos:**
  Se realizará una auditoria integral del servidor Debian (192.168.1.34), incluyendo los servicios asociados y la integridad del sistema de archivos.
- **Exclusiones:**
  Se excluye de la auditoria la capa de presentación/interfaz del CMS WordPress debido a errores de disponibilidad del servicio. Sin embargo, se incluye la auditoria de su base de datos y de los permisos de archivos, ya que contiene información crítica de la organización.
- **Límites de la prueba:**
  La actividad se restringe a la IP 192.168.1.34. No se autorizan ataques de denegación de servicios masivos, ni técnicas de ingeniería social sobre el personal de 4geeks Academy.

IP	192.168.1.34 (DHCP)
Usuario	debian
Contraseña	Conocida por el equipo auditor

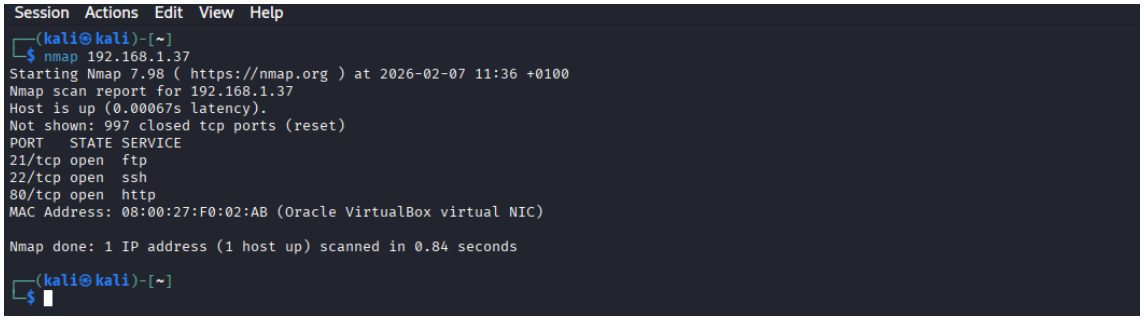
3.3. Recolección de información

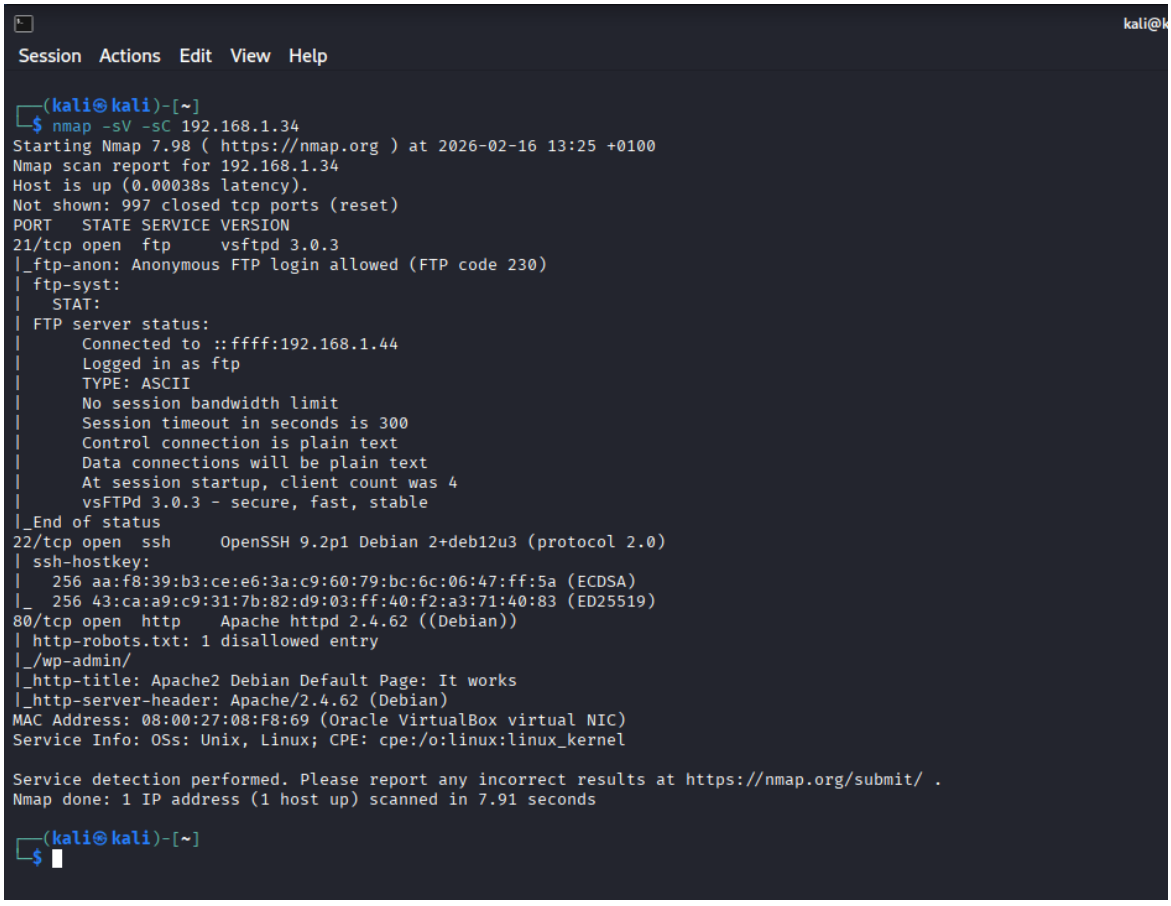
Se ha realizado un escaneo con NMAP para identificar puertos abiertos, servicios y versiones, así como una fase de enumeración dirigida a aplicaciones web y servicios de red.

3.3.1. Hallazgos de Red:

- **21/TCP (FTP):**
  Servicio vsftpd 3.0.3. Se detectó configuración de acceso anónimo activa y falta de cifrado en la transmisión.
- **22/TCP (SSH):**
  Servicio OpenSSH 9.2p1. Configuración que permite intentos de autenticación administrativa (root).

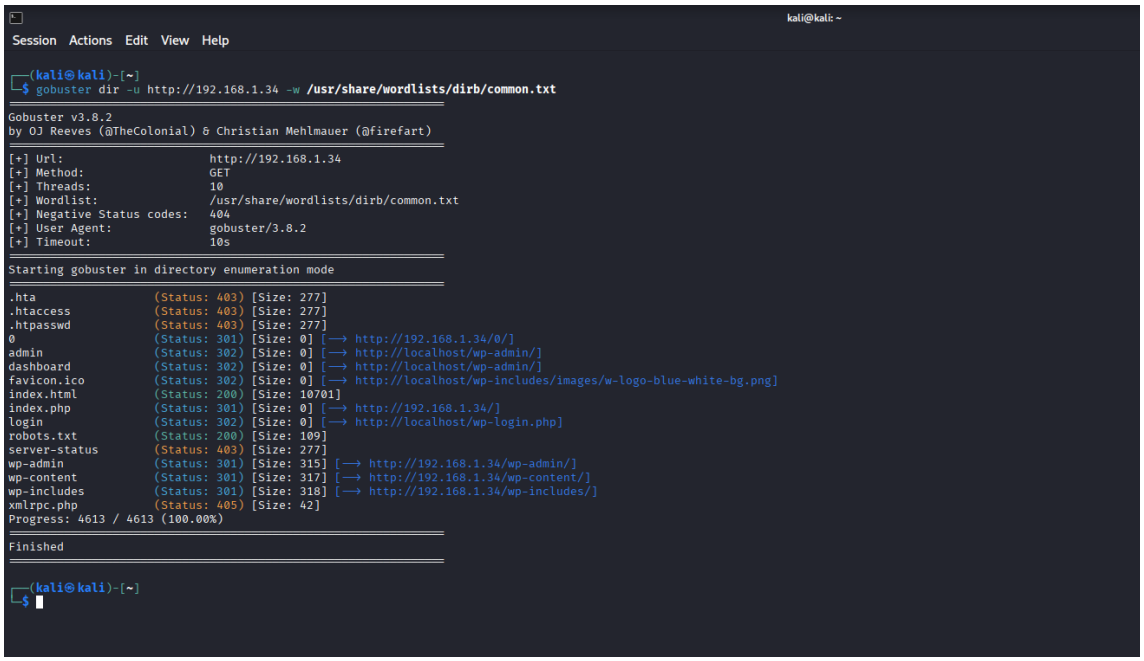
- **80/TCP (HTTP):**
  Servidor Apache 2.4.62. Aloja un gestor de contenidos (CMS) WordPress mal configurado, el cual apunta las rutas al localhost.





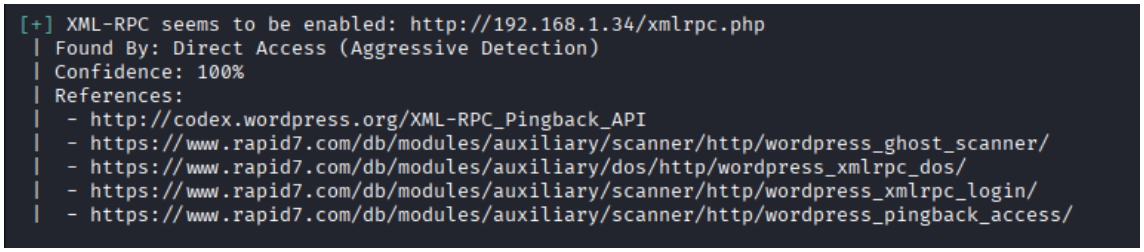
3.3.2. Enumeración de Directorios (Fuzzing)

Utilizando la herramienta Gobuster con el diccionario common.txt, se identificaron rutas críticas como /wp-admin/, /wp-content/ y, específicamente los directorios/wp-includes/ y /wp-content/uploads/ con el listado de archivos habilitado.



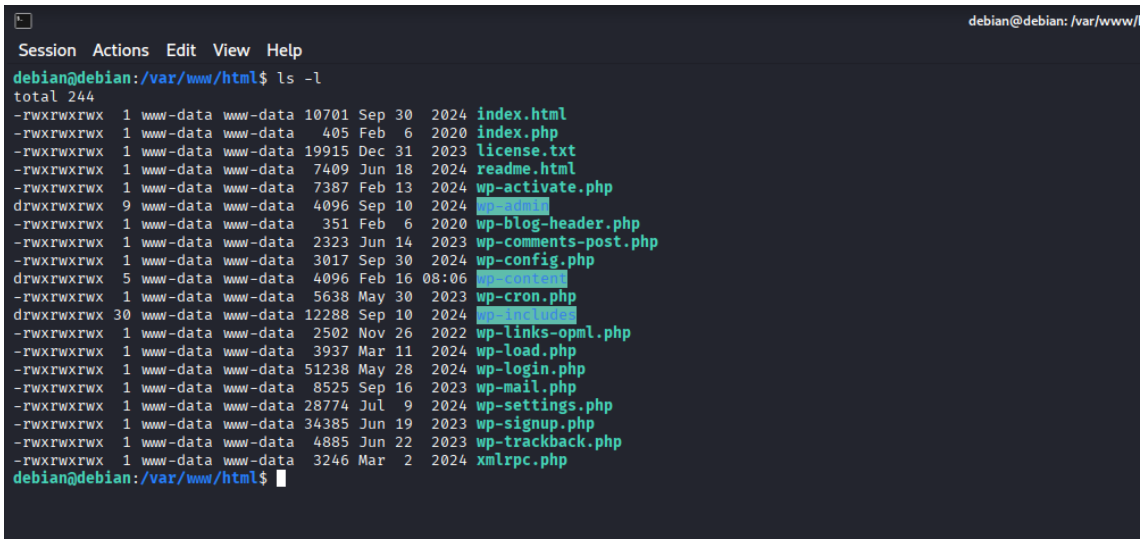
3.3.3. Auditoría de CMS (WPScan)

Se ejecutó un escaneo especializado sobre el servicio web, identificando la versión de WordPress 6.6.2 y la existencia del archivo xmlrpc.php, el cual es vulnerable a ataques de fuerza bruta y denegación de servicio (DoS).



3.3.4. Auditoría de permisos del sistema de archivos

Tras obtener acceso mediante el usuario "debian", se realizó una inspección detallada del directorio raíz del servidor web (/var/www/html/). Se detectó una configuración crítica de seguridad donde la totalidad de los archivos y directorios poseen permisos 777



3.3.5. Configuraciones inseguras del servidor WEB

A través de la herramienta Nikto, se determinó que el servidor presenta una superficie de ataque informativa. Aunque se validó que métodos peligrosos como TRACE están deshabilitados, el servidor web apache está configurado con valores por defecto que exponen la versión exacta del software.

3.4. Vulnerabilidades reportadas

Puerto	Servicio	Versión	Severidad	Detalles
21	FTP	vsftpd 3.0.3	Media	Exposición de archivos y captura de credenciales.
22	SHH	OpenSSH 9.2p1	Crítica	El servidor permite acceso directo a root con contraseña '123456'.
80	HTTP	Apache httpd 2.4.62	Media	Ataques de fuerza bruta y posible DoS distribuido.

80	HTTP	Apache httpd 2.4.62	Media	Fuga de información y exposición de archivos sensibles.
-	Sistema	Debian Release 12	Alta	Directorios y archivos críticos con permisos 777 (rwxrwxrwx).
3306	MySQL	MariaDB 10.11.6	Crítica	Robo de base de datos y acceso a datos de usuarios.

Esta tabla de vulnerabilidades está clasificada por niveles de severidad basada en CVSS

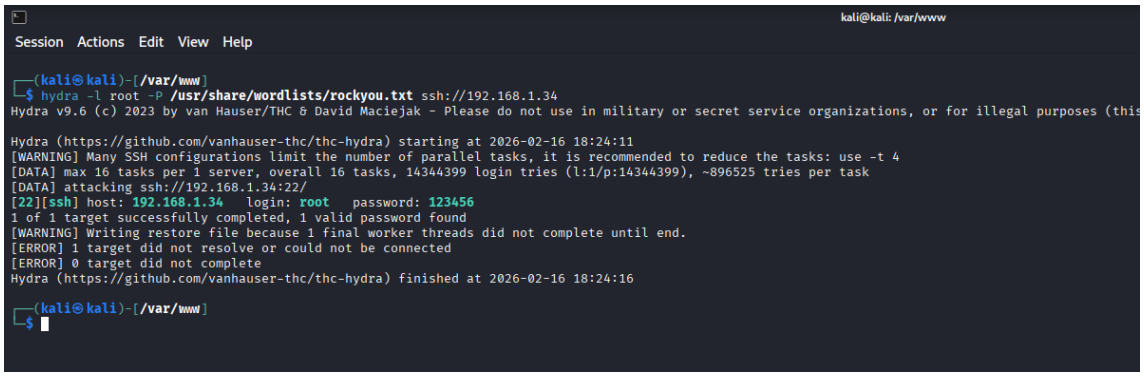
3.5. Pruebas de vulnerabilidad

3.5.1. Fuerza bruta contra SSH

Tras confirmar mediante el escaneo de red que el servicio SSH se encontraba operativo, se auditó la robustez de las credenciales del superusuario root mediante un ataque de diccionario utilizando la herramienta Hydra.

Se utilizó el diccionario estándar rockyou.txt para realizar intentos de autenticación automatizados sobre el usuario root. Se ha podido descubrir que el servidor tiene habilitada la directiva PermitRootLogin, lo que permite intentos de acceso remoto directo al administrador del sistema.

Se identificó la contraseña válida en tan solo 5 segundos. La clave se encontraba en las primeras posiciones del diccionario, lo que evidencia una vulnerabilidad crítica por el uso de contraseñas de "baja entropía" o por defecto.



3.5.2. Exposición de credenciales en wp-config.php

Se comprobó que el archivo wp-config.php no solo posee permisos de lectura global, sino que además contiene una contraseña débil para el usuario de la base de datos. Esta vulnerabilidad permite a cualquier usuario con acceso al servidor (o mediante una vulnerabilidad web de lectura de archivos) comprometer la integridad de la base de datos MySQL de la organización.

3.5.3. Explotación de la base de Datos MySQL

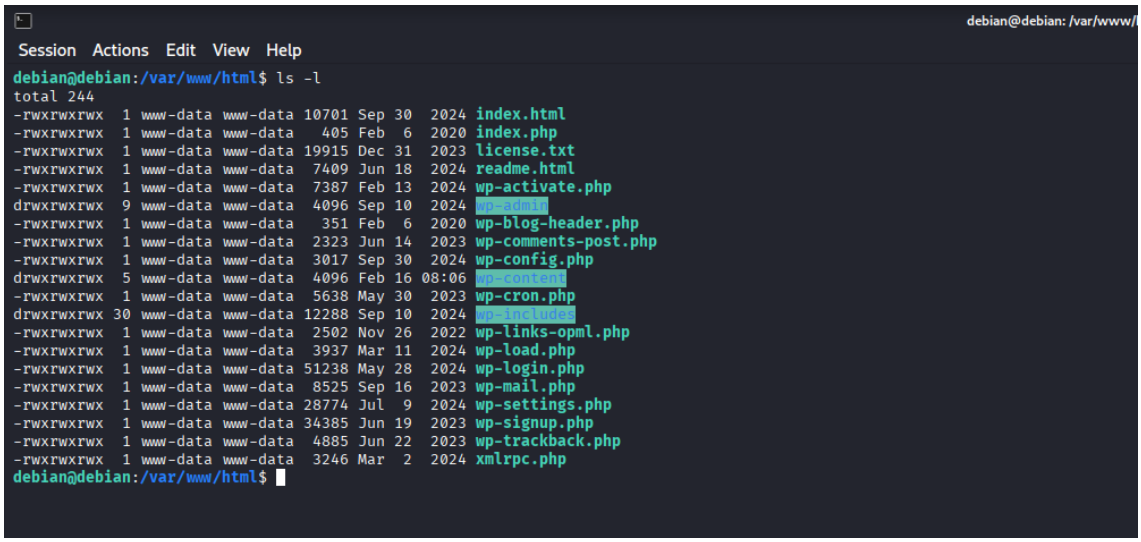
Utilizando la contraseña '123456' (identificada como una contraseña débil y extraída del archivo wp-config.php), se logró el acceso al gestor de base de datos MariaDB.

- Acceso: Se utilizó el usuario wordpressuser.
- Exfiltración: Como se muestra en la Imagen, se ejecutó una consulta exitosa a la tabla wp_users de la base de datos wordpress.
- Datos Obtenidos: Se recuperó el hash de la contraseña y el correo electrónico (rosinnicuentas@gmail.com) del usuario administrador wordpress-user.

```
debian@debian:~$ mysql -u wordpressuser -p'123456'
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 280230
Server version: 10.11.6-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SELECT user_login, user_pass, user_email FROM wp_users
    -> ;
ERROR 1046 (3D000): No database selected
MariaDB [(none)]> use wordpress
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [wordpress]> SELECT user_login, user_pass, user_email FROM wp_users
    -> ;
+---------------+------------------------------------+-----------------------+
| user_login    | user_pass                          | user_email            |
+---------------+------------------------------------+-----------------------+
| wordpress-user | $P$BM4LABXxcoawTZfuH8QRU5dcnKT2IC. | rosinnicuentas@gmail.com |
+---------------+------------------------------------+-----------------------+
```

3.5.4. Integridad del sistema de archivos

Mediante la enumeración local, se confirma que el administrador aplico de forma recursiva el permiso chmod 777 sobre /var/www/html. Esta es la vulnerabilidad raíz que permite que, una vez comprometido cualquier usuario, el atacante pueda leer el wp-config.php y saltar a la base de datos MySQL sin necesidad de explotar vulnerabilidades web complejas.

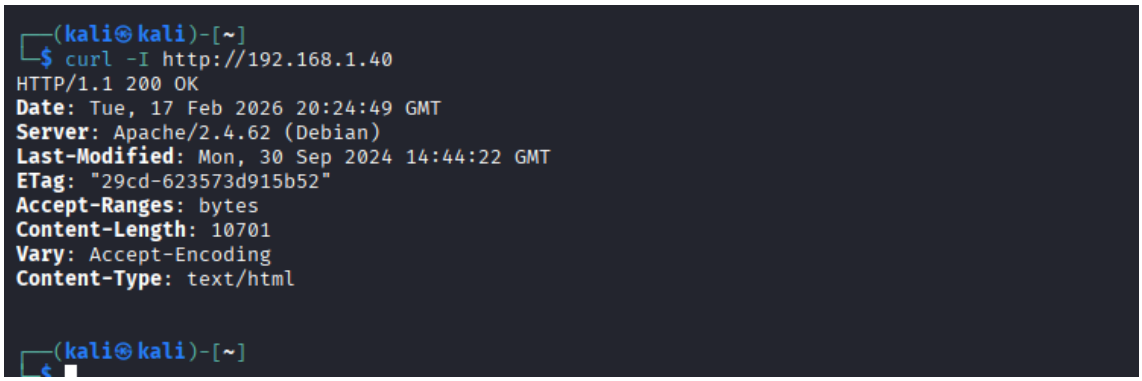


3.5.5. Configuraciones inseguras del servidor WEB

Para confirmar los hallazgos reportados por las herramientas automáticas y evaluar su impacto real, se realizó pruebas manuales utilizando la utilidad cURL.

- Validación del banner grabbing
  Se ejecutó la petición de cabeceras HTTP para inspeccionar la respuesta del servidor y verificar si la configuración de apache revela información sobre su arquitectura interna.
  - curl -I http://192.168.1.40

  Al ejecutar el comando, el servidor respondió a la cabecera con información del servidor. Esta respuesta confirma que el servidor tiene activo el Banner Grabbing, una configuración por defecto que expone tanto la versión exacta del servicio como la distribución del sistema operativo. Al conocer la versión del programa, un atacante puede investigar exploits conocidos de dicha versión para realizar ataques específicos contra las vulnerabilidades encontradas.

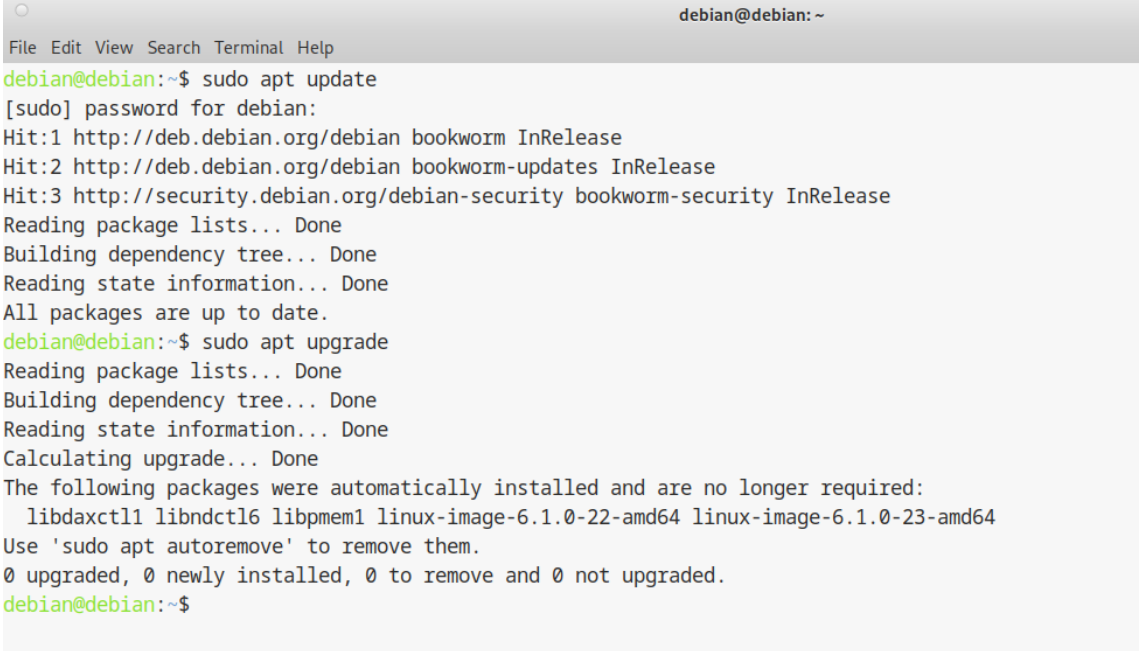


Mitigación de vulnerabilidades y propuesta de mejoras

(Blue Team)

4. Mitigación de vulnerabilidades

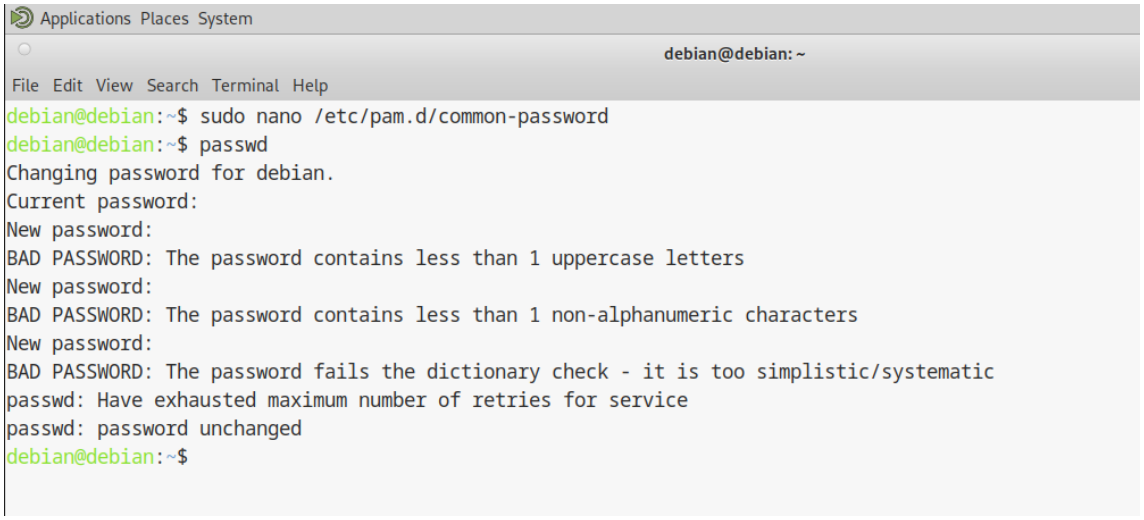
Como medida de respuesta inmediata tras la identificación de las brechas de seguridad, la primera fase del plan de remediación se centró en asegurar la base del sistema operativo (OS Hardening). Se realizó una actualización crítica de todos los paquetes y servicios del sistema mediante los comandos sudo apt update y sudo apt upgrade. Esta acción garantiza que el servidor cuente con los últimos parches de seguridad distribuidos por Debian, mitigando vulnerabilidades conocidas (CVEs) y reduciendo la superficie de ataque.

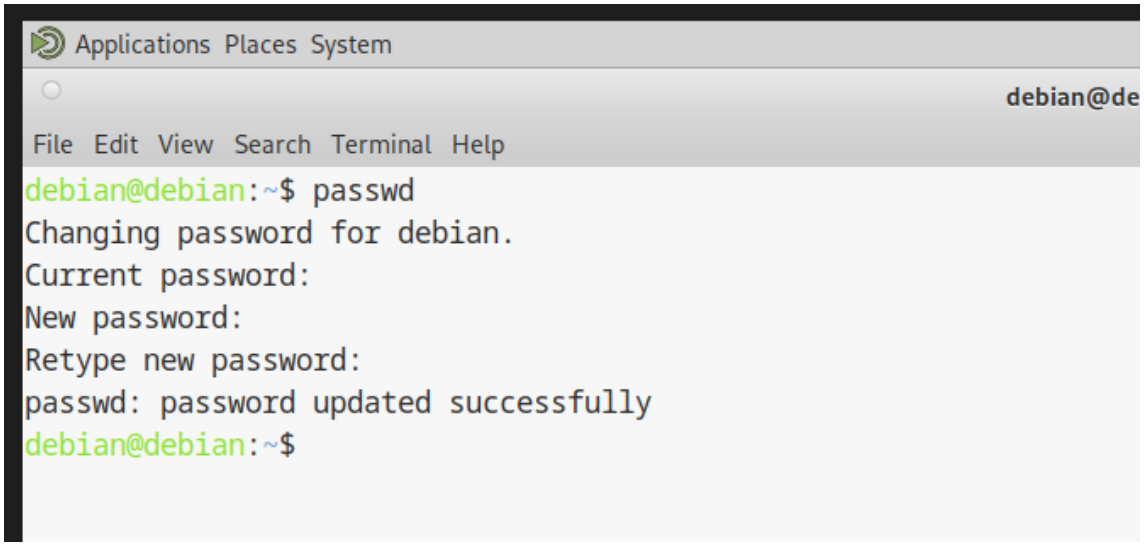


Una vez consolidada la integridad del software base, se procedió a la ejecución de acciones de mejora (hardening) específicas para cada una de las incidencias documentadas en la fase de auditoría:

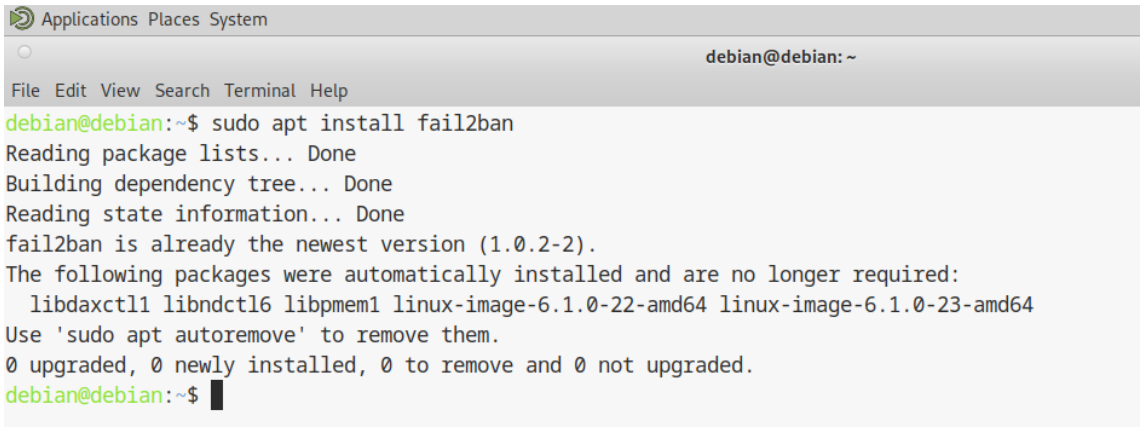
4.1. Fuerza bruta con SSH

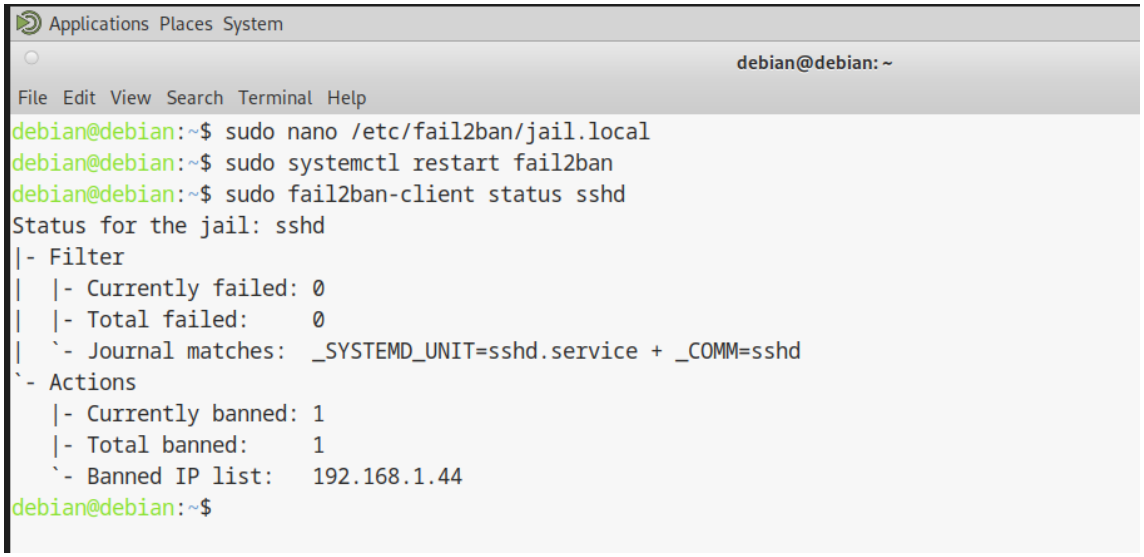
- Incidente:
  Detección de múltiples intentos de acceso fallidos automatizados (fuerza bruta) utilizando herramientas como Hydra.
- Soluciones:
  - Enforcement de Contraseñas: Se configuró el módulo pam_pwquality en /etc/pam.d/common-password. Se ha establecido una política estricta que exige un mínimo de 12 caracteres, incluyendo mayúsculas, minúsculas, números y símbolos, rechazando cualquier credencial débil o diccionario.

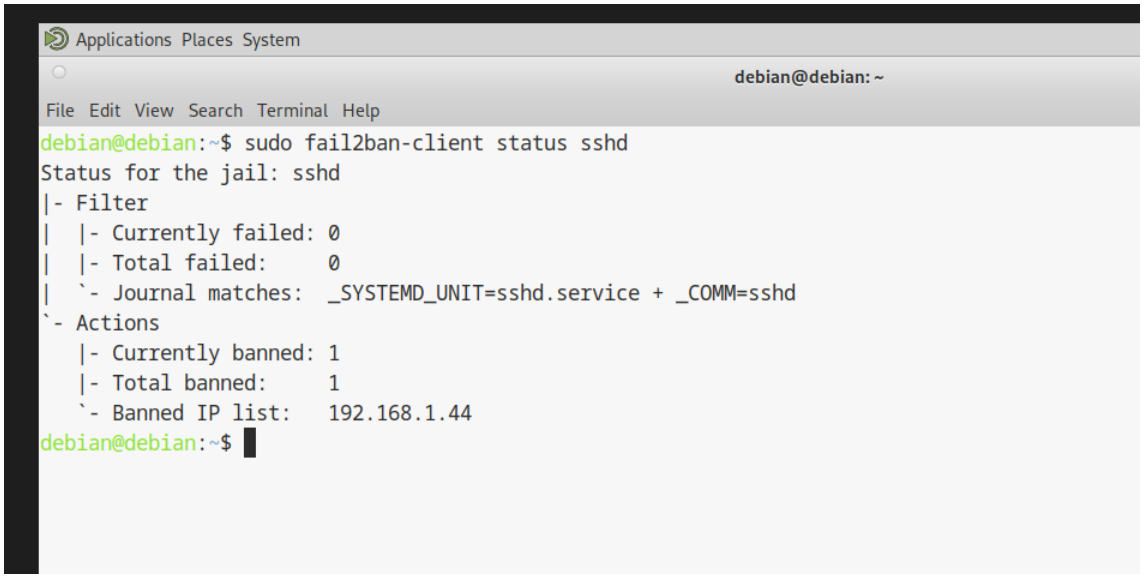


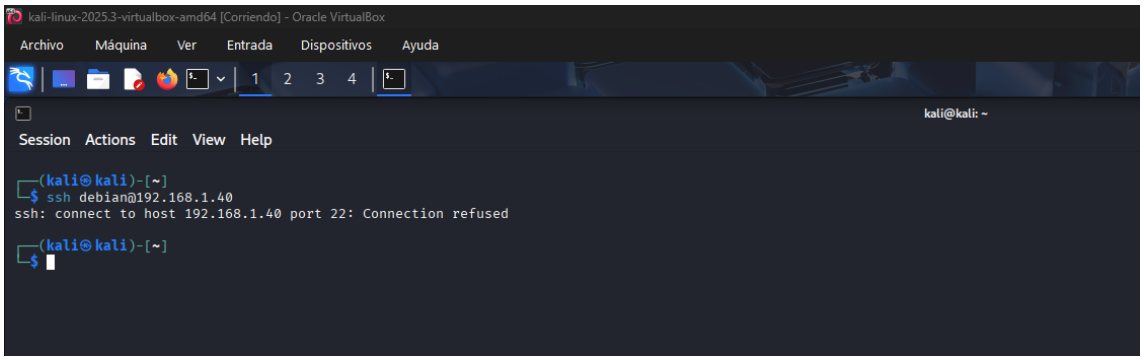


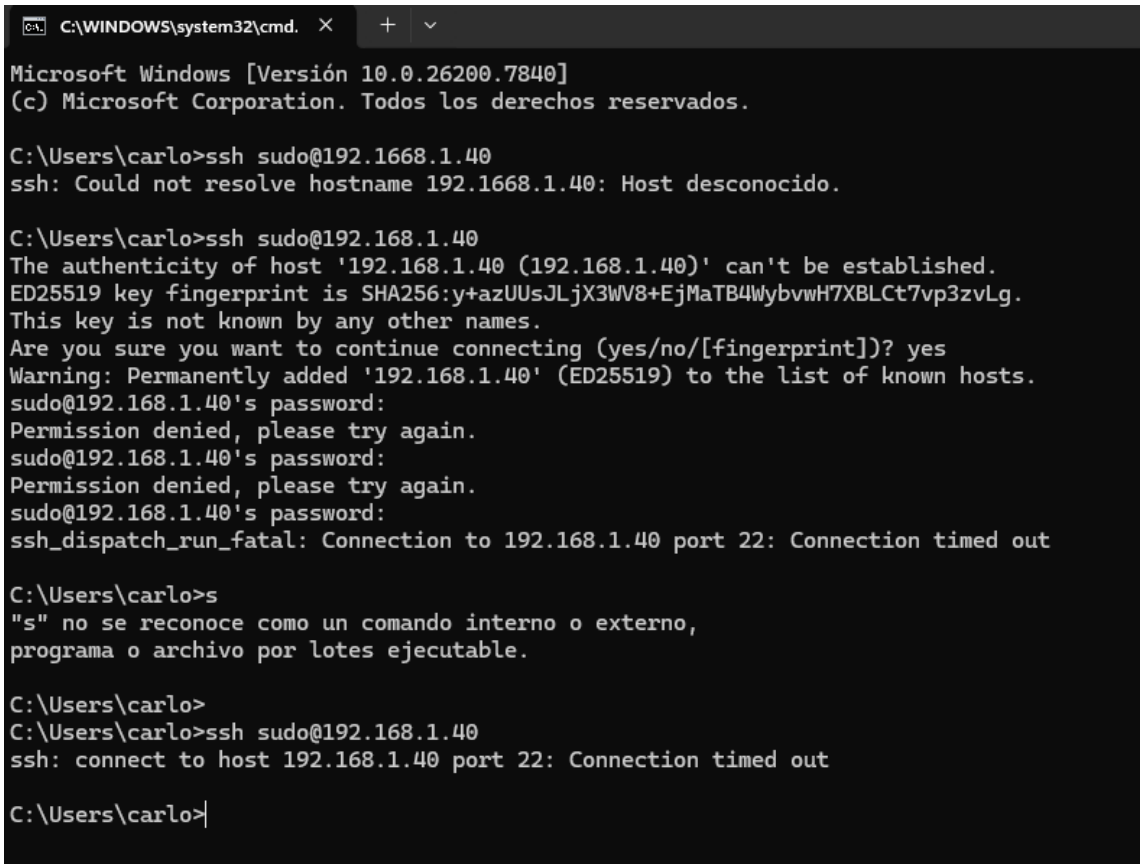
- Defensa Activa (Fail2Ban): Instalación y configuración del servicio Fail2Ban. Se definió una "jail" para SSH que bloquea la IP del atacante tras 3 intentos fallidos en una ventana de 10 minutos, aplicando una regla de rechazo en el firewall por 1 hora.


Applications Places System
debian@debian: ~
File Edit View Search Terminal Help
debian@debian:~$ sudo apt install fail2ban
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
fail2ban is already the newest version (1.0.2-2).
The following packages were automatically installed and are no longer required:
libdaxctl1 libndctl6 libpmem1 linux-image-6.1.0-22-amd64 linux-image-6.1.0-23-amd64
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
debian@debian:~$


Applications Places System
debian@debian: ~
File Edit View Search Terminal Help
debian@debian:~$ sudo nano /etc/fail2ban/jail.local
debian@debian:~$ sudo systemctl restart fail2ban
debian@debian:~$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
| |- Currently failed: 0
| |- Total failed: 0
| `- Journal matches: _SYSTEMD_UNIT=sshd.service + _COMM=sshd
`- Actions
|- Currently banned: 1
|- Total banned: 1
`- Banned IP list: 192.168.1.44
debian@debian:~$


Applications Places System
debian@debian: ~
File Edit View Search Terminal Help
debian@debian:~$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
| |- Currently failed: 0
| |- Total failed: 0
| `- Journal matches: _SYSTEMD_UNIT=sshd.service + _COMM=sshd
`- Actions
|- Currently banned: 1
|- Total banned: 1
`- Banned IP list: 192.168.1.44
debian@debian:~$





4.2. Seguridad en la red

- Incidente:
  Detección de protocolos de comunicación en texto plano: HTTP (puerto 80) y FTP (puerto 21), permitiendo la intercepción de credenciales y datos.
- Soluciones
  - Cambiar estos protocolos sin cifrado por sus análogos con cifrado.
    - HTTP → HTTPS
      Actualmente, el servidor web opera bajo el protocolo HTTP (puerto 80), el cual transmite toda la información en texto plano. Esto incluye datos sensibles de los usuarios, credenciales de administración de WordPress y cookies de sesión. La ausencia de cifrado expone a la organización a ataques de interceptación de datos (Sniffing) y ataques de hombre en el medio (Man-in-the-Middle).

Se urge la migración inmediata hacia el protocolo HTTPS mediante la implementación de un certificado SSL/TLS. Esta medida garantiza un canal de comunicación cifrado entre el navegador del usuario y el servidor web.

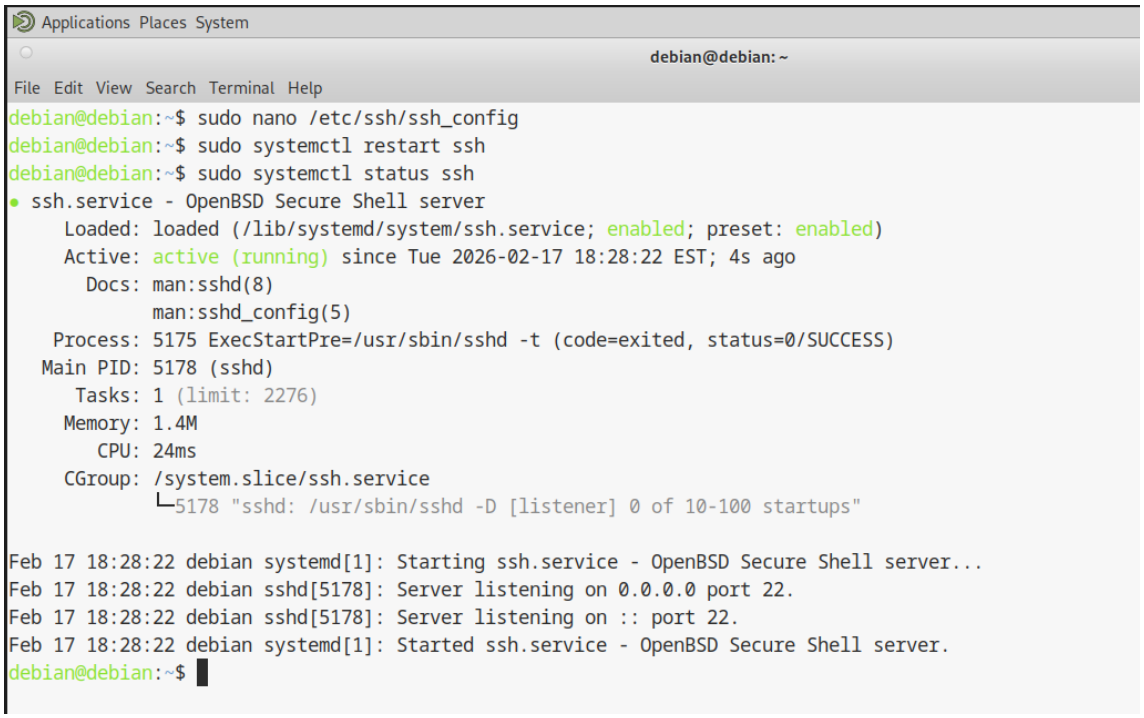
- FTP → SFTP:
  Se ha deshabilitado el servicio FTP inseguro y se ha migrado la operativa al protocolo SFTP (sobre SSH, puerto 22), asegurando que tanto la autenticación como la transferencia de datos viajen por un túnel cifrado.

```
debian@debian:~$ sudo apt purge vsftpd -y
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following packages were automatically installed and are no longer required:
  libdaxctl1 libndctl6 libpmem1 linux-image-6.1.0-22-amd64 linux-image-6.1.0-23-amd64
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  vsftpd*
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.
After this operation, 351 kB disk space will be freed.
(Reading database ... 176905 files and directories currently installed.)
Removing vsftpd (3.0.3-13+b2) ...
Processing triggers for man-db (2.11.2-2) ...
(Reading database ... 176851 files and directories currently installed.)
Purging configuration files for vsftpd (3.0.3-13+b2) ...
```

```
Session  Actions  Edit  View  Help
debian@debian: ~   kali@kali: ~
debian@debian:~$ sudo nano /etc/ssh/sshd_config
debian@debian:~$ sudo systemctl restart ssh
debian@debian:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
     Active: active (running) since Tue 2026-02-17 19:14:40 EST; 5s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 10766 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 10768 (sshd)
      Tasks: 1 (limit: 2276)
     Memory: 1.4M
        CPU: 26ms
     CGroup: /system.slice/ssh.service
             └─10768 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Feb 17 19:14:40 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server ...
Feb 17 19:14:40 debian sshd[10768]: Server listening on 0.0.0.0 port 22.
Feb 17 19:14:40 debian sshd[10768]: Server listening on :: port 22.
Feb 17 19:14:40 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
debian@debian:~$
```

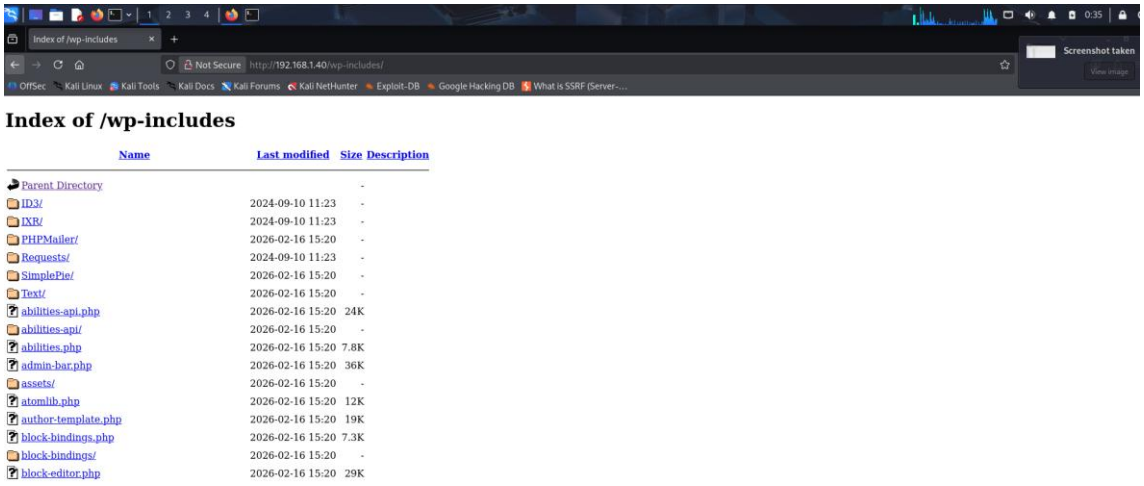
4.3. Acceso directo Root por SSH

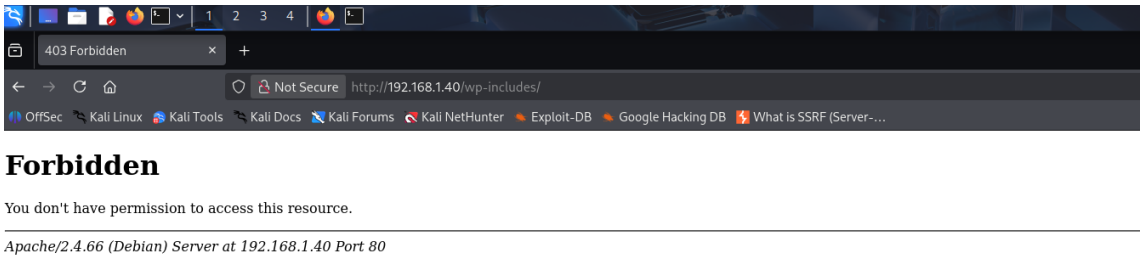
- Incidente:
  Configuración permisiva que permitía el inicio de sesión remoto directo del superusuario.
- Restricción de Privilegios:
  Se modificó el archivo /etc/ssh/sshd_config estableciendo la directiva PermitRootLogin no. Esto obliga a los administradores a autenticarse primero con un usuario estándar y escalar privilegios posteriormente, garantizando la trazabilidad y auditoría de las acciones.

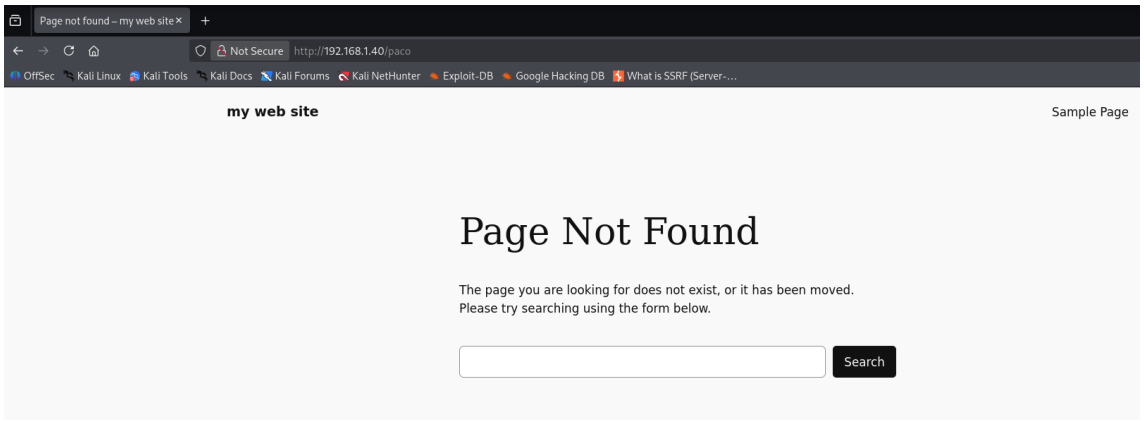


4.4. Enumeración de directorios

- Incidente:
  El servidor web listaba el contenido de directorios sensibles al no encontrar un archivo índice, exponiendo la estructura interna del sitio.
- Soluciones:
  - Hardening de Apache: Se aplicó la directiva Options -Indexes en la configuración global de Apache y se reforzó mediante reglas en el archivo .htaccess, devolviendo un error "403 Forbidden" ante intentos de exploración de carpetas.

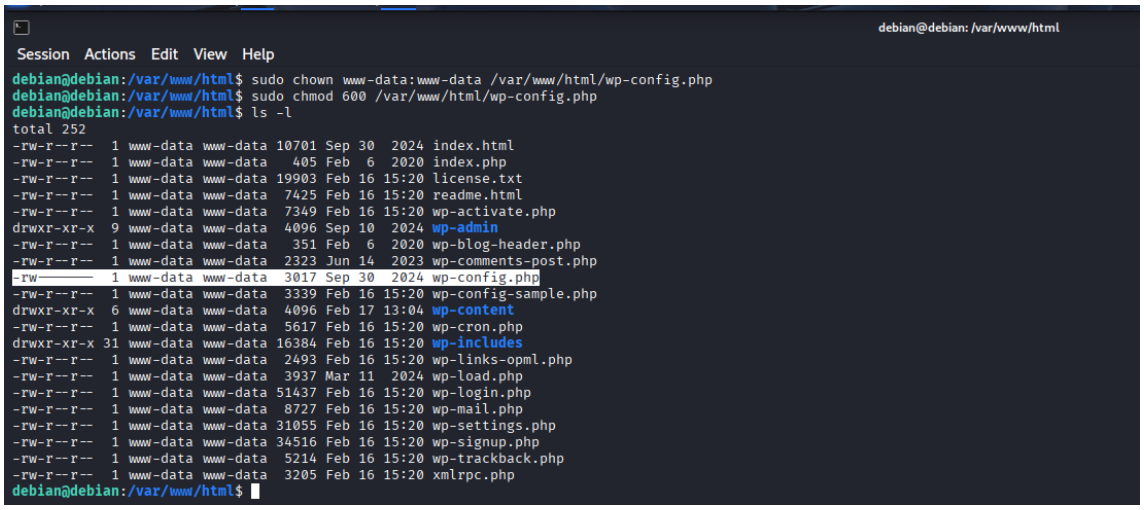






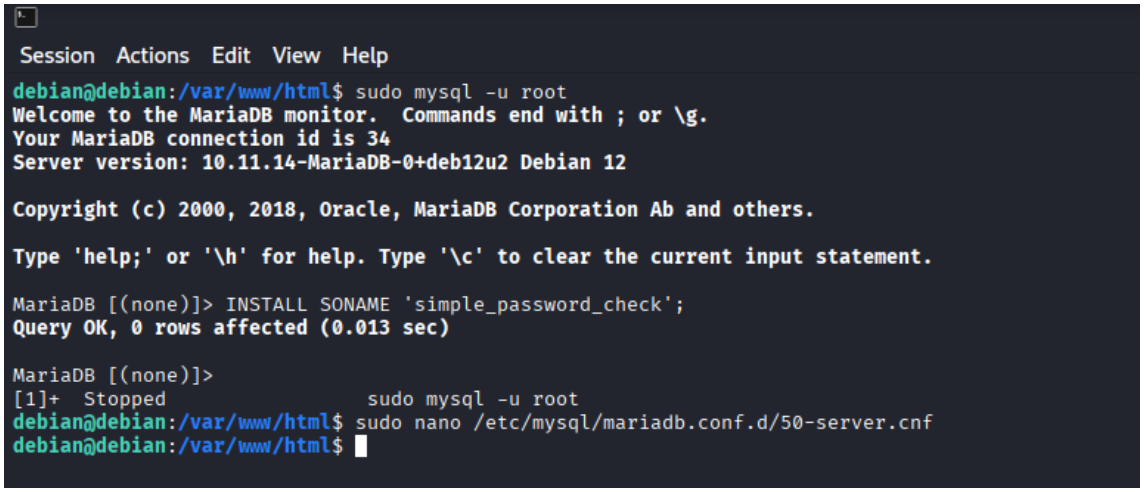
4.5. Exposición de credenciales en wp-config

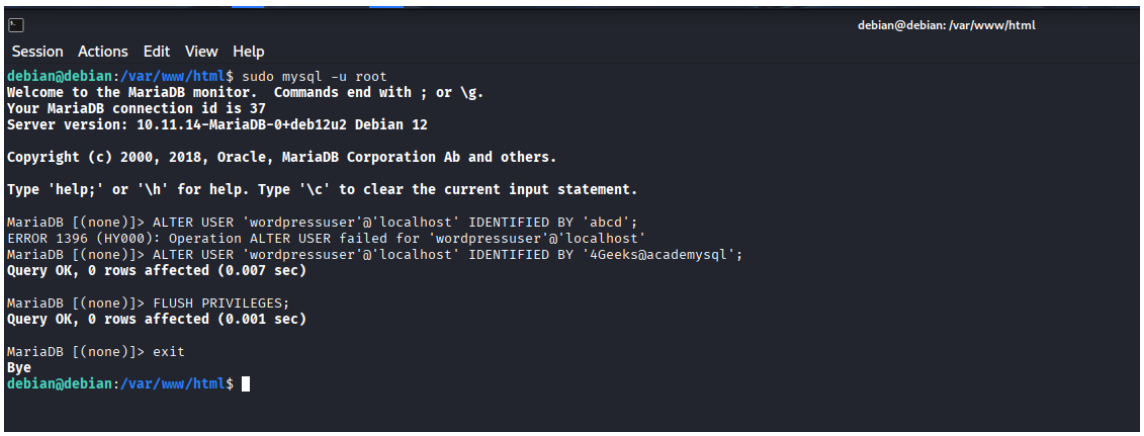
- Incidente:
  Credenciales de base de datos legibles por cualquier usuario del sistema dentro del archivo de configuración.
- Soluciones:
  - Permisos de Archivo: Se modificaron los permisos del archivo wp-config.php a 600 (chmod 600), restringiendo la lectura y escritura exclusivamente al propietario del archivo (usuario del servidor web), impidiendo el acceso a otros usuarios locales o atacantes con acceso parcial.



4.6. Explotación de la base de datos MySQL

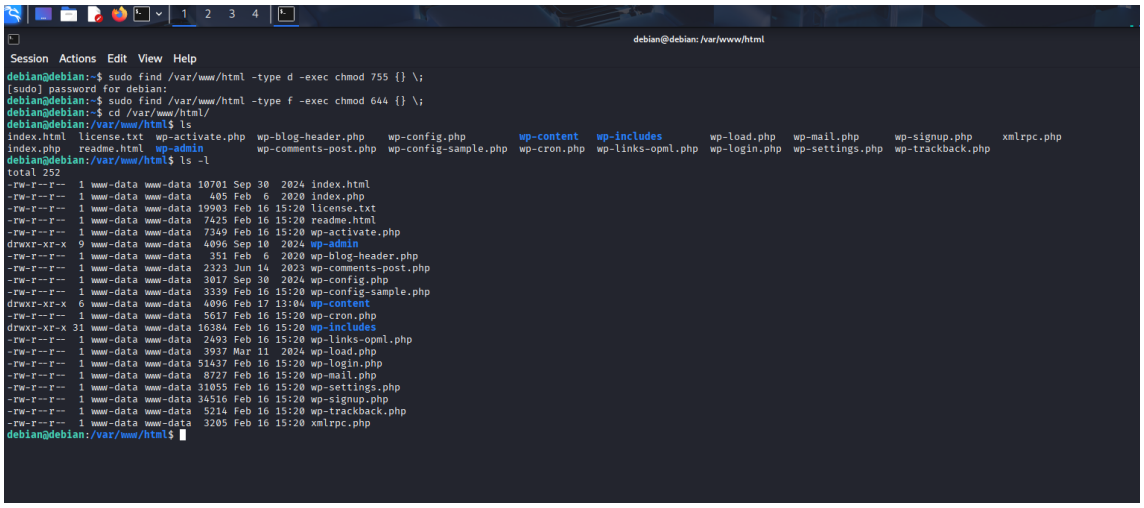
- Incidente:
  Acceso a la base de datos mediante conexión con el cliente MySQL con credenciales comprometidas.
- Soluciones:
  - Saneamiento de Credenciales: Se modificó la contraseña del usuario de la aplicación por una generada aleatoriamente con alta entropía.
  - Aplicar políticas de contraseña.



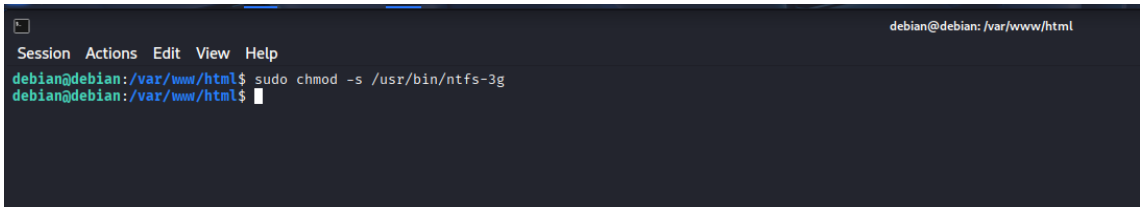


4.7. Integridad del sistema de archivos

- Incidente:
  Detección de archivos con permisos universales (777) y binarios innecesarios con bit SUID activo.
- Soluciones:
  - Corrección Masiva de Permisos: Se ejecutó un script de normalización estableciendo permisos 755 para directorios y 644 para archivos en la raíz web, eliminando la capacidad de escritura pública.



  - Eliminación de bit SUID: Se auditó y retiró el bit de set-user-ID (chmod -s) en binarios no esenciales (ej. ntfs-3g) para mitigar riesgos de escalada de privilegios local.



4.8. Configuraciones inseguras del servidor

- Incidente:
  El servidor revelaba versiones exactas del software (Apache/Debian) en las cabeceras HTTP (Banner Grabbing).
- Soluciones:
  - Ocultación de Banner: Se configuraron las directivas ServerTokens Prod y ServerSignature Off en Apache. Esto aplica el principio de "seguridad por oscuridad", entregando la mínima información posible al atacante y dificultando la búsqueda de exploits específicos por versión.

```
┌──(kali㉿kali)-[~]
└─$ curl -I http://192.168.1.40
HTTP/1.1 200 OK
Date: Wed, 18 Feb 2026 00:04:40 GMT
Server: Apache/2.4.66 (Debian)
Last-Modified: Mon, 30 Sep 2024 14:44:22 GMT
ETag: "29cd-623573d915b52"
Accept-Ranges: bytes
Content-Length: 10701
Vary: Accept-Encoding
Content-Type: text/html

┌──(kali㉿kali)-[~]
└─$
```

```
Session  Actions  Edit  View  Help
debian@debian: ~   kali@kali: ~
debian@debian:~$ sudo nano /etc/apache2/conf-enabled/security.conf
debian@debian:~$ sudo systemctl restart apache2
debian@debian:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Tue 2026-02-17 19:05:31 EST; 4s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 10554 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 10559 (apache2)
      Tasks: 6 (limit: 2276)
     Memory: 16.2M
        CPU: 86ms
     CGroup: /system.slice/apache2.service
             ├─10559 /usr/sbin/apache2 -k start
             ├─10561 /usr/sbin/apache2 -k start
             ├─10562 /usr/sbin/apache2 -k start
             ├─10563 /usr/sbin/apache2 -k start
             ├─10564 /usr/sbin/apache2 -k start
             └─10565 /usr/sbin/apache2 -k start

Feb 17 19:05:31 debian systemd[1]: Starting apache2.service - The Apache HTTP Server ...
Feb 17 19:05:31 debian systemd[1]: Started apache2.service - The Apache HTTP Server.
debian@debian:~$
```

```
┌──(kali㉿kali)-[~]
└─$ curl -I http://192.168.1.40
HTTP/1.1 200 OK
Date: Wed, 18 Feb 2026 00:04:40 GMT
Server: Apache/2.4.66 (Debian)
Last-Modified: Mon, 30 Sep 2024 14:44:22 GMT
ETag: "29cd-623573d915b52"
Accept-Ranges: bytes
Content-Length: 10701
Vary: Accept-Encoding
Content-Type: text/html

┌──(kali㉿kali)-[~]
└─$ curl -I http://192.168.1.40
HTTP/1.1 200 OK
Date: Wed, 18 Feb 2026 00:06:08 GMT
Server: Apache
Last-Modified: Mon, 30 Sep 2024 14:44:22 GMT
ETag: "29cd-623573d915b52"
Accept-Ranges: bytes
Content-Length: 10701
Vary: Accept-Encoding
Content-Type: text/html

┌──(kali㉿kali)-[~]
└─$
```

5. Propuesta de mejora

Para llevar la seguridad de este servidor a un nivel corporativo, se proponen las siguientes implementaciones adicionales que no se cubrieron en esta primera fase:

5.1. Implementación de WAF (Web Application Firewall)

- Situación Actual:
  El servidor protege la red (puertos), pero no inspecciona el tráfico HTTP/HTTPS en busca de ataques específicos a la web (SQL Injection, XSS).
- Propuesta:
  Desplegar ModSecurity con las reglas OWASP Core Rule Set (CRS). Esto filtraría peticiones maliciosas antes de que lleguen a WordPress.

5.2. Autenticación de Doble Factor (2FA) en SSH

- Situación Actual:
  La seguridad depende de una contraseña robusta. Si esta es robada (keylogger), el acceso es total.
- Propuesta:
  Integrar Google Authenticator (libpam-google-authenticator) en el login de SSH. Incluso si el atacante tiene la contraseña, necesitará el código temporal del móvil del administrador.

5.3. Certificados SSL de Confianza (Let's Encrypt)

- Situación Actual:
  Se utiliza un certificado autofirmado, lo que genera advertencias de "Sitio no seguro" en los navegadores, afectando la confianza del usuario.
- Propuesta:
  Automatizar la gestión de certificados con Certbot (Let's Encrypt) para obtener un candado verde válido y renovación automática cada 90 días.

5.4. Sistema de Auditoría y Monitorización (SIEM/Logs)

- Situación Actual:
  Los logs existen, pero no hay alertas proactivas si algo "raro" ocurre fuera de los bloqueos de Fail2Ban.
- Propuesta: Instalar Wazuh para monitorizar cambios en archivos críticos (como /etc/shadow o wp-config.php) en tiempo real y tener una mejor visibilidad y control del sistema.

5.5. Copias de Seguridad Automatizadas

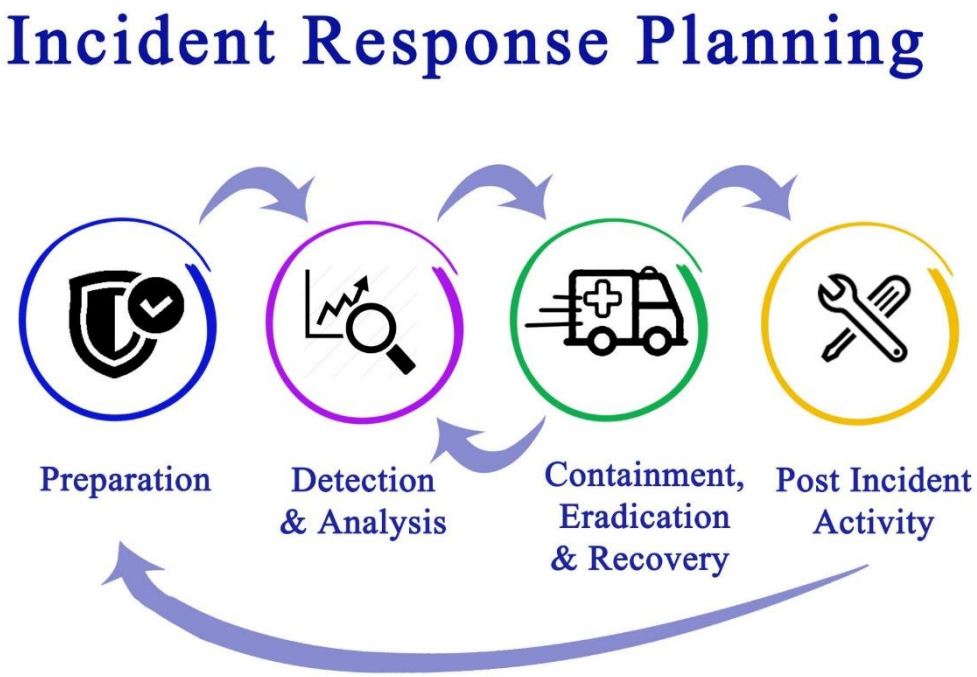
- Situación Actual:
  El servidor es seguro, pero no resiliente ante un borrado accidental o un fallo de disco.
- Propuesta:

Configurar un script (cron) que realice un mysqldump de la base de datos y comprima el directorio /var/www/html enviándolo diariamente a un servidor externo (S3, otro VPS) vía SFTP/Rsync.

Plan de respuesta de incidentes y certificación

6. Plan de Respuesta a Incidentes (NIST SP 800-61)

Este plan establece los procedimientos para detectar, responder y limitar los efectos de un ataque cibernético contra la infraestructura crítica (Servidor Debian / WordPress).



6.1. Fase de Preparación

- **Equipo de Respuesta (CSIRT):**
  - **Líder:** Responsable de Seguridad (CISO) - Toma decisiones críticas (apagar servidor, notificar legal).
  - **Técnico:** Administrador de Sistemas (SysAdmin) - Ejecuta comandos de contención y análisis.
  - **Comunicación:** Encargado de notificar a usuarios afectados.
- **Herramientas Preventivas Implementadas:**
  - **Monitorización:** Logs centralizados (/var/log/auth.log, syslog) y alertas de Fail2Ban.
  - **Hardening:** Servidor bastionado (SSH securizado, permisos corregidos).
  - **Sincronización:** NTP configurado para asegurar la integridad de la línea de tiempo forense.

6.2. Fase de Detección y Análisis (Detection & Analysis)

- **Vectores de Ataque Identificados (Escenario Anterior):**

  - Fuerza bruta contra SSH (Hydra).
  - Explotación de vulnerabilidades Web (Directory Listing, exposición de wp-config.php).

- **Indicadores de Compromiso (IoCs):**
  - **Sistema:** Alto consumo de CPU/RAM injustificado (posible criptominado).
  - **Red:** Múltiples conexiones fallidas desde una misma IP (Alertas de Fail2Ban).
  - **Integridad:** Cambios en ficheros críticos detectados por Tripwire o AIDE.
  - **Logs:** Presencia de "Accepted password for root" en horas inusuales o IPs desconocidas.
- **Procedimiento de Análisis:**
  - Validar la alerta (¿es un falso positivo?).
  - Clasificar la severidad (Baja, Media, Crítica). El acceso root no autorizado se clasifica como Crítico.
  - Documentar cada hallazgo (capturas de pantalla, copias de logs).

6.3. Fase de Contención, Erradicación y Recuperación

6.3.1. Contención (Frenar el ataque)

- **Contención a Corto Plazo:**
  - Bloquear la IP atacante inmediatamente: sudo iptables -A INPUT -s <IP_ATACANTE> -j DROP.
  - Si el atacante tiene acceso root: Desconectar el servidor de la red (cable o vSwitch).
- **Contención a Largo Plazo:**
  - Cambiar todas las contraseñas administrativas (SSH, Base de Datos, WordPress) inmediatamente.
  - Aplicar reglas de firewall restrictivas (Allowlist) solo para IPs de administración conocidas.

6.3.2. Erradicación (Eliminar la amenaza)

- Identificar y eliminar binarios maliciosos (Rootkits/Malware) usando rkhunter o chkrootkit.
- Eliminar cuentas de usuario desconocidas creadas por el atacante (/etc/passwd).
- Parchear la vulnerabilidad explotada (ej. actualizar WordPress, corregir permisos 777).

6.3.3. Recuperación (Restaurar el servicio)

- Restaurar archivos afectados desde una **copia de seguridad limpia** (previa al incidente).
- Reiniciar servicios (systemctl restart apache2, ssh).
- Monitorizar el tráfico intensivamente durante las siguientes 48 horas para detectar persistencia.

6.4. Actividad Post-Incidente (Post-Incident Activity)

- **Reunión de Lecciones Aprendidas:** ¿Qué funcionó? ¿Qué falló?
- **Informe:** Generar un reporte ejecutivo con la línea de tiempo del ataque y el impacto económico/reputacional.
- **Actualización del Plan:** Mejorar las reglas de detección (ej. bajar el umbral de Fail2Ban de 5 a 3 intentos).

7. Mecanismos de Protección de Datos

Para prevenir la recurrencia de ataques similares, se documentan los siguientes controles técnicos:

7.1. Respaldos (Backups) - Estrategia 3-2-1:

- **Frecuencia:** Diaria (Base de datos) y Semanal (Sistema completo).
- **Ubicación:** Una copia local y una copia **inmutable** en la nube (AWS S3 con Object Lock) para protección contra Ransomware.
- **Pruebas:** Realizar un simulacro de restauración trimestral ("No tienes backups si no has probado restaurarlos").

7.2. Cifrado de Información:

- **En Tránsito:** Uso obligatorio de HTTPS (TLS 1.2/1.3) y SFTP.
- **En Reposo:** Cifrado de disco completo con algún sistema de cifrado en el servidor y cifrado de columnas sensibles en la base de datos (ej. tarjetas de crédito, contraseñas hash con bcrypt).

7.3. Control de Acceso (IAM):

- **Principio de Mínimo Privilegio:** Los desarrolladores no tienen acceso root. Solo sudo para tareas específicas.
- **MFA (Autenticación Multifactor):** Implementación futura de Google Authenticator para acceso SSH.

8. Sistema de Gestión de Seguridad de la Información (SGSI - ISO 27001)

8.1. Alcance y Contexto

- **Alcance:** Infraestructura tecnológica que soporta el sitio web corporativo, incluyendo servidor web, base de datos y datos de clientes almacenados.
- **Partes Interesadas:** Clientes (protección de datos), Administración (continuidad de negocio), Equipo Técnico (disponibilidad).

8.2. Análisis de Riesgos

Metodología: Activo -> Amenaza -> Vulnerabilidad -> Impacto.

Activo	Amenaza	Vulnerabilidad	Riesgo	Tratamiento (Plan de Acción)
Base de Datos Clientes	Robo de información (SQL Injection)	Software desactualizado / Código inseguro	Alto	Implementar WAF, Hardening de MySQL, Cifrado de datos.
Servidor Web	Acceso no autorizado (Fuerza Bruta)	SSH expuesto con password débil	Crítico	Fail2Ban, PermitRootLogin no, Política de contraseñas robustas.
Sitio WordPress	Mala configuración	Permisos 777 en /var/www	Medio	Auditoría de permisos (755/644), eliminación de usuarios admin por defecto.

8.3. Políticas de Seguridad

Se definen las siguientes políticas mandatarías para la organización:

- **Política de Seguridad de la Información:** Documento marco aprobado por la dirección que establece el compromiso con la seguridad.
- **Gestión de Acceso de Usuarios:**
    - Regla: "Todo usuario debe tener una cuenta nominativa única. Las contraseñas deben rotarse cada 90 días y cumplir complejidad (12 caracteres)".
    - Regla: "El acceso de root está prohibido vía SSH directo".
- **Copias de Seguridad:**
    - Regla: "La información crítica debe respaldarse cada 24 horas y cifrarse antes de salir del servidor".

- **Continuidad de Negocio:**
    - Plan: "En caso de caída del servidor principal, se activará el servidor de contingencia en un plazo máximo de 4 horas (RTO)".

8.4. Evaluación del Desempeño y Mejora

- **Auditorías Internas:** Semestralmente se ejecutará un escaneo de vulnerabilidades (OWASP/Nessus) y una revisión de permisos.
- **Revisión por la Dirección:** Anualmente, se revisará el SGSI para adaptar el presupuesto de seguridad a nuevas amenazas (ej. auge de IA ofensiva).

9. Conclusión

El presente proyecto de auditoría y reparación de vulnerabilidades ha permitido transformar una infraestructura crítica inicialmente vulnerable en un entorno robusto, resiliente y alineado con los estándares de la industria. A través de una metodología integral que abarca desde el análisis forense y la simulación de amenazas (Red Team) hasta la implementación de controles defensivos y remediación (Blue Team), se ha logrado reducir drásticamente la superficie de ataque del servidor Debian y la aplicación WordPress.

Desde el punto de vista técnico, la transición de protocolos inseguros (HTTP/FTP) hacia canales cifrados (HTTPS/SFTP), junto con el endurecimiento de los servicios de acceso (SSH/Fail2Ban) y la gestión granular de privilegios en el sistema de archivos y base de datos, ha mitigado eficazmente los vectores de ataque más comunes, como la fuerza bruta, la intercepción de tráfico y la escalada de privilegios.

Sin embargo, la protección de los activos de información no termina con la configuración del servidor. La integración de un Plan de Respuesta a Incidentes basado en la normativa NIST y el diseño de un Sistema de Gestión de Seguridad de la Información (SGSI) conforme a la norma ISO 27001, dotan a la organización de la madurez necesaria para gestionar riesgos, responder ante crisis y garantizar la continuidad del negocio.

En definitiva, este trabajo evidencia una realidad fundamental en la ciberseguridad: la seguridad no es un producto, es un proceso. La implementación de este SGSI garantiza que las medidas técnicas aplicadas (Hardening) se mantengan en el tiempo y evolucionen frente a nuevas amenazas, asegurando así la integridad, confidencialidad y disponibilidad de la información a largo plazo.

10. Glosario

10.1. Banner Grabbing

Técnica utilizada por atacantes para obtener información sobre un sistema informático (como el nombre y la versión del software del servidor) analizando los mensajes de bienvenida o cabeceras que envía el servicio. En este proyecto, se mitigó ocultando la versión de Apache.

10.2. Hardening

Proceso de asegurar un sistema reduciendo su superficie de vulnerabilidad. Incluye acciones como cambiar contraseñas predeterminadas, eliminar software innecesario, deshabilitar servicios no utilizados y corregir permisos de archivos.

10.3. Blue Team

Equipo de seguridad defensiva responsable de proteger la infraestructura de la organización. Su labor incluye la monitorización de logs, la configuración de firewalls (Fail2Ban), la gestión de parches y la respuesta ante incidentes.

10.4. CIA (Confidencialidad, Integridad y Disponibilidad)

Triángulo fundamental de la seguridad de la información. Modelo que guía las políticas de seguridad para asegurar que los datos solo sean accesibles por autorizados (C), no sean alterados (I) y estén accesibles cuando se necesiten (A).

10.5. CVE (Common Vulnerabilities and Exposures)

Lista de vulnerabilidades de seguridad cibernética divulgadas públicamente. Se utiliza para identificar y catalogar fallos de seguridad en software específico.

10.6. Directory Listing (Listado de Directorios)

Vulnerabilidad en servidores web que ocurre cuando, ante la ausencia de un archivo índice (index.php/html), el servidor muestra una lista completa de todos los archivos y carpetas del directorio, exponiendo información sensible.

10.7. Fail2Ban

Herramienta de prevención de intrusiones que monitoriza los archivos de registro (logs) en busca de actividades sospechosas, como múltiples intentos fallidos de contraseña, y actualiza las reglas del firewall para bloquear las direcciones IP atacantes.

10.8. Forense Digital

Rama de la ciencia forense que abarca la recuperación e investigación de material encontrado en dispositivos digitales. Se utilizó en la primera fase para analizar cómo se comprometió el servidor.

10.9. Fuerza Bruta

Método de ataque que consiste en probar sistemáticamente todas las combinaciones posibles de contraseñas hasta encontrar la correcta. Herramientas como Hydra automatizan este proceso.

10.10. Hydra

Herramienta de hacking utilizada por el Red Team para realizar ataques rápidos de fuerza bruta contra servicios de autenticación como SSH, FTP o bases de datos.

10.11. IoC (Indicador de Compromiso)

Evidencia digital (como una IP sospechosa, un hash de archivo malicioso o una entrada extraña en un log) que indica que un sistema ha sido comprometido.

10.12. ISO 27001

Norma internacional que permite el aseguramiento, la confidencialidad e integridad de los datos y de la información, así como de los sistemas que la procesan, mediante la implementación de un Sistema de Gestión de Seguridad de la Información (SGSI).

10.13. Man-in-the-Middle (MitM)

Tipo de ciberataque donde el atacante intercepta secretamente y posiblemente altera la comunicación entre dos partes que creen que se están comunicando directamente entre sí. El uso de FTP y HTTP aumenta este riesgo.

10.14. NIST SP 800-61

Guía desarrollada por el Instituto Nacional de Estándares y Tecnología (EE.UU.) que proporciona instrucciones paso a paso para establecer y gestionar un plan de respuesta a incidentes de seguridad informática.

10.15. PAM (Pluggable Authentication Modules)

Mecanismo en sistemas Linux que integra múltiples esquemas de autenticación de bajo nivel en una API de alto nivel. Se utilizó pam_pwquality para imponer políticas de complejidad de contraseñas.

10.16. Principio de Mínimo Privilegio

Concepto de seguridad que establece que un usuario o proceso solo debe tener los permisos estrictamente necesarios para realizar su función, y nada más (ej. crear un usuario específico para la base de datos en lugar de usar root).

10.17. Red Team

Equipo de seguridad ofensiva que simula ataques reales para evaluar la eficacia de las defensas de una organización. En este proyecto, se encargó de la auditoría y explotación de vulnerabilidades.

10.18. SGSI (Sistema de Gestión de Seguridad de la Información)

Conjunto de políticas y procedimientos para gestionar sistemáticamente los datos sensibles de una organización. Es el núcleo de la norma ISO 27001.

10.19. SUID (Set User ID)

Permiso especial en Linux que permite a un usuario ejecutar un archivo con los permisos del propietario del archivo (generalmente root). Si se configura mal, puede permitir la escalada de privilegios.

11. Referencias

11.1. Normativas y Estándares Internacionales (Para la parte de Gestión/SGSI)

- National Institute of Standards and Technology (NIST)
- International Organization for Standardization (ISO)
- OWASP Foundation

11.2. Documentación Técnica

- Fail2Ban Project
  - GitHub
  - RaiolaNetworks
- 4Geeks Academy

12. Anexo

El presente informe se entrega acompañado de un repositorio digital que contiene las evidencias forenses, los registros de auditoría y el estado final del servidor tras el proceso de bastionado. A continuación, se detalla la estructura de dicho repositorio:

12.1. Inventario de Activos Digitales

- **Máquina Virtual (Post-Hardening):**
    - Archivo de imagen (OVA/VMDK) del servidor Debian 12 con todas las medidas de seguridad y correcciones implementadas (Blue Team) ya aplicadas.
- **Evidencias Gráficas (Imágenes):**
    - Carpeta que contiene la totalidad de las capturas de pantalla generadas durante las fases de Análisis Forense, Red Team y Blue Team, numeradas según el índice del proyecto.
- **Registros del Sistema (Logs):**
    - Copia de seguridad de los ficheros de log originales analizados durante la auditoría (auth.log, syslog, access.log, error.log), preservando la cadena de custodia para su revisión.

A continuación, se facilitan las credenciales de acceso actualizadas para la validación de las medidas de seguridad implementadas en la Máquina Virtual entregada:

Ámbito / Servicio	Usuario	Contraseña	Notas
Sistema Operativo (Debian)	debian	4Geeks@cademy	Usuario con permisos sudo.
Base de Datos (MariaDB)	wordpressuser	4Geeks@academysql	Acceso local restringido.