PROYECTO FINAL

Análisis Forense

Reconocimiento y Recolección de Evidencias

Ramón Tirado Fernández

ÍNDICE

1. INTRODUCCIÓN

Este informe detalla los hallazgos y acciones tomadas tras una intrusión en un servidor Debian 11 que alojaba un sitio WordPress. El análisis reveló múltiples fallos de seguridad que permitieron el acceso no autorizado.

El documento no solo describe las vulnerabilidades explotadas, sino que también presenta un plan de remediación ejecutado y recomendaciones proactivas para evitar futuros incidentes. El objetivo es servir como referencia para fortalecer entornos similares, enfatizando que la mayoría de los vectores de ataque podrían haberse mitigado con configuraciones básicas de seguridad.

2. METODOLOGIA Y ANALISIS

El análisis forense se realizó siguiendo una metodología estructurada para identificar y documentar las vulnerabilidades explotadas en el servidor Debian 11 con WordPress. El proceso incluyó:

- **Recolección de evidencias:** Examinar logs del sistema, procesos en ejecución, configuraciones de servicios y archivos sospechosos.
- **Identificación de compromisos:**
  - Acceso SSH no autorizado como root desde IP 192.168.0.134 mediante contraseña débil.
  - Configuración insegura de servicios (FTP con acceso anónimo, MySQL con contraseñas reutilizadas).
  - Permisos excesivos (777) en archivos de WordPress, incluyendo wp-config.php.
  - Procesos inusuales (orca) y posibles rootkits detectados.
- **Análisis de impacto:** Evaluación de cómo estas vulnerabilidades podrían permitir el control completo del servidor, robo de datos y persistencia de atacantes.

3. HERRAMIENTAS UTILIZADAS

- **Comandos nativos de Linux:** journalctl, ps aux, chmod, chown, crontab -l.
- **Análisis de logs:** Revisión de logs de autenticación SSH y servicios.
- **Escaneo de malwares:**
    - chkrootkit (detectó archivos sospechosos y modo promiscuo en NetworkManager).
    - rkhunter (identificó rootkits potenciales y configuraciones inseguras).
- **Gestión de bases de datos:** Comandos SQL para auditar usuarios y contraseñas en MySQL.
- **Firewall:** iptables para bloquear IPs maliciosas.

4. IDENTIFICACION DE VULNERABILIDADES

4.1. búsqueda inicial de logs:

A) Procedimiento:

- Revisión del listado de archivos, y carpetas de la carpeta log

```
ls  /var/log/
```

B) Hallazgo:

- Ausencia de logs tradicionales, presencia de" /var/log/journal/".
- Descubrimiento de archivo "README" explicando el uso de systemdjournal.

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

```
debian@debian:/$ cd /var/log/
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!
```

4.2. Análisis de Logs del Sistema

A) Procedimiento:

- Revisión de logs de autenticación

```
journalctl _SYSTEMD_UNIT=ssh.service --no-pager
```

B) Hallazgo:

- Acceso exitoso como root desde IP 192.168.0.134, por el puerto 45623
- Autenticación por contraseña habilitada. El método de acceso fue una contraseña (no clave SSH), lo que sugiere que:
  - La contraseña de root era débil o predecible.
  - La configuración de SSH permitía autenticación por contraseña para root.

```
root@debian:/home/debian# journalctl _SYSTEMD_UNIT=ssh.service --no-pager
Sep 30 12:25:16 debian sshd[51422]: Server listening on 0.0.0.0 port 22.
Sep 30 12:25:16 debian sshd[51422]: Server listening on :: port 22.
Sep 30 12:27:50 debian sshd[51422]: Received signal 15; terminating.
-- Boot 46ff5cf6df3d4f0e86b315592aaba2d0 --
Sep 30 15:09:51 debian sshd[560]: Server listening on 0.0.0.0 port 22.
Sep 30 15:09:51 debian sshd[560]: Server listening on :: port 22.
Oct 08 16:14:16 debian sshd[560]: Received signal 15; terminating.
Oct 08 16:14:16 debian sshd[5341]: Server listening on 0.0.0.0 port 22.
Oct 08 16:14:16 debian sshd[5341]: Server listening on :: port 22.
-- Boot 342683d8f35244b08c4f3863f2978eca --
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.
-- Boot d28e179bf5884b25bf94452c79fd0afa --
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.
-- Boot af0a79f76920440c8e08594d6547449b --
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 17:40:59 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 4aecefb286074af397b24901664a9259 --
```

4.3. Análisis de Procesos en Ejecución (ps aux)

A) Procedimiento:

```
ps aux | grep -E "(httpd|apache|mysql|ftp|ssh|python|perl|nc|netcat|bash|sh)"
```

```
root@debian:/home/debian# ps aux | grep -E "(httpd|apache|mysql|ftp|ssh|python|perl|nc|netcat|bash|sh)
"
root           1  0.0  0.5 102372 10384 ?        Ss   06:31   0:01 /sbin/init splash
root           5  0.0  0.0      0     0 ?        I<   06:31   0:00 [slub_flushwq]
root          65  0.0  0.0      0     0 ?        I<   06:31   0:00 [zswap-shrink]
systemd+     289  0.0  0.2  90104  5676 ?        Ssl  06:31   0:00 /lib/systemd/systemd-timesyncd
root         570  0.0  0.1  10196  3384 ?        Ss   06:32   0:00 /usr/sbin/vsftpd /etc/vsftpd.conf
mysql        669  0.0  2.3 1481040 46780 ?       Ssl  06:32   0:02 /usr/sbin/mariadbd
root         671  0.0  0.3  15432  7312 ?        Ss   06:32   0:00 sshd: /usr/sbin/sshd -D [listener]
0 of 10-100 startups
root         694  0.0  0.8 268684 17344 ?        Ss   06:32   0:00 /usr/sbin/apache2 -k start
www-data     713  0.0  1.8 269952 37032 ?        S    06:32   0:00 /usr/sbin/apache2 -k start
www-data     714  0.0  1.7 271976 34740 ?        S    06:32   0:01 /usr/sbin/apache2 -k start
www-data     715  0.0  2.3 272104 47768 ?        S    06:32   0:00 /usr/sbin/apache2 -k start
www-data     716  0.0  4.0 382800 81864 ?        S    06:32   0:06 /usr/sbin/apache2 -k start
www-data     717  0.0  1.6 269944 32256 ?        S    06:32   0:00 /usr/sbin/apache2 -k start
debian      1089  0.0  0.0   7684    44 ?        Ss   06:32   0:00 /usr/bin/ssh-agent x-session-manage
r
debian      1090  0.0  0.2 311140  6020 ?        Ssl  06:32   0:00 /usr/libexec/at-spi-bus-launcher
debian      1096  0.3  0.2  10288  5332 ?        S    06:32   0:39 /usr/bin/dbus-daemon --config-file=
/usr/share/defaults/at-spi2/accessibility.conf --nofork --print-address 11 --address=unix:path=/run/us
er/1000/at-spi/bus_0
debian      1184  0.0  1.6 539880 32612 ?        Sl   06:32   0:08 /usr/lib/mate-panel/wnck-applet
debian      1192  3.7  3.9 412692 79652 ?        Sl   06:32   7:51 /usr/bin/python3 /usr/bin/orca
```

B) Hallazgo:

- **Servicios legítimos:**
  - **SSH:**

```
root         671  0.0  0.3  15432  7312 ?        Ss   06:32   0:00 sshd: /usr/sbin/sshd -D [listener]
0 of 10-100 startups
```

- **Apache (HTTP):**
  - Riesgo: Si el servidor web aloja aplicaciones (ej. WordPress), podría ser el vector de ataque

```
root         694  0.0  0.8 268684 17344 ?        Ss   06:32   0:00 /usr/sbin/apache2 -k start
www-data     713  0.0  1.8 269952 37032 ?        S    06:32   0:00 /usr/sbin/apache2 -k start
www-data     714  0.0  1.7 271976 34740 ?        S    06:32   0:01 /usr/sbin/apache2 -k start
www-data     715  0.0  2.3 272104 47768 ?        S    06:32   0:00 /usr/sbin/apache2 -k start
www-data     716  0.0  4.0 382800 81864 ?        S    06:32   0:06 /usr/sbin/apache2 -k start
www-data     717  0.0  1.6 269944 32256 ?        S    06:32   0:00 /usr/sbin/apache2 -k start
```

- **MySQL:**
  - Riesgo: Credenciales débiles o inyección SQL.

```
mysql        669  0.0  2.3 1481040 46780 ?       Ssl  06:32   0:02 /usr/sbin/mariadbd
```

- **FTP (vsftpd):**
  - Riesgo: Acceso anónimo o permisos inseguros.

```
root         570  0.0  0.1  10196  3384 ?        Ss   06:32   0:00 /usr/sbin/vsftpd /etc/vsftpd.conf
```

- **Procesos sospechosos:**
  - **orca (Python):**
    - ¿Qué es?: Herramienta de accesibilidad para usuarios con discapacidad visual.
    - Riesgo: Inusual en un servidor. Podría ser un script malicioso camuflado.

4.4. Análisis del proceso orca (Python):

A) Procedimiento:

```
cat /usr/bin/orca
```

B) Hallazgo:

- Es un componente legítimo de accesibilidad para discapacitados visuales (normalmente en entornos de escritorio, no en servidores)

4.5. Análisis de Servidor Web (Apache/WordPress)

A) Procedimiento:

- Comprobación de correcto funcionamiento de apache:

```
sudo systemctl status apache2
```

```
root@debian:/home/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 Wed 2025-05-28 06:32:02 EDT; 4h 6min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 563 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 694 (apache2)
      Tasks: 10 (limit: 2284)
     Memory: 166.7M
        CPU: 9.971s
     CGroup: /system.slice/apache2.service
             ├─ 694 /usr/sbin/apache2 -k start
             ├─ 713 /usr/sbin/apache2 -k start
             ├─ 714 /usr/sbin/apache2 -k start
             ├─ 715 /usr/sbin/apache2 -k start
             ├─ 716 /usr/sbin/apache2 -k start
             ├─ 717 /usr/sbin/apache2 -k start
             ├─3233 /usr/sbin/apache2 -k start
             ├─3270 /usr/sbin/apache2 -k start
             ├─3278 /usr/sbin/apache2 -k start
             └─3279 /usr/sbin/apache2 -k start
```

- Revisión de archivos subidos recientemente en /var/www/html

```
ps aux | grep -E "(httpd|apache|mysql|ftp|ssh|python|perl|nc|netcat|bash|sh)"
```

```
root@debian:/home/debian# ls -la /var/www/html/
total 260
drwxrwxrwx  5 www-data www-data  4096 May 28 07:10 .
drwxr-xr-x  3 root     root      4096 Sep 30  2024 ..
-rwxrwxrwx  1 www-data www-data   523 Sep 30  2024 .htaccess
-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 May 28 07:09 license.txt
-rwxrwxrwx  1 www-data www-data  7425 May 28 07:09 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 May 28 07:09 wp-config-sample.php
drwxrwxrwx  6 www-data www-data  4096 May 28 07:10 wp-content
-rwxrwxrwx  1 www-data www-data  5617 May 28 07:09 wp-cron.php
drwxrwxrwx 30 www-data www-data 12288 May 28 07:09 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 May 28 07:09 wp-login.php
-rwxrwxrwx  1 www-data www-data  8727 May 28 07:09 wp-mail.php
-rwxrwxrwx  1 www-data www-data 30081 May 28 07:09 wp-settings.php
-rwxrwxrwx  1 www-data www-data 34516 May 28 07:09 wp-signup.php
-rwxrwxrwx  1 www-data www-data  5102 May 28 07:09 wp-trackback.php
-rwxrwxrwx  1 www-data www-data  3205 May 28 07:09 xmlrpc.php
```

B) Hallazgo:

- El servicio de Apache esta activo, y WordPress funciona correctamente
- Directorio /var/www/html:
  - Contiene archivos típicos de WordPress (wp-admin, wp-includes, wp-config.php, etc.).
  - Permisos inseguros: Todos los archivos tienen permisos 777 (rwxrwxrwx), lo que permite a cualquier usuario modificarlos.
- Archivo wp-config.php:
  - Contiene credenciales de la base de datos. Debe tener permisos 600 (solo lectura/escritura para el dueño):

4.6. Análisis de MySQL (mariadb)

A) Procedimiento:

- Verificar usuarios y contraseñas débiles:

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

```
root@debian:/home/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 |
+---------------+-----------+-------------------------------------------+
```

B) Hallazgo:

- Usuarios con Contraseñas Débiles/Reutilizadas:
  - root y wordpressuser comparten el mismo hash de contraseña (*6BB4837EB74329105...), indicando:
    - Reutilización de contraseñas entre cuentas privilegiadas
    - Posible contraseña común vulnerable (ej: "123456", "password")
  - User tiene una contraseña diferente, pero sin política de complejidad verificable.
- Cuentas con Acceso sin Contraseña
  - Riesgo: Ataques de escalamiento de privilegios locales
- Exposición de Hashes
  La consulta muestra hashes en texto plano (aunque cifrados), lo que permite ataques de fuerza bruta.

4.7. Análisis de servidor FTP (vsftpd)

A) Procedimiento

- Revisar configuración:

```
cat /etc/vsftpd.conf | grep -E "anonymous_enable|local_enable|write_enable"
```

```
root@debian:/home/debian# cat /etc/vsftpd.conf | grep -E "anonymous_enable|local_enable|write_enable"
anonymous_enable=YES
local_enable=YES
write_enable=YES
#anon_mkdir_write_enable=YES
```

B) Hallazgo

- Configuraciones actuales:
  - anonymous_enable=YES
  - local_enable=YES
  - write_enable=YES

- Riesgos:
  - Acceso anónimo: Cualquiera puede conectarse sin credenciales.
  - Escritura habilitada: Usuarios locales (o atacantes que roben credenciales) pueden modificar archivos.

4.8. Escaneo de Malwares con chkrootkit

A) Procedimiento

- Escaneo con chkrootkit

```
sudo chkrootkit
```

B) Hallazgo

- Archivos sospechosos identificados:

```
WARNING: The following suspicious files and directories were found:
/usr/lib/ruby/gems/3.1.0/gems/typeprof-0.21.2/vscode/.vscodeignore
/usr/lib/ruby/gems/3.1.0/gems/typeprof-0.21.2/vscode/.gitignore
/usr/lib/ruby/gems/3.1.0/gems/typeprof-0.21.2/vscode/.vscode
/usr/lib/ruby/vendor_ruby/rubygems/ssl_certs/.document
/usr/lib/ruby/vendor_ruby/rubygems/tsort/.document
/usr/lib/ruby/vendor_ruby/rubygems/optparse/.document
/usr/lib/libreoffice/share/.registry
```

  - Directorios Ruby con configuraciones ocultas.

```
/usr/lib/ruby/gems/3.1.0/gems/typeprof-0.21.2/vscode/.vscodeignore
/usr/lib/ruby/gems/3.1.0/gems/typeprof-0.21.2/vscode/.gitignore
```

    TypeProf es una herramienta legítima de análisis estático para Ruby, pero:
    - Estos archivos ocultos (.gitignore, .vscodeignore) no son típicos en instalaciones globales de gems
    - Podrían indicar:
      - ❖ Modificación no autorizada del paquete
      - ❖ Intentos de ocultar archivos maliciosos

  - Archivos de documentación inusuales:

```
/usr/lib/ruby/vendor_ruby/rubygems/ssl_certs/.document
```

    El archivo “.document” no es estándar en este directorio

Posibles escenarios:
- Artefacto de instalación legítimo pero inusual
- Intento de documentar actividades maliciosas
- Pista dejada por atacante (OpSec fallido)

- Sniffer detectado:
  - NetworkManager ejecutándose en modo promiscuo:

```
WARNING: Output from ifpromisc:
lo: not promisc and no packet sniffer sockets
enp0s3: PACKET SNIFFER(/usr/sbin/NetworkManager[46083])
```

Modo promiscuo permite capturar todo el tráfico de red

Comportamiento inusual para NetworkManager. Pondría indicar:
- Configuración legítima para diagnóstico
- Compromiso del servicio
- Interceptación activa de tráfico

4.9. Escaneo de Malwares con rkhunter:

A) Procedimiento

- Escaneo con rkhunter:

```
sudo rkhunter --check
```

B) Hallazgo

- Configuración Insegura de SSH (Alerta Crítica):
  - Evidencia en logs:

```
Performing system configuration file checks
  Checking for an SSH configuration file                  [ Found ]
  Checking if SSH root access is allowed                  [ Warning ]
  Checking if SSH protocol v1 is allowed                  [ Not set ]
  Checking for other suspicious configuration settings    [ None found ]
  Checking for a running system logging daemon            [ Found ]
  Checking for a system logging configuration file        [ Found ]
```

- Segmentos de Memoria Compartida Sospechosos:
  - Evidencia en logs:

```
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 ]
```

- Archivo /usr/bin/lwp-request Marcado como Sospechoso
  - Evidencia en logs:

```
/usr/bin/lwp-request                                       [ Warning ]
```

- Rootkits Detectados (7 Posibles)
  - Evidencia en logs:

```
Rootkit checks...
    Rootkits checked : 497
    Possible rootkits: 7
```

4.10. Análisis de tareas programadas

A) Procedimiento

- Ver los crontabs del sistema mediante “crontab -l”

B) Hallazgo

- No se han encontrado tareas programadas.

```
root@debian:/home/debian# crontab -l
no crontab for root
```

5. PLAN DE RESPUESTA A INCIDENTES Y CORRECCION DE VULNERABILIDADES

5.1. Actualizar sistema

A) Incidencia: El sistema esta desactualizado, lo que lo expone a vulnerabilidades conocidas.

B) Acciones:

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

5.2. Acceso no autorizado como root

A) Incidencia: Autenticación exitosa como root desde la IP 192.168.0.134, mediante contraseña débil.

B) Acciones:

- Cambiar contraseña de root.
  Procedimiento:
  - Ejecutando el comando “passwd root”. Se utilizará una contraseña compleja (mínimo 12 caracteres, mezcla de mayúsculas, números y símbolos).
- Deshabilitar el login SSH como root
  Procedimiento:
  - Ejecutando el comando “nano /etc/ssh/sshd_config”
  - Cambiar línea “PermitRootLogin yes” por “PermitRootLogin no”

```
  GNU nano 7.2                    /etc/ssh/sshd_config
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

```
  GNU nano 7.2                    /etc/ssh/sshd_config *
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes
```

- Guardamos los cambios, y reiniciamos servicio con el comando "systemctl restart ssh"

- Bloquear la IP del atacante
  Procedimiento:
  - Instalamos el firewall "iptables", mediante el comando "apt install iptables -y"
  - Se agrega una regla al firewall para bloquear la IP 192.168.0.134, con el comando "sudo iptables -A INPUT -s 192.168.0.134 -j DROP"
  - Se guardan las reglas permanentemente (requiere instalar un paquete adicional). Usaremos los comandos:
    - "Sudo apt install iptables-persistent -y"
    - "Sudo netfilter-persistent save"

```
root@debian:/home/debian# sudo netfilter-persistent save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables save
```

- Se verifica que la IP está bloqueada, mediante el comando "sudo iptables -L -n -v | grep 192.168.0.134".

```
root@debian:/home/debian# sudo iptables -L -n -v | grep 192.138.0.134
    0     0 DROP       0    --  *      *       192.138.0.134        0.0.0.0/0
```

5.3. Eliminación de proceso orca (Python)

A) Incidencia: No es necesario en servidores.

B) Acciones:

- Desinstalación del proceso orca mediante el comando “apt remove orca -y”

5.4. Modificación de permisos del servidor web (Apache/WordPress)

A) Incidencia:

- Todos los archivos tienen permisos 777 (rwxrwxrwx), lo que permite a cualquier usuario modificarlos.
- El archivo wp-config.php contiene credenciales de la base de datos. Debe tener permisos 600 (lectura/escritura para el dueño)

B) Acciones:

- Restringir permisos (ejecución sólo para dueño/grupo):

```
chmod -R 750 /var/www/html
```

- Aseguramos propiedad correcta:

```
chown -R www-data:www-data /var/www/html
```

5.5. Protección de cuentas de usuarios en MySQL

A) Incidencia:

- Usuarios con Contraseñas Débiles/Reutilizadas:
- Cuentas con Acceso sin Contraseña
- Exposición de Hashes

B) Acciones:

- Accedemos al cliente de MySQL/MariaDB:

```
mysql -u root -p
```

- Rotación de credenciales:
  - Cambiamos la contraseña de root:

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'nuevacontraseña;
```

- Cambiamos la contraseña de wordpressuser:

```
ALTER USER 'wordpressuser'@'localhost' IDENTIFIED BY 'nuevacontraseña';
```

- Se elimina usuario genérico (innecesario):

```
DROP USER 'user'@'localhost';
```

```
root@debian:/home/debian# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 32
Server version: 10.11.11-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> ALTER USER 'root'@'localhost' IDENTIFIED BY 'contraseñamuysegura';
Query OK, 0 rows affected (0.126 sec)

MariaDB [(none)]> ALTER USER 'wordpressuser'@'localhost' IDENTIFIED BY 'contraseñamuysegura';
Query OK, 0 rows affected (0.016 sec)

MariaDB [(none)]> DROP USER 'user'@'localhost'
```

- Aseguramos cuentas del sistema:
  - Se deshabilita cuenta “mysql” (uso interno):

```
ALTER USER 'mysql'@'localhost' ACCOUNT LOCK;
```

- Creamos script de monitoreo diario:
  - Se crea el documento monitoreo.sh

```
Sudo nano /usr/local/bin/mysql_audit.sh
```

- Se utilizará el siguiente script de monitoreo diario:

```
#!/bin/bash
mysql -e "SELECT user, host, password_expired, account_locked FROM mysql.user" > /var/log/mysql_users.log
```

```
 GNU nano 7.2                          /usr/local/bin/mysql_audit.sh
#!/bin/bash
mysql -e "SELECT user, host, password_expired, account_locked FROM mysql.user" > /var/log/mysql_users.log
```

5.6. Configuraciones inseguras en FTP

A) Incidencia

- Acceso anónimo
- Escritura habilitada

B) Acciones

- Deshabilitar acceso anónimo:
  Editamos "/etc/vsftpd.conf":
  - Cambiamos "anonymous_enable=NO"
  - Restringimos escritura "write_enable=NO"
  - Reiniciamos servicio con "systemctl restart vsftpd"

```
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=NO
```

- Actualizar WordPress, y Plugins, ya que las versiones antiguas pueden tener vulnerabilidades explotadas.
  - Comando para ver versión actual:

```
cat /var/www/html/wp-includes/version.php | grep '$wp_version'
```

6. RECOMENDACIONES FINALES

- **Proteger el acceso**
  - Activar la autenticación en dos pasos para SSH y el panel de WordPress.
  - Limitar el acceso remoto solo a direcciones IP fiables.
- **Vigilar el servidor**
  - Instalar herramientas como Fail2Ban para bloquear intentos de acceso sospechosos.
  - Configurar alertas por correo cuando ocurran incidencias importantes (ej: logins fuera de horario).
- **Mantener todo actualizado**
  - Programar actualizaciones automáticas para el sistema, WordPress y sus plugins.
  - Realiza escaneos mensuales para detectar vulnerabilidades.
- **Copiar de seguridad fuera del servidor**
  - Guardar copias de seguridad en la nube o en otro equipo, y comprobar que se puedan restaurar.
- **Reducir riesgos innecesarios**
  - Eliminar servicios que no uses (como FTP si no es esencial).
  - Usar contraseñas únicas y complejas para cada cuenta importante.
- **Preparar un plan de acción**
  - Tener documentado qué hacer si ocurre un ataque: cómo aislar el servidor y recuperar datos.