Informe Técnico de Respuesta a Incidente

Título: Análisis y mitigación de ataque en servidor Linux (debian) con WordPress

Fecha de elaboración: 9 de junio de 2025

Responsable del análisis: Alan Alvarado Ramírez

1. Resumen del incidente

Durante la revisión de los registros del sistema y servicios críticos, se detectó actividad anómala atribuida a un usuario sin privilegios root que intentó ejecutar comandos con sudo, alterando y eliminando procesos y archivos. También se detectaron configuraciones débiles en WordPress, FTP y SSH, así como usuarios MySQL con contraseñas vulnerables.

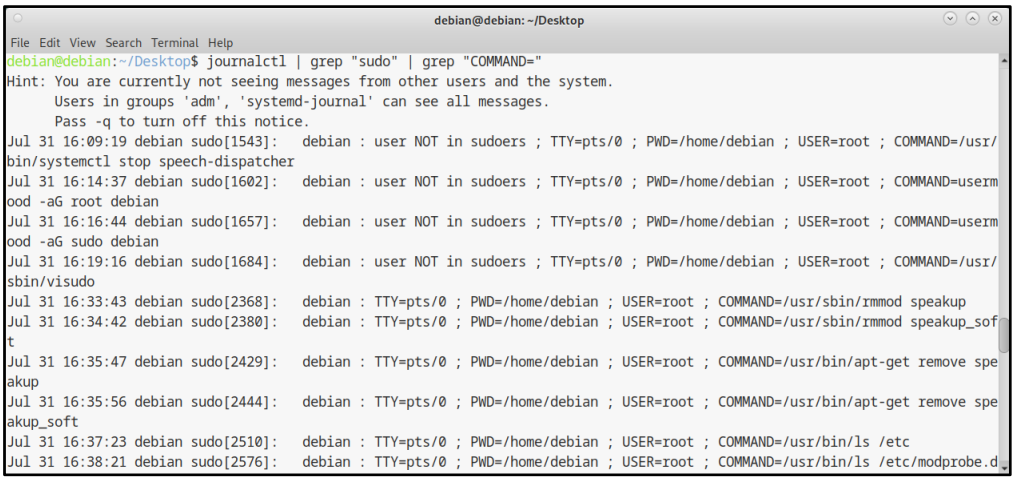
2. Análisis técnico

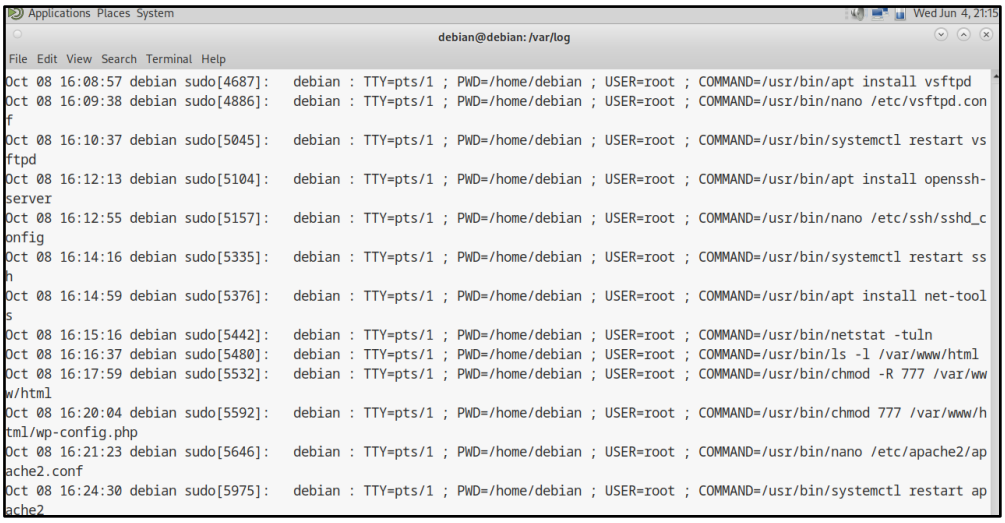
1. Detección de comandos sospechosos con SUDO

- Se ejecutó:

  journalctl | grep "sudo" | grep "COMMAND="

- Resultado: se detectaron intentos de escalación de privilegios y eliminación de procesos por un usuario sin permisos.





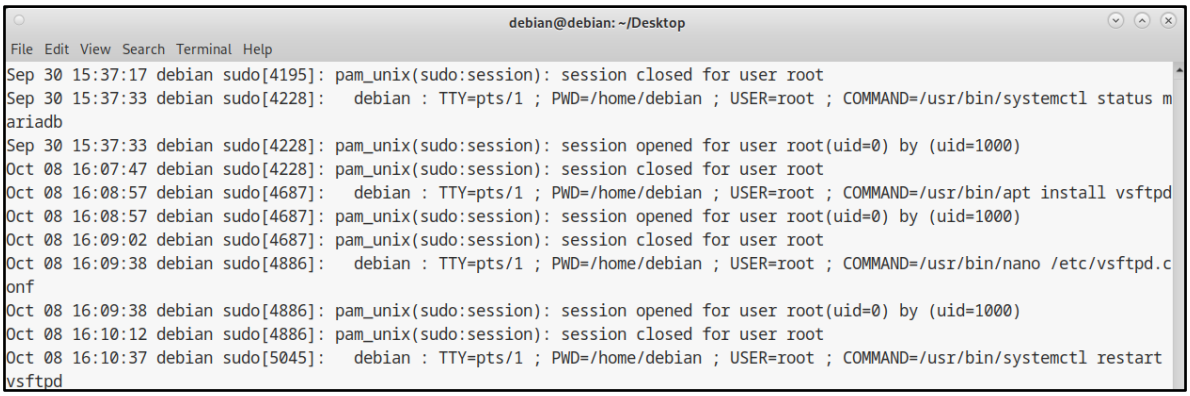

```
Sep 30 11:57:49 debian sudo[38553]:   debian : TTY=pts/0 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/chown -R www-dat
a:www-data /var/www/html/
Sep 30 11:58:23 debian sudo[38598]:   debian : TTY=pts/0 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/chmod -R 755 /va
r/www/html/
Sep 30 11:59:38 debian sudo[38666]:   debian : TTY=pts/0 ; PWD=/var/www/html ; USER=root ; COMMAND=/usr/bin/mv wp-c
onfig-sample.php wp-config.php
Sep 30 12:00:08 debian sudo[38693]:   debian : TTY=pts/0 ; PWD=/var/www/html ; USER=root ; COMMAND=/usr/bin/nano wp
-config.php
Sep 30 12:05:46 debian sudo[39731]:   debian : TTY=pts/0 ; PWD=/var/www/html ; USER=root ; COMMAND=/usr/bin/systemc
tl restart apache2
Sep 30 12:06:00 debian sudo[39831]:   debian : TTY=pts/0 ; PWD=/var/www/html ; USER=root ; COMMAND=/usr/bin/systemc
tl status apache2
Sep 30 12:11:37 debian sudo[40186]:   debian : TTY=pts/0 ; PWD=/var/www/html ; USER=root ; COMMAND=/usr/bin/apt ins
tall php libapache2-mod-php php-mysqli php-gd php-xml php-mbstring php-curl -y
Sep 30 12:15:25 debian sudo[50261]:   debian : TTY=pts/0 ; PWD=/ ; USER=root ; COMMAND=/usr/bin/nano /etc/apache2/s
ites-available/000-default.conf
Sep 30 12:18:19 debian sudo[50495]:   debian : TTY=pts/0 ; PWD=/ ; USER=root ; COMMAND=/usr/bin/nano /etc/apache2/s
ites-available/000-default.conf
Sep 30 12:19:00 debian sudo[50603]:   debian : TTY=pts/0 ; PWD=/ ; USER=root ; COMMAND=/usr/bin/systemctl restart a
pache2
Sep 30 12:19:23 debian sudo[50659]:   debian : TTY=pts/0 ; PWD=/ ; USER=root ; COMMAND=/usr/bin/nano /var/www/html/
info.php
```

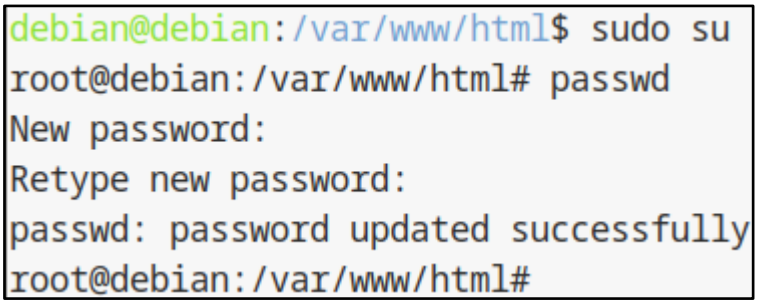
2. Revisión de inicios de sesión con SUDO

- Comandos:

  journalctl | grep "sudo"
  sudo passwd root

- Resultado: confirmación de intentos no autorizados y se cambia al contraseña de root por una más fuerte.



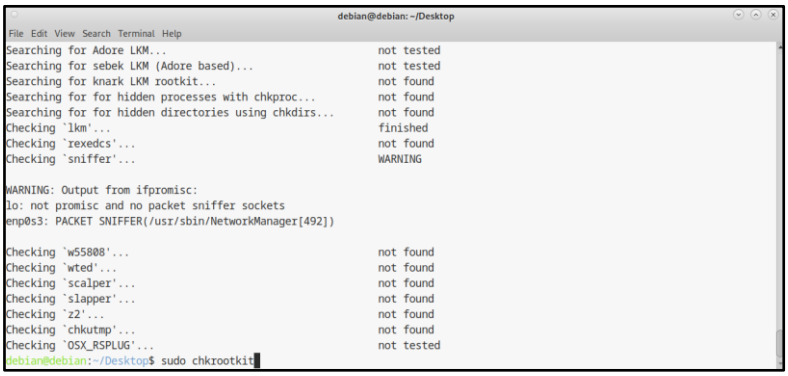


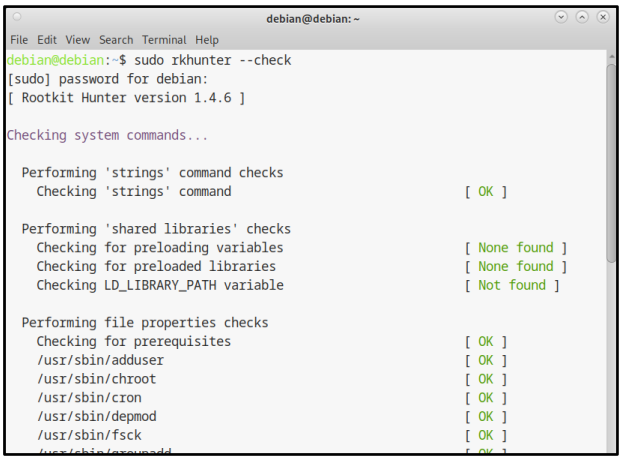
3. Búsqueda de rootkits

- Herramientas y comandos utilizados:

  chkrootkit
  rkhunter --check

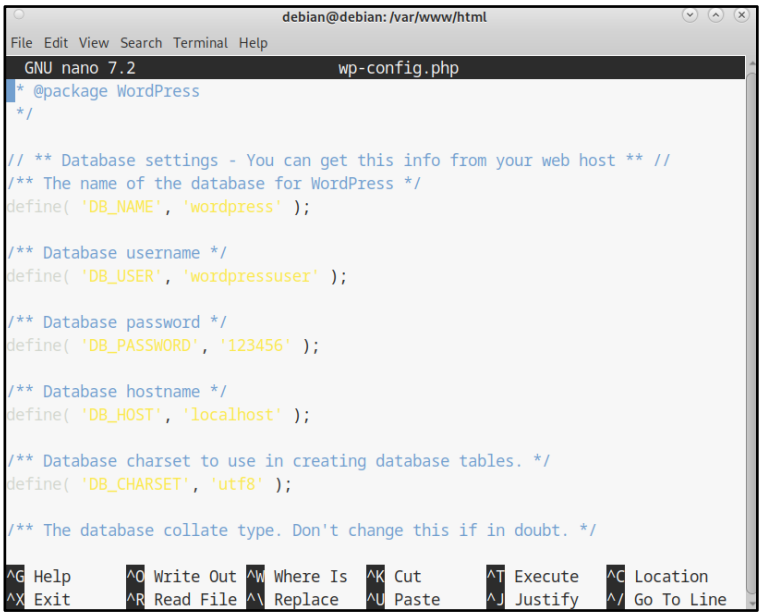
- Resultado: no se encontraron rootkits activos.





4. Análisis de WordPress (wp-config.php)

- Se detectó una contraseña débil del usuario wordpressuser.
- Acción tomada:
  - Cambio del usuario y contraseña en wp-config.php.
  - Refuerzo de la contraseña con criterios seguros.



```
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'wpusernew' );

/** Database password */
define( 'DB_PASSWORD', 'Jh7yt5GG13' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );
```

5. Revisión de privilegios en MySQL

- Comando:

  SHOW GRANTS FOR 'wordpressuser'@'localhost';

- Resultado: el usuario tenía ALL PRIVILEGES sobre la base de datos.
- Acción: el usuario fue eliminado.

```
MariaDB [(none)]> SELECT User, Host, plugin FROM mysql.user;
+---------------+-----------+-----------------------+
| User          | Host      | plugin                |
+---------------+-----------+-----------------------+
| mariadb.sys   | localhost | mysql_native_password |
| root          | localhost | mysql_native_password |
| mysql         | localhost | mysql_native_password |
| wordpressuser | localhost | mysql_native_password |
| user          | localhost | mysql_native_password |
+---------------+-----------+-----------------------+
5 rows in set (0.101 sec)

MariaDB [(none)]> SHOW GRANTS FOR 'wordpress'@'localhost';
ERROR 1141 (42000): There is no such grant defined for user 'wordpress' on host 'localhost'
MariaDB [(none)]> SHOW GRANTS FOR 'wordpressuser'@'localhost';
+-----------------------------------------------------------------------------------------------------------------------+
| Grants for wordpressuser@localhost                                                                                    |
+-----------------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO `wordpressuser`@`localhost` IDENTIFIED BY PASSWORD '*6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9' |
| GRANT ALL PRIVILEGES ON `wordpress`.* TO `wordpressuser`@`localhost`                                                  |
+-----------------------------------------------------------------------------------------------------------------------+
2 rows in set (0.003 sec)

MariaDB [(none)]>
```

```
MariaDB [(none)]> SELECT User , Host FROM mysql.user;
+---------------+-----------+
| User          | Host      |
+---------------+-----------+
| mariadb.sys   | localhost |
| mysql         | localhost |
| root          | localhost |
| user          | localhost |
| wordpressuser | localhost |
+---------------+-----------+
5 rows in set (0.026 sec)

MariaDB [(none)]> DROP USER 'wordpressuser'@'localhost';
Query OK, 0 rows affected (0.079 sec)

MariaDB [(none)]> SELECT User, Host FROM mysql.user WHERE User = 'wordpressuser';
Empty set (0.006 sec)
```

6. Auditoría de hashes de contraseñas MySQL

- Comando:

  SELECT User, Host, plugin, authentication_string FROM mysql.user;

- Herramienta de auditoría: hashes.com (ya que mysql usa doble cifrado y john the ripper no puede descifrarlas).
- Resultado: se descifraron varias contraseñas, confirmando que eran débiles.
- Acción: recomendación de fortalecer políticas de contraseñas y rotarlas periódicamente.

```
MariaDB [(none)]> SELECT User, Host, plugin, authentication_string FROM mysql.user;
+-------------+-----------+-----------------------+-------------------------------------------+
| User        | Host      | plugin                | authentication_string                     |
+-------------+-----------+-----------------------+-------------------------------------------+
| mariadb.sys | localhost | mysql_native_password |                                           |
| root        | localhost | mysql_native_password | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| mysql       | localhost | mysql_native_password | invalid                                   |
| user        | localhost | mysql_native_password | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 |
+-------------+-----------+-----------------------+-------------------------------------------+
4 rows in set (0.019 sec)
```

7. Revisión de configuración FTP

- Archivo revisado:

/etc/vsftpd.conf

- Problema: acceso anónimo habilitado.
- Acción: acceso anónimo deshabilitado.

```
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
```

```
# 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=YES
```

8. Revisión de configuración SSH

- Archivo revisado:

  /etc/ssh/sshd_config

- Problemas:
  - Autenticación por contraseña habilitada.
  - Login como root habilitado.
- Acción: ambas opciones desactivadas y se migró a autenticación con llave SSH.

```
# Authentication:

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

#PubkeyAuthentication yes
```

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
```

```
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
```

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no
```

9. Actualización del sistema

- Comandos ejecutados:

  sudo apt update
  sudo apt upgrade -y
  sudo apt full-upgrade

- Resultado: sistema actualizado con los últimos parches de seguridad.

```
Removing linux-image-6.1.0-23-amd64 (6.1.99-1) ...
/etc/kernel/postrm.d/initramfs-tools:
update-initramfs: Deleting /boot/initrd.img-6.1.0-23-amd64
/etc/kernel/postrm.d/zz-update-grub:
Generating grub configuration file ...
Found background image: /usr/share/images/desktop-base/desktop-grub.png
Found linux image: /boot/vmlinuz-6.1.0-37-amd64
Found initrd image: /boot/initrd.img-6.1.0-37-amd64
Found linux image: /boot/vmlinuz-6.1.0-25-amd64
Found initrd image: /boot/initrd.img-6.1.0-25-amd64
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
debian@debian:/var/www/html$
```

3. Medidas tomadas para mitigar el ataque

- Cierre del acceso SUDO al usuario sospechoso mediante un cambio de contraseña.
- Eliminación del usuario wordpressuser con permisos elevados.

- Reforzamiento de contraseñas en WordPress y MySQL.
- Desactivación del acceso anónimo por FTP y endurecimiento de su configuración.
- Desactivación del login por contraseña y del acceso root por SSH.
- Actualización completa del sistema.

4. Recomendaciones para prevenir futuros ataques

1. Implementar políticas de contraseñas fuertes a nivel de sistema y MySQL, que incluyan:
    - Longitud mínima de 12 caracteres.
    - Uso de mayúsculas, minúsculas, números y símbolos.
    - Cambios obligatorios de forma periódica.

2. Deshabilitar acceso SSH por contraseña y usar exclusivamente autenticación por llave pública.

3. Eliminar usuarios innecesarios en el sistema y en la base de datos.

4. Activar el plugin “validate_password” en MySQL para forzar contraseñas seguras.

5. Auditar regularmente archivos sensibles como wp-config.php, /etc/passwd, /etc/sudoers, etc.

6. Instalar y configurar fail2ban para bloquear intentos de fuerza bruta en SSH, FTP y WordPress.

7. Establecer respaldos automáticos y almacenarlos fuera del servidor.

8. Realizar escaneos periódicos de rootkits y malware con rkhunter y chkrootkit.

9. Revisar periódicamente los servicios en ejecución y cerrar puertos no utilizados.

10. Se debe realizar el control y monitoreo de usuarios privilegiados.