

Informe de análisis forense

Fase 1: Reconocimiento y recolección de evidencias. Corrección de un hackeo

Camila Aranibar Pozo
Proyecto final de ciberseguridad

ÍNDICE

1. Introducción.....	3
2. Objetivo y Alcance.....	3
3. Herramientas y Técnicas utilizadas.....	3
4. Identificación de las vulnerabilidades.....	4
4.1. Análisis de Logs y Accesos.....	4
4.2. Identificación de modificaciones inusuales.....	6
4.3. Detección de rootkits o malware.....	8
4.4. Servidor FTP.....	10
4.5. WordPress.....	11
4.6. Directorios web.....	12
5. Corrección y mitigación de las vulnerabilidades.....	13
5.1. Acceso SSH al servidor como root.....	13
5.2. Usuario mysql en la base de datos.....	15
5.3. Actualización.....	16
5.4. Servidor FTP.....	16
5.5. WordPress.....	17
5.6 Directorios web.....	18
6. Recomendaciones para Prevención Futura.....	19

1. Introducción

En este informe se realizará un análisis forense del servidor crítico que ha sido comprometido para determinar todas las vulnerabilidades explotadas por el atacante. También se detallarán todas los bloqueos del exploit, las mitigaciones y correcciones de las vulnerabilidades que se han realizado en el servidor afectado.

2. Objetivo y Alcance

El objetivo de este documento es identificar todas las vulnerabilidades y servicios comprometidos que presenta el servidor crítico de 4Geeks Academy, en este caso corresponde a la máquina hackeada Debian proporcionada.

Este informe de seguridad se ha realizado en un entorno virtual (mediante el VirtualBox) y como alcance se ha centrado únicamente en el análisis de la Máquina hackeada, concretamente es una máquina virtual Debian (IP 192.168.56.50/24).

3. Herramientas y Técnicas utilizadas

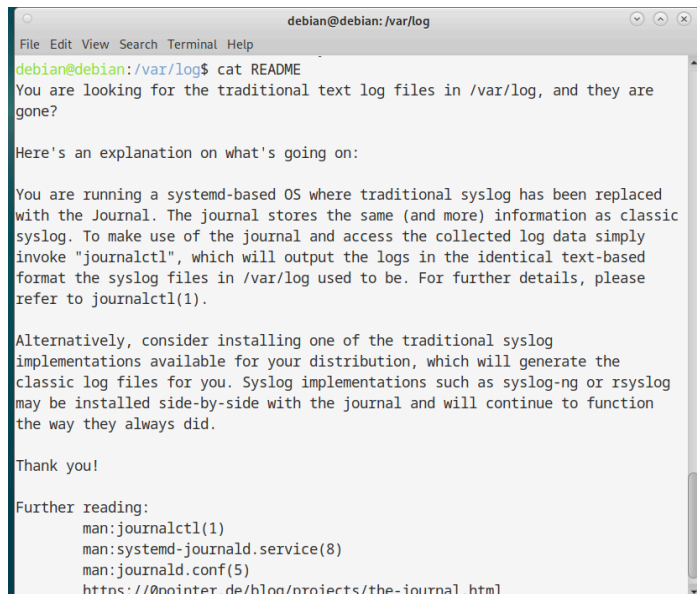
La técnica utilizada para la recuperación de evidencias que se ha utilizado es la adquisición en caliente (la máquina virtual Debian estaba encendida) y por tanto no se ha bloqueado la escritura. Además, no se ha realizado una imagen forense o clonado de la máquina virtual, dado que la máquina virtual comprometida el ejercicio la proporciona y sería el clonado.

4. Identificación de las vulnerabilidades

4.1. Análisis de Logs y Accesos

Se han revisado los logs de autenticación del sistema para encontrar accesos sospechosos y así determinar los servicios que han sido comprometidos. Para ello realizado el siguiente comandos: `cat /var/log/auth.log`

Sin embargo, hemos descubierto que se ha reemplazado el “syslog” por el “journal”.



```
debian@debian: /var/log
File Edit View Search Terminal Help
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:

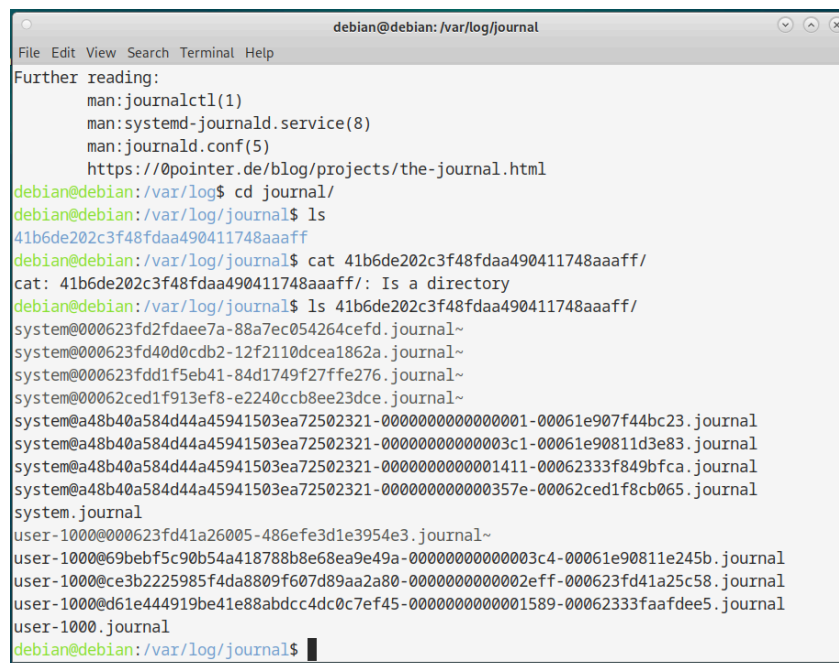
You are running a systemd-based OS where traditional syslog has been replaced with the Journal. The journal stores the same (and more) information as classic syslog. To make use of the journal and access the collected log data simply invoke "journalctl", which will output the logs in the identical text-based 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
```

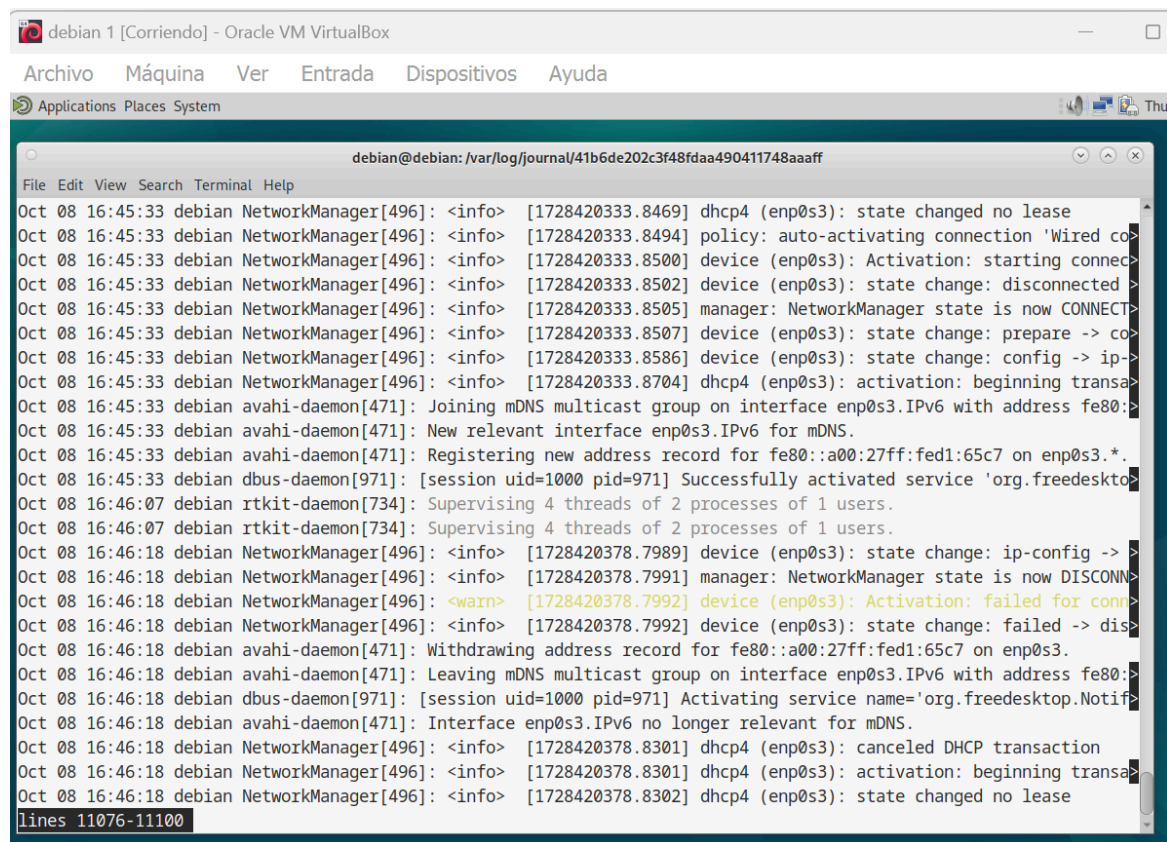
Por tanto accedemos al directorio con el comando: `cd /var/log/journal`



```
debian@debian: /var/log/journal
File Edit View Search Terminal Help
Further reading:
  man:journalctl(1)
  man:systemd-journald.service(8)
  man:journald.conf(5)
  https://0pointer.de/blog/projects/the-journal.html
debian@debian: /var/log$ cd journal/
debian@debian: /var/log/journal$ ls
41b6de202c3f48fdaa490411748aaaff
debian@debian: /var/log/journal$ cat 41b6de202c3f48fdaa490411748aaaff/
cat: 41b6de202c3f48fdaa490411748aaaff/: Is a directory
debian@debian: /var/log/journal$ ls 41b6de202c3f48fdaa490411748aaaff/
system@000623fd2fdae7a-88a7ec054264cefd.journal~
system@000623fd40d0cdb2-12f2110dcea1862a.journal~
system@000623fdd1f5eb41-84d1749f27ffe276.journal~
system@00062ced1f913ef8-e2240ccb8ee23dce.journal~
system@a48b40a584d44a45941503ea72502321-0000000000000001-00061e907f44bc23.journal
system@a48b40a584d44a45941503ea72502321-000000000000003c1-00061e90811d3e83.journal
system@a48b40a584d44a45941503ea72502321-00000000000001411-00062333f849bfca.journal
system@a48b40a584d44a45941503ea72502321-0000000000000357e-00062ced1f8cb065.journal
system.journal
user-1000@000623fd41a26005-486efe3d1e3954e3.journal~
user-1000@69bebf5c90b54a418788b8e68ea9e49a-000000000000003c4-00061e90811e245b.journal
user-1000@ce3b2225985f4da8809f607d89aa2a80-00000000000002eff-000623fd41a25c58.journal
user-1000@d61e444919be41e88abdcc4dc0c7ef45-0000000000001589-00062333faafdee5.journal
user-1000.journal
debian@debian: /var/log/journal$
```

Después revisamos el contenido de todos los archivos para encontrar algún acceso sospechoso. Para obtener una información detallada nos cambiamos a root con comando `su -` y luego analizamos solo los logs dentro de este directorio específico con el comando:

```
journalctl --directory=/var/log/journal/41b6/
```

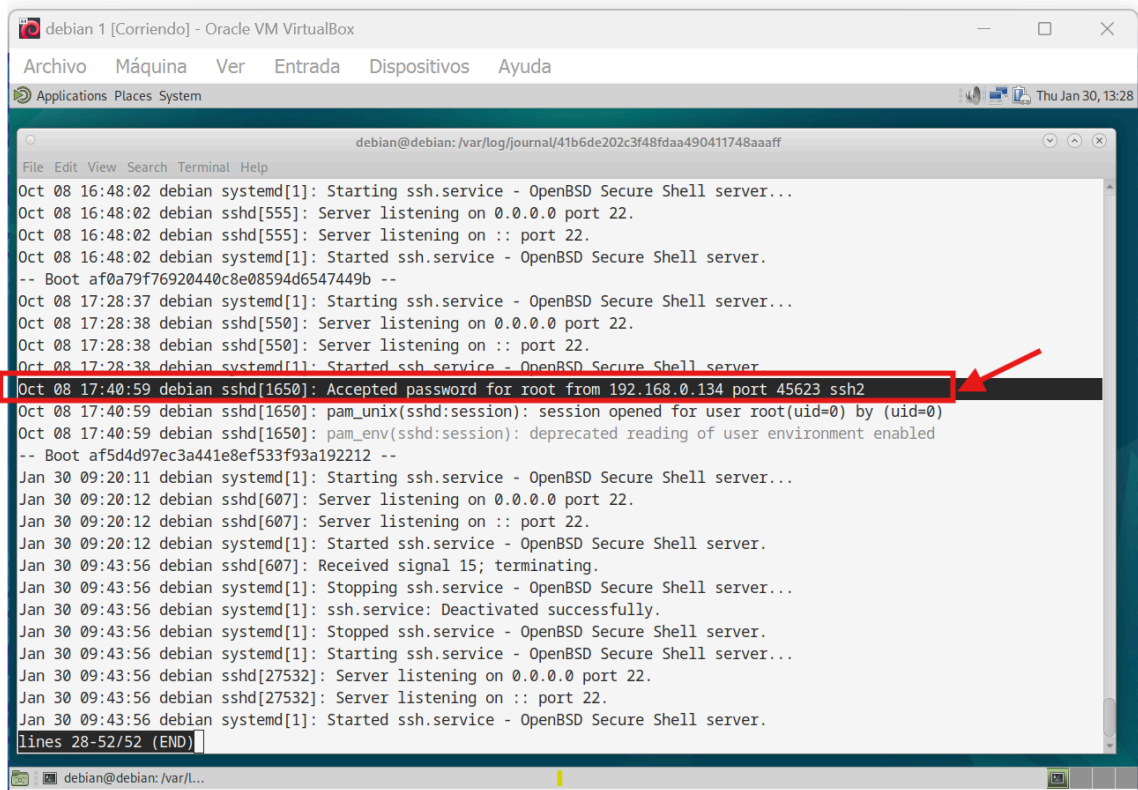


```
debian 1 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Applications Places System
debian@debian: /var/log/journal/41b6de202c3f48fdaa490411748aaaff
File Edit View Search Terminal Help
Oct 08 16:45:33 debian NetworkManager[496]: <info> [1728420333.8469] dhcp4 (enp0s3): state changed no lease
Oct 08 16:45:33 debian NetworkManager[496]: <info> [1728420333.8494] policy: auto-activating connection 'Wired co
Oct 08 16:45:33 debian NetworkManager[496]: <info> [1728420333.8500] device (enp0s3): Activation: starting connec
Oct 08 16:45:33 debian NetworkManager[496]: <info> [1728420333.8502] device (enp0s3): state change: disconnected >
Oct 08 16:45:33 debian NetworkManager[496]: <info> [1728420333.8505] manager: NetworkManager state is now CONNECT
Oct 08 16:45:33 debian NetworkManager[496]: <info> [1728420333.8507] device (enp0s3): state change: prepare -> co
Oct 08 16:45:33 debian NetworkManager[496]: <info> [1728420333.8586] device (enp0s3): state change: config -> ip->
Oct 08 16:45:33 debian NetworkManager[496]: <info> [1728420333.8704] dhcp4 (enp0s3): activation: beginning transa
Oct 08 16:45:33 debian avahi-daemon[471]: Joining mDNS multicast group on interface enp0s3.IPv6 with address fe80::
Oct 08 16:45:33 debian avahi-daemon[471]: New relevant interface enp0s3.IPv6 for mDNS.
Oct 08 16:45:33 debian avahi-daemon[471]: Registering new address record for fe80::a00:27ff:fed1:65c7 on enp0s3.*.
Oct 08 16:45:33 debian dbus-daemon[971]: [session uid=1000 pid=971] Successfully activated service 'org.freedesktop
Oct 08 16:46:07 debian rtkit-daemon[734]: Supervising 4 threads of 2 processes of 1 users.
Oct 08 16:46:07 debian rtkit-daemon[734]: Supervising 4 threads of 2 processes of 1 users.
Oct 08 16:46:18 debian NetworkManager[496]: <info> [1728420378.7989] device (enp0s3): state change: ip-config -> >
Oct 08 16:46:18 debian NetworkManager[496]: <info> [1728420378.7991] manager: NetworkManager state is now DISCONN
Oct 08 16:46:18 debian NetworkManager[496]: <warn> [1728420378.7992] device (enp0s3): Activation: failed for conn
Oct 08 16:46:18 debian NetworkManager[496]: <info> [1728420378.7992] device (enp0s3): state change: failed -> dis
Oct 08 16:46:18 debian avahi-daemon[471]: Withdrawing address record for fe80::a00:27ff:fed1:65c7 on enp0s3.
Oct 08 16:46:18 debian avahi-daemon[471]: Leaving mDNS multicast group on interface enp0s3.IPv6 with address fe80::
Oct 08 16:46:18 debian dbus-daemon[971]: [session uid=1000 pid=971] Activating service name='org.freedesktop.Notif
Oct 08 16:46:18 debian avahi-daemon[471]: Interface enp0s3.IPv6 no longer relevant for mDNS.
Oct 08 16:46:18 debian NetworkManager[496]: <info> [1728420378.8301] dhcp4 (enp0s3): canceled DHCP transaction
Oct 08 16:46:18 debian NetworkManager[496]: <info> [1728420378.8301] dhcp4 (enp0s3): activation: beginning transa
Oct 08 16:46:18 debian NetworkManager[496]: <info> [1728420378.8302] dhcp4 (enp0s3): state changed no lease
lines 11076-11100
```

Observamos que han realizado cambios en la configuración de la máquina como la desactivación de dhcp de la máquina (por lo que hemos tenido que poner una ip fija para el análisis). Dado que hay varios logs nos centramos en revisar los más sospechosos:

- **Eventos relacionados con SSH:**

El comando utilizado es: `journalctl --directory=/var/log/journal/41b6/ -u ssh`. En los resultados observamos que el 8 de Octubre se inició sesión desde 192.168.0.134 mediante el puerto 45623. Esto implica que se abrió una sesión SSH con privilegios de root, por lo que el atacante accedió al servidor utilizando las credenciales comprometidas.



```
debian 1 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Applications  Places  System
Thu Jan 30, 13:28

debian@debian: /var/log/journal/41b6de20c3f48fdaa490411748aaaff
File  Edit  View  Search  Terminal  Help
Oct 08 16:48:02 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Oct 08 16:48:02 debian sshd[555]: Server listening on 0.0.0.0 port 22.
Oct 08 16:48:02 debian sshd[555]: Server listening on :: port 22.
Oct 08 16:48:02 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
-- Boot af0a79f76920440c8e08594d6547449b --
Oct 08 17:28:37 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Oct 08 17:28:38 debian sshd[550]: Server listening on 0.0.0.0 port 22.
Oct 08 17:28:38 debian sshd[550]: Server listening on :: port 22.
Oct 08 17:28:38 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
Oct 08 17:40:59 debian sshd[1650]: Accepted password for root from 192.168.0.134 port 45623 ssh2
Oct 08 17:40:59 debian sshd[1650]: pam_unix(sshd:session): session opened for user root(uid=0) by (uid=0)
Oct 08 17:40:59 debian sshd[1650]: pam_env(sshd:session): deprecated reading of user environment enabled
-- Boot af5d4d97ec3a441e8ef533f93a192212 --
Jan 30 09:20:11 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Jan 30 09:20:12 debian sshd[607]: Server listening on 0.0.0.0 port 22.
Jan 30 09:20:12 debian sshd[607]: Server listening on :: port 22.
Jan 30 09:20:12 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
Jan 30 09:43:56 debian sshd[607]: Received signal 15; terminating.
Jan 30 09:43:56 debian systemd[1]: Stopping ssh.service - OpenBSD Secure Shell server...
Jan 30 09:43:56 debian systemd[1]: ssh.service: Deactivated successfully.
Jan 30 09:43:56 debian systemd[1]: Stopped ssh.service - OpenBSD Secure Shell server.
Jan 30 09:43:56 debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Jan 30 09:43:56 debian sshd[27532]: Server listening on 0.0.0.0 port 22.
Jan 30 09:43:56 debian sshd[27532]: Server listening on :: port 22.
Jan 30 09:43:56 debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
lines 28-52/52 (END)
```

4.2. Identificación de modificaciones inusuales

Como hemos comprobado previamente, se ha producido un acceso no autorizado por SSH. Este atacante podría haber modificado procesos, archivos o crear usuarios, con lo que nos centraremos en identificar estas modificaciones sospechosas.

- **Modificación de procesos y archivos:**

Se ha verificado que no hay ninguna tarea programada en el cron (`sudo crontab -l`). También se ha revisado los procesos de ejecución (`ps aux --sort=-%cpu`) y no se ha encontrado nada extraño.

- **Creación de usuarios:**

Verificamos si hay nuevos usuarios con el comando: `cat /etc/passwd | grep "/bin/bash"` y nos aseguramos que no se ha modificado el archivo donde se guardan los usuarios (`/etc/passwd`).

```
debian@debian:/$ cat /etc/passwd |grep "/bin/bash"
root:x:0:0:root:/root:/bin/bash
debian:x:1000:1000:4geeks,,,:/home/debian:/bin/bash
debian@debian:/$ ls -l /etc/passwd
-rw-r--r-- 1 root root 2004 Oct  8 16:09 /etc/passwd
debian@debian:/$
```

Teniendo en cuenta los resultados, confirmamos que no se han creado nuevos usuarios ya que solo están el usuario root (administrador) y el usuario debian (sistema).

Cuando se ha revisado los procesos de ejecución se ha detectado que se usa MariaDB (MySQL) por lo que también revisamos los usuarios de la base de datos mediante el comando:

```
sudo mysql -e "SELECT user, host, password FROM mysql.user;"
```

```
debian@debian:/$ sudo mysql -e "SELECT user, host, password FROM mysql.user;"
+-----+-----+-----+
| User      | Host      | Password                                     |
+-----+-----+-----+
| mariadb.sys | localhost |                                             |
| root       | localhost | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| mysql      | localhost | invalid                                  |
| wordpressuser | localhost | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| user       | localhost | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 |
+-----+-----+-----+
debian@debian:/$
```

Observamos que el usuario "mysql" tiene las credenciales inválidas. Verificamos si lo ha creado el atacante:

```
debian@debian:/$ sudo mysql -e "SELECT user, host, plugin FROM mysql.user WHERE user='mysql';"
+-----+-----+-----+
| User | Host      | plugin                |
+-----+-----+-----+
| mysql | localhost | mysql_native_password |
+-----+-----+-----+
debian@debian:/$
```

Este usuario está usando el plugin *mysql_native_password*, lo que permite la autenticación con contraseña en texto plano y facilita los ataques de fuerza bruta. Para mayor seguridad verificamos los permisos de este usuario:

```

debian@debian:/$ sudo mysql -e "SHOW GRANTS FOR 'mysql'@'localhost';"
+-----+
| Grants for mysql@localhost |
+-----+
| GRANT ALL PRIVILEGES ON *.* TO 'mysql'@'localhost' IDENTIFIED VIA mysql_native_password USING 'invalid' OR unix_socket WITH GRANT OPTION |
| GRANT PROXY ON ''@'%' TO 'mysql'@'localhost' WITH GRANT OPTION |
+-----+
debian@debian:/$

```

Este usuario tiene acceso total al servidor, por lo que es bastante peligroso ya que puede hacer cambios irreversibles en la base de datos y es una puerta trasera que el atacante ha dejado.

También revisamos la contraseña de este usuario:

```

debian@debian:/$ sudo mysql -e "SELECT user, host, authentication_string FROM mysql.user WHERE user='mysql';"
+-----+-----+-----+
| User | Host | authentication_string |
+-----+-----+-----+
| mysql | localhost | invalid |
+-----+-----+-----+
debian@debian:/$

```

Comprobamos que tiene las credenciales dañadas o mal configuradas. A continuación revisamos si este usuario permite acceso remoto:

```

debian@debian:/$ sudo mysql -e "SELECT user, host FROM mysql.user WHERE user='mysql';"
+-----+-----+
| User | Host |
+-----+-----+
| mysql | localhost |
+-----+-----+
debian@debian:/$

```

Confirmamos que este usuario solo puede autenticarse desde la misma máquina debian (localhost) y no puede conectarse de forma remota desde cualquier otra IP. A pesar de esto sigue siendo peligroso ya que sus permisos le dan acceso total al servidor.

4.3. Detección de rootkits o malware

Los atacantes pueden haber instalado **rootkits o malware** para mantener accesos ocultos, por lo que escaneamos el servidor (`sudo chkrootkit`):


```
WARNING: Output from ifpromisc:
lo: not promisc and no packet sniffer sockets
enp0s8: not promisc and no packet sniffer sockets
enp0s9: PACKET SNIFFER(/usr/sbin/NetworkManager[479], /usr/sbin/NetworkManager[479])
```

Observamos que solo hay un warning que indica que la interfaz de red enp0s9 está en modo promiscuo, es decir, que puede leer tráfico de otras interfaces. Esto en principio no es un problema de seguridad.

También realizamos un escaneo con rkhunter (`sudo rkhunter --check --sk`). Como resultados hemos obtenido 2 warnings:

```
/usr/bin/gawk           [ OK ]
/usr/bin/lwp-request    [ Warning ]
/usr/bin/mail.mailutils [ OK ]
/usr/bin/dash           [ OK ]
```

```
Performing malware checks
Checking running processes for suspicious files [ None found ]
Checking for login backdoors                  [ None found ]
Checking for sniffer log files                 [ None found ]
Checking for suspicious directories            [ None found ]
Checking for suspicious (large) shared memory segments [ Warning ]
Checking for Apache backdoor                  [ Not found ]
```

El primer warning, lwp-request, es parte de libwww-perl que es una biblioteca usada en Perl para hacer solicitudes HTTP. En principio no es malicioso, sin embargo los atacantes pueden usarlo para descargar archivos o ejecutar exploits. Por tanto lo verificamos:

```
debian@debian:/$ ls -l /usr/bin/lwp-request
-rwxr-xr-x 1 root root 16202 Mar  1 2023 /usr/bin/lwp-request
```

El archivo pertenece a root y los permisos que tiene son normales, así que descartamos como vulnerabilidad.

Por otra parte, el segundo warning indica que hay segmentos de memoria compartida de gran tamaño en uso. Vemos los segmentos de memoria compartida activos:

```

debian@debian:/$ ipcs -m

----- Shared Memory Segments -----
key          shmid      owner      perms      bytes      nattch     status
0x00000000 8          debian     600        524288     2          dest
0x00000000 13         debian     600        4194304   2          dest
0x00000000 16         debian     600        524288     2          dest
0x00000000 131090    debian     600        524288     2          dest
0x00000000 19         debian     600        524288     2          dest
0x00000000 22         debian     600        524288     2          dest
0x00000000 25         debian     600        524288     2          dest
0x00000000 28         debian     600        524288     2          dest
0x00000000 31         debian     600        524288     2          dest
0x00000000 36         debian     600        33554432  2          dest
0x00000000 65586     debian     600        33554432  2          dest
0x00000000 65590     debian     600        33554432  2          dest
0x00000000 59         debian     600        4194304   2          dest

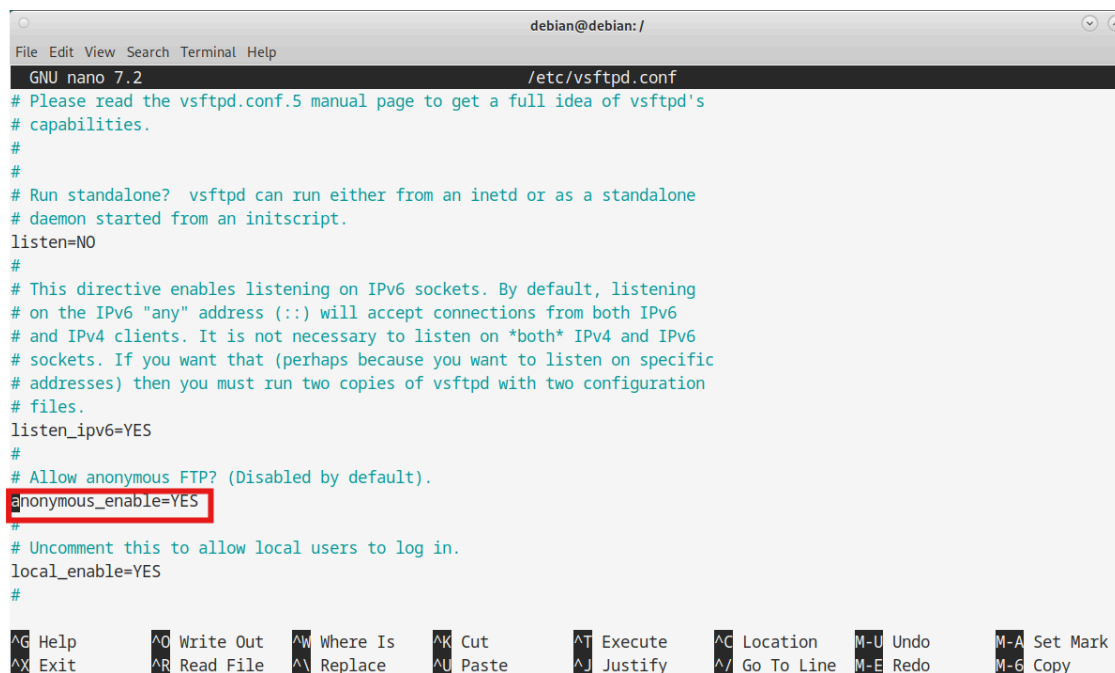
debian@debian:/$

```

No observamos ningún segmento sospechoso, por lo que también lo descartamos como vulnerabilidad.

4.4. Servidor FTP

Nos hemos asegurado que el servidor FTP esté operativo y hemos revisado su configuración (`sudo nano /etc/vsftpd.conf`). Detectamos que se permite acceso anónimo sin autenticación:



```

debian@debian: /
File Edit View Search Terminal Help
GNU nano 7.2 /etc/vsftpd.conf
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
#
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location  M-U Undo     M-A Set Mark
^X Exit      ^R Read File ^A Replace   ^U Paste     ^J Justify  ^_ Go To Line M-E Redo     M-G Copy

```

También podemos observar que se permite el acceso externo (`listen=NO` y `listen_ipv6=YES`). Esto no es una vulnerabilidad si tienes bien protegida tu zona perimetral, pero por si acaso se puede restringir también.

4.5. WordPress

Verificamos que el apache está activo para ver si en esta máquina debian está el servicio de wordpress operativo:

```
debian@debian:/$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-01-30 09:44:16 EST; 10h ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 30975 (apache2)
      Tasks: 6 (limit: 2284)
     Memory: 24.5M
        CPU: 4.328s
    CGroup: /system.slice/apache2.service
            └─30975 /usr/sbin/apache2 -k start
              └─30993 /usr/sbin/apache2 -k start
                └─30994 /usr/sbin/apache2 -k start
                  └─30995 /usr/sbin/apache2 -k start
                    └─30996 /usr/sbin/apache2 -k start
                      └─30997 /usr/sbin/apache2 -k start

Jan 30 09:44:16 debian systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jan 30 09:44:16 debian systemd[1]: Started apache2.service - The Apache HTTP Server.
```

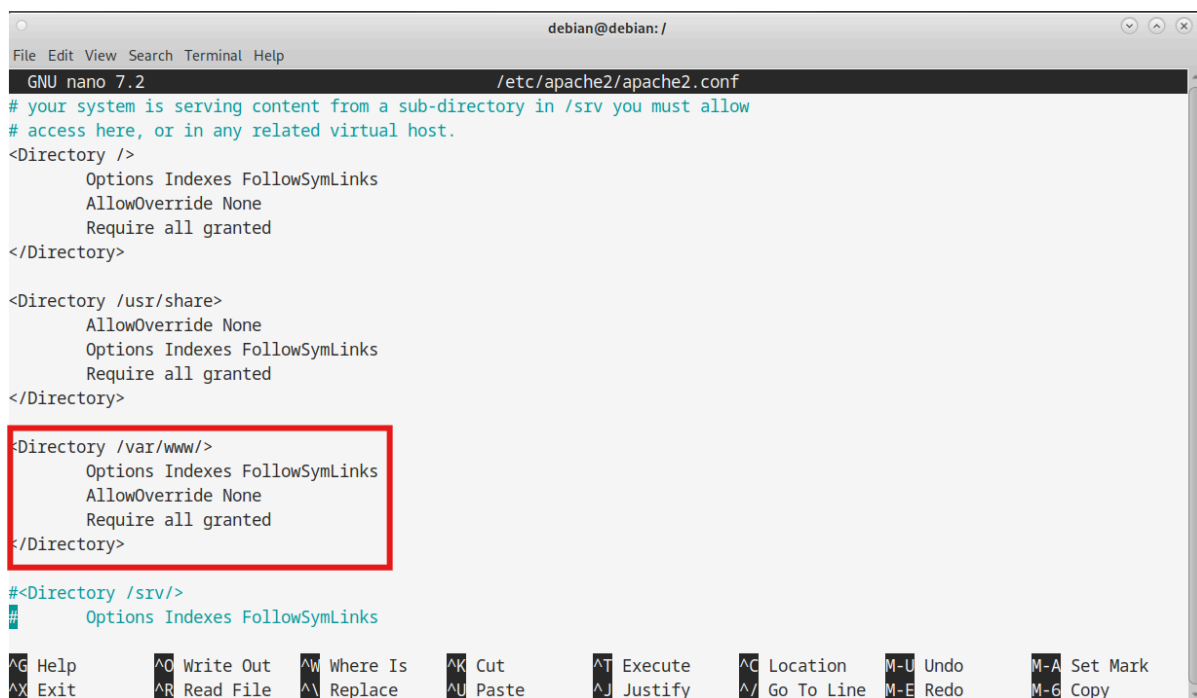
A continuación verificamos que Wordpress está instalado mirando en detalle sus archivos del directorio:

```
debian@debian:/$ ls -l /var/www/html/
total 244
-rwxrwxrwx 1 www-data www-data 10701 Sep 30 10:44 index.html
-rwxrwxrwx 1 www-data www-data  405 Feb  6 2020 index.php
-rwxrwxrwx 1 www-data www-data 19915 Dec 31 2023 license.txt
-rwxrwxrwx 1 www-data www-data  7409 Jun 18 2024 readme.html
-rwxrwxrwx 1 www-data www-data  7387 Feb 13 2024 wp-activate.php
drwxrwxrwx 9 www-data www-data  4096 Sep 10 11:23 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 12:02 wp-config.php
drwxrwxrwx 5 www-data www-data  4096 Oct  8 16:49 wp-content
-rwxrwxrwx 1 www-data www-data  5638 May 30 2023 wp-cron.php
drwxrwxrwx 30 www-data www-data 12288 Sep 10 11:23 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 51238 May 28 2024 wp-login.php
-rwxrwxrwx 1 www-data www-data  8525 Sep 16 2023 wp-mail.php
-rwxrwxrwx 1 www-data www-data 28774 Jul  9 2024 wp-settings.php
-rwxrwxrwx 1 www-data www-data  34385 Jun 19 2023 wp-signup.php
-rwxrwxrwx 1 www-data www-data  4885 Jun 22 2023 wp-trackback.php
-rwxrwxrwx 1 www-data www-data  3246 Mar  2 2024 xmlrpc.php
debian@debian:/$
```

Observamos que en general todos los archivos tienen permisos excesivos, dando lugar a que cualquier usuario los pueda modificar sin respetar el principio del menor privilegio. Esto es una vulnerabilidad considerable, sobre todo en el caso del archivo wp-config.php, que contiene credenciales críticas de la base de datos.

4.6. Directorios web

Es importante que el servidor web tenga una correcta configuración, dado que hemos visto previamente que el apache está activo revisamos su configuración (`sudo nano /etc/apache2/apache2.conf`).



```
debian@debian: /
GNU nano 7.2 /etc/apache2/apache2.conf
# your system is serving content from a sub-directory in /srv you must allow
# access here, or in any related virtual host.
<Directory />
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory /usr/share>
    AllowOverride None
    Options Indexes FollowSymLinks
    Require all granted
</Directory>

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

#<Directory /srv/>
#     Options Indexes FollowSymLinks

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^G Location  M-U Undo     M-A Set Mark
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line M-E Redo     M-6 Copy
```

Podemos observar que el servidor web permite la indexación de directorios, es decir, los directorios se pueden listar y un atacante puede aprovechar esto para explorar los archivos internos sin ninguna restricción.

5. Corrección y mitigación de las vulnerabilidades

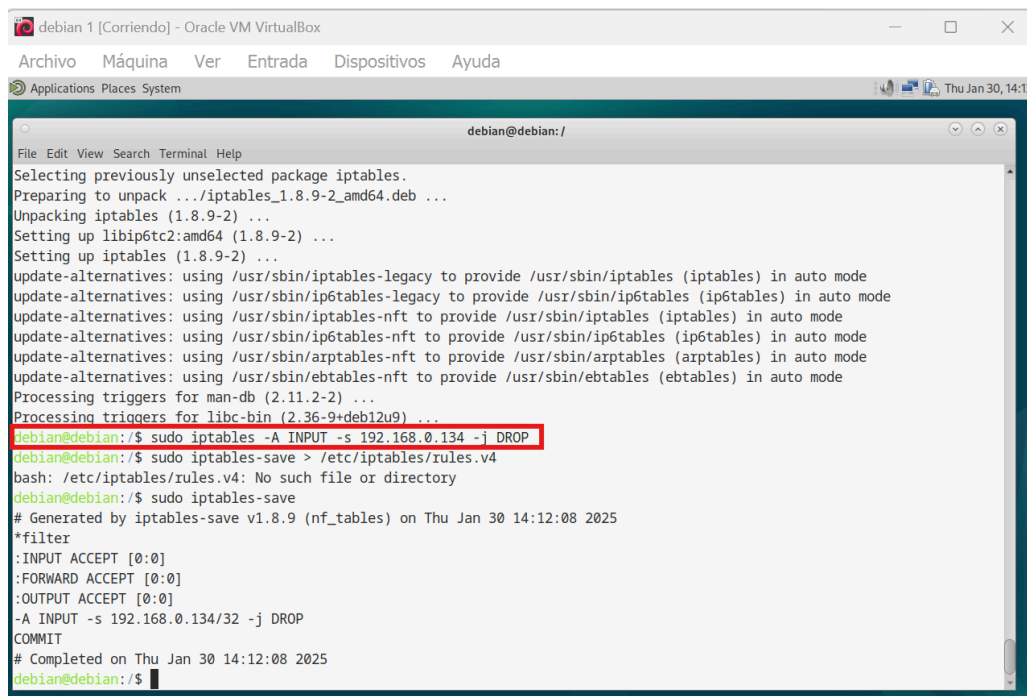
5.1. Acceso SSH al servidor como root

Para corregir esta vulnerabilidad vamos a aplicar diferentes medidas:

- **Bloqueo de la IP sospechosa:** Bloqueamos mediante reglas en el firewall la IP sospechosa 192.168.0.134 para que no se vuelva a recibir más tráfico proveniente de esta IP. Instalamos el firewall "Iptables" y aplicamos los siguientes comandos:

```
sudo iptables -A INPUT -s 192.168.0.134 -j DROP
```

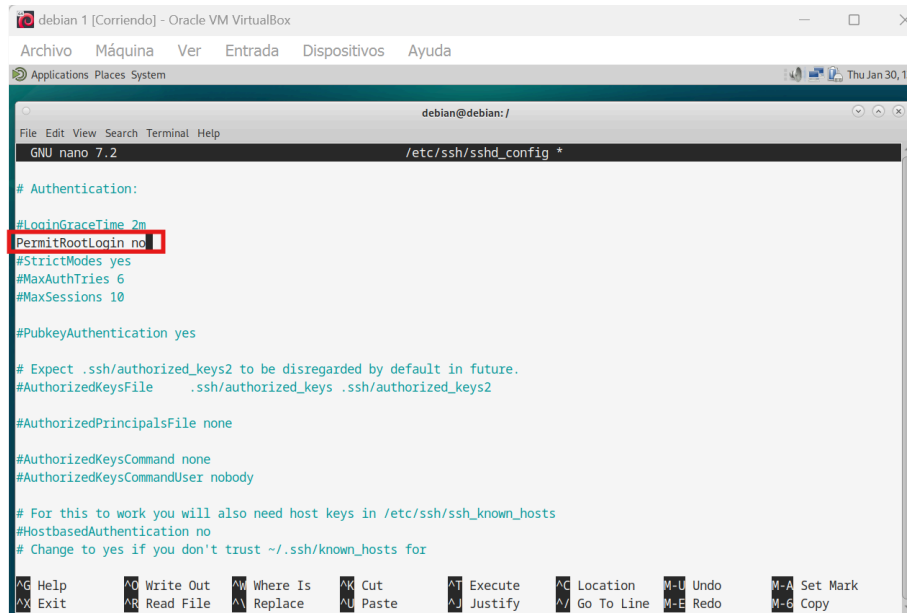
```
sudo iptables-save
```



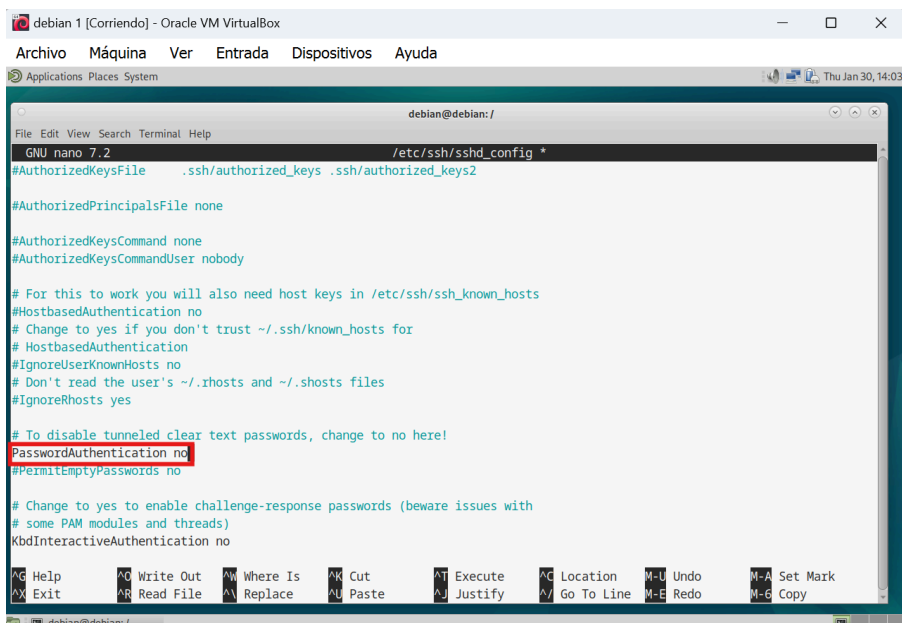
The screenshot shows a terminal window titled "debian 1 [Corriendo] - Oracle VM VirtualBox". The terminal output shows the installation of the iptables package, including unpacking and setting up alternatives. The command `sudo iptables -A INPUT -s 192.168.0.134 -j DROP` is highlighted with a red box. Subsequent commands show an attempt to save the rules to `/etc/iptables/rules.v4` (which fails with "No such file or directory") and then to `/etc/iptables/rules.v4` (which also fails). Finally, the command `sudo iptables-save` is executed, resulting in a new ruleset being generated and committed.

```
File Edit View Search Terminal Help
Selecting previously unselected package iptables.
Preparing to unpack .../iptables_1.8.9-2_amd64.deb ...
Unpacking iptables (1.8.9-2) ...
Setting up libip6tc2:amd64 (1.8.9-2) ...
Setting up iptables (1.8.9-2) ...
update-alternatives: using /usr/sbin/iptables-legacy to provide /usr/sbin/iptables (iptables) in auto mode
update-alternatives: using /usr/sbin/ip6tables-legacy to provide /usr/sbin/ip6tables (ip6tables) in auto mode
update-alternatives: using /usr/sbin/iptables-nft to provide /usr/sbin/iptables (iptables) in auto mode
update-alternatives: using /usr/sbin/ip6tables-nft to provide /usr/sbin/ip6tables (ip6tables) in auto mode
update-alternatives: using /usr/sbin/arptables-nft to provide /usr/sbin/arptables (arptables) in auto mode
update-alternatives: using /usr/sbin/ebtables-nft to provide /usr/sbin/ebtables (ebtables) in auto mode
Processing triggers for man-db (2.11.2-2) ...
Processing triggers for libc-bin (2.36-9+deb12u9) ...
debian@debian:/$ sudo iptables -A INPUT -s 192.168.0.134 -j DROP
debian@debian:/$ sudo iptables-save > /etc/iptables/rules.v4
bash: /etc/iptables/rules.v4: No such file or directory
debian@debian:/$ sudo iptables-save
# Generated by iptables-save v1.8.9 (nf_tables) on Thu Jan 30 14:12:08 2025
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -s 192.168.0.134/32 -j DROP
COMMIT
# Completed on Thu Jan 30 14:12:08 2025
debian@debian:/$
```

- **Deshabilitar acceso con contraseña:** Editamos la configuración del archivo SSH (`sudo nano /etc/ssh/sshd_config`) para deshabilitar el acceso con contraseña y root. De esta forma solo se permite la autenticación con claves SSH para acceder y así se puede evitar el uso de la contraseña del root y los ataques de fuerza bruta.



```
debian 1 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Applications Places System
debian@debian: /
File Edit View Search Terminal Help
GNU nano 7.2 /etc/ssh/sshd_config *
# Authentication:
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
#PubkeyAuthentication yes
# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2
#AuthorizedPrincipalsFile none
#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody
# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# Help Write Out Where Is Cut Execute Location M-U Undo M-A Set Mark
# Exit Read File Replace Paste Justify Go To Line M-E Redo M-G Copy
```



```
debian 1 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Applications Places System
debian@debian: /
File Edit View Search Terminal Help
GNU nano 7.2 /etc/ssh/sshd_config *
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2
#AuthorizedPrincipalsFile none
#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody
# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no
# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
KbdInteractiveAuthentication no
# Help Write Out Where Is Cut Execute Location M-U Undo M-A Set Mark
# Exit Read File Replace Paste Justify Go To Line M-E Redo M-G Copy
```

- **Cambio de contraseña:** Dado que la contraseña del usuario root se ha comprometido, realizamos el cambio de la contraseña “123456” a la nueva contraseña “debian” mediante el comando `sudo passwd root`

5.2. Usuario mysql en la base de datos

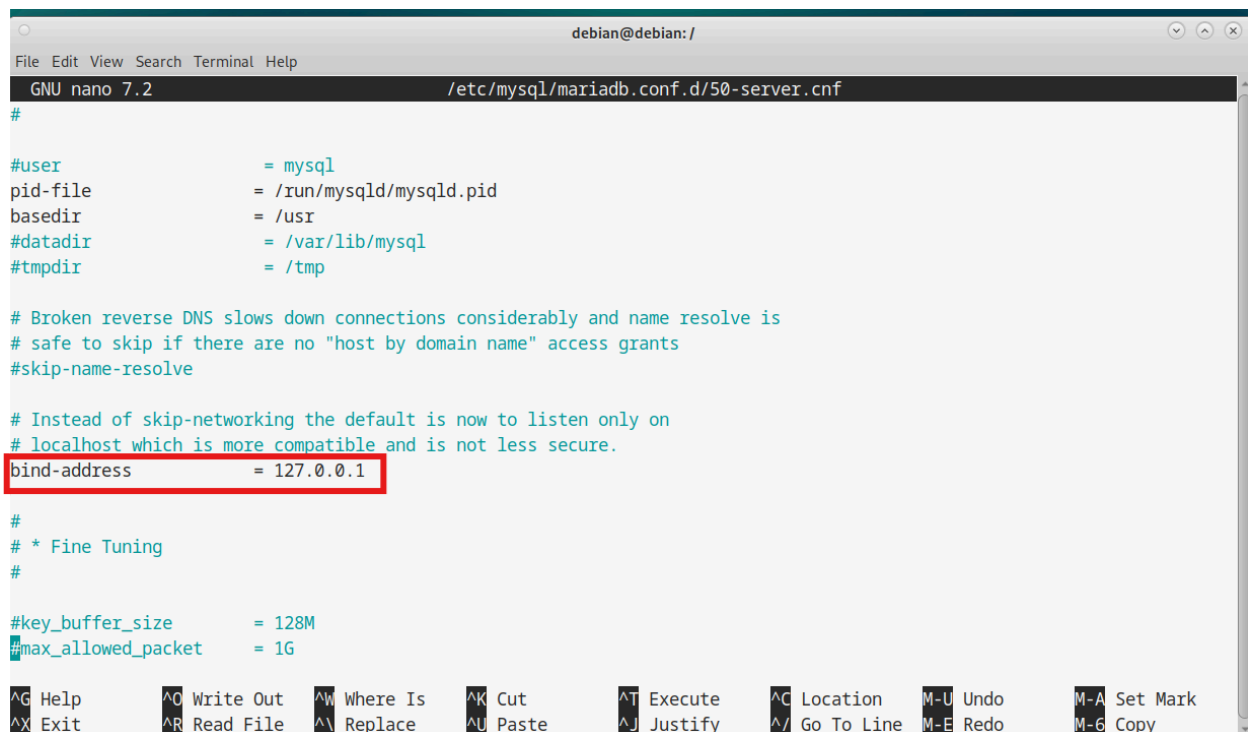
De acuerdo a lo comentado previamente, el usuario “mysql” tiene privilegios elevados y es peligroso. Por tanto se elimina a este usuario:

```
debian@debian:/$ sudo mysql -e "DROP USER 'mysql'@'localhost';"
debian@debian:/$ sudo mysql -e "SELECT user, host, password FROM mysql.user;"
```

User	Host	Password
mariadb.sys	localhost	
root	localhost	*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9
wordpressuser	localhost	*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9
user	localhost	*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19

Por si acaso, también modificamos la configuración para que MySQL/MariaDB no acepte conexiones remotas. Las modificaciones las realizamos en:

```
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```



```
debian@debian: /
GNU nano 7.2 /etc/mysql/mariadb.conf.d/50-server.cnf
#
#user                    = mysql
pid-file                 = /run/mysqld/mysqld.pid
basedir                  = /usr
#datadir                 = /var/lib/mysql
#tmpdir                   = /tmp
# Broken reverse DNS slows down connections considerably and name resolve is
# safe to skip if there are no "host by domain name" access grants
#skip-name-resolve
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address              = 127.0.0.1
#
# * Fine Tuning
#
#key_buffer_size          = 128M
#max_allowed_packet       = 1G
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo     M-A Set Mark
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line M-E Redo     M-G Copy
```

5.3. Actualización

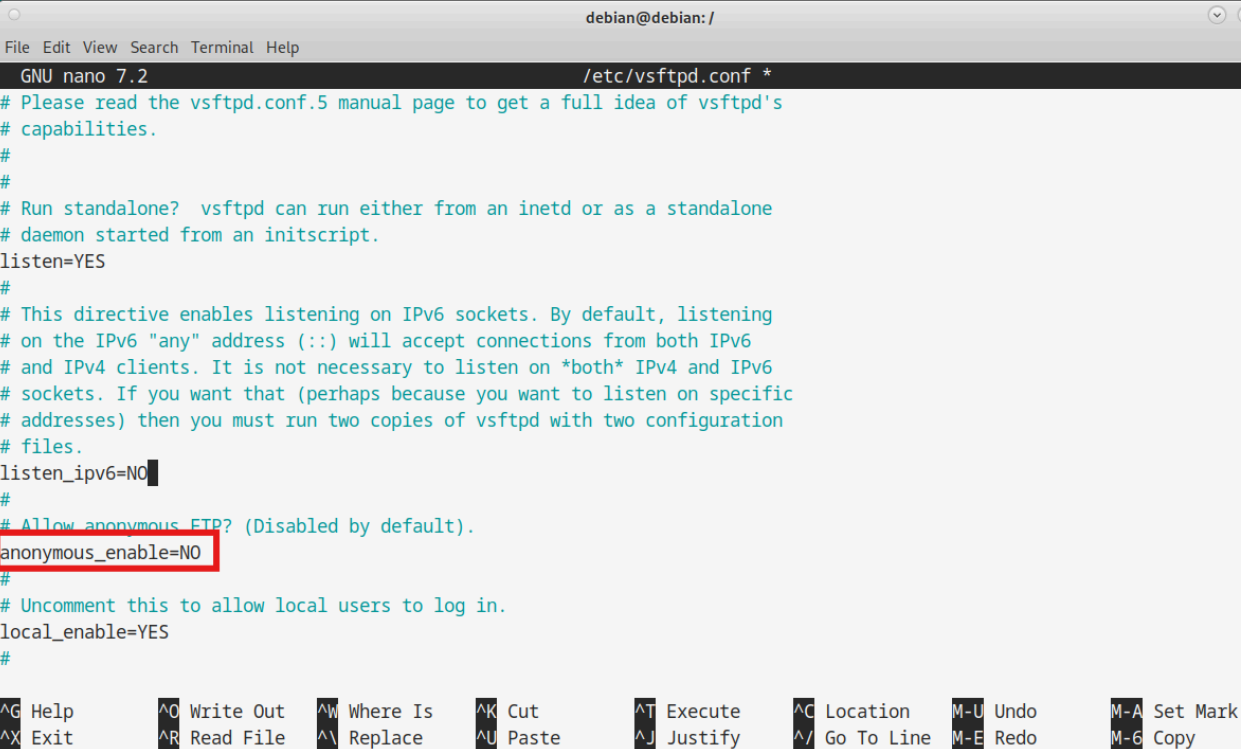
Realizamos actualizaciones de paquetes para que se apliquen parches de seguridad utilizando los comandos:

```
sudo apt update
```

```
sudo apt upgrade -y
```

5.4. Servidor FTP

Cambiamos la configuración del servidor (`sudo nano /etc/vsftpd.conf`) para que no se permita el acceso anónimo sin autenticación y todo pueda tener una trazabilidad o registro (`anonymos_enable=NO`):



```
debian@debian: /
File Edit View Search Terminal Help
GNU nano 7.2 /etc/vsftpd.conf *
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=YES
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=NO
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location  M-U Undo     M-A Set Mark
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^/ Go To Line M-E Redo     M-G Copy
```

También restringimos el acceso solo a usuarios locales (cambiamos `listen=YES` y `listen_ipv6=NO`) para incrementar su seguridad.

5.5. WordPress

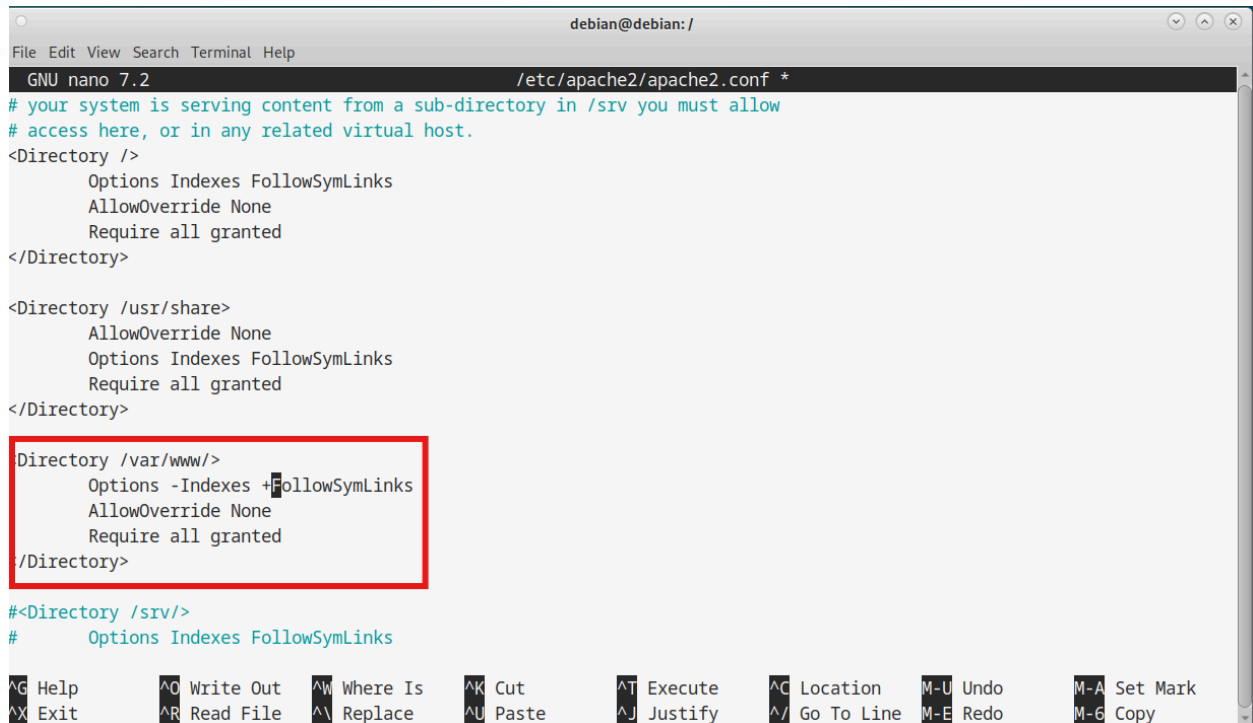
Modificamos los permisos excesivos del archivo wp-config.php, que contiene credenciales críticas de la base de datos. De esta forma se respeta el principio de menor privilegio y se reduce el riesgo de que cualquier usuario pueda modificarlo sin restricciones.

Por tanto, le asignamos los permisos correctos y limitados que le corresponden (`sudo chmod 600 /var/www/html/wp-config.php`) y verificamos el resultado:

```
debian@debian:/$ sudo chmod 600 /var/www/html/wp-config.php
[sudo] password for debian:
debian@debian:/$ ls -l /var/www/html/
total 244
-rwxrwxrwx 1 www-data www-data 10701 Sep 30 10:44 index.html
-rwxrwxrwx 1 www-data www-data 405 Feb 6 2020 index.php
-rwxrwxrwx 1 www-data www-data 19915 Dec 31 2023 license.txt
-rwxrwxrwx 1 www-data www-data 7409 Jun 18 2024 readme.html
-rwxrwxrwx 1 www-data www-data 7387 Feb 13 2024 wp-activate.php
drwxrwxrwx 9 www-data www-data 4096 Sep 10 11:23 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
-rw----- 1 www-data www-data 3017 Sep 30 12:02 wp-config.php
drwxrwxrwx 5 www-data www-data 4096 Oct 8 16:49 wp-content
-rwxrwxrwx 1 www-data www-data 5638 May 30 2023 wp-cron.php
drwxrwxrwx 30 www-data www-data 12288 Sep 10 11:23 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 51238 May 28 2024 wp-login.php
-rwxrwxrwx 1 www-data www-data 8525 Sep 16 2023 wp-mail.php
-rwxrwxrwx 1 www-data www-data 28774 Jul 9 2024 wp-settings.php
-rwxrwxrwx 1 www-data www-data 34385 Jun 19 2023 wp-signup.php
-rwxrwxrwx 1 www-data www-data 4885 Jun 22 2023 wp-trackback.php
-rwxrwxrwx 1 www-data www-data 3246 Mar 2 2024 xmlrpc.php
debian@debian:/$
```

5.6 Directorios web

Modificamos la configuración del servidor apache (`sudo nano /etc/apache2/apache2.conf`) para que los directorios ya no sean listables. Con lo cual ningún atacante podrá visualizar estos archivos internos y confidenciales.



```
debian@debian: /
GNU nano 7.2 /etc/apache2/apache2.conf *
# your system is serving content from a sub-directory in /srv you must allow
# access here, or in any related virtual host.
<Directory />
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory /usr/share>
    AllowOverride None
    Options Indexes FollowSymLinks
    Require all granted
</Directory>

Directory /var/www/>
    Options -Indexes +FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

#<Directory /srv/>
#     Options Indexes FollowSymLinks

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo     M-A Set Mark
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify   ^_ Go To Line M-E Redo     M-G Copy
```

Tras la modificación reiniciamos el servidor para que se apliquen los cambios (`sudo systemctl restart apache2`).

6. Recomendaciones para Prevención Futura

En este informe se ha explicado diferentes vulnerabilidades en 6 servicios diferentes. A pesar que se han aplicado medidas correctivas para mitigar las vulnerabilidades detectadas, recomendamos que la empresa aplique las siguientes medidas a toda su infraestructura tecnológica:

- Implementar el 2FA para accesos remotos y si es posible incluir el PAM para conexiones de proveedores. De esta forma se evitarán conexiones remotas no deseadas.
- Añadir reglas en los firewalls o hacer listas blancas de IP's para limitar accesos.
- Incrementar la monitorización de los servicios críticos o sensibles, mediante sondas o el SIEM. El objetivo es detectar cualquier actividad inusual como la creación de usuarios en la base de datos de MySQL.
- Hacer guías de hardening con las configuraciones básicas e imprescindibles de los servidores. Los técnicos podrán seguirlas y se reducirá el riesgo de malas configuraciones como el acceso al servidor FTP sin autenticación o la posibilidad de listar el directorio del servidor apache.
- Hacer una política donde se enfatice el principio de menor privilegio en todos los sistemas y evitar los permisos excesivos.
- Realizar un seguimiento y actualización constante de todos los servicios y dispositivos para que estén al día respecto a parches de seguridad.
- Realizar auditorías internas de forma periódica para revisar y detectar posibles fallos.
- Aprender de esta situación en que el servidor crítico ha sido comprometido para aplicar las mejoras oportunas y concientizar al personal de la importancia de cumplir con todas las medidas de seguridad.