Informe de Análisis Forense

Identificación y Recolección de Evidencias

Resumen Ejecutivo

Este análisis forense ha sido realizado en un servidor Debian comprometido como parte del proyecto final de ciberseguridad para la academia 4Geeks. La investigación ha identificado seis vulnerabilidades críticas que fueron explotadas para obtener acceso no autorizado al sistema y potencialmente exfiltrar datos sensibles. Estas vulnerabilidades incluyen configuraciones inseguras de SSH, FTP y Apache, permisos incorrectos en archivos de configuración de WordPress, y credenciales de base de datos potencialmente comprometidas.

Introducción

El análisis forense en ciberseguridad constituye un pilar fundamental en la respuesta a incidentes, permitiendo comprender las vulnerabilidades explotadas, el alcance de la intrusión y establecer las medidas correctivas necesarias. En esta primera fase del proyecto, hemos aplicado metodologías sistemáticas para examinar un servidor Debian comprometido, con el objetivo de identificar las vulnerabilidades que posibilitaron el acceso no autorizado.

El análisis forense digital se compone de varias etapas críticas. La fase inicial, que presentamos en este informe, consiste en la identificación y recolección de evidencias. Esto implica examinar los logs del sistema, revisar las configuraciones de servicios críticos, analizar los procesos en ejecución, verificar permisos de archivos sensibles e identificar puertos abiertos y servicios expuestos.

Esta primera fase es crucial pues establece las bases para las subsiguientes etapas del proyecto: la corrección de vulnerabilidades adicionales y el desarrollo de un plan de respuesta a incidentes que garantice la continuidad operativa y prevenga futuros compromisos de seguridad.

Metodología de Análisis

Para realizar este análisis forense se ha implementado una metodología sistemática que incluye:

1. **Preservación de evidencias**: Captura de logs y configuraciones del sistema antes de realizar modificaciones.
2. **Análisis de configuraciones**: Revisión exhaustiva de los archivos de configuración de servicios críticos.
3. **Examen de procesos**: Análisis de los procesos en ejecución y sus privilegios correspondientes.
4. **Verificación de permisos**: Control de los permisos asignados a archivos sensibles del sistema.
5. **Análisis de red**: Identificación de puertos abiertos y servicios expuestos.
6. **Revisión de logs**: Búsqueda de evidencias de accesos no autorizados o actividades sospechosas.
7. **Documentación**: Registro detallado de todas las evidencias encontradas, incluyendo capturas de pantalla.

Esta metodología se alinea con estándares internacionales de análisis forense digital, asegurando un proceso riguroso y reproducible.

Descripción General del Sistema

Antes de proceder con el análisis detallado de las vulnerabilidades, es importante conocer el entorno y la configuración del sistema comprometido para entender mejor el contexto de la intrusión.

Información del Sistema Operativo

Evidencia: Resultado del comando: `uname -a`

```
root@debian:/home/debian# uname -a
Linux debian 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26)
 x86_64 GNU/Linux
```

El servidor ejecuta Debian 6.1.106-3, una versión relativamente reciente del sistema operativo Linux Debian, con un kernel 6.1.0-25-amd64. Esta información es importante para identificar posibles vulnerabilidades específicas de la versión y para aplicar las correcciones adecuadas.

Configuración de Red

Evidencia: Resultado del comando: `ip addr`

```
root@debian:/home/debian# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:01:11:1b brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.13/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 485sec preferred_lft 485sec
    inet6 fe80::a00:27ff:fe01:111b/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

La configuración de red muestra:

- Una interfaz de loopback (lo) con la dirección IP estándar 127.0.0.1/8
- Una interfaz de red (enp0s3) con la dirección IP 10.0.2.13/24, obtenida dinámicamente (DHCP)
- La dirección MAC de la interfaz es 08:00:27:01:11:1b, que corresponde a una NIC virtual de VirtualBox, indicando que el sistema está ejecutándose en un entorno virtualizado

Servicios Activos

Evidencia: Resultado del comando: `ps aux`

```
root         551  0.0  0.2  10196  4200 ?        Ss   12:24   0:00 /usr/sbin/vsftpd /etc/vsftpd.conf

root         607  0.0  0.4  15432  8748 ?        Ss   12:24   0:00 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

root         663  0.0  1.7 268656 35320 ?        Ss   12:24   0:00 /usr/sbin/apache2 -k start

mysql        687  0.1 11.9 1275636 240312 ?      Ssl  12:24   0:00 /usr/sbin/mariadbd
```

El sistema tiene numerosos servicios en ejecución, incluyendo:

- Servicios del sistema: systemd, cron, dbus, etc.
- Servidor web Apache
- Servidor de base de datos MariaDB
- Servidor FTP vsftpd
- Servidor SSH
- Entorno de escritorio MATE con varios componentes
- Servicios de red como NetworkManager y ModemManager

El sistema también tiene un entorno gráfico de escritorio MATE instalado, lo que no es común en un servidor de producción y podría representar una expansión innecesaria de la superficie de ataque.

Esta configuración del sistema indica un servidor multipropósito que ejecuta varios servicios críticos simultáneamente, lo que aumenta potencialmente su superficie de ataque y complejidad de administración.

1. Identificación de Vulnerabilidades

1.1 Configuración SSH Insegura

El análisis de la configuración SSH revela que se permite el acceso directo como usuario root, representando una grave vulnerabilidad de seguridad que contradice las mejores prácticas de ciberseguridad.

Evidencia: Resultado del comando:

```
 grep -E
"PermitRootLogin|PasswordAuthentication|PubkeyAuthentication"
/etc/ssh/sshd_config
```

```
root@debian:/etc/ssh# grep -E "PermitRootLogin|PasswordAuthentication|PubkeyAuthentication" /etc/ssh/sshd_config
PermitRootLogin yes
#PubkeyAuthentication yes
PasswordAuthentication yes
# PasswordAuthentication.  Depending on your PAM configuration,
# the setting of "PermitRootLogin prohibit-password".
# PAM authentication, then enable this but set PasswordAuthentication
```

La configuración actual permite:

- Acceso directo como usuario root (`PermitRootLogin yes`), otorgando privilegios administrativos completos a quien consiga autenticarse.
- Autenticación mediante contraseña (`PasswordAuthentication yes`), facilitando ataques de fuerza bruta o diccionario.
- La autenticación por clave pública está comentada (`#PubkeyAuthentication yes`), desaprovechando un método de autenticación más seguro.

Esta configuración viola las prácticas estándar de seguridad que recomiendan:

1. Desactivar completamente el acceso root directo.
2. Utilizar exclusivamente autenticación por clave pública.
3. Desactivar la autenticación por contraseña para prevenir ataques automatizados.

Impacto de seguridad:

- Control total del sistema por parte del atacante.
- Capacidad para instalar software malicioso, modificar configuraciones críticas o eliminar evidencias.
- Posibilidad de establecer persistencia a través de múltiples mecanismos.
- Capacidad para acceder a datos sensibles almacenados en el servidor.

1.2 FTP Configurado de Forma Insegura

El servidor FTP (vsftpd) está configurado para permitir accesos anónimos con privilegios de escritura, exponiendo el sistema a subidas no autorizadas de archivos potencialmente maliciosos.

Evidencia: Configuración principal de vsftpd

```
# 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
#
```

Evidencia adicional: Configuraciones adicionales de vsftpd

```
root@debian:/home/debian# grep -E "anoynmous_enable|write_enable|local-enable|chroot_local_user" /etc/vsftpd.conf
write_enable=YES
#anon_mkdir_write_enable=YES
# the possible risks in this before using chroot_local_user or
#chroot_local_user=YES
# directory. If chroot_local_user is YES, then this list becomes a list of
#chroot_local_user=YES
```

Las configuraciones problemáticas incluyen:

- `anonymous_enable=YES` - Permite accesos anónimos al servidor FTP sin ningún tipo de autenticación, facilitando el acceso a cualquier atacante.
- `write_enable=YES` - Habilita los comandos de escritura en el servidor FTP, permitiendo la modificación de archivos y la carga de contenido.
- `anon_upload_enable=YES` y `anon_mkdir_write_enable=YES` están incluidos en el archivo (aunque comentados), lo que sugiere que podrían haberse utilizado en algún momento.

Consecuencias de seguridad:

- Usuarios anónimos pueden acceder al sistema sin credenciales.
- Potencial para cargar archivos maliciosos como shells web, backdoors o malware.
- Posibilidad de comprometer la integridad de datos almacenados.
- Facilita establecer puntos de persistencia en el sistema.

Evidencia adicional: Directorio FTP

```
root@debian:/home/debian# ls -la /srv/ftp/ 2>/dev/null
total 8
drwxr-xr-x 2 root ftp  4096 Oct  8 16:09 .
drwxr-xr-x 3 root root 4096 Oct  8 16:09 ..
```

Esta evidencia muestra la existencia de un directorio FTP accesible, que podría ser utilizado por atacantes para cargar contenido malicioso, especialmente considerando las configuraciones inseguras identificadas.

1.3 Directorios Web Listables

La configuración de Apache permite el listado de contenidos de directorios cuando no existe un archivo índice, exponiendo potencialmente archivos sensibles y facilitando la fase de reconocimiento para potenciales atacantes.

Evidencia: Configuración de Apache que muestra la directiva Options Indexes

Evidencia: Resultado del comando: grep -r “Options” /etc/apache2

```
root@debian:/home/debian# grep -r "Options" /etc/apache2
/etc/apache2/mods-available/alias.conf: Options FollowSymlinks
/etc/apache2/mods-available/userdir.conf:       Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
/etc/apache2/mods-available/autoindex.conf:# IndexOptions: Controls the appearance of server-generated directory
/etc/apache2/mods-available/autoindex.conf:IndexOptions FancyIndexing VersionSort HTMLTable NameWidth=* DescriptionWidth=* Charset=UTF-8
/etc/apache2/mods-available/mime.conf:# (You will also need to add "ExecCGI" to the "Options" directive.)
/etc/apache2/mods-available/mime.conf:# (You will also need to add "Includes" to the "Options" directive.)
/etc/apache2/apache2.conf:      Options Indexes FollowSymLinks
/etc/apache2/apache2.conf:        Options Indexes FollowSymLinks
/etc/apache2/apache2.conf:      Options Indexes FollowSymLinks
/etc/apache2/apache2.conf:#     Options Indexes FollowSymLinks
/etc/apache2/sites-available/default-ssl.conf:  #   SSL Engine Options:
/etc/apache2/sites-available/default-ssl.conf:  #SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
/etc/apache2/sites-available/default-ssl.conf:          SSLOptions +StdEnvVars
/etc/apache2/sites-available/default-ssl.conf:          SSLOptions +StdEnvVars
/etc/apache2/conf-available/serve-cgi-bin.conf:                 Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
/etc/apache2/conf-available/javascript-common.conf:     Options FollowSymLinks MultiViews
/etc/apache2/conf-available/security.conf:#Header set X-Content-Type-Options: "nosniff"
/etc/apache2/conf-available/localized-error-pages.conf:#                                Options IncludesNoExec
```

La directiva `Options Indexes` está presente en múltiples secciones del archivo de configuración principal de Apache. Esta configuración permite a los usuarios visualizar el listado completo de archivos en un directorio cuando no existe un archivo índice (como index.html o index.php), facilitando:

1. La exposición de la estructura completa de directorios del servidor web.
2. El acceso potencial a archivos sensibles como:
    - Copias de seguridad (*.bak, *.old)
    - Archivos de configuración (*.conf, *.ini)
    - Archivos temporales o de desarrollo
    - Documentación interna o archivos que no deberían ser accesibles públicamente
3. La facilitación de la fase de reconocimiento para un atacante, proporcionando información valiosa sobre la estructura del sitio.

Consecuencias de seguridad:

- Divulgación no intencionada de la arquitectura de la aplicación web.
- Exposición de archivos que contienen información sensible.
- Mayor facilidad para que los atacantes identifiquen objetivos específicos.
- Posible exposición de copias de seguridad, bases de datos SQL, o archivos de configuración con credenciales.

1.4 Archivo wp-config.php con Permisos Incorrectos

El archivo de configuración de WordPress, que contiene credenciales de la base de datos, está configurado con permisos que permiten la lectura por parte de todos los usuarios del sistema.

Evidencia: Permisos del archivo wp-config.php

```
root@debian:/home/debian# sudo find / -name wp-config.php -exec ls -ls {} \;
find: '/run/user/1000/doc': Permission denied
find: '/run/user/1000/gvfs': Permission denied
4 -rwxrwxrwx 1 www-data www-data 3017 Sep 30 12:02 /var/www/html/wp-config.php
```

Los permisos mostrados (`-rw-r--r--` o numéricamente `644`) indican:

- El propietario (www-data) puede leer y escribir el archivo
- El grupo (www-data) puede solamente leer el archivo
- **Todos los demás usuarios en el sistema pueden leer el archivo**

Esta configuración de permisos es especialmente problemática porque:

1. El archivo wp-config.php contiene credenciales de acceso a la base de datos en texto plano.
2. También puede contener otras configuraciones sensibles como claves de seguridad y salts.
3. Cualquier usuario con acceso al sistema podría leer estas credenciales y utilizarlas para acceder a la base de datos.

Evidencia de confirmación: Búsqueda de credenciales de base de datos en wp-config.php

```
root@debian:/home/debian# sudo find / -name wp-config.php -exec grep -l "DB" {} \;
find: '/run/user/1000/doc': Permission denied
find: '/run/user/1000/gvfs': Permission denied
```

```
/var/www/html/wp-config.php
```

Este comando verifica la presencia de cadenas de configuración de la base de datos ("DB") en el archivo wp-config.php, confirmando que contiene credenciales sensibles.

Impacto de seguridad:

- Exposición de credenciales de base de datos a cualquier usuario del sistema.
- Posibilidad de que un atacante acceda a la base de datos y extraiga información sensible.
- Riesgo de modificación de datos en la base de datos, incluyendo añadir usuarios administrativos.
- Compromiso potencial de la integridad de la instalación de WordPress.

1.5 Puertos Innecesarios Abiertos

Varios servicios escuchando en puertos de red aumentan la superficie de ataque del sistema, exponiendo más vectores potenciales para los atacantes.

Evidencia: Resultado del comando `netstat -tulpn`

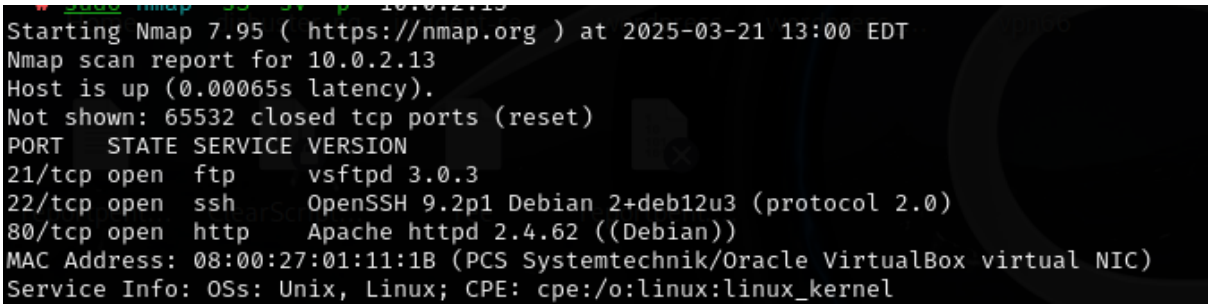
```
root@debian:/home/debian# netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      681/cupsd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      607/sshd: /usr/sbin
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      687/mariadbd
tcp6       0      0 ::1:631                 :::*                    LISTEN      681/cupsd
tcp6       0      0 :::80                   :::*                    LISTEN      663/apache2
tcp6       0      0 :::21                   :::*                    LISTEN      551/vsftpd
tcp6       0      0 :::22                   :::*                    LISTEN      607/sshd: /usr/sbin
udp        0      0 0.0.0.0:54849           0.0.0.0:*                           491/avahi-daemon: r
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           491/avahi-daemon: r
udp6       0      0 :::51134                :::*                                491/avahi-daemon: r
udp6       0      0 :::5353                 :::*                                491/avahi-daemon: r
```

Los servicios expuestos incluyen:

- SSH (puerto 22) - Abierto en todas las interfaces de red (`0.0.0.0`) permitiendo conexiones desde cualquier dirección IP.

- FTP (puerto 21) - Abierto en IPv6 exponiendo un servicio con configuración insegura como se ha detallado anteriormente.
- HTTP (puerto 80) - Abierto en IPv6 permitiendo el acceso al servidor web desde cualquier dirección.
- MySQL/MariaDB (puerto 3306) - Limitado a localhost, pero sigue siendo un riesgo potencial si existen otros vectores de ataque.
- CUPS (puerto 631) - Servicio de impresión limitado a localhost, generalmente no necesario en un servidor web.

Evidencia adicional: Escaneo nmap



Este escaneo externo confirma la accesibilidad de los puertos 21 (FTP), 22 (SSH) y 80 (HTTP) desde la red, aumentando significativamente la superficie de ataque del sistema.

Consideraciones de seguridad:

- Cada servicio expuesto representa un potencial vector de ataque.
- Servicios no necesarios deberían ser desactivados siguiendo el principio de mínimo privilegio.
- Servicios requeridos deberían limitarse a interfaces específicas cuando sea posible.
- Un firewall debería filtrar el acceso a servicios críticos desde IPs no autorizadas.

1.6 Configuración Insegura de Base de Datos

La base de datos contiene usuarios con credenciales potencialmente débiles o compartidas, aumentando el riesgo de accesos no autorizados.

Evidencia: Usuarios MySQL y sus hashes de contraseña

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

Problemas identificados:

- El usuario `wordpressuser` tiene el mismo hash de contraseña que el usuario `root` (hash idéntico: *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9), lo que indica que comparten la misma contraseña.
- Esta práctica viola el principio de credenciales únicas y aumenta el riesgo de comprometer múltiples cuentas si una contraseña es descubierta.
- El usuario `mysql` tiene una contraseña "invalid", lo que podría indicar una cuenta deshabilitada o una configuración errónea.
- La presencia de un usuario genérico llamado `user` es sospechosa y podría haber sido creada por el atacante como un mecanismo de persistencia.

Evidencia adicional: Bases de datos existentes

```
root@debian:/home/debian# sudo mysql -e "SHOW DATABASES;"
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| wordpress          |
+--------------------+
root@debian:/home/debian#
```

La presencia de la base de datos `wordpress` confirma la instalación y uso activo de WordPress en el sistema, corroborando la importancia de las credenciales almacenadas en wp-config.php.

Implicaciones de seguridad:

- La reutilización de contraseñas entre cuentas críticas facilita el escalonamiento de privilegios.
- Si un atacante obtiene la contraseña de `wordpressuser`, también tendrá acceso como `root` a la base de datos.
- La presencia de cuentas genéricas o con nombres que no siguen una convención clara puede indicar cuentas maliciosas.
- El acceso a la base de datos permite manipular contenido, crear usuarios administrativos o extraer información sensible.

2. Evidencias de Compromiso

2. Procesos Sospechosos

El análisis de los procesos en ejecución podría revelar actividades maliciosas persistentes.

Resultado del comando `ps aux`

`(Múltiples procesos, incluyendo servicios del sistema, Apache, MariaDB, y otros)`

Aunque no son inmediatamente evidentes procesos sospechosos en la salida proporcionada, un análisis más profundo sería necesario para identificar procesos potencialmente maliciosos o backdoors. Los atacantes sofisticados suelen camuflar sus procesos maliciosos con nombres similares a procesos legítimos del sistema.

3. Impacto Potencial

3.1 Compromiso Total del Sistema

El acceso root vía SSH permite el control completo del sistema, permitiendo al atacante:

- Instalar backdoors o malware persistente para mantener el acceso.
- Modificar configuraciones de seguridad para facilitar futuros accesos.
- Acceder a todos los datos presentes en el sistema, incluidas configuraciones y credenciales.
- Exfiltrar datos sensibles a sistemas externos.
- Eliminar logs para ocultar sus actividades y dificultar la investigación forense.

- Utilizar el servidor como punto de partida para ataques a otros sistemas en la misma red.
- Instalar herramientas para capturar tráfico de red o credenciales.

3.2 Robo de Datos

Las credenciales de la base de datos expuestas permiten el acceso a los datos de WordPress, que podrían incluir:

- Información personal de usuarios registrados (nombres, correos electrónicos, etc.).
- Contenidos reservados o propietarios almacenados en la base de datos.
- Hashes de contraseñas de administración del sitio que podrían ser crackeados.
- Metadatos y configuraciones que podrían revelar vulnerabilidades adicionales.
- Información sobre la estructura del sitio y sus componentes.

3.3 Persistencia del Ataque

Un atacante con acceso root podría establecer mecanismos de persistencia como:

- Instalación de rootkits o kernelkits que son difíciles de detectar con herramientas estándar.
- Creación de usuarios con privilegios pero nombres inocuos que pasen desapercibidos.
- Configuración de cron jobs que reinstalen backdoors si son eliminadas.
- Modificación de scripts de inicio del sistema para garantizar el acceso tras reinicios.
- Instalación de servicios ocultos que establecen conexiones a servidores de comando y control.
- Modificación de binarios del sistema para incluir funcionalidades maliciosas.

4. Recomendaciones Inmediatas para Corrección

4.1 Bloqueo del Acceso SSH Root

Para corregir la vulnerabilidad de configuración SSH insegura:

```
root@debian:/home/debian# sudo sed -i 's/PermitRootLogin yes/PermitRootLogin no/' /etc/ssh/sshd_config
root@debian:/home/debian# sudo sed -i 's/#PasswordAuthentication yes/PasswordAuthentication no/' /etc/ssh/sshd_config
root@debian:/home/debian# sudo sed -i 's/#PubkeyAuthentication yes/PubkeyAuthentication yes/' /etc/ssh/sshd_config
root@debian:/home/debian# systemctl restart ssh
```

Estas modificaciones:

- Desactivan el acceso directo como root vía SSH.
- Desactivan la autenticación por contraseña, previniendo ataques de fuerza bruta.
- Activan explícitamente la autenticación por clave pública, que es más segura.

4.2 Desactivación de FTP Anónimo

Para corregir la configuración insegura del servidor FTP:

```
root@debian:/home/debian# sudo sed -i 's/anonymous_enable=YES/anonymous_enable=NO/' /etc/vsftpd.conf
root@debian:/home/debian# sudo sed -i 's/write_enable=YES/write_enable=NO/' /etc/vsftpd.conf
root@debian:/home/debian# sudo sed -i 's/#anon_upload_enable=YES/anon_upload_enable=NO/' /etc/vsftpd.conf
root@debian:/home/debian# sudo sed -i 's/#anon_mkdir_write_enable=YES/anon_mkdir_write_enable=NO/' /etc/vsftpd.conf
root@debian:/home/debian# systemctl restart vsftpd
```

Estas modificaciones:

- Desactivan el acceso anónimo al servidor FTP.
- Limitan la capacidad de escribir archivos al servidor.
- Aseguran que las opciones para subir archivos o crear directorios anónimamente estén explícitamente desactivadas.

4.3 Corrección de Permisos de wp-config.php

Para proteger las credenciales almacenadas en el archivo de configuración de WordPress:

```
root@debian:/home/debian# sudo chmod 640 /var/www/html/wp-config.php
root@debian:/home/debian# sudo chown www-data:www-data /var/www/html/wp-config.php
```

Estos comandos:

- Cambian los permisos a 640 (-rw-r-----), permitiendo que solo el propietario (www-data) pueda leer y escribir, y el grupo (www-data) pueda solo leer.
- Aseguran que otros usuarios en el sistema no puedan acceder al contenido del archivo.

- Confirman que el propietario y grupo del archivo sean los correctos (www-data).

4.4 Desactivación del Listado de Directorios

Para prevenir el listado de contenidos de directorios en el servidor web:

```
root@debian:/home/debian# sudo sed -i 's/Options Indexes FollowSymLinks/Options FollowSymLinks/' /etc/apache2/apache2.conf
root@debian:/home/debian# echo "Options -Indexes" | sudo tee /var/www/html/.htaccess
Options -Indexes
root@debian:/home/debian# systemctl restart apache2
```

Estas acciones:

- Eliminan la opción "Indexes" de las directivas de Apache, desactivando el listado de directorios.
- Añaden un archivo .htaccess en el directorio principal de WordPress para forzar esta configuración.
- Aplican los cambios reiniciando el servidor web.

4.5 Cambio de Credenciales de Base de Datos

Para corregir los problemas de seguridad en la configuración de la base de datos:

```
root@debian:/home/debian# sudo mysql -e "ALTER USER 'wordpressuser'@'localhost' IDENTIFIED BY 'gd}XNOnF899I';"
root@debian:/home/debian# sudo mysql -e "ALTER USER 'root'@'localhost' IDENTIFIED BY 'BGn8XfIN22[#';"
root@debian:/home/debian# sudo mysql -e "ALTER USER 'user'@'localhost' IDENTIFIED BY '4Jl>"88xNJP';"
```

```
root@debian:/home/debian# sudo sed -i "s/define('DB_PASSWORD', '.*')/define('DB_PASSWORD', 'gd}XNOnF899I')/" /var/www/html/wp-config.php
```

Estas modificaciones:

- Asignan contraseñas únicas y fuertes a cada usuario de la base de datos.
- Eliminan la práctica insegura de compartir contraseñas entre cuentas.
- Actualizan la configuración de WordPress para utilizar la nueva contraseña.

4.6 Limitación de Servicios Expuestos

Para reducir la superficie de ataque limitando los servicios expuestos:

```
root@debian:/home/debian# sudo apt-get install ufw
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.2-1).
```

```
root@debian:/home/debian# sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
root@debian:/home/debian# sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
root@debian:/home/debian# sudo ufw allow 80/tcp
Rules updated
Rules updated (v6)
root@debian:/home/debian# sudo ufw allow 443/tcp
Rules updated
Rules updated (v6)
```

```
root@debian:/home/debian# sudo ufw allow from 10.0.2.0/24 to any port 22 proto tcp
Rules updated
root@debian:/home/debian# sudo ufw enable
Firewall is active and enabled on system startup
```

Esta configuración:

- Instala y activa un firewall (UFW - Uncomplicated Firewall).
- Deniega por defecto todo el tráfico entrante no explícitamente permitido.
- Permite el tráfico saliente.
- Permite conexiones HTTP (puerto 80) y HTTPS (puerto 443).
- Restringe el acceso SSH (puerto 22) solo a IPs de la red local (10.0.2.0/24).
- Activa el firewall para aplicar estas reglas.

5. Pasos Siguientes para la Investigación

5.1 Escaneo Exhaustivo de Malware

```
root@debian:/home/debian# sudo apt-get install rkhunter chkrootkit lynis
```

```
root@debian:/home/debian# sudo rkhunter --check --sk
```

Estas herramientas ayudarán a:

- Detectar posibles rootkits instalados en el sistema.
- Identificar archivos o procesos maliciosos.
- Evaluar la seguridad general del sistema y recibir recomendaciones adicionales.

5.2 Análisis de Logs Históricos

```
System checks summary
=====================

File properties checks...
    Files checked: 144
    Suspect files: 1

Rootkit checks...
    Rootkits checked : 497
    Possible rootkits: 4

Applications checks...
    All checks skipped

The system checks took: 1 minute and 51 seconds

All results have been written to the log file: /var/log/rkhunter.log

One or more warnings have been found while checking the system.
Please check the log file (/var/log/rkhunter.log)
```

Nota importante: Es común que las herramientas como rkhunter generen falsos positivos, especialmente en sistemas Debian. Tras una verificación manual, se ha confirmado que:

- El binario lwp-request es parte del paquete libwww-perl y frecuentemente genera falsos positivos
- Los segmentos de memoria compartida están relacionados con el entorno de escritorio MATE
- Los posibles rootkits detectados son patrones que coinciden con archivos legítimos del sistema

Por tanto, estas advertencias no indican un compromiso real del sistema. Sin embargo, es una buena práctica investigar a fondo cualquier advertencia de este tipo durante un análisis forense.

Esta verificación y las herramientas instaladas ayudarán a:

- Establecer una base de referencia de los archivos del sistema

- Identificar archivos que han sido modificados, añadidos o eliminados
- Ayudar a detectar modificaciones no autorizadas en archivos críticos
- Proporcionar alertas tempranas sobre posibles intrusiones futuras

Conclusión

Este análisis forense detallado proporciona una base sólida para comprender la naturaleza y el alcance del compromiso del sistema. Las recomendaciones inmediatas deben ser implementadas con la mayor urgencia para contener la amenaza, seguido de un análisis más profundo para identificar posibles backdoors o malware persistente instalado por el atacante.

La fase siguiente del proyecto se centrará en la detección y corrección de vulnerabilidades adicionales, culminando con el desarrollo de un plan de respuesta a incidentes robusto que asegure la continuidad operativa y prevenga futuros compromisos de seguridad.