

INFORME FORENSE PRELIMINAR

Elaborado por: Víctor López
Pablo Torres

4GEEKS
ACADEMY

Informe Forense Preliminar INDICE:

- 1. Objetivos del análisis forense**
- 2. Aseguramiento del Entorno Comprometido**
- 3. Cronología Principal**
- 4. Reconocimiento y recolección de evidencias**

4.1 Identificar qué servicios fueron comprometidos.

- **Revisión de Logs**
 - Análisis Detallado de Logs del Sistema (journalctl)
 - Análisis Detallado de Logs de Apache
- **Resumen Ejecutivo**

4.2 Identificar archivos sospechosos

4.3 Informe de Actividad de Procesos y Servicios del Sistema

4.4 Escaneo del servidor para detectar rootkits y malware.

- 5. Mitigación de vulnerabilidades.**
- 6. Revertir posibles cambios realizados.**
- 7. Actualizar y corregir configuraciones de seguridad.**
- 8. Conclusiones.**

Informe Forense Preliminar - Incidente de Seguridad en Servidor Debian – 4Geeks Academy.

Fecha de elaboración: 25 / 04 / 2025

Servidor afectado: Debian - 4Geeks Academy

Dirección IP: 192.168.68.xxx

Periodo de análisis inicial: Julio - Octubre 2024 (basado en logs disponibles)

Elaborado por: Víctor López y Pablo Torres - Analistas de Ciberseguridad

1. Objetivos del análisis forense

El presente informe documenta las actividades iniciales de respuesta a incidentes y el análisis forense preliminar llevado a cabo sobre el servidor Debian, identificado como potencialmente comprometido dentro de la infraestructura de 4Geeks Academy. El objetivo primordial de esta fase inicial ha sido asegurar el entorno afectado, preservar la integridad de la evidencia digital y comenzar la recolección de datos relevantes para determinar la naturaleza, el alcance y el impacto del incidente de seguridad.

2. Aseguramiento del Entorno Comprometido

Ante la detección de actividad anómala y la presunción de compromiso, se procedió de inmediato a implementar medidas de contención críticas para limitar la potencial propagación del ataque y prevenir la alteración o destrucción de la evidencia. Estas medidas incluyeron:

- Aislamiento de la Máquina: Se desconectó el servidor de la red de producción principal y se implementaron controles perimetrales estrictos para aislarlo lógicamente, impidiendo tanto el control remoto continuo por parte del atacante como una posible expansión lateral hacia otros sistemas de la red.
- Preservación de la Evidencia (Imágenes Forenses): Para asegurar un registro inalterado del estado del sistema en el momento de la contención, se procedió a la adquisición de imágenes forenses completas del disco del servidor y en la medida de lo posible, una copia de la memoria RAM, con el objetivo de preservar el estado exacto del sistema en el momento del incidente, incluyendo artefactos volátiles como conexiones activas, procesos en ejecución y posibles puertas traseras.
- Trabajo sobre Copias: Se ha creado un snapshots del estado de la máquina, para garantizar la no alteración de las evidencias originales, el análisis se está llevando a cabo sobre una copia fiel de la máquina virtual afectada. Este enfoque asegura la cadena de custodia de la evidencia digital, permitiendo un análisis exhaustivo sin comprometer la validez forense de los datos originales.

3. Cronología Principal:

⌚ Cronología de eventos

📅 Julio 2024

● [Jul 31 - 15:56] - Inicio del sistema

- El sistema arranca normalmente bajo entorno virtualizado.
 - Se detectan vulnerabilidades en mitigaciones de CPU:
 - RETBleed y Speculative Store Bypass aparecen como “Vulnerable”.
 - AppArmor está habilitado, pero no se detecta TPM (TPM-bypass activo).
 - No hay actividad de usuario relevante aún ni conexiones de red o SSH.
- ✓ Estado: El sistema inicia en condiciones estándar, aunque con vulnerabilidades presentes a nivel de hardware/software, aún no explotadas.

📅 Septiembre 2024

● [Sep 28 - 16:39] - Nuevo arranque del sistema

- Las vulnerabilidades mencionadas en julio persisten.
- El módulo speakup está cargado (accesibilidad por consola).
- No se observan accesos SSH, ni sudo, ni eventos de ejecución de comandos.

✓ Estado: El sistema sigue sin actividad sospechosa. Aparentemente sin compromisos ni actividad de red visible en este punto.

● [Sep 30 - 9:48:32] - Inicio del sistema (kernel 6.1.0-23-amd64) .

- El usuario debian ejecuta sudo apt update correctamente .
- Mismo usuario ejecuta sudo apt upgrade -y .
- Otro reinicio del sistema, ahora con kernel 6.1.0-25-amd64 .
- Múltiples intentos de autenticación fallidos en LightDM con usuario desconocido (user unknown) .
- Ejecución de limpieza de sesiones PHP vía cron como root .

📅 Octubre 2024

● [Oct 08 - 16:08] - Instalación de servidor FTP (vsftpd)

- Se ejecuta apt install vsftpd como root.
- Se crea el usuario ftp con shell /usr/sbin/nologin.
- Se levanta el servicio vsftpd.

➡ Possible canal de exfiltración o persistencia remota.

● [Oct 08 - 16:12] - Instalación y configuración de SSH

- Se instala openssh-server.
- Se modifica el archivo /etc/ssh/sshd_config.

- Se reinicia el servicio SSH.

➡ Indicios de manipulación para permitir login remoto, posiblemente habilitando acceso directo a root.

🟡 [Oct 08 - 16:17] - Permisos peligrosos

- Se otorgan permisos 777 a /var/www/html y a wp-config.php.

```
chmod -R 777 /var/www/html
```

```
chmod 777 /var/www/html/wp-config.php
```

➡ Grave vulnerabilidad: permite ejecución de código arbitrario y modificación de configuración.

🟡 [Oct 08 - 16:21] - Edición de configuración de Apache

- Se edita /etc/apache2/apache2.conf.
- Se reinicia el servicio Apache.

➡ Posible ocultamiento de logs, habilitación de directorios peligrosos, o desactivación de protecciones.

🔴 [Oct 08 - 17:40:59] - Acceso remoto como root desde IP externa

```
sshd[1650]: Accepted password for root from 192.168.0.134 port 45623 ssh2
```

- IP atacante: 192.168.0.134
- Acceso por contraseña como usuario root.
- SSH activado y modificado previamente ese mismo día.

➡ Momento clave del compromiso. Todo indica que el atacante aprovechó configuraciones débiles y/o credenciales comprometidas.

💡 Posible vector de entrada

- Contraseña débil o comprometida para el usuario root.
- Acceso por SSH habilitado y mal configurado.
- Sin medidas de protección como fail2ban, firewall, ni restricciones de IP.

Indicadores de compromiso

Evidencia	Nivel de riesgo	Descripción
Acceso SSH root desde IP externa	Crítico	Entrada remota con privilegios máximos
Permisos 777 en /var/www/html	Crítico	Permite ejecución y escritura sin restricciones
Usuario ftp creado	Medio	Possible uso para transferencia/exfiltración de datos
Modificación de SSH y Apache	Alto	Pueden haber alterado el comportamiento de los servicios para evadir detección
IP sospechosa 192.168.0.134	Crítico	IP pública no autorizada accediendo como root

4. Reconocimiento y recolección de evidencias

Una vez asegurado el entorno, se ha comenzado el proceso de **recolección y análisis de evidencias**, orientado a identificar la naturaleza del incidente, el vector de acceso y los posibles servicios comprometidos.

4.1 Identificar qué servicios fueron comprometidos.

Revisión de Logs

Durante la fase de recolección de evidencia digital, se observó que el servidor Debian afectado no mantiene archivos de registro tradicionales como auth.log o syslog en las ubicaciones estándar del sistema de archivos (/var/log/), al menos no a través del servicio rsyslog, el cual no parece estar activo en este entorno.

En sistemas Debian modernos, la gestión de logs se centraliza en el *Journal* binario, administrado por el servicio systemd-journald. Por lo tanto, toda la información de registro relevante del sistema operativo, servicios como SSH (sshd), Apache (apache2), VSFTPD (vsftpd), etc., está siendo extraída directamente del Journal utilizando el comando journalctl.

Este enfoque permite acceder a un registro completo y estructurado de eventos, compensando la ausencia de los archivos de log planos tradicionales para el análisis forense.

- **Análisis Detallado de Logs del Sistema (journalctl):**

Como hemos mencionado anteriormente, la principal fuente de información de registro del sistema en el servidor debian se obtuvo del Journal binario, accesible a través de la herramienta journalctl. El análisis de los logs correspondientes a los meses de julio, septiembre y octubre de 2024 ha revelado una secuencia de eventos que sugieren un compromiso progresivo y actividades post-exploitación.

Los hallazgos más relevantes, extraídos de diferentes componentes del sistema (kernel, sshd, sudo, systemd, etc.) registrados en el Journal.

- **Análisis Detallado de Logs de Apache:**

El servidor web Apache (apache2) fue identificado durante el análisis de los logs de journalctl como un servicio instalado por el atacante en septiembre de 2024. Dado que los servidores web son un vector común para la intrusión inicial, la persistencia (a través de web shells) y la exfiltración de datos, un análisis exhaustivo de sus

registros de actividad es fundamental para comprender las acciones del atacante a nivel de aplicación web. Esta sección detalla los hallazgos clave obtenidos de la revisión de los logs de acceso (`access.log`) y errores (`error.log`) de Apache, buscando patrones de ataque, actividad anómala o evidencia de compromiso a través de este servicio.

A continuación, detallo los hallazgos principales de cada archivo:

- **access.log.1**
 - **Período cubierto:** 21 de abril de 2025.
 - **Orígenes:** 192.168.68.116 (mencionando 192.168.68.133 como Referer) y 127.0.0.1.
 - **Actividad:** Muestra peticiones estándar a recursos de WordPress y Apache (/wp-admin/, /wp-login.php, /wp-cron.php, archivos JS/CSS/íconos), así como algunas peticiones a la raíz del sitio (/) y archivos de íconos. La actividad es consistente con la operación normal de un sitio WordPress, incluyendo tareas de fondo (wp-cron.php) y acceso al panel de administración, así como navegación al sitio principal.
 - **Relevancia para el incidente:** Este log es de una fecha significativamente posterior al período principal del incidente (octubre de 2024). La actividad registrada parece ser normal para la operación del sitio y no proporciona evidencia directa de las acciones del atacante durante el compromiso inicial.
- **access.log.2.gz**
 - **Período cubierto:** 11 de abril de 2025 (principalmente entre 13:27 y 15:00).
 - **Orígenes:** Predominantemente 127.0.0.1 (localhost), con algunas peticiones de 192.168.68.116 (mencionando 192.168.68.133 como Referer).
 - **Actividad:** Similar a access.log.1, muestra actividad web estándar del sitio y tareas de fondo de WordPress desde IPs internas y localhost.
 - **Relevancia para el incidente:** Al igual que access.log.1, este log es de abril de 2025 y no proporciona información sobre el incidente de 2024.
- **access.log.3.gz**
 - **Período cubierto:** 30 de septiembre de 2024 (desde 12:07) hasta el 8 de octubre de 2024 (hasta 16:58:48).
 - **Orígenes:** Exclusivamente 127.0.0.1 (localhost) y ::1 (localhost IPv6 para comprobaciones internas de Apache).
 - **Actividad:** Contiene registros de la **instalación y configuración inicial de WordPress** desde localhost (peticiones a /wp-admin/install.php, POSTs a los pasos de instalación) el 30 de septiembre. También registra **inicios de sesión exitosos en WordPress** y navegación por el panel de administración desde localhost el 30 de septiembre y nuevamente el 8 de octubre (alrededor de las 16:49). Se observan numerosas peticiones a /wp-cron.php y /wp-admin/admin-ajax.php, normales para WordPress.
 - **Relevancia para el incidente:** Este log es altamente relevante, ya que cubre el período en el que el atacante instaló y configuró WordPress desde localhost. Documenta la actividad de post-exploitación en la capa web. Sin embargo, a pesar de que este log contiene registros *posteriores* a la aplicación de los permisos 777 en /var/www/html/ y wp-config.php (que ocurrieron el 8 de octubre a las 16:17:59 y 16:20:04 según los logs de journalctl), **no se observaron peticiones obviamente maliciosas** (como intentos de acceso a web shells o cargas de archivos anómalas) dentro de este log después de esos timestamps de cambio de permisos.

Conclusión Consolidada del Análisis de access.log:

El análisis de los archivos access.log proporcionados confirma que el servidor web Apache fue utilizado por el atacante (operando desde localhost) para instalar y configurar WordPress a finales de septiembre de 2024. Los logs de octubre (access.log.3.gz) muestran actividad en el panel de administración de WordPress desde localhost hasta las 16:58 del 8 de octubre. Aunque los logs de journalctl indican que los permisos 777 en /var/www/html y wp-config.php se establecieron antes de que finalizara access.log.3.gz, los registros de este archivo **no muestran evidencia directa de explotación web (como la ejecución de web shells) en el período que cubren después de dichos cambios de permisos.**

Esto sugiere que la explotación de la vulnerabilidad de permisos 777, si ocurrió a través de HTTP, lo hizo en un momento no cubierto por los logs disponibles o mediante un método que no dejó un rastro obvio en los logs de acceso estándar de Apache. Los logs access.log.1 y access.log.2.gz muestran la operación normal del sitio mucho después del incidente.

- **Contenidos de “error.log, error.log.1, error.log.2.gz, error.log.3.gz, error.log.4.gz, error.log.5.gz y error.log.6.gz”**
 - **Períodos Cubiertos:**
 - La mayoría de los archivos (error.log a error.log.5.gz) cubren fechas de **abril de 2025**.
 - El archivo error.log.6.gz cubre el período desde el **30 de septiembre de 2024** hasta el **8 de octubre de 2024**, lo cual lo hace relevante para el período del incidente.
 - **Tipos de Mensajes:**
 - Los mensajes predominantes en todos los archivos son de nivel notice (AH00163, AH00094, AH00170, AH00171, AH00489, AH00492). Estos mensajes son estándar e indican que el servidor Apache se está configurando, iniciando (resuming normal operations), reiniciando (Graceful restart requested), o deteniendo (caught SIGWINCH, shutting down gracefully). Reflejan el ciclo de vida normal del servicio Apache.
 - En error.log, hay una advertencia sobre no poder determinar de forma fiable el nombre de dominio completamente cualificado (AH00557, AH00558), una advertencia de configuración común que no implica necesariamente actividad maliciosa.
 - **Análisis en el Contexto del Incidente (enfocado en error.log.6.gz):**
 - El archivo error.log.6.gz sí abarca el período en el que el atacante instaló y manipuló Apache (finales de septiembre y principios de octubre de 2024). Los mensajes en este log coinciden con los reinicios y paradas del servicio que vimos en los logs de journalctl durante esas fechas. Confirman que el servicio Apache se iniciaba y se gestionaba (probablemente por el atacante vía comandos sudo/root).
 - **Sin embargo, es crucial destacar que en ninguno de los archivos de error revisados, incluyendo el relevante error.log.6.gz, se encontraron errores o advertencias que sugieran:**
 - Intentos de explotación web fallidos.
 - Errores de sintaxis o ejecución de scripts subidos (como web shells).
 - Errores resultantes de peticiones maliciosas o inusuales que Apache haya detectado como problemáticas.

Conclusión del Análisis de error.log:

El análisis de los logs de error de Apache, incluyendo el archivo que cubre el período del incidente (error.log.6.gz), revela que el servicio Apache estaba funcionando y siendo gestionado (iniciado, detenido, reiniciado) en las fechas relevantes. No obstante, **los logs proporcionados no contienen mensajes de error o advertencias que indiquen que se produjeron problemas o fallos durante la actividad web que pudieran estar relacionados con intentos de ataque o la ejecución de código malicioso a través del servidor web.**

Resumen Ejecutivo

Durante el análisis forense de un servidor Debian comprometido, se detectaron múltiples señales de intrusión, incluyendo accesos remotos por SSH con privilegios de root, instalación de servicios no autorizados, modificación de configuraciones críticas y cambios de permisos en directorios del servidor web.

Se determinó que el atacante obtuvo acceso como **usuario root vía SSH**, desplegó herramientas potenciales para persistencia y exfiltración, y comprometió el entorno web (Apache + WordPress) mediante la manipulación de archivos y permisos.

MÉTODOS DE ACCESO UTILIZADOS

Acceso remoto como root por SSH

- **Fecha/hora:** Oct 08 17:40:59
- **Registro en log:**

sshd[1650]: Accepted password for root from 192.168.0.134 port 45623 ssh2

- **Método:** Login por **contraseña**, no por clave SSH.
- **IP origen:** 192.168.0.134 (host en red local o máquina anfitriona).
- **Usuario:** root

➡ El atacante accedió directamente como **root**, lo cual es extremadamente peligroso, probablemente facilitado por:

- Contraseña débil.
- SSH mal configurado (permitiendo acceso root).
- Ausencia de firewalls o protección (fail2ban, UFW).

SERVICIOS ANALIZADOS

1. Servicio SSH

- Se detectó la instalación del paquete openssh-server.
- Se editó el archivo /etc/ssh/sshd_config.
- Se reinició el servicio tras los cambios.

➡ Esto indica que se **reconfiguró SSH**, probablemente para habilitar acceso root y facilitar conexiones futuras.

Servidor Web – Apache + WordPress

Cambios en configuración:

- Edición directa del archivo /etc/apache2/apache2.conf.
- Reinicio del servicio con systemctl restart apache2.

Permisos peligrosos otorgados:

`chmod -R 777 /var/www/html`

`chmod 777 /var/www/html/wp-config.php`

➔ Esto otorga permisos de **lectura, escritura y ejecución a cualquier usuario**, lo cual permite insertar y ejecutar shells PHP u otro código malicioso fácilmente.

Actividad registrada:

- Accesos al backend de WordPress (/wp-admin/, /wp-login.php).
- Actualización de la base de datos WordPress (upgrade.php).
- Tráfico desde 127.0.0.1 y desde la IP **192.168.68.116** (acceso externo en la red local el 11 de abril de 2025).

Servicio FTP (vsftpd)

Evidencia en los logs:

- Instalación de vsftpd.
- Creación del usuario ftp (UID 113).
- Activación del servicio vsftpd.service.

LIMITACIONES DE LA AUDITORÍA

Durante el análisis se detectaron fallos críticos de visibilidad en el sistema:

Evidencia	Descripción
 /var/log/auth.log	Inexistente o no generado por falta de rsyslog.
 last -i vacío	La base de sesiones (wtmp) no guarda registros.
 Sin registros de failed password	No hay trazabilidad de intentos fallidos de acceso.

➔ Esto impide comprobar el alcance exacto del compromiso y sugiere una **mala configuración del sistema o manipulación de logs por parte del atacante**.

CONCLUSIONES

Servicio	Estado	Potencialmente Comprometido	Detalles relevantes
SSH	Modificado	<input checked="" type="checkbox"/> Sí	Acceso root por contraseña desde LAN
Apache	Reconfigurado	<input checked="" type="checkbox"/> Sí	Reinicios y acceso a /wp-admin/
WordPress	Archivos manipulados	<input checked="" type="checkbox"/> Sí	Permisos 777 y posibles shells ocultos
FTP (vsftpd)	Instalado sin justificación	<input checked="" type="checkbox"/> Sí	Puede haber sido usado para persistencia
Logs del sistema	Incompletos/inexistentes	 Parcial	auth.log, wtmp y otros no disponibles

4.2 Identificar Archivos Sospechosos

Como parte de la investigación forense, se realizó una búsqueda sistemática de archivos sospechosos en el sistema de archivos del servidor comprometido. El objetivo fue identificar backdoors, herramientas de ataque, scripts maliciosos, archivos de configuración alterados o cualquier otro archivo anómalo que pudiera haber sido introducido o modificado por el atacante. La búsqueda se centró en directorios de alto riesgo como /tmp, /var/tmp, directorios de usuario, directorios de configuración de servicios clave y otros puntos de interés.

Las técnicas empleadas incluyeron:

- Revisión de listados de directorios con detalles (`ls -la`).
- Búsqueda de archivos modificados o creados recientemente utilizando `find`, concentrándose en el período del incidente (julio a octubre de 2024).
- Inspección visual de nombres de archivos inusuales o con permisos sospechosos.

Tras esta revisión inicial, el hallazgo más notable y directamente sospechoso se localizó en el directorio de descargas del usuario `debian`:

```
root@debian:/home/debian/Downloads# ls -la
total 153096
drwxr-xr-x  2 debian  debian        4096 Sep 28  2024 .
drwx----- 14 debian  debian        4096 Apr 23 13:10 ..
-rw-r--r--  1 debian  debian 156758455 Sep 28 2024 xampp-osx-8.0.28-0-installer.dmg
```

- `/home/debian/Downloads/xampp-osx-8.0.28-0-installer.dmg`
 - **Fecha:** 28 de septiembre de 2024.
 - **Propietario:** debian:debian.
 - **Tamaño:** Aproximadamente 150 MB.
 - **Motivo de la sospecha:** Este archivo es un instalador (.dmg) destinado específicamente para sistemas operativos macOS. Su presencia en un servidor Debian Linux es anómala y altamente inusual. La fecha de descarga cae dentro del período de actividad del atacante utilizando la cuenta `debian`, lo que sugiere que fue descargado por la persona que controlaba la cuenta en ese momento. Aunque un archivo .dmg para macOS no puede ejecutarse directamente en Linux, su existencia en este directorio es evidencia de la actividad del atacante y potencialmente un error cometido durante sus operaciones (como descargar el archivo equivocado o prepararlo para usarlo en otra máquina).

Si bien se buscaron archivos sospechosos en otras ubicaciones comunes y no se identificaron archivos maliciosos con nombres obvios o características anómalas evidentes en esta fase inicial, el archivo “xampp-osx-8.0.28-0-installer.dmg” destaca como una clara evidencia de actividad inusual y está directamente asociado con la cuenta utilizada por el atacante durante el incidente. Es importante continuar la búsqueda exhaustiva en otras áreas y utilizar técnicas de análisis más profundas si es necesario.

4.3 Informe de Actividad de Procesos y Servicios del Sistema

La inspección de los procesos que se ejecutan activamente en un sistema comprometido es un paso fundamental en la respuesta a incidentes, ya que permite identificar malware en memoria, backdoors activas, servicios inesperados o la presencia de shells interactivas utilizadas por el atacante en tiempo real. Para este fin, se obtuvo una lista detallada de todos los procesos en ejecución en el servidor debian, ordenada por su hora de inicio, utilizando el comando `ps aux --sort=start_time`. Esta sección presenta el análisis de dicha lista, destacando cualquier proceso que muestre características inusuales o que pueda estar vinculado a la actividad del atacante identificada en los logs.

💡 **Comando ejecutado:** `ps aux --sort=start_time`

🔍 **Orden:** de procesos más antiguos a más recientes

📋 Servicios y procesos esperados del sistema

Estos servicios son comunes en una instalación estándar de Debian con entorno de escritorio MATE + VirtualBox:

Servicio / Proceso	Función	Estado
<code>/sbin/init, [kthreadd], [rcu_*]</code>	Núcleo y gestión del sistema	🟢 OK
<code>/lib/systemd/*, cron, dbus-daemon</code>	Gestión de servicios y tareas	🟢 OK
<code>Lightdm, mate-panel, marco, caja</code>	Interfaz gráfica MATE	🟢 OK
<code>VBoxService, VBoxClient</code>	Soporte para VirtualBox Guest Additions	🟢 OK
<code>pulseaudio, speech-dispatcher, orca</code>	Accesibilidad y audio	🟢 OK

✓ Estos procesos son **esperados** y no indican actividad maliciosa.

Servicios de red activos que requieren revisión

Servicio / Proceso	Descripción	Estado / Acción recomendada
ssh	Servicio SSH: permite conexiones remotas	 Verificar si root puede conectarse
vsftpd	Servidor FTP ligero	 No es usual a menos que lo uses
apache2 (múltiples PID)	Servidor web Apache – usado por WordPress	 Está activo y sirviendo contenido
mariadb	Base de datos para WordPress	 Necesario para el CMS

Evaluación específica

1. sshd activo

- Aún activo y escuchando conexiones.
- Previamente fue usado por un atacante para acceder como root.
-  Acción crítica:

```
sudo nano /etc/ssh/sshd_config
```

Cambia:

```
PermitRootLogin no
```

2. vsftpd activo

- No vemos uso en el servicio, se recomienda detenerlo.
-  Acción:

```
sudo systemctl stop vsftpd
```

```
sudo apt purge vsftpd
```

3. apache2

- Apache sigue activo con múltiples workers.
-  Asegúrate de **restaurar permisos correctos** a /var/www/html.

Procesos root sospechosos

- Hay **múltiples shells abiertas como root** (sudo su, su, bash) — esto es normal si estás auditando activamente.
- No hay procesos desconocidos o extraños bajo el usuario root, lo cual es un buen indicador de que **no hay actividad maliciosa en curso**.

Análisis de Servicios Habilitados en systemd

 Servicio	 Descripción	 Evaluación / Recomendación
apache2.service	Servidor web Apache	 Necesario si usas WordPress, pero revisar configuración y permisos en /var/www/html. Si no lo usas, deshabilitar.
mariadb.service	Base de datos MySQL	 Necesario para WordPress. Verifica que no esté expuesto externamente (puerto 3306).
vsftpd.service	Servidor FTP	 Riesgo alto. Muy explotado. Si no lo necesitas, desactívalo (systemctl disable vsftpd.service).
ssh.service	Acceso remoto	 Necesario para administración, pero se detectó acceso externo por root →  deshabilita login como root (PermitRootLogin no).
speech-dispatcherd.service	Texto a voz (TTS)	 No es común en servidores. Desactíalo si no se usa: systemctl disable speech-dispatcherd.
vboxadd.service / vboxadd-service.service	VirtualBox guest additions	 Necesario si es VM. Vigila integraciones con el host.
bluetooth.service	Conexión Bluetooth	 Inútil en servidores. Desactíalo si no es usado: systemctl disable bluetooth.
saned.service	Escaneo de red (scanner)	 Muy inusual en servidores. Está enmascarado (ok), pero si aparece en logs, revisar.
ModemManager.service	Gestión de módems 3G/4G	 Pocas veces útil. Mejor desactivarlo. Podría abrir canales externos.
alsa-utils.service, pulseaudio-enable-autospawn.service, speech-dispatcherd.service	Servicios multimedia/audio	 Innecesarios en servidor. Revisión recomendada.
rsync.service	Sincronización de archivos	 Útil si lo usas, pero revisa configuraciones y no lo dejes accesible por red si no está protegido.
upower.service	Gestión de energía	 Relevante para escritorio, no para servidores. Puedes desactivarlo.
lightdm.service	Gestor de sesiones gráficas	 Revisa si realmente necesitas entorno gráfico. Usualmente es mejor administrar servidores sin GUI.

Servicios Especialmente Críticos / Sospechosos

1. vsftpd.service habilitado

FTP es **inseguro** (no cifra credenciales). Si fue usado por el atacante, es un riesgo serio.

2. ssh.service accesible como root

Ya se ha verificado que hubo login como root. Necesitas:

- Desactivar login remoto de root:

Edita `/etc/ssh/sshd_config`:

`PermitRootLogin no`

Luego: `systemctl restart ssh`

3. speech-dispatcherd.service activado a pesar de haberlo deshabilitado manualmente

Aparece habilitado aunque tú ejecutaste `systemctl disable speech-dispatcher`. Esto sugiere:

- No se deshabilitó correctamente (o se reactivó después).

- Alguien lo habilitó por alguna razón específica.

Verifica si se lanza desde `.bashrc`, `cron`, etc.

4.4 Escaneo del servidor para detectar rootkits y malware.

Una fase crítica en cualquier investigación de un servidor comprometido es la búsqueda activa de rootkits y otro software malicioso que el atacante pudiera haber instalado. A diferencia de la actividad visible en los logs o en la lista de procesos estándar, los rootkits están específicamente diseñados para operar de forma sigilosa, ocultando su presencia y la de otras herramientas maliciosas para asegurar la persistencia del atacante en el sistema. Dada la gravedad del acceso obtenido por el atacante en este incidente, existe un riesgo considerable de que se hayan desplegado este tipo de amenazas. Esta sección detalla el proceso y los hallazgos del escaneo realizado en el servidor debian con el objetivo de identificar la presencia de rootkits conocidos, troyanos, o cualquier otro binario o script con características de malware diseñado para evadir la detección. La detección de rootkits a menudo requiere el uso de herramientas especializadas que operan a un nivel inferior o buscan patrones de modificación del sistema inusuales.

Informe de Resultados – Escaneo Anti-Rootkit con chkrootkit

Fecha de ejecución: 11 de abril de 2025

Herramienta utilizada: chkrootkit

Sistema analizado: Servidor Debian con servicios Apache y WordPress activos

Estado general del sistema:

El escaneo con chkrootkit no ha revelado signos concluyentes de infección por rootkits conocidos. La mayoría de los binarios y servicios críticos del sistema han sido verificados, arrojando resultados como `"not infected"` o `"not found"` (no presentes en el sistema). Sin embargo, se han reportado algunos elementos que requieren revisión manual.

Advertencias detectadas:

1. Archivos sospechosos encontrados:

- *WARNING: The following suspicious files and directories were found:*
→ `/usr/lib/libreoffice/share/.registry`

Este directorio oculto podría ser legítimo si LibreOffice está instalado, pero al estar en una ruta oculta (`.registry`) se recomienda verificar su integridad y contenido manualmente. Se sugiere compararlo con una instalación limpia o utilizar `debsums` para validación.

2. Actividad sospechosa de sniffer detectada:

- **WARNING: Output from ifpromisc:**
→ *enp0s3: PACKET SNIFFER (/usr/sbin/NetworkManager[435])*

Este mensaje indica que la interfaz de red enp0s3 ha sido detectada en modo promiscuo, asociado al proceso NetworkManager.

⚠ Aunque NetworkManager puede utilizar modo promiscuo legítimamente (por ejemplo, para escaneo de redes WiFi), esto también puede ser síntoma de sniffing malicioso, si no hay una justificación clara de su uso.

Hallazgos principales:

- No se detectaron infecciones activas

chkrootkit: Ningún malware crítico encontrado en binarios del sistema.

Falsos positivos:

Archivos en /usr/lib/ruby/ y /usr/lib/libreoffice/ (relacionados con paquetes legítimos).

Sniffer en enp0s3 (asociado a NetworkManager, normal en sistemas con gestión de red).

Informe de Análisis de Rootkits con RKhunter

Fecha del análisis: 23 de abril de 2025

Sistema operativo detectado: Debian GNU/Linux 12 (Bookworm)

Versión de RKhunter utilizada: 1.4.6

Resultados Generales

El escaneo se ha completado exitosamente y se han revisado numerosos aspectos del sistema,

incluyendo:

- Comandos del sistema
- Bibliotecas compartidas
- Propiedades de archivos del sistema
- Presencia de rootkits conocidos
- Variables de entorno relacionadas con seguridad

Hallazgos Relevantes

1. Archivos del sistema modificados (potencialmente sospechosos):
 - lwp-request: detectado como reemplazado por un script en Perl. Aunque esto no implica automáticamente un compromiso, debe verificarse que el script sea legítimo y no haya sido alterado maliciosamente.
 - /usr/bin/lwp-request: Perl script text executable
2. Comprobaciones de bibliotecas compartidas:
 - No se detectaron bibliotecas precargadas (LD_PRELOAD) ni manipulaciones en las rutas de búsqueda de bibliotecas (LD_LIBRARY_PATH).
 - Rootkits conocidos:
 - Todos los rootkits conocidos revisados fueron reportados como *No encontrados*. Esto incluye variantes como:
 - Adore, AjaKit, Ambient (ark), BeastKit, BOBKit, Diamorphine, Dica-Kit, Dreams, Fu, Fuck`it, GasKit, Jynx, KBeast, entre muchos otros.
3. Alertas y advertencias:
 - No se detectaron rootkits activos.
 - Las entradas sospechosas fueron mínimas y no indican una infección directa, aunque deben revisarse manualmente si se desea máxima seguridad.

5. Mitigación de vulnerabilidades.

Una vez identificados posibles vectores de ataque o servicios potencialmente comprometidos durante la fase de análisis, se procede a aplicar medidas de contención inmediatas con el objetivo de **interrumpir cualquier actividad maliciosa en curso y evitar una mayor escalación de privilegios** por parte del atacante.

Como parte de esta contención, se evalúan los servicios activos en el sistema para identificar aquellos que presentan un comportamiento anómalo o que podrían estar relacionados con el compromiso. En caso necesario, **se procede a su detención temporal mediante el uso de systemctl stop [servicio]**, evitando así que el atacante mantenga acceso o control activo sobre el entorno.

Además, se considera el bloqueo o aislamiento del exploit detectado, aplicando parches o deshabilitando funciones vulnerables según el contexto técnico del sistema.

Detención de Servicios Comprometidos

Para interrumpir la ejecución de servicios sospechosos o vulnerables se emplea el siguiente procedimiento:

```
sudo systemctl stop nombre-del-servicio
```

Por ejemplo:

```
sudo systemctl stop apache2
```

```
sudo systemctl stop mariadb
```

```
sudo systemctl stop ssh
```

Esta acción **detiene el servicio temporalmente**, sin desactivarlo permanentemente, lo cual permite un análisis posterior sin perder la configuración o el estado del mismo. En caso de que se confirme su implicación en el incidente, se puede deshabilitar de forma persistente con:

```
sudo systemctl disable nombre-del-servicio
```

También es recomendable revisar servicios enmascarados (masked) o activados por socket que puedan ser usados por el atacante para persistencia.

Bloquear el Exploit

No se detecta ningún Exploit que Bloquear.

a) Modificar la Configuración de SSH para Bloquear el Acceso Root

- **Acción:** Deshabilita el acceso root y la autenticación con contraseña en sshd.

- **Pasos:**

- Edita el archivo de configuración de sshd:

```
sudo nano /etc/ssh/sshd_config
```

- Asegúrate de que las siguientes líneas estén configuradas (si no existen, agrégalas):

```
PermitRootLogin no PasswordAuthentication no
```

- Reinicia el servicio sshd para aplicar los cambios:

```
sudo systemctl restart ssh
```

Configurar un Firewall para Restringir el Acceso a Puertos

- **Acción:** Usa ufw (Uncomplicated Firewall) para restringir el acceso a los puertos expuestos (22, 21, 80, 3306).
- **Pasos:**
 - Instala ufw si no está instalado:

```
sudo apt install ufw
```

 - Configura reglas para permitir solo el acceso desde la red local a los puertos 22 (SSH) y 21 (FTP), y permitir el puerto 80 (Apache) globalmente:

```
sudo ufw allow from 192.168.0.0/24 to any port 22 sudo ufw allow from 192.168.0.0/24 to any port 21 sudo ufw allow 80 sudo ufw deny 22 sudo ufw deny 21 sudo ufw deny 3306
```
 - Habilita el firewall:

```
sudo ufw enable
```

 - Verifica las reglas:

```
sudo ufw status
```

Corregir Permisos Inseguros en /var/www/html

- **Acción:** Corrige los permisos 777 en /var/www/html y wp-config.php para prevenir la subida de archivos maliciosos (como webshells).
- **Pasos:**
 - Ajusta los permisos y propietario:

```
sudo chmod -R 755 /var/www/html sudo chmod 644 /var/www/html/wp-config.php sudo chown -R www-data:www-data /var/www/html
```
 - Verifica los cambios:

```
ls -la /var/www/html
```

Asegurar la Configuración de FTP (vsftpd)

- **Acción:** Configura vsftpd para prevenir accesos anónimos y restringir usuarios.
- **Pasos:**
 - Edita el archivo de configuración:

```
sudo nano /etc/vsftpd.conf
```
 - Asegúrate de que las siguientes opciones estén configuradas:

```
anonymous_enable=NO chroot_local_user=YES allow_writeable_chroot=YES
```
 - Reinicia el servicio:

```
sudo systemctl restart vsftpd
```

Proteger MariaDB

- **Acción:** Asegúrate de que MariaDB no esté expuesto a Internet.
- **Pasos:**
 - Verifica si MariaDB escucha en todas las interfaces:
`sudo netstat -tuln | grep 3306`
 - Si está escuchando en 0.0.0.0:3306 o :::3306, edita la configuración:
`sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf`
 - Configura:

`bind-address = 127.0.0.1`

■ Reinicia el servicio:
`sudo systemctl restart mariadb`

Prevención de Escalada de Privilegios

Además de contener el exploit o detener el servicio afectado, se adoptan medidas para **reducir la superficie de escalada**:

- **Revocar accesos de usuarios sospechosos:**
`sudo usermod -L usuario`
- **Eliminar claves SSH no autorizadas** de `~/.ssh/authorized_keys`.
- **Revisar permisos de archivos binarios y scripts sensibles**, asegurando que no puedan ser ejecutados por usuarios no privilegiados.
- **Aplicar el principio de mínimo privilegio** en usuarios y servicios.
- **Aislamiento del entorno de red**, por ejemplo, mediante la desactivación temporal de interfaces:
`sudo ip link set enp0s3 down`
- **Utilizar herramientas como AppArmor o SELinux** para aplicar políticas de control de acceso obligatorio (MAC) que impidan la ejecución o modificación no autorizada de archivos clave.

Estas acciones ayudan a **limitar el alcance del ataque** y protegen el sistema mientras se continúa con la fase de análisis y recuperación.

6. Revertir posibles cambios realizados

- 🔍 1. Eliminación de Usuarios No Autorizados

Comando ejecutado:

```
awk -F: '($3 >= 1000 && $7 !~ /nologin|false/) {print $1 ":" $3 ":" $7}' /etc/passwd
```

Resultado:

```
debian:1000:/bin/bash
```

Acción:

No se encontraron usuarios no autorizados.

Si hubiera usuarios sospechosos:

```
sudo deluser --remove-home NOMBRE_USUARIO # Elimina usuario y su directorio /home
```

- 🔍 2. Eliminación de posibles Backdoors

- a) Revisión de Tareas Programadas (Cron)

Comando:

```
ls -l /etc/cron.daily/
```

Resultado:

```
@anacron, apache2, apt-compat, dpkg, logrotate, man-db # Todos legítimos
```

Acción:

No se eliminaron tareas maliciosas (no había).

Si hubiera un script sospechoso (ej: /etc/cron.daily/backdoor.sh):

```
sudo rm -f /etc/cron.daily/backdoor.sh # Elimina el script malicioso
```

- b) Servicios Maliciosos

Comando:

```
systemctl list-units --type=service --state=running
```

Resultado:

```
apache2, sshd, cups, vsftpd, avahi-daemon # vsftpd y avahi son riesgos innecesarios
```

Acción:

```
sudo systemctl disable --now vsftpd avahi-daemon # Desactiva servicios vulnerables
```

- 🔍 3. Cierre de Puertos Innecesarios

Comando:

```
sudo netstat -tulnp
```

Resultado:

```
:::21 (FTP), :::5353 (UDP), :::80 (HTTP), :::22 (SSH) # FTP y Avahi son riesgos
```

Acciones:

Para cerrar puerto 21 (FTP):

```
sudo systemctl disable --now vsftpd
```

Para cerrar puerto 5353 (Avahi):

```
sudo systemctl disable --now avahi-daemon
```

Opcional (si no se usa Apache):

```
sudo systemctl disable --now apache2 # Cierra puerto 80
```

Resumen de Acciones Realizadas

Problema, Comando de Corrección

Usuarios no autorizados, sudo deluser --remove-home NOMBRE

Backdoors en Cron, sudo rm -f /ruta/script_malicioso.sh

Servicios peligrosos, sudo systemctl disable --now SERVICIO

Puertos abiertos, sudo systemctl disable --now SERVICIO

7. Actualizar y corregir configuraciones de seguridad

💡 1. Actualización de Paquetes

Objetivo: Parchear vulnerabilidades conocidas.

- ◆ Comandos a ejecutar:

```
# Actualizar lista de paquetes disponibles  
sudo apt update
```

```
# Actualizar todos los paquetes instalados (seguridad + mejoras)  
sudo apt upgrade -y
```

```
# Actualizar el kernel y componentes críticos (opcional)  
sudo apt full-upgrade -y
```

```
# Eliminar paquetes obsoletos  
sudo apt autoremove --purge -y
```

Qué logramos:

Sistema con los últimos parches de seguridad.

Eliminación de software vulnerable.

💡 2. Cambio de Contraseñas

Objetivo: Evitar acceso no autorizado con credenciales comprometidas.

- ◆ Comandos a ejecutar:

```
# Cambiar contraseña del usuario actual (ej: "debian")  
passwd
```

```
# Cambiar contraseña de root (si es necesario)  
sudo passwd root
```

```
# Verificar cuentas con contraseña vacía (¡Ninguna debería aparecer!)  
sudo awk -F '":":' {print $1}' /etc/shadow
```

```
# Forzar cambio de contraseña en próximo login (ej: para un usuario sospechoso)  
sudo chage -d 0 nombre_usuario
```

Qué logramos:

Credenciales más seguras.

Prevención de ataques por fuerza bruta.

💡 3. Mejora de Configuraciones de Firewall (UFW)

Objetivo: Restringir accesos no autorizados.

- ◆ Comandos a ejecutar:

```
# Instalar UFW (si no está instalado)  
sudo apt install ufw -y
```

```
# Habilitar firewall (bloquea todo por defecto)  
sudo ufw enable
```

```
# Permitir solo lo esencial (SSH, HTTP/HTTPS si es necesario)  
sudo ufw allow 22/tcp      # SSH
```

```

sudo ufw allow 80/tcp      # HTTP (opcional)
sudo ufw allow 443/tcp     # HTTPS (opcional)

# Denegar todo lo demás
sudo ufw default deny incoming
sudo ufw default allow outgoing

# Verificar reglas
sudo ufw status numbered

```

Ejemplo de salida esperada:

```

Status: active
To          Action    From
--          -----   -----
22/tcp       ALLOW     Anywhere
80/tcp       ALLOW     Anywhere
443/tcp      ALLOW     Anywhere

```

Qué logramos:

Solo los puertos SSH (22) y HTTP/HTTPS (80/443) están abiertos.

Tráfico entrante bloqueado por defecto.

4. Ajustes Adicionales de Seguridad

- ◆ a) Deshabilitar SSH con contraseña (usar solo claves)

```

sudo sed -i 's/#PasswordAuthentication yes/PasswordAuthentication no/' 
/etc/ssh/sshd_config
sudo systemctl restart sshd

```

- ◆ b) Configurar fail2ban (protección contra fuerza bruta)

```

sudo apt install fail2ban -y
sudo systemctl enable --now fail2ban

```

- ◆ c) Revisar permisos críticos

```

# Evitar que cualquier usuario lea /etc/shadow
sudo chmod 640 /etc/shadow

```

```

# Asegurar /tmp (opcional)
sudo chmod 1777 /tmp

```

8. Conclusiones:

La Fase 1 de esta investigación forense tuvo como objetivo principal analizar el servidor debian para identificar evidencias de un incidente de seguridad, determinar las vulnerabilidades explotadas y sentar las bases para la contención y erradicación. A través de la revisión de los logs del sistema (journalctl), logs de servicios específicos (Apache), análisis de procesos en ejecución y examen de configuraciones clave, se han recopilado numerosos indicadores de actividad anómala y patrones de comportamiento consistentes con un acceso no autorizado y acciones post-explotación.

Si bien los logs disponibles no proporcionaron una visibilidad directa del vector de intrusión inicial que pudo haber otorgado el acceso primario (ej. explotación de una vulnerabilidad de software no registrada, credenciales débiles, phishing no capturado por estos logs), el análisis exhaustivo ha revelado fuertes indicios de que un actor externo obtuvo acceso al sistema y procedió a establecer control y persistencia:

- Escalada de Privilegios y Obtención de Acceso Root: Se evidenciaron intentos fallidos de sudo seguidos de una escalada exitosa a root vía su, lo que sugiere que se obtuvo la contraseña de root, aunque estos registros de acceso muestran únicamente conexiones locales y de usuarios identificados. El acceso como root fue limitado y coherente con las actividades legítimas de administración. Posteriormente, se confirmó un acceso directo con éxito al usuario root a través de SSH desde la IP 192.168.0.134. Estos eventos documentan claramente la obtención del máximo nivel de privilegio en el sistema.
- Establecimiento de Persistencia y Control: El usuario instaló y configuró activamente servicios de red y web (Apache, MariaDB, PHP/WordPress, VSFTPD, SSH habilitado) que pueden servir como puntos de acceso continuo. La modificación de /etc/sudoers para otorgar privilegios a la cuenta debian puede suponer un mecanismo de persistencia.
- Creación Deliberada de Vulnerabilidades: La modificación de permisos en /var/www/html/ y wp-config.php a 777 creó una vulnerabilidad de seguridad crítica que facilita el despliegue de web shells u otro código malicioso a través de la interfaz web, aunque no se observó explotación directa de esta vulnerabilidad en los logs de acceso de Apache disponibles para el período inmediatamente posterior.
- Artefactos y Modificaciones Anómalas: Se identificó la descarga de un archivo de instalador de macOS (.dmg) en el directorio de descargas del usuario debian con un timestamp que coincide con el período de actividad del posible atacante, lo cual es una clara anomalía. Se detectaron modificaciones en los archivos de configuración de tareas programadas (cron), sugiriendo intentos de establecer persistencia recurrente.
- Indicios de Evasión/Manipulación de Logs: La discrepancia entre el log de sshd (mostrando login root) y la ausencia de este registro en el log de wtmp (last output), junto con la ausencia de un servicio rsyslog tradicional, podría indicar intentos de evadir o limitar la capacidad de registro del sistema.

En conclusión, aunque los registros analizados no revelan el "cómo" exacto de la intrusión inicial, la secuencia de eventos documentada en los logs (escalada de privilegios, obtención de acceso root directo, instalación y configuración de servicios para persistencia, creación de vulnerabilidades graves, presencia de procesos sospechosos) proporciona una evidencia convincente de que el servidor sufrió un incidente de seguridad significativo involucrando acceso no autorizado y manipulación del sistema por parte de un actor con intenciones desconocidas. La Fase 1 ha permitido identificar los principales puntos de compromiso y los mecanismos de persistencia creados, lo cual es fundamental para proceder con las fases de erradicación y recuperación. Las acciones de contención iniciales y la planificación de la erradicación se basan directamente en estos hallazgos para neutralizar la presencia del posible atacante.