

# Informe de Incidente de Seguridad

Análisis forense y respuesta al compromiso de un servidor Debian GNU/Linux

**Autor:**

**Alejandro Winter**

**Contenido:**

**Este informe documenta el análisis forense del acceso no autorizado, la identificación del vector de ataque y las medidas de contención y erradicación aplicadas.**

### 3.1 RESUMEN TÉCNICO DEL ENTORNO

El presente proyecto se ha desarrollado sobre un entorno de laboratorio controlado, compuesto por dos sistemas principales con roles claramente diferenciados.

El sistema objetivo es un servidor **Debian GNU/Linux**, configurado como servidor web y de servicios, el cual presenta diversas configuraciones inseguras que permiten simular un escenario realista de compromiso de seguridad. Este sistema actúa como máquina víctima durante las fases de análisis forense y validación de las medidas correctivas.

Por otro lado, se ha utilizado una máquina **Kali Linux** como sistema atacante, desde la cual se han realizado las pruebas de escaneo, detección y explotación de vulnerabilidades, simulando el comportamiento de un atacante externo sin privilegios previos sobre el sistema objetivo.

El análisis se ha dividido en tres fases principales:

**Análisis forense y contención del incidente** sobre el sistema Debian. **Detección y explotación de vulnerabilidades** mediante técnicas de pentesting desde Kali Linux.

**Definición de un plan de respuesta a incidentes y medidas preventivas**, alineadas con buenas prácticas de seguridad.

# Fase 1 : Análisis forense y recolección de evidencias.

## 1) Identificar como los servicios fueron comprometidos y como el atacante vulneró y accedió al servidor.

**Servicio comprometido:** SSH (puerto 22, servicio sshd).

**Cómo accedió el atacante:**

```
debian@debian:~$ tail -f /var/log/auth.log
Oct 08 16:14:16 debian sshd[5341]: Server listening on :: port 22.
Oct 08 16:43:16 debian kernel: tsc: Marking TSC unstable due to check_tsc_sync_source failed
Oct 08 16:43:16 debian kernel: [dm:vmw_host_printf [vmwgfx]] *ERROR* Failed to send host log message.
Oct 08 16:43:17 debian udisksd[485]: failed to load module mdraid: libbb_mdraid.so.2: cannot open shared object file: No such file or directory
Oct 08 16:43:17 debian udisksd[485]: Failed to load the 'mdraid' libblockdev plugin
Oct 08 16:43:18 debian alsactl[59]: alsa-lib main.c:1541:[snd_use_case_mgr_open] error : failed to import hw::use_case configuration -2
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 apache2[1579]: AH00557: apache2: apr_socaddr_info_get() failed for debian
Oct 08 16:44:03 debian NetworkManager[496]: <info> [1728420243.7989] device (enp0s3): state change: ip-config -> failed (reason 'ip-config-unavailable', sys-iface-state: 'managed')
Oct 08 16:44:03 debian NetworkManager[496]: <warn> [1728420243.7993] device (enp0s3): Activation: failed for connection 'Wireless connection 1'
Oct 08 16:44:03 debian NetworkManager[496]: <info> [1728420243.7994] device (enp0s3): state change: failed -> disconnected (reason 'none', sys-iface-state: 'managed')
Oct 08 16:44:07 debian xdg-desktop-por[1159]: Failed connect to PipeWire: Couldn't crea
te PipeWire context
Oct 08 16:44:18 debian systemd[1]: NetworkManager-wait-online.service: Failed with resu
```

Login **correcto** sobre el usuario root.

Autenticación por **contraseña** (Accepted password).

Desde la IP **192.168.0.134** y puerto de origen 45623.

**Vulnerabilidades implícitas:**

-Root tiene permitido el login por SSH

-La contraseña de root es débil o se ha filtrado.

-SSH escuchando en 0.0.0.0:22 (**expuesto a la red**).

A partir de los logs de **journctl** se confirma que el servicio SSH (**sshd**) se encuentra activo y escuchando en el puerto 22 sobre todas las interfaces (**0.0.0.0**). Esto expone el acceso remoto a la máquina y constituye el vector de ataque **principal**.

Supone un posible ataque mediante **fuerza bruta** o explotación de **credenciales débiles**. Es correcto que esté abierto, pero **riesgo alto** si no está endurecido.

## 2) Identificar archivos sospechosos, procesos en ejecución y cualquier modificación inusual en el sistema.

Ejecuté **ps aux --sort=-%cpu | head** para buscar procesos activos.

```
debian@debian:~$ ps aux --sort=-%cpu | head -n 15
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND
debian     6470 100  0.2 11224 4824 pts/1    R+ 13:41 0:00 ps aux --sort=-%cpu
root      1440  1.0  6.8 514368 137372 ttys0  Ssl+ 09:31 2:30 /usr/lib/xorg/Xorg :0
seat seat0 -auth /var/run/lightdm/root/:0 -nolisten tcp vt7 -novoswitch
debian    1762  0.3  0.1 282904 2184 ?        S1 09:31 0:58 /usr/bin/VBoxClient --
draganddrop
root      5334  0.3  0.0      0 ?        I 11:33 0:24 [kworker/1:0-events]
debian    2245  0.2  2.5 560316 51800 ?
root      1754  0.1  0.1 216852 2208 ?
seamless
root     1133  0.1  0.0 349848 1044 ?
root     1136  0.1  0.1 355500 2812 ?
--pidfile /var/run/vboxadd-service.sh
debian    1819  0.0  2.5 817620 51020 ?
message+ 499  0.0  0.2 10092 5488 ?
ssd      09:30 0:12 /usr/bin/dbus-daemon -
-system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
root      6346  0.0  0.0      0 ?
root      6339  0.0  0.0      0 ?
root      6462  0.0  0.0      0 ?
polkitd   501  0.0  0.6 313128 12736 ?
ssd      09:30 0:10 /usr/lib/polkit-1/polkitd --no-debug
```

-No hay procesos maliciosos activos visibles (nada raro en **/tmp**, nada con nombres sospechosos, ningún minero, ningún script extraño...).

-No se dejó un proceso persistente ejecutándose.

-No se detectan **procesos anómalos** ni consumo excesivo de CPU que indique actividad **maliciosa** o **ataques DoS**.

---

En la memoria (**ps aux –sort=-%mem**). El servicio de base de datos **MariaDB** presenta el **mayor consumo** de memoria del sistema, lo cual es esperable pero podría **optimizarse** o **deshabilitarse si no es necesario**.

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
mysql	649	0.0	11.8	1349436	239372	?	Ssl	09:30	0:08	/usr/sbin/mariadb
root	1440	1.0	6.8	514368	137372	ttv7	Ssl+	09:31	2:30	/usr/lib/xorg/Xorg :0

### **UDP y sockets locales**

Los servicios UDP detectados corresponden mayoritariamente a procesos locales del entorno gráfico y **no representan** un vector de ataque remoto significativo.

- /run/user/1000
- dbus
- pulseaudio
- avahi
- cups
- mdns

Usando el comando “**ss -lptn**” pude ver que el puerto 80 -HTTP (Apache) está activo y es una superficie de ataque clara.

State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port	Process
LISTEN	0	80	127.0.0.1:3306	0.0.0.0:*	
LISTEN	0	128	0.0.0.0:22	0.0.0.0:*	
LISTEN	0	20	127.0.0.1:25	0.0.0.0:*	
LISTEN	0	128	127.0.0.1:631	0.0.0.0:*	
LISTEN	0	511	*:80	*:*	
LISTEN	0	128	[::]:22	[::]:*	
LISTEN	0	32	*:21	*:*	
LISTEN	0	20	[::1]:25	[::]:*	
LISTEN	0	128	[::1]:631	[::]:*	

Luego se puede observar que en el puerto ( \*:21) el **servicio FTP** se encuentra activo y **accesible públicamente**, lo que representa un **riesgo elevado** al no **cifrar** las credenciales.

Mientras que **ss -lptn** permite identificar servicios **TCP** expuestos a la red y potenciales vectores de ataque remoto, el comando **ss -lpx**, usado en la imagen de abajo muestra principalmente **sockets locales** y **servicios UDP** asociados al entorno del sistema, los cuales no representan un riesgo significativo de acceso externo.

```
u_str LISTEN 0      80          /run/mysql/mysqld.sock 15193 *
```

Para encontrar archivos sospechosos y modificaciones recientes, ejecuté estos comandos:

- sudo find / -type f -mtime -7 2>/dev/null | head -n 40
- sudo find / -type f \(-iname "\*shell\*" -o -iname "\*cmd\*" -o -iname "\*backdoor\*" -o -iname "\*upload\*"\) 2>/dev/null

## MEDIDAS DE MITIGACIÓN

No se detectaron binarios maliciosos, shells persistentes ni backdoors. Sin embargo, el compromiso evidenciado en logs indica acceso no autorizado mediante credenciales, por lo que las acciones de erradicación se centraron en revocar el vector de acceso (endurecimiento de SSH), revisar cuentas/credenciales y reducir superficie de ataque.

-Bloquear el exploit y prevenir escalación (SSH inseguro)

Empleando **autenticación por contraseña**.

Este vector representa un riesgo crítico, ya que permite acceso completo al servidor.

Para **bloquear completamente este método de entrada** y evitar que el atacante pueda volver a conectarse, se aplicaron varias medidas de hardening en el archivo de configuración de SSH (`/etc/ssh/sshd_config`):

- 1) Se deshabilitó totalmente el acceso SSH del usuario root. (impide cualquier intento de iniciar sesión como root por SSH, incluso utilizando claves).
- 2) Se deshabilitó la autenticación por contraseña. (Esta medida asegura que **ningún usuario del sistema pueda autenticarse mediante password**, reduciendo drásticamente los ataques de fuerza bruta y evitando el mismo tipo de intrusión).
- 3) Se deshabilitó el login interactivo por teclado. (Evita que PAM o métodos alternativos puedan aceptar contraseñas aunque estén bloqueadas explícitamente).
- 4) **PAM permanece habilitado** para la gestión interna de sesiones y políticas del sistema.
- 5) Permite seguir usando los módulos de autenticación del sistema sin exponer contraseñas vía **SSH**.  
**UsePam yes**

- 6) Se reinició el servicio **SSH** para aplicar los cambios.

**sudo systemctl restart sshd**

- Se bloquea por completo el vector de acceso utilizado por el atacante.
- Ningún usuario puede autenticarse por contraseña.
- Root no puede conectarse vía SSH de ninguna forma.
- Se reduce significativamente la superficie de ataque del servidor.