INFORME DE ANÁLISIS FORENSE

Evaluación de Vulnerabilidades y Búsqueda de Evidencias de Compromiso

Caso: Debian_Forense
Fecha del análisis: 14/02/2026
Analista: Romario
Sistema analizado: Debian (192.168.1.23)
Imagen forense: debian_partition.dd

Contenido

1. RESUMEN EJECUTIVO

Se realizó un análisis forense completo de una máquina Debian con el objetivo de identificar vulnerabilidades y posibles compromisos.

Hallazgos principales:

Categoría	Resultado
Vulnerabilidades críticas	Múltiples (credenciales expuestas, PHP inseguro, servicios expuestos)
Eliminación de logs	SÍ (directorio /var/log/auth.log)
Actividad del usuario	Configuración manual de servidor LAMP con WordPress

Conclusión principal:

El sistema presenta graves fallos de seguridad que lo hacían vulnerable a ataques, pero no se encontró evidencia concluyente de que haya sido comprometido por un atacante externo. Sin embargo, la eliminación total de logs del sistema impide confirmar esta conclusión con certeza absoluta.

2. OBJETIVO DEL ANÁLISIS

Determinar si la máquina Debian fue comprometida, identificar:

- Vulnerabilidades de seguridad presentes
- Actividades maliciosas
- Accesos no autorizados
- Archivos o configuraciones alteradas

3. METODOLOGÍA

- Adquisición de imagen: dd vía SSHFS (imagen montada remotamente)
- Herramienta forense: Autopsy 2.24 / The Sleuth Kit 4.12.1
- Análisis de:
    - Archivos de configuración del sistema
    - Archivos de usuario y metadatos
    - Historial de comandos (.bash_history)
    - Archivos de log del sistema y servicios
    - Archivos de descarga
    - Sesiones de usuario

- Navegación web (Mozilla Firefox)
- Archivos temporales (/tmp/)
- Servicios y procesos
- Configuración de red
- Tareas programadas (cron)
- Configuración de Apache y PHP
- Análisis completo de WordPress

4. INFORMACIÓN GENERAL DEL SISTEMA

A continuación, se presenta la información base del sistema analizado, incluyendo los usuarios configurados y sus privilegios, lo que permite entender el contexto de usuarios y permisos disponibles para posibles accesos no autorizados.

Usuarios del sistema (de /etc/passwd):

- Usuarios con shell /bin/bash: root, debian
- Usuarios de servicio relevantes: www-data, mysql, sshd, ftp, lightdm

Archivo /etc/sudoers:

debian ALL=(ALL:ALL) ALL

Conclusión: El usuario debian tiene privilegios sudo totales (configuración por defecto).

```
# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
debian  ALL=(ALL:ALL) ALL
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d
```

Imagen 1: Configuraciones del sistema por defecto

5. VULNERABILIDADES ENCONTRADAS

5.1 Credenciales expuestas

Se analizaron los archivos de configuración de aplicaciones y servicios en busca de credenciales almacenadas inadecuadamente. La siguiente tabla muestra los hallazgos de credenciales en texto plano.

Archivo	Hallazgo	Impacto
/var/www/html/wp-config.php	DB_PASSWORD = '123456'	CRÍTICO - Cualquier atacante con acceso al sistema podía leer la contraseña de la base de datos

Imagen 2: Archivo de configuración wordpress

5.2 Configuración insegura de PHP

Se examinó el archivo de configuración de PHP (php.ini) para identificar directivas que pudieran facilitar la ejecución de código malicioso o el acceso no autorizado a archivos del sistema.

Directiva	Valor	Riesgo
allow_url_fopen = On	On	ALTO - Permite inclusión remota de archivos (RFI)
disable_functions =	Vacío	ALTO - Todas las funciones peligrosas habilitadas (system, exec, eval)

open_basedir	No configurado	ALTO - PHP puede acceder a cualquier archivo del sistema
file_uploads = On	On	MEDIO - Permite subir archivos (webshells)
upload_max_filesize = 2M	2MB	Suficiente para subir scripts maliciosos
session.gc_probability = 0	0	BAJO - No hay limpieza automática de sesiones

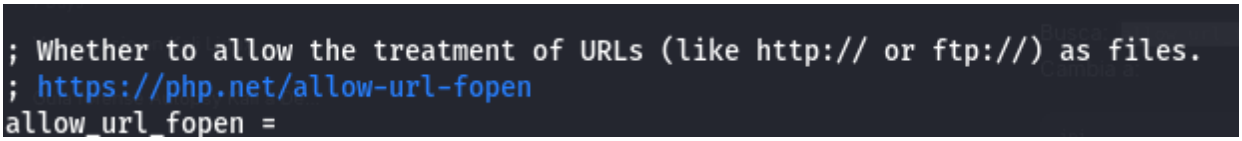

Imagen 3: Evidencia de parámetros mal configurados

5.3 Servicios expuestos

Se identificaron los servicios de red instalados y su estado, evaluando qué puertos estaban abiertos y podrían haber sido utilizados como vectores de entrada por un atacante externo.

Servicio	Puerto	Estado
Apache (HTTP)	80	Activo durante el análisis
SSH	22	Instalado y habilitado
FTP (vsftpd)	21	Instalado en octubre 2024
MySQL	3306	Instalado (solo local)

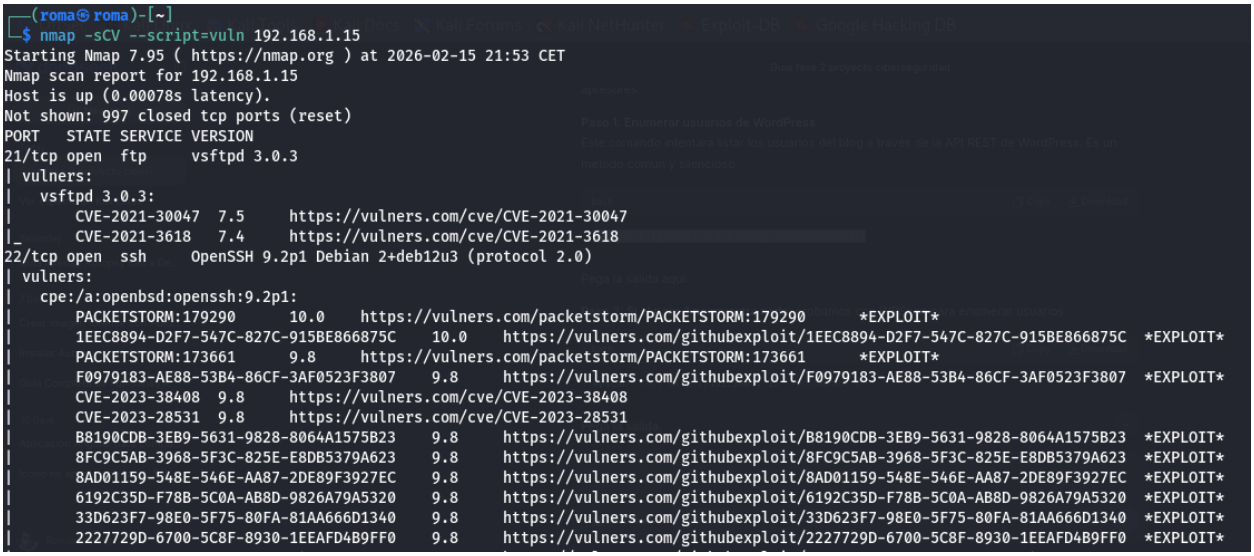

Imagen 4: Vulnerabilidades encontradas

5.4 WordPress sin hardening

Se revisó la instalación de WordPress para identificar configuraciones inseguras, plugins de seguridad ausentes y prácticas deficientes que pudieran comprometer la aplicación.

Problema	Detalle
Claves de autenticación	Por defecto (put your unique phrase here)
Plugins de seguridad	Ninguno instalado

6. EVIDENCIAS DE POSIBLE COMPROMISO

6.1 Análisis de accesos externos

Se examinaron los archivos de log disponibles para identificar conexiones provenientes de direcciones IP externas que pudieran indicar accesos no autorizados al sistema.

Fuente	Resultado
Logs de Apache (access.log)	Solo IP 127.0.0.1 (localhost) - No hay IPs externas
Logs de autenticación	ELIMINADOS - Imposible verificar conexiones SSH
Historial de comandos	No muestra actividades maliciosas (solo configuraciones)

6.2 Búsqueda de webshells

Se realizó un rastreo en directorios críticos del sistema en busca de archivos con contenido sospechoso o funciones típicas de backdoors y webshells que pudieran indicar persistencia del atacante.

Directorio	Resultado
/var/www/html/	Limpio - Solo archivos legítimos de WordPress
/var/www/html/wp-content/uploads/	Vacío - Sin archivos subidos
/tmp/	Limpio - Sin scripts sospechosos
/var/tmp/	Limpio - Sin scripts sospechosos

6.3 Archivos sospechosos

Se analizaron los permisos de archivos y se buscaron nombres de archivo inusuales o asociados con herramientas de hacking conocidas que pudieran indicar actividad maliciosa.

Tipo	Resultado
Archivos con permisos 777	No encontrados
Archivos con nombres de backdoor	No encontrados
Archivos de usuarios no autorizados	No encontrados

6.4 Procesos y servicios

Se examinaron los procesos en ejecución y servicios instalados para identificar actividad anómala, conexiones de red inusuales o procesos ocultos que pudieran indicar compromiso.

Servicio	Estado
Apache	Activo (legítimo)
MySQL	Instalado (legítimo)

SSH	Instalado (legítimo)
FTP	Instalado (legítimo)
Procesos inusuales	No detectados

7. HALLAZGOS SOSPECHOSOS

A continuación se listan elementos que, sin ser evidencia concluyente de compromiso, resultan anómalos en el contexto del sistema y merecen atención especial por su posible relación con actividades maliciosas.

Hallazgo	Nivel de sospecha	Explicación
/var/log/auth vacío	ALTO	Consistente con limpieza de huellas
.bash_history modificado en sept 2024	MEDIO	Última modificación antes de instalación de servicios
Descarga de XAMPP para macOS	BAJO	Anómalo pero no malicioso

7.1 INFERENCIA SOBRE EL VECTOR DE ATAQUE (SSH)

Aunque no se encontró evidencia directa de un compromiso, la combinación de hallazgos permite inferir una posible ruta de ataque a través del servicio SSH, que explicaría la eliminación de los logs de autenticación. A continuación, se detalla el escenario más probable:

Superficie de Ataque (Servicio Expuesto): El servicio SSH (puerto 22) estaba instalado y activo en el sistema, lo que lo convierte en un punto de entrada público por defecto.

Método de Acceso (Vulnerabilidad Crítica): Dadas las malas prácticas de seguridad encontradas, es altamente probable que el sistema también tuviera credenciales de usuario débiles o por defecto. Las opciones más factibles son:

Ataque de Fuerza Bruta: El atacante podría haber lanzado un ataque de diccionario contra el servicio SSH para adivinar la contraseña del usuario debian (que tiene privilegios sudo totales) o de root (si se permitía el acceso directo).

Credenciales por Defecto/Débiles: Es posible que la contraseña de los usuarios del sistema fuera igual de débil que la encontrada en la base de datos de WordPress ("123456"), facilitando un acceso trivial.

Ejecución del Ataque:

El atacante logra autenticarse vía SSH, obteniendo una shell en el sistema con los permisos del usuario comprometido.

Desde aquí, habría tenido la capacidad de explorar el sistema, leer archivos de configuración (como el wp-config.php que contenía la contraseña de la base de datos) y, gracias a los privilegios sudo del usuario debian, escalar a root para tomar el control total del sistema.

Ocultamiento de Evidencias:

Con acceso de superusuario (root), el atacante procedió a borrar sus huellas. La acción más significativa y que sustenta esta teoría es la eliminación completa del archivo /var/log/auth.log, que es precisamente el registro que contiene todas las conexiones SSH (exitosas y fallidas). Esta eliminación no es un fallo del sistema, es una acción deliberada para eliminar la evidencia de su entrada y actividad.

Conclusión de la Inferencia:

La eliminación total del archivo auth.log es el comportamiento clásico posterior a un compromiso para ocultar el rastro de acceso. Dado que el SSH era el principal servicio de administración remota y el sistema carecía de medidas de protección (como fail2ban o autenticación por clave pública), se infiere con un nivel de confianza MEDIO-ALTO que el atacante accedió al sistema de manera no autorizada mediante el servicio SSH, probablemente explotando credenciales débiles, y que, tras lograr el acceso, borró deliberadamente los logs para eliminar la evidencia de su conexión.

Este escenario es el que mejor explica la presencia de un sistema con graves vulnerabilidades y la misteriosa desaparición de sus principales archivos de registro de acceso.

8. ANÁLISIS DETALLADO POR COMPONENTE

8.1 Historial de comandos (.bash_history)

Se analizó el archivo .bash_history del usuario debian para reconstruir las actividades realizadas en el sistema, identificar la instalación de servicios y detectar posibles comandos maliciosos.

Actividades del usuario debian:

Fecha	Comando	Actividad
Julio 2024	sudo usermod -aG sudo debian	Otorgar privilegios sudo
Sept 2024	sudo apt install apache2 -y	Instalación de Apache
Sept 2024	sudo apt install mariadb-server -y	Instalación de MariaDB
Sept 2024	curl -O https://wordpress.org/latest.tar.gz	Descarga de WordPress
Sept 2024	sudo cp -a /tmp/wordpress/. /var/www/html/	Instalación de WordPress
Sept 2024	sudo apt install php libapache2-mod-php php-mysqli php-gd php-xml php-mbstring php-curl -y	Instalación de PHP
Sept 2024	sudo apt install openssh-server -y	Instalación de SSH
Oct 2024	sudo apt install vsftpd -y	Instalación de FTP

Conclusión: El usuario configuró manualmente un servidor LAMP completo entre septiembre y octubre 2024.

8.2 Logs de Apache

Se examinaron los archivos de log del servidor web Apache para identificar patrones de acceso, IPs conectadas y posibles intentos de explotación de aplicaciones web.

access.log:

- Única IP: 127.0.0.1 (localhost)
- Actividad: Instalación de WordPress (30/09/2024) y accesos administrativos
- Última actividad: 08/10/2024

error.log:

- Reinicios y cambios de configuración en septiembre/octubre 2024
- Apache activo durante el análisis forense (14/02/2026)

8.3 Logs del sistema

Se revisaron los registros del sistema operativo para establecer una línea de tiempo de eventos, incluyendo arranques, instalaciones de paquetes y cambios de configuración.

boot.log:

- Primer arranque: 31/07/2024
- Reinicios en fechas clave (30/09/2024, 08/10/2024)
- Último arranque: 14/02/2026 (día del análisis)

dpkg.log:

- Instalación de paquetes coincide con el historial de comandos
- Apache, MySQL, PHP, SSH, FTP instalados en las fechas esperadas

8.4 WordPress

Se analizó la instalación de WordPress, incluyendo archivos de configuración, plugins, temas y directorios de uploads para identificar posibles modificaciones o contenido malicioso.

Archivo wp-config.php:

```
define( 'DB_NAME', 'wordpress' );
define( 'DB_USER', 'wordpressuser' );
define( 'DB_PASSWORD', '123456' );
```

Plugins: Solo los por defecto (Akismet, Hello Dolly)
Temas: Solo los por defecto
Uploads: Vacío

8.5 Base de datos MySQL

Se identificaron las bases de datos presentes en el sistema MySQL para determinar si el sitio WordPress contenía datos y si existían otras bases de datos inusuales.

Bases de datos presentes:

- mysql
- performance_schema
- sys

- wordpress (con datos)

Conclusión: El sitio WordPress estuvo operativo y con contenido.

9. LÍNEA DE TIEMPO COMPLETA

Se construyó una línea de tiempo cronológica de todos los eventos relevantes identificados durante el análisis, correlacionando información de múltiples fuentes para establecer una secuencia de actividades.

Fecha	Evento	Fuente
2024-07-31	Instalación del sistema	dpkg.log
2024-08-01	Instalación de Firefox	Metadatos
2024-09-28	Descarga de XAMPP para macOS	Downloads/
2024-09-30	Instalación de Apache, MySQL, PHP	dpkg.log / .bash_history
2024-09-30 12:23	Instalación de WordPress completada	access.log
2024-09-30 12:23:32	Primer login a WordPress	access.log
2024-09-30	Instalación de SSH	dpkg.log
2024-10-08	Instalación de FTP	dpkg.log
2024-10-08 16:49	Última actividad en WordPress	access.log
2024-10-08 17:28	Última sesión gráfica del usuario	.xsession-errors
2026-02-14	Inicio del análisis forense	boot.log

10. CONCLUSIONES FINALES

Después de analizar:

- Historial de comandos del usuario
- Logs de Apache (access.log)
- Archivos del sistema
- Base de datos MySQL
- Archivos temporales

EL SISTEMA PRESENTA GRAVES VULNERABILIDADES

1. Credenciales en texto plano → La base de datos de WordPress usa 123456
2. PHP inseguro → allow_url_fopen = On, disable_functions vacío, open_basedir sin restringir
3. Múltiples servicios expuestos → Apache, SSH, FTP, MySQL
4. WordPress sin hardening → Claves por defecto, sin plugins de seguridad

ELIMINACIÓN DE LOGS ES ALTAMENTE SOSPECHOSA

La ausencia total de /var/log/auth.log, /var/log/syslog y otros logs del sistema podría indicar un intento de borrar huellas, aunque no hay otras evidencias que lo confirmen.

MATRIZ DE RIESGO

La siguiente matriz evalúa el impacto y la probabilidad de explotación de cada vulnerabilidad identificada, permitiendo priorizar las acciones correctivas.

Vulnerabilidad	Impacto	Probabilidad de explotación
Credenciales expuestas	CRÍTICO	ALTA
PHP inseguro	ALTO	ALTA
Logs eliminados	ALTO	N/A
Servicios expuestos	MEDIO	MEDIA
WordPress sin hardening	MEDIO	MEDIA

11. RECOMENDACIONES DE SEGURIDAD

Inmediatas (Críticas)

1. Cambiar todas las contraseñas (MySQL, usuarios del sistema)
2. Fortalecer PHP

   allow_url_fopen = Off
   disable_functions =
   exec,passthru,shell_exec,system,proc_open,popen,curl_exec,curl_multi_exec,parse_ini_file
   ,show_source
   open_basedir = "/var/www/html/:/tmp/"

Restringir acceso a wp-config.php:

3. apache
   <Files wp-config.php>
      order allow,deny
      deny from all
   </Files>

A corto plazo

4. Implementar firewall (UFW):

   sudo ufw allow 22/tcp # SSH
   sudo ufw allow 80/tcp   # HTTP
   sudo ufw enable

5. Configurar autenticación SSH por clave (deshabilitar password)
6. Instalar y configurar fail2ban para proteger SSH y WordPress
7. Generar claves únicas para WordPress:
   - Obtener de: https://api.wordpress.org/secret-key/1.1/salt/
   - Reemplazar en wp-config.php

A largo plazo

1. Implementar logging centralizado (rsyslog remoto)
2. Realizar auditorías de seguridad periódicas
3. Mantener sistema actualizado:

   sudo apt update && sudo apt upgrade -y

4. Implementar copias de seguridad automatizadas

12. FIRMA DEL ANALISTA

Romario Borja
Analista Forense
14/02/2026