



4Geeks
Academy

INFORME DE INCIDETE DE SEGURIDAD

TFM



Alejandro Gil Sánchez

Mayo 2025

Informe de Incidente de Seguridad

1.	Introducción	1
1.1	Objetivo y alcance	1
1.2	Descripción del Servidor Debian.....	1
2.	Análisis Forense	1
2.1	Análisis de Logs y Accesos	1
2.2	Revisión de Historial de Comandos	4
2.3	Identificación de modificaciones inusuales	5
2.4	Revisión de MySQL	7
2.5	Búsqueda de procesos sospechosos.....	8
2.6	Detección de rootkits o malware	9
3.	Mitigación de vulnerabilidades	10
3.1	Actualizar el sistema.....	10
3.2	Cambio de contraseñas.....	10
3.3	Servicio SSH	10
3.4	Servicio wordpress	11
3.5	Configuración de Red	12
4.	Conclusión	13
4.1	Timeline	13
4.2	Impacto del Incidente.....	13
4.3	Reflexión Final	13

1. Introducción

1.1 Objetivo y alcance

En este informe llevaremos a cabo un análisis exhaustivo de una Máquina Virtual Debian la cual aloja un Servidor Crítico perteneciente a 4 Geeks Academy. Dicho Servidor ha sido vulnerado, por lo que procederemos a realizar un análisis forense e identificar la o las vulnerabilidades explotadas por el atacante, qué hizo este una vez accedió al sistema y posteriormente aplicar las medidas y configuraciones necesarias para mejorar la seguridad del Servidor e impedir que pueda ser explotada de nuevo. Para ello se nos ha proporcionado la Máquina Afectada y contamos con todos los privilegios para poder analizarla.

1.2 Descripción del Servidor Debian

- **Sistema operativo:** Linux debían 6.1.0-25-amd64
- **Dirección IP:** 192.168.1.150
- **Dirección MAC:** 08:00:27:F6:50:4C
- **Servicios corriendo:**
 - **FTP** – vsftpd 3.0.3
 - **SSH** – OpenSSH 9.2p1 Debian
 - **HTTP** – Apache httpd 2.4.62

2. Análisis Forense

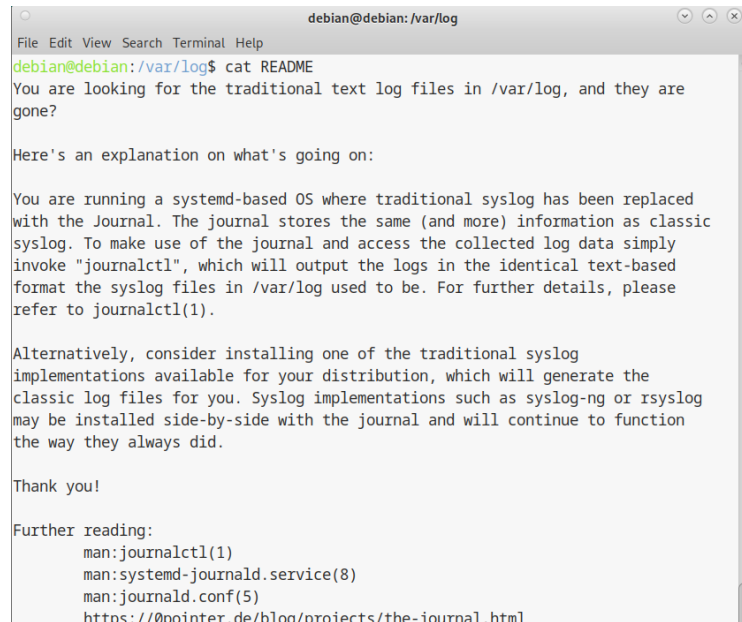
2.1 Análisis de Logs y Accesos

Inicialmente nos dirigimos al directorio `/var/log/` donde se alojan todos los logs del sistema, entre ellos buscamos el archivo `auth.log`, que registra los logs de autenticación del sistema de modo que podemos investigar Access sospechosos. Tras una revisión del directorio `/var/log/` nos percatamos de que no hay archivo `auth.log` y tampoco hay registros de interés en los demás archivos logs, el

```
debian@debian:/var/log$ ls -la
.          boot.log      dpkg.log.1    lightdm        Xorg.0.log
..         boot.log.1    faillog       private        Xorg.0.log.old
alternatives.log  btmp          fontconfig.log  README
alternatives.log.1 btmp.1        installer       runit
apache2          cups          journal         speech-dispatcher
apt              dpkg.log      lastlog         wtmp
```

También revisamos el archivo **README**, el cual nos informa que se ha reemplazado el “Syslog” por el “Journal”.

Los logs de Journal se encuentran en la carpeta con su mismo nombre, nos ayudaremos del comando **journalctl** para interpretar los datos.



```
debian@debian: /var/log$ cat README
You are looking for the traditional text log files in /var/log, and they are gone?

Here's an explanation on what's going on:

You are running a systemd-based OS where traditional syslog has been replaced with the Journal. The journal stores the same (and more) information as classic syslog. To make use of the journal and access the collected log data simply invoke "journalctl", which will output the logs in the identical text-based format the syslog files in /var/log used to be. For further details, please refer to journalctl(1).

Alternatively, consider installing one of the traditional syslog implementations available for your distribution, which will generate the classic log files for you. Syslog implementations such as syslog-ng or rsyslog may be installed side-by-side with the journal and will continue to function the way they always did.

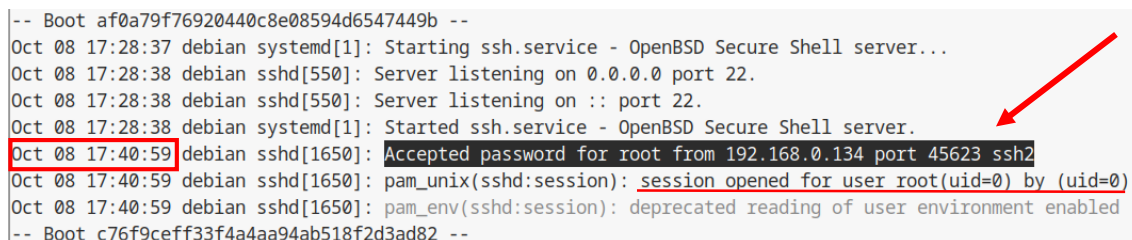
Thank you!

Further reading:
man:journalctl(1)
man:systemd-journald.service(8)
man:journald.conf(5)
https://0pointer.de/blog/projects/the-journal.html
```

➤ Entrada del atacante al sistema

Los primeros registros son del 31 de julio, como hay infinidad de datos e información, procederemos a hacer un filtrado buscando accesos con ssh por ejemplo:

Comando: `sudo journal -u ssh.service`



```
-- 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 c76f9ceff33f4a4aa94ab518f2d3ad82 --
```

No hay registros relevantes hasta este, un usuario desconocido inicia sesión como **root** mediante el Servicio **SSH** el día **8 de octubre a las 17:40:59** con **Ip 192.168.0.134** por el **puerto de red 45623**.

Observamos que el denominado atacante realizó un inicio de sesión sin intentos fallidos a la hora de autenticarse, esto nos lleva a barajar las siguientes opciones:

- Credenciales filtradas o comprometidas por lo que el atacante disponía de los datos de inicio de sesión previamente.

- La contraseña es muy débil, evidentemente hay que cambiarla por una más compleja, y el atacante uso una herramienta y diccionario de fuerza bruta, como por ejemplo la herramienta hydra, y la primera opción fue la correcta.

```
> hydra -L /usr/share/wordlists/metasploit/unix users.txt -P /usr/share/wordlists/rockyou.txt ssh://192.168.1.150
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-17 20:48:52
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 2524614224 login tries (l:176/p:14344399), ~157788389 tries per task
[DATA] attacking ssh://192.168.1.150:22/
[22][ssh] host: 192.168.1.150 login: root password: 123456
[22][ssh] host: 192.168.1.150 login: debian password: 123456
```

➤ Cambios en la configuración de red

Una vez confirmamos que el atacante inició sesión el 8 de octubre, procedemos a analizar más en concreto ese día y encontramos:

```
Oct 08 17:57:51 debian NetworkManager[498]: <info> [1728424671.0885] dhcp6 (enp0s3): state changed new lease
, address=2800:810:458:173:8879:c1b2:f8ba:ceb5
Oct 08 17:58:38 debian NetworkManager[498]: <info> [1728424718.3542] dhcp4 (enp0s3): state changed new lease
, address=192.168.0.137
```

Podemos apreciar como a las **17:57:51** el atacante desactiva el DHCPv6 y el DHCPv4 para establecer direcciones estáticas:

- **IPv6 address** → 2800:810:458:173:8879:c1b2:f8ba:ceb5
La dirección 2800:81 pertenece al bloque de direcciones IPv6 asignado a América Latina y el Caribe (LACNIC). Por lo tanto, se trata de una dirección IPv6 global o agregable única (ULA).
- **IPv4 address** → 192.168.0.137
Podemos apreciar de que la Ip que cambia está en la subred 192.168.0.0/24, la misma subred de la IP del atacante (192.168.0.134) por lo que tendríamos que contemplar si el atacante estaba conectado a la misma subred.

➤ Desconexión y conexión de USBs

A las **18:03:07**, se registra como se desconecta un USB con device number 2 y conecta un dispositivo USB con device number 3.

1. Oct 08 18:03:07 debian kernel: usb 2-1: USB disconnect, device number 2
2. Oct 08 18:03:07 debian kernel: usb 2-1: new full-speed USB device number 3 using ohci-pci
- Oct 08 18:03:08 debian kernel: usb 2-1: New USB device found, idVendor=80ee, idProduct=0021, bcdDevice= 1.00
- Oct 08 18:03:08 debian kernel: usb 2-1: New USB device strings: Mfr=1, Product=3, SerialNumber=0
- Oct 08 18:03:08 debian kernel: usb 2-1: Product: USB Tablet
- Oct 08 18:03:08 debian kernel: usb 2-1: Manufacturer: VirtualBox
- Oct 08 18:03:08 debian kernel: input: VirtualBox USB Tablet as /devices/pci0000:00/0000:00:06.0/usb2/2-1/2-1:1.0/0003:80EE:0021.0002/input/input9
- Oct 08 18:03:08 debian kernel: hid-generic 0003:80EE:0021.0002: input,hidraw0: USB HID v1.10 Mouse [VirtualBox x USB Tablet] on usb-0000:00:06.0-1/input0
- Oct 08 18:03:08 debian mtp-probe[1707]: checking bus 2, device 3: "/sys/devices/pci0000:00/0000:00:06.0/usb2/2-1"
- Oct 08 18:03:08 debian mtp-probe[1707]: bus: 2, device: 3 was not an MTP device
- Oct 08 18:03:08 debian mtp-probe[1712]: checking bus 2, device 3: "/sys/devices/pci0000:00/0000:00:06.0/usb2/2-1"
- Oct 08 18:03:08 debian mtp-probe[1712]: bus: 2, device: 3 was not an MTP device

1. **18:03:07**: Se desconecta un USB device number 2.
2. **18:03:07**: Inmediatamente, en el mismo puerto, se conecta un USB device number 3, mediante u
3. **18:03:08**: El sistema reconoce el dispositivo como un periférico de entrada (mouse/tableta).

2.2 Revisión de Historial de Comandos

Procedemos a analizar el historial de comandos de los usuarios con Shell:

➤ Usuario root:

```
debian@debian:~$ sudo cat /root/.bash_history
[sudo] password for debian:
sudo visudo
sudo systemctl stop speech-dispatcher
sudo systemctl disable speech-dispatcher
systemctl list-units --type=service
cd apache2/
ls
systemctl stop ssh
ls
cat access.log
ls
cat error.log
exit
```

Podemos observar como no hay ningún indicio sospechoso en los comandos emitidos por el usuario root, aunque es posible que el atacante borrase parte del historial antes de abandonar el sistema.

➤ Usuario debían:

```
debian@debian:~$ cat ~/.bash_history
sudo systemctl stop speech-dispatcher
sudo usermood -aG root debian
pwd
sudo usermood -aG sudo debian
whoami
sudo visudo
su
sudo imod speakup
sudo immod speakup
sudo immod speakup_soft
sudo apt-get remove speakup
sudo apt-get remove speakup_soft
sudo ls /etc
sudo ls /etc/modprobe.d/
sudo nano /etc/modprobe.d/blacklist-speakup.conf
sudo nano /etc/default/grub
sudo update-grub
sudo reboot
```

Tampoco observamos comportamientos sospechosos en los comandos del usuario debían, aunque al igual que comente anteriormente el atacante pudo limpiar su rastro antes de abandonar el sistema.

2.3 Identificación de modificaciones inusuales

Teniendo en cuenta que sabemos que el atacante accedió al sistema con privilegios de root, procedemos a analizar todos los archivos y configuraciones sensibles:

➤ Revisión de usuarios en /etc/passwd/

Este archivo contiene a los usuarios del sistema, revisamos si se ha creado un usuario con shell nuevo a partir de la fecha y hora del ataque:

```
debian@debian:~$ cat /etc/passwd | grep "sh"
root:x:0:0:root:/root:/bin/bash
debian:x:1000:1000:4geeks,,,:/home/debian:/bin/bash
sshd:x:112:65534:./run/sshd:/usr/sbin/nologin
```

Como vemos solo están el usuario **root**, el usuario **debian** y el usuario **sshd**.

- **Usuario Root:** es una cuenta de usuario especial con privilegios administrativos sin restricciones que se crea automáticamente en cualquier Sistema Linux.
- **Usuario Sshd** no es una cuenta de usuario real en un sistema. sshd es el nombre del dominio del servidor SSH, que es el proceso que escucha las conexiones SSH entrantes. No es un usuario con el que se pueda iniciar sesión.

- **Usuario Debian:** cuenta creada manualmente, comprobamos que su directorio home fue creado el 31-07-2024, antes de la entrada del atacante del sistema mediante el comando **stat**. Por lo que llegamos a la conclusión de que no se crearon nuevos usuarios a raíz del ataque.

```

debian@debian:/home$ stat debian
  File: debian
  Size: 4096          Blocks: 8          IO Block: 4096   directory
Device: 8,1      Inode: 1179651      Links: 14
Access: (0700/drwx-----)  Uid: ( 1000/  debian)   Gid: ( 1000/  debian)
Access: 2024-07-31 18:18:15.491021442 -0400
Modify: 2025-05-16 04:50:48.992000000 -0400
Change: 2025-05-16 04:50:48.992000000 -0400
Birth: 2024-07-31 14:18:55.772602000 -0400

```

➤ Revisión de permisos en directorios/archivos sensibles:

Analizamos si se han modificado los permisos de directorios sensibles y observamos permisos inusuales en `/var/www/html`:

Podemos apreciar como todos los directorios o archivos tienen todos los permisos (lectura, escritura y ejecución) habilitados, lo cual es una falla de seguridad muy importante y no es la configuración predeterminada, por lo que deducimos que ha sido modificado por el atacante.

```

debian@debian:/var/www/html$ ls -l
total 248
-rwxrwxrwx 1 www-data www-data 10701 Sep 30 2024 index.html
-rwxrwxrwx 1 www-data www-data 405 Feb 6 2020 index.php
-rwxrwxrwx 1 www-data www-data 19903 Apr 21 19:35 license.txt
-rwxrwxrwx 1 www-data www-data 7425 Apr 21 19:35 readme.html
-rwxrwxrwx 1 www-data www-data 7387 Feb 13 2024 wp-activate.php
drwxrwxrwx 9 www-data www-data 4096 Sep 10 2024 wp-admin/
-rwxrwxrwx 1 www-data www-data 351 Feb 6 2020 wp-blog-header.php
-rwxrwxrwx 1 www-data www-data 2323 Jun 14 2023 wp-comments-post.php
-rwxrwxrwx 1 www-data www-data 3017 Sep 30 2024 wp-config.php
-rwxrwxrwx 1 www-data www-data 3336 Apr 14 13:38 wp-config-sample.php
drwxrwxrwx 6 www-data www-data 4096 Apr 21 19:35 wp-content/
-rwxrwxrwx 1 www-data www-data 5617 Apr 14 13:38 wp-cron.php
drwxrwxrwx 10 www-data www-data 12288 Apr 21 19:35 wp-includes/
-rwxrwxrwx 1 www-data www-data 2502 Nov 26 2022 wp-links-opml.php
-rwxrwxrwx 1 www-data www-data 3937 Mar 11 2024 wp-load.php
-rwxrwxrwx 1 www-data www-data 51414 Apr 21 19:35 wp-login.php
-rwxrwxrwx 1 www-data www-data 8727 Apr 21 19:35 wp-mail.php
-rwxrwxrwx 1 www-data www-data 30081 Apr 21 19:35 wp-settings.php
-rwxrwxrwx 1 www-data www-data 34516 Apr 21 19:35 wp-signup.php
-rwxrwxrwx 1 www-data www-data 5102 Apr 14 13:38 wp-trackback.php
-rwxrwxrwx 1 www-data www-data 3205 Apr 21 19:35 xmlrpc.php

```


Una vez detectado esto, comprobamos el archivo wp-config.php, que aloja la configuración y credenciales para acceder a MySQL desde el equipo.

```
// ** Database settings - You can get this info from your web host ** //  
/** The name of the database for WordPress */  
define( 'DB_NAME', 'wordpress' );  
  
/** Database username */  
define( 'DB_USER', 'wordpressuser' );  
  
/** Database password */  
define( 'DB_PASSWORD', '123456' );
```

Vemos como aparte de que el atacante tenía acceso a dicho archivo, las credenciales son muy débiles y ha podido acceder a MySQL.

2.4 Revisión de MySQL

➤ Historial de root de MySQL

Analizamos el archivo /root/.mysql_history

```
1 _HiStoRy_V2_  
2 CREATE DATABASE wordpress DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;  
3 CREATE USER wordpressuser@'localhost' IDENTIFIED BY '123456';  
4 GRANT ALL PRIVILEGES ON wordpress.* TO wordpress@'localhost';  
5 GRANT ALL PRIVILEGES ON wordpress.* TO wordpressuser@'localhost';  
6 FLUSH PRIVILEGES;  
7 FLUSH PRIVILEGES;  
8 EXIT;  
9 CREATE USER user@'localhost' IDENTIFIED BY 'password';  
10 GRANT ALL PRIVILEGES ON *.* TO user@'localhost' WITH GRANT OPTION;  
11 FLUSH PRIVILEGES;  
12 EXIT;
```

En las primeras líneas observamos la creación de:

- La Base de Datos Wordpress
- Un nuevo usuario y contraseña → **wordpressuser ; 123456**
A pesar de tener una contraseña tan débil, se otorga la creación al usuario root, no al atacante.
- Un nuevo usuario y asignación de todos los privilegios, **y que pueda otorgar privilegios a terceros**. En este caso si indica que el atacante ha realizado esta acción:
Usuario y contraseña → **user ; password**

➤ Información sensible de la Base de Datos

Analizando la base de datos para analizar contenido sensible, ya que tenemos pruebas de que ha sido vulnerada y posiblemente se haya extraído toda la información.

En la Base de Datos wordpress, en la tabla wp_users encontramos:

`SELECT user_nicename, user_pass, user_email, user_registered FROM wp_users;`

```
MariaDB [wordpress]> SELECT user_nicename, user_pass, user_email, user_registered FROM wp_users;
+-----+-----+-----+-----+
| user_nicename | user_pass | user_email | user_registered |
+-----+-----+-----+-----+
| wordpress-user | $P$BM4LABXxcoawTZfuH8QRU5dcnKT2IC. | rosinnicuentas@gmail.com | 2024-09-30 16:23:12 |
+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

Damos por hecho que el atacante ha accedido a la base de datos y robó todos los datos, hemos encontrado una dirección Gmail en la Base de Datos wordpress

→ rosinnicuentas@gmail.com.

La contraseña está cifrada pero habría que dar por hecho su filtración, cambiarla inmediatamente y asegurarse de que el usuario no tiene repetida la contraseña en otras redes sociales.

2.5 Búsqueda de procesos sospechosos

➤ Listar procesos en ejecución

Revisamos procesos inusuales o ejecutados por usuarios desconocidos. Prestando atención a rutas raras, scripts, o binarios fuera de lugar.
No se encuentran procesos sospechosos corriendo en el sistema.

```
debian@debian:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.6 102312 12364 ?        Ss   06:02   0:00 /sbin/init sp
root         2  0.0  0.0      0     0 ?        S    06:02   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        I<   06:02   0:00 [rcu_gp]
root         4  0.0  0.0      0     0 ?        I<   06:02   0:00 [rcu_par_gp]
root         5  0.0  0.0      0     0 ?        I<   06:02   0:00 [slub_flushwq]
root         6  0.0  0.0      0     0 ?        I<   06:02   0:00 [netns]
root        10  0.0  0.0      0     0 ?        I<   06:02   0:00 [mm_percpu_wq]
root        11  0.0  0.0      0     0 ?        I    06:02   0:00 [rcu_tasks_kt
root        12  0.0  0.0      0     0 ?        I    06:02   0:00 [rcu_tasks_ru
root        13  0.0  0.0      0     0 ?        I    06:02   0:00 [rcu_tasks_tr
root        14  0.0  0.0      0     0 ?        S    06:02   0:00 [ksoftirqd/0]
root        15  0.0  0.0      0     0 ?        I    06:02   0:00 [rcu_preempt]
root        16  0.0  0.0      0     0 ?        S    06:02   0:00 [migration/0]
```

➤ Búsqueda de tareas programadas

- **El Crontab** de Linux es una herramienta que te permite configurar y programar tareas repetitivas para que se ejecuten de manera automática en segundo plano. No se encuentra ninguna configuración sospechosa.

```
debian@debian:~$ crontab -l
no crontab for debian
debian@debian:~$ sudo crontab -l
[sudo] password for debian:
no crontab for root
```

- Revisamos si hay **servicios sospechosos** cuando iniciamos el sistema.
`systemctl list-unit-files --state=enabled`
No se observa nada inusual.

```
debian@debian:~$ systemctl list-unit-files --state=enabled
```

UNIT FILE	STATE	PRESET
cups.path	enabled	enabled
accounts-daemon.service	enabled	enabled
anacron.service	enabled	enabled
apache2.service	enabled	enabled
apparmor.service	enabled	enabled
avahi-daemon.service	enabled	enabled
bluetooth.service	enabled	enabled
console-setup.service	enabled	enabled
cron.service	enabled	enabled
cups-browsed.service	enabled	enabled
cups.service	enabled	enabled
e2scrub_reap.service	enabled	enabled
getty@.service	enabled	enabled

- Comprobamos si hay conexiones de red activas
`ss -tunlp`

```
debian@debian:~$ ss -tunlp
```

Netid	State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port	Process
udp	UNCONN	0	0	0.0.0.0:5353	0.0.0.0:*	
udp	UNCONN	0	0	0.0.0.0:46919	0.0.0.0:*	
udp	UNCONN	0	0	:::36870	:::*	
udp	UNCONN	0	0	:::5353	:::*	
tcp	LISTEN	0	128	127.0.0.1:631	0.0.0.0:*	
tcp	LISTEN	0	80	127.0.0.1:3306	0.0.0.0:*	
tcp	LISTEN	0	128	0.0.0.0:22	0.0.0.0:*	
tcp	LISTEN	0	128	:::1:631	:::*	
tcp	LISTEN	0	32	*:21	*:*	
tcp	LISTEN	0	128	:::22	:::*	
tcp	LISTEN	0	511	*:80	*:*	

2.6 Detección de Rookits o Malware

Los atacantes pueden haber instalado **rookits o malware** para mantener accesos ocultos, por lo que procedemos a escanear el servidor con la herramienta chkrootkit.

```
debian@debian:/etc/ssh$ sudo rkhunter --checkall
[ Rootkit Hunter version 1.4.6 ]

Checking system commands...

Performing 'strings' command checks
Checking 'strings' command [ OK ]

Performing 'shared libraries' checks
Checking for preloading variables [ None found ]
Checking for preloaded libraries [ None found ]
Checking LD_LIBRARY_PATH variable [ Not found ]

Performing file properties checks
Checking for prerequisites [ OK ]
/usr/bin/systemctl [ OK ]
/usr/bin/gawk [ OK ]
/usr/bin/lwp-request [ Warning ]
```

El único aviso que tenemos es con `/usr/bin/lwp-request`, que es un script legítimo que forma parte del paquete `libwww-perl` en sistemas Linux. Es una herramienta de línea de comandos escrita en Perl para realizar peticiones HTTP y consultas web, similar a `curl` o `wget`.

3. Mitigación de vulnerabilidades

Servicio	Vulnerabilidad	Riesgo
SSH	Acceso directo como root desde 192.168.0.134 (Oct 08).	Escalada de privilegios total.

3.1 Actualizar el sistema

Gran parte de las vulnerabilidades de debe, aparte de las configuraciones, por contar con versiones desactualizadas de los distintos servicios que corre el servidor:

Comando: `sudo apt update -y && sudo apt upgrade -y`

3.2 Cambio de contraseñas

Una de las claves del acceso del atacante fue una contraseña muy débil, aunque vatamos a hacer modificaciones en la configuración del SSH, es una buena práctica de seguridad asegurarse de tener contraseñas robustas en el sistema y más la del usuario con totales privilegios como root:

`passwd root`

Contraseña recomendada → m@nt€niMiento_\$ibar1T@_7593

`passwd benian`

Contraseña recomendada → p@7aR1to_€xTrañ@do 6532

3.3 Servicio SSH

➤ Cambios en la configuración

En el archivo Esto lo haremos en el archivo `/etc/ssh/sshd_config` realizaremos los siguientes cambios:

- `PermitRootLogin no` → Prohibir acceso root directo
- `PasswordAuthentication no` → Deshabilitar contraseñas, solo se aceptará autenticación por claves.
- `MaxAuthTries 3` → Un máximo de 3 intentos para autenticarse.
- `LoginGraceTime 30` → Tiempo límite para autenticarse 30 s

➤ Implementación de Firewall

Para ello nos ayudaremos de **iptables**, es una herramienta de firewall para sistemas Linux que actúa como una especie de “guardia fronteriza” entre tu computadora e internet.

- Permitir SSH solo desde IPs autorizadas, por ejemplo 192.168.0.100:
`sudo iptables -A INPUT -p tcp -dport 22 -s 192.168.0.100 -j ACCEPT`
- Bloquear IP del atacante
`sudo iptables -A INPUT -p tcp -dport 22 -s 192.168.0.134 -j DROP`
- Bloquear escaneos de fuerza bruta:
`sudo iptables -A INPUT -p tcp -dport 22 -m recent --name SH_ATTACK --set`
`sudo iptables -A INPUT -p tcp --dport 22 -m recent --name SH_ATTACK --update --seconds 60 --hitcount 5 -j DROP`

3.4 Servicio wordpress

➤ Permisos de /var/www/html

Como mencionamos anteriormente el atacante cambió la asignación de permisos de dicho directorio, le asignamos los permisos correspondientes:

```
cd /var/www
```

```
sudo chmod 755 /html
```

Asignación de permisos mínimos:

```
cd /html/
```

```
sudo chmod 611 *
```

```
sudo chmod 755 wp-admin/ wp-content/ wp-includes/
```

```
debian@debian:/var/www/html$ ls -l
total 248
-rw-r--r-- 1 www-data www-data 10701 Sep 30 2024 index.html
-rw-r--r-- 1 www-data www-data 405 Feb 6 2020 index.php
-rw-r--r-- 1 www-data www-data 19903 May 17 21:58 license.txt
-rw-r--r-- 1 www-data www-data 7425 May 17 21:58 readme.html
-rw-r--r-- 1 www-data www-data 7387 Feb 13 2024 wp-activate.php
drwxr-xr-x 9 www-data www-data 4096 Sep 10 2024 wp-admin
-rw-r--r-- 1 www-data www-data 351 Feb 6 2020 wp-blog-header.php
-rw-r--r-- 1 www-data www-data 2323 Jun 14 2023 wp-comments-post.php
-rw-r--r-- 1 www-data www-data 3017 Sep 30 2024 wp-config.php
-rw-r--r-- 1 www-data www-data 3336 May 17 21:58 wp-config-sample.php
drwxr-xr-x 6 www-data www-data 4096 May 17 22:05 wp-content
-rw-r--r-- 1 www-data www-data 5617 May 17 21:58 wp-cron.php
drwxr-xr-x 30 www-data www-data 12288 May 17 21:58 wp-includes
-rw-r--r-- 1 www-data www-data 2502 Nov 26 2022 wp-links-opml.php
-rw-r--r-- 1 www-data www-data 3937 Mar 11 2024 wp-load.php
-rw-r--r-- 1 www-data www-data 51414 May 17 21:58 wp-login.php
-rw-r--r-- 1 www-data www-data 8727 May 17 21:58 wp-mail.php
-rw-r--r-- 1 www-data www-data 30081 May 17 21:58 wp-settings.php
-rw-r--r-- 1 www-data www-data 34516 May 17 21:58 wp-signup.php
-rw-r--r-- 1 www-data www-data 5102 May 17 21:58 wp-trackback.php
-rw-r--r-- 1 www-data www-data 3205 May 17 21:58 xmlrpc.php
```

➤ Usuario creado en MySQL

Como averiguamos anteriormente el atacante creó un usuario con totales privilegios, procedemos a eliminarlo:

```
DROP USER 'user'@'%';  
FLUSH PRIVILEGES;
```

Y por último establecemos una contraseña segura para los usuarios legítimos:

```
ALTER USER 'wordpressuser'@'localhost' IDENTIFIED BY  
'Nuev@Contras€ña_$egura!2024';
```

3.5 Configuración de Red

Se detectó como el usuario malicioso estableció IPs estáticas tanto IPv4 como IPv6, procedemos a activar el DHCP en ambas versiones de nuevo, para ello modificaremos el archivo `/etc/network/interfaces`

```
# This file describes the network interfaces available on your system  
# and how to activate them. For more information, see interfaces(5).  
  
source /etc/network/interfaces.d/*  
  
# The loopback network interface  
auto lo  
iface lo inet loopback  
  
auto enp0s3  
iface enp0s3 inet dhcp  
iface enp0s3 inet6 dhcp
```

A continuación, reiniciamos el servicio de red y DHCP Client:

```
sudo systemctl restart networking
```

```
sudo dhclient -r enp0s3 && sudo dhclient enp0s3
```


4. Conclusión

4.1 Timeline

Los sucesos ocurren el **8 de octubre de 2024**, a continuación, detallamos qué hizo el atacante una vez penetró en el sistema:

- **17:40:59** → el atacante entra al sistema mediante el Servicio SSH autenticándose como root. Con dirección IP **192.168.0.134** por el puerto de red 45623.
- **17:51:57** → el atacante desactiva el DHCPv6 y el DHCPv4 para establecer direcciones estáticas:
 - **IPv6 address** → 2800:810:458:173:8879:c1b2:f8ba:ceb5
 - **IPv4 address** → 192.168.0.137
- **18:03:07** → se desconecta un USB con device number 2 y conecta un dispositivo USB con device number 3.

4.2 Impacto del Incidente

- **Confidencialidad:** Fuga de datos de WordPress (usuarios, hashes de contraseñas).
- **Integridad:** Configuraciones alteradas (red, permisos, servicios).
- **Persistencia:** Riesgo de reentrada si no se revierten todos los cambios maliciosos.

4.3 Reflexión Final

Este incidente pone de manifiesto cómo la combinación de contraseñas débiles, configuraciones inseguras y servicios desactualizados puede facilitar un acceso no autorizado y comprometer seriamente la integridad del sistema. El atacante aprovechó un acceso root mal protegido para tomar control del servidor, alterar configuraciones clave y extraer posibles datos sensibles. Este caso refuerza la necesidad de aplicar medidas de seguridad básicas, mantener una política de actualizaciones continua y limitar el acceso privilegiado. En definitiva, la prevención sigue siendo el pilar fundamental de la ciberseguridad.