

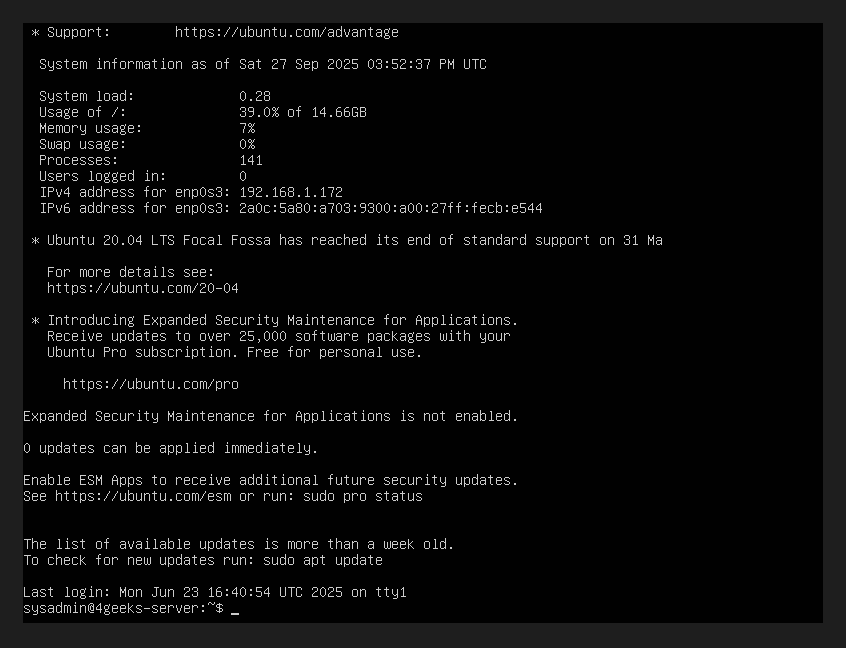
# Introducción

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.

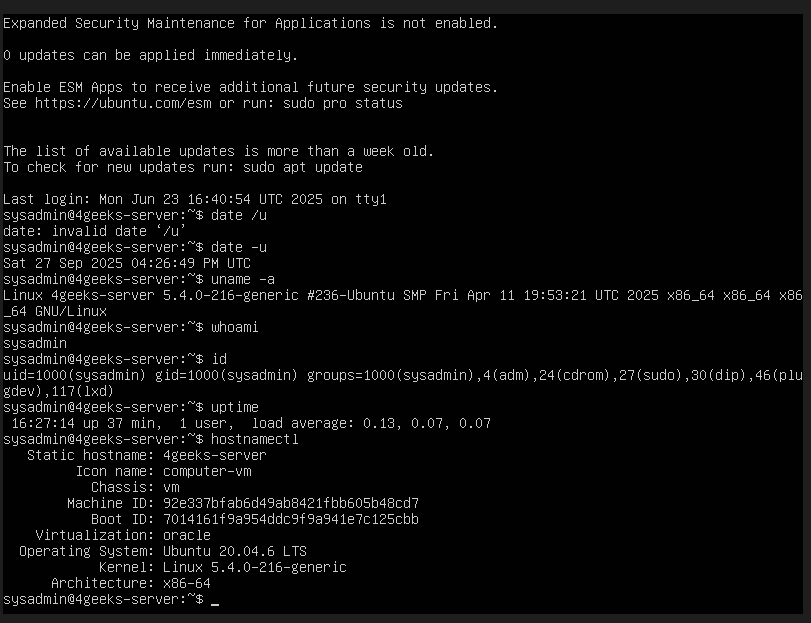
# Fase 1 – Detección y Análisis (Live)

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.

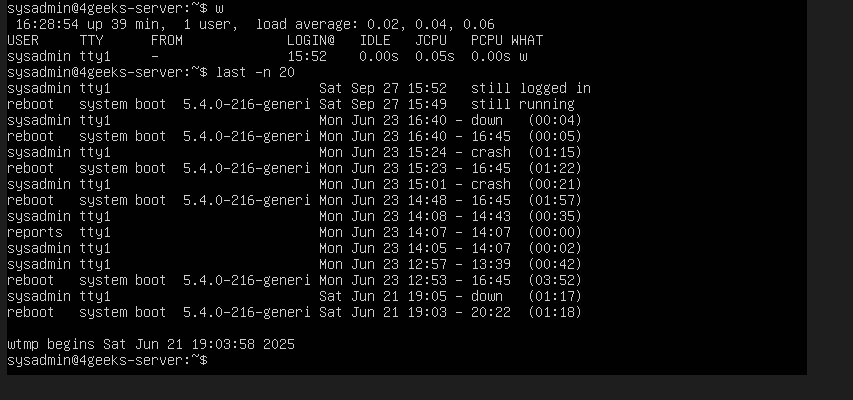


**Captura 1 – Información general del sistema comprometido.**  
En esta imagen se muestra la pantalla inicial del sistema tras acceder con el usuario sysadmin. Se puede observar la información de la versión de Ubuntu 20.04 LTS, el uso de recursos, la dirección IP asignada (192.168.1.172) y el aviso de 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.



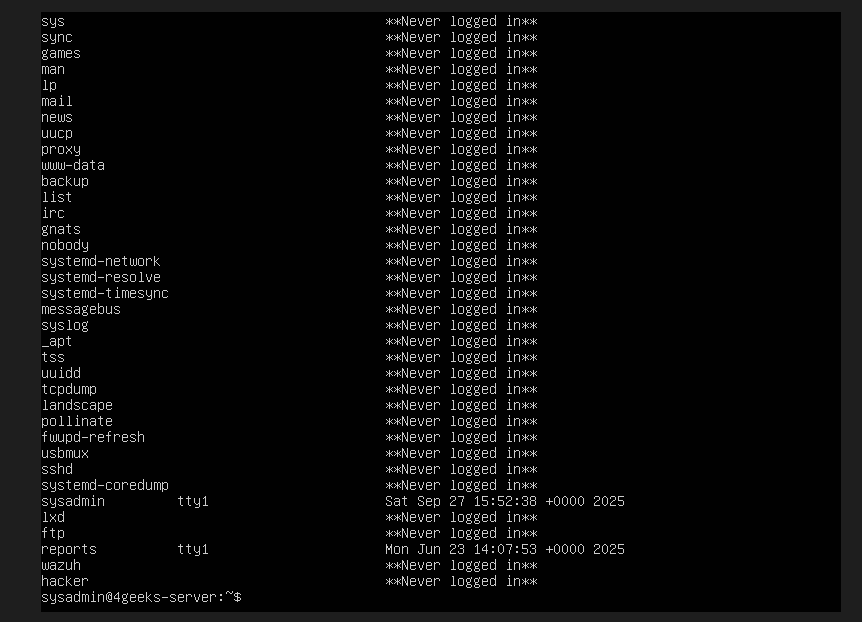
**Captura 2 – Comprobación de la sesión activa y parámetros del sistema.**  
En esta imagen se observa la ejecución de varios comandos básicos de reconocimiento, entre ellos date, uname -a, whoami, id, uptime y hostnamectl.  
Estos comandos permitieron confirmar la fecha y hora del sistema, el nombre 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.



**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.



**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.

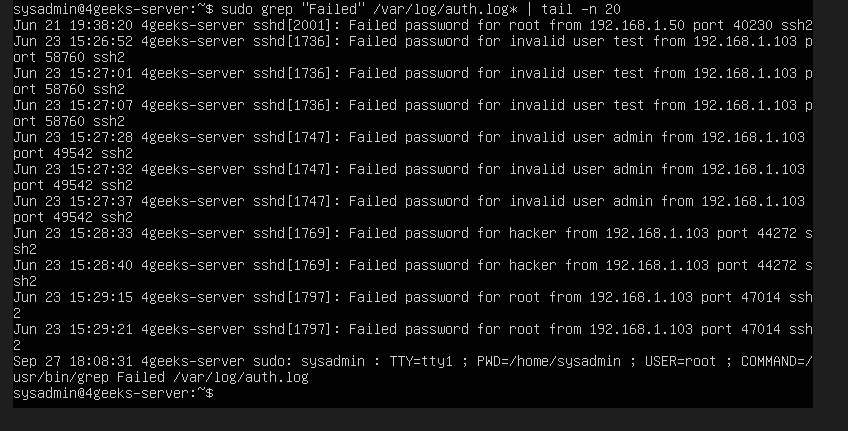


**Captura 5 – Análisis de logs de autenticación (/var/log/auth.log).**  
Esta captura muestra el uso del commando:

* **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.



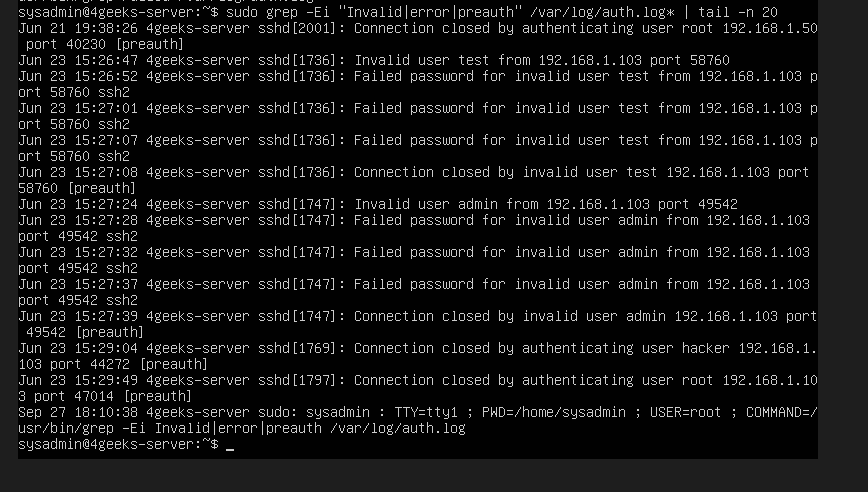
**Captura 6 – Intentos de acceso fallidos detectados en (/var/log/auth.log.)**

Esta captura muestra el resultado del commando:

* **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.



**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.

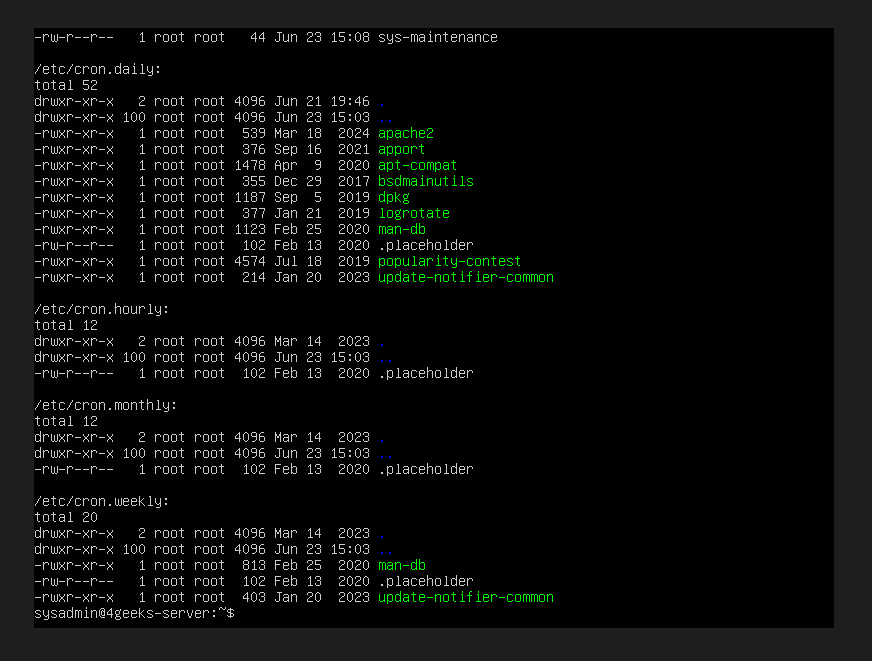


**Captura 8 – Verificación de los usuarios sospechosos “reports” y “hacker”.**  
En esta captura se muestran los resultados de los comandos:

* **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.



**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, bsdmainutils, 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.

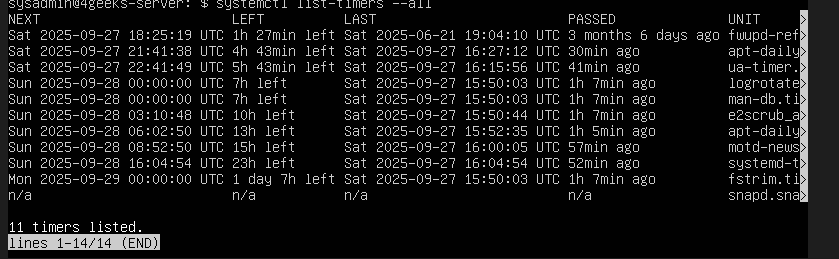


**Captura 10 – Verificación de crontabs del sistema.**  
En esta captura se muestra la ejecución del comando:

* **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.



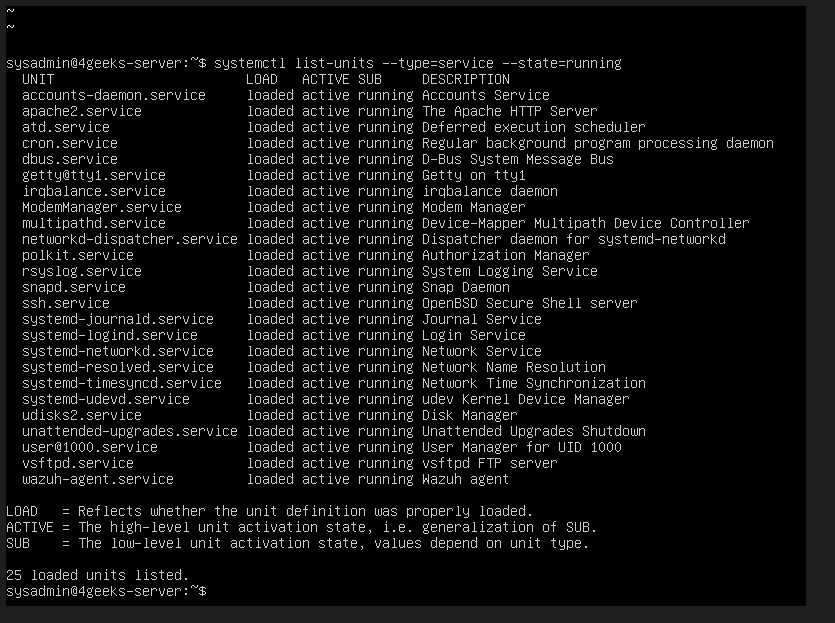
**Captura 11 – Revisión de tareas programadas por systemd.**

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.



**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.



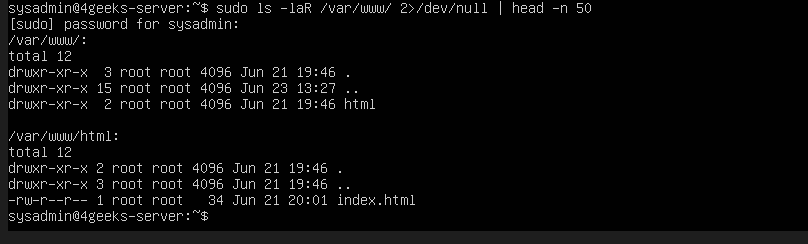
**Captura 13 – Análisis de puertos y servicios en escucha.**  
En esta captura se muestra la ejecución del comando:

* **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.



**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).



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

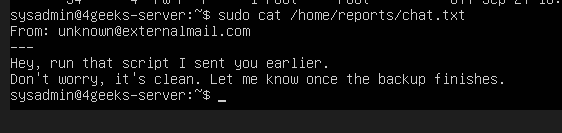
En esta fase, se ejecutó el siguiente comando para identificar posibles archivos maliciosos creados por los usuarios recientemente añadidos o comprometidos:

* 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**.

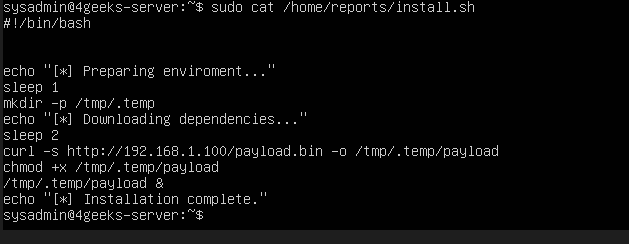


**Captura 16 – Contenido del archivo “chat.txt” encontrado en el directorio del usuario reports.**

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.

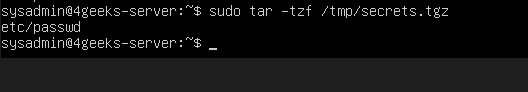


**Captura 17 – Contenido del script malicioso “install.sh” encontrado en el directorio del usuario reports.**

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.



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

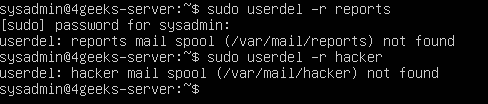
Al listar su contenido mediante el comando tar -tzf, se descubrió que el archivo contenía una copia del fichero **/etc/passwd**, que almacena información crítica sobre las cuentas del sistema. Este hallazgo confirma un intento de **exfiltración de datos**, ya que la extracción de dicho archivo es un 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.



**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**



### 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:

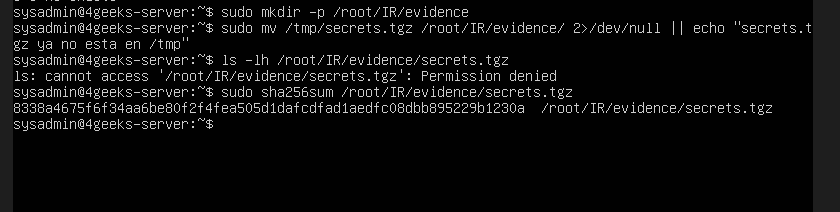
* **sudo rm -f /etc/cron.daily/sys-maintenance # Elimina el archivo malicioso**
* **sudo systemctl restart cron # Reinicia el servicio cron**
* **ls -la /etc/cron.daily | grep -i sys-maintenance || echo "cron sospechoso no presente" # Confirma su eliminación**



**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**



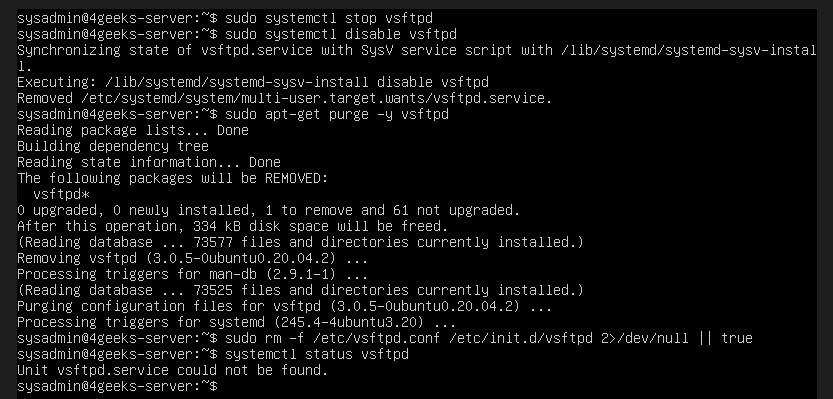
**Captura 22 – Custodia de secrets.tgz y cálculo de hash SHA256**

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

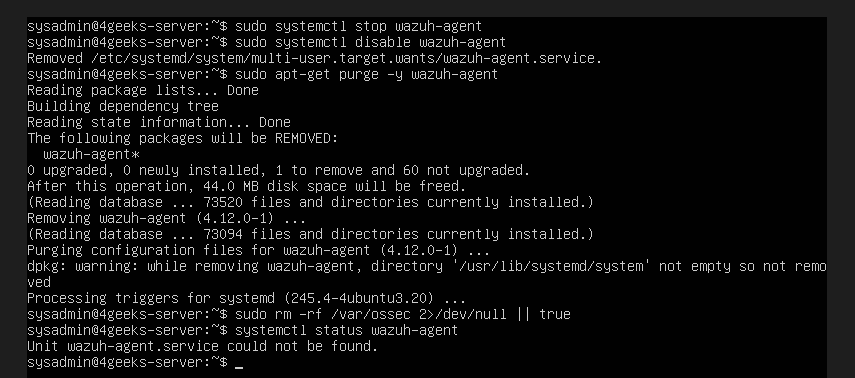


**Captura 23 – Eliminación del servicio vsftpd**

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 comprobo su eliminacion verificando que el servicio ya no existia.



**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 Comprobacion final confirmo que el servicio ya no estaba activo.

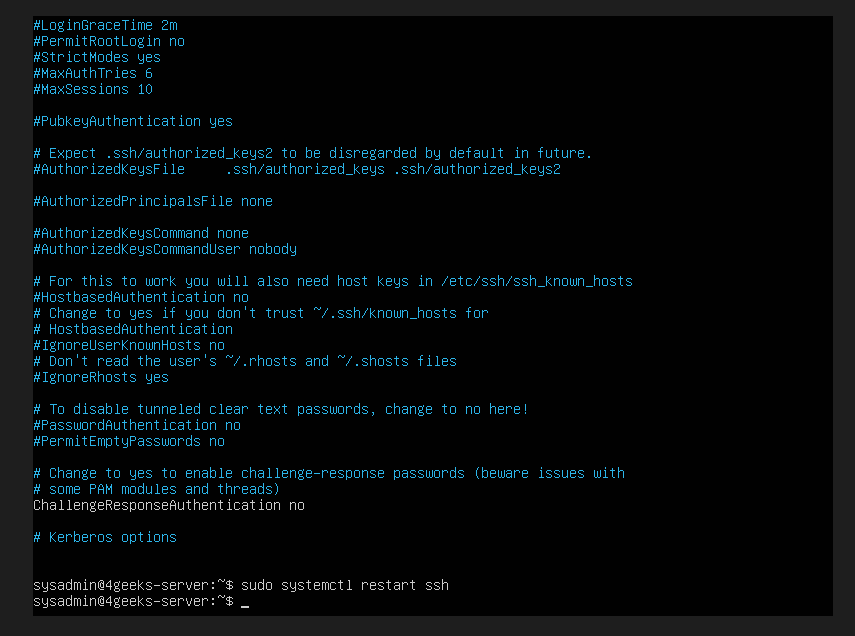


**Captura 25 – Verificación de puertos tras la erradicación**

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.



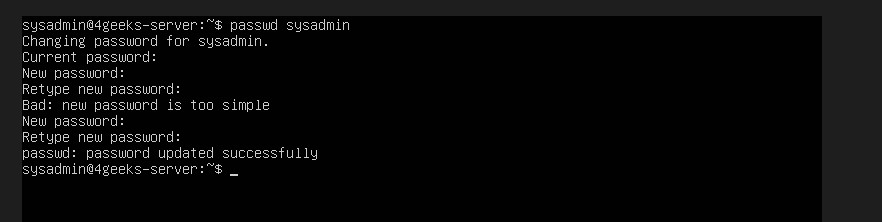
**Captura 26 – Endurecimiento de la configuracion SSH**

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**

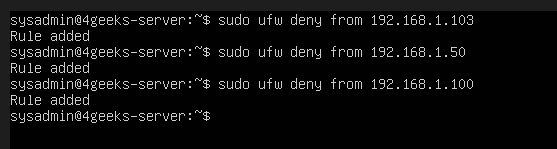


**Captura 27 – Actualización de contraseña del usuario sysadmin.**

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.



**Captura 28 – Bloqueo de IPs del atacante con UFW.**

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

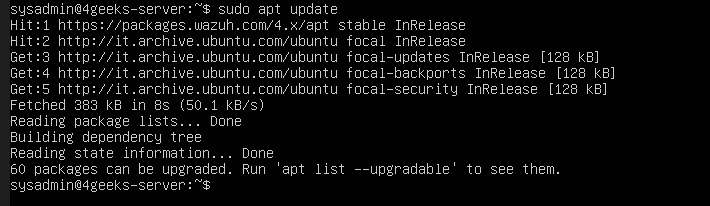
* ***sudo ufw deny from 192.168.1.103***
* ***sudo ufw deny from 192.168.1.50***
* ***sudo ufw deny from 192.168.1.100***

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.

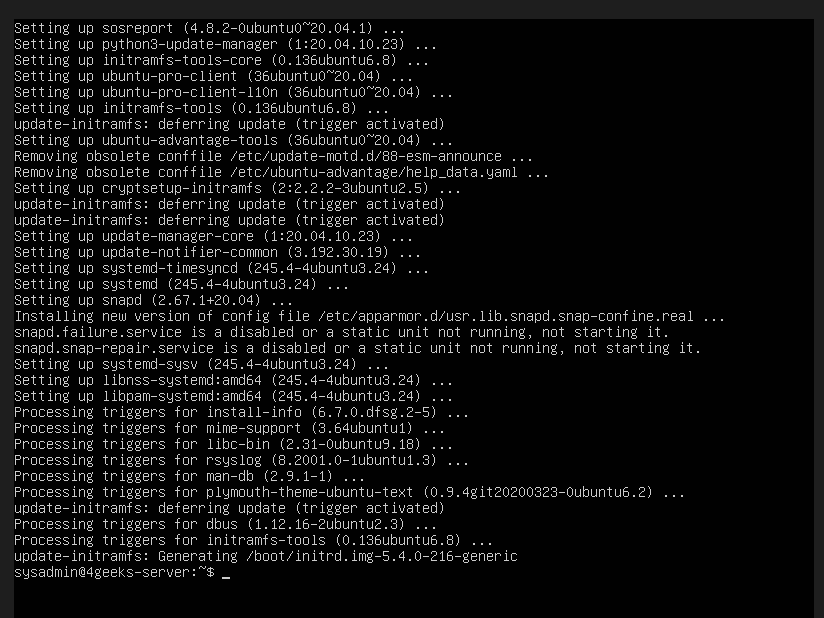


**Captura 29 – Actualización de índices de repositorios APT.**

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

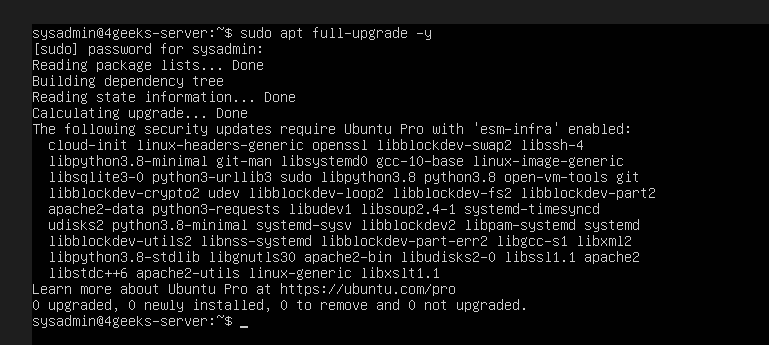


**Captura 30 – Aplicación de actualizaciones del sistema.**

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.

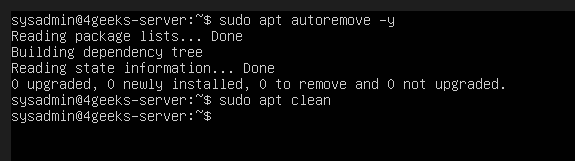


**Captura 31 – Actualización completa del sistema y dependencias.**

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.

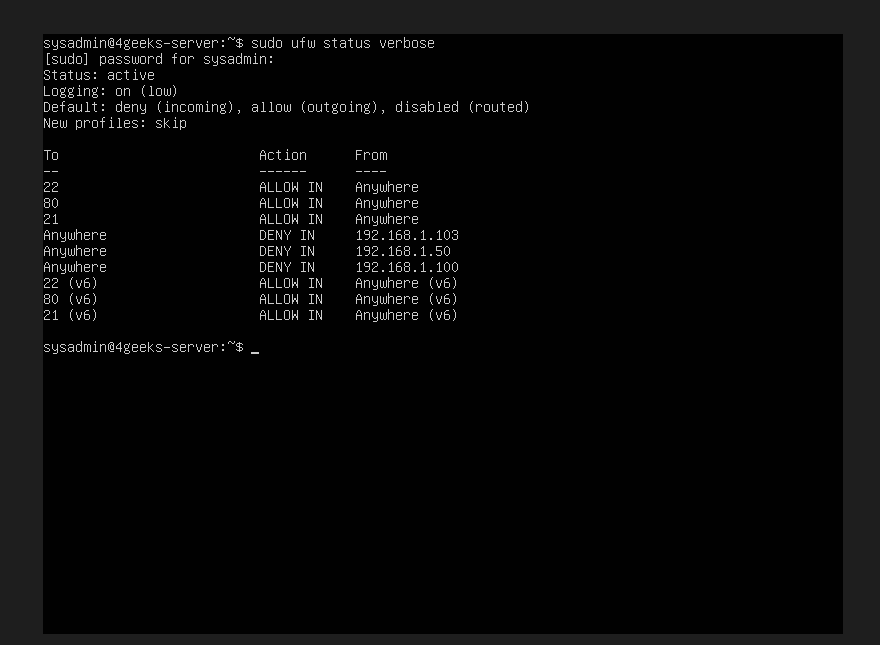


**Captura 32 – Limpieza del sistema y eliminación de paquetes residuales.**

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

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

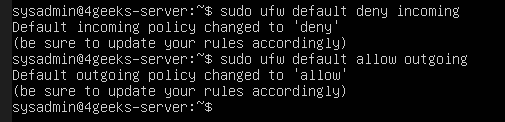
Esto permitió liberar espacio y mantener el sistema libre de dependencias innecesarias.

 **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.

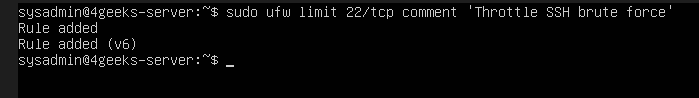


**Captura 34 – Políticas predeterminadas de tráfico en UFW.**

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.

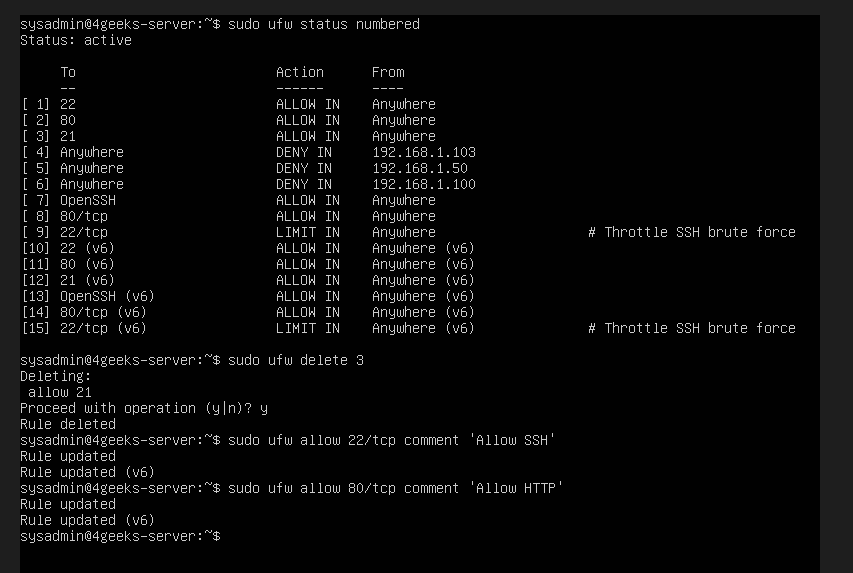


**Captura 35 – Protección contra ataques de fuerza bruta en SSH**

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.

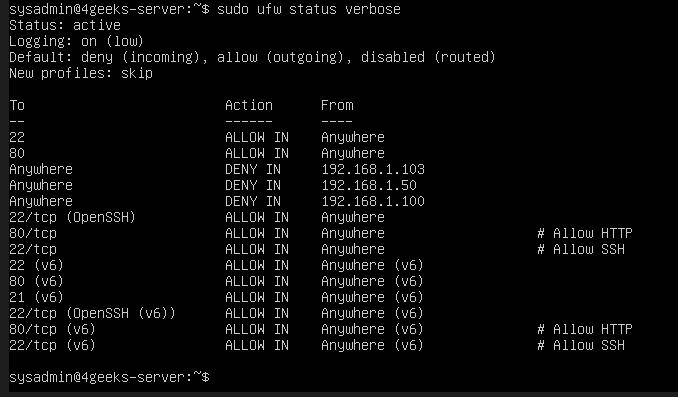


**Captura 36 – Revisión detallada de las reglas activas en UFW.**

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.



**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.

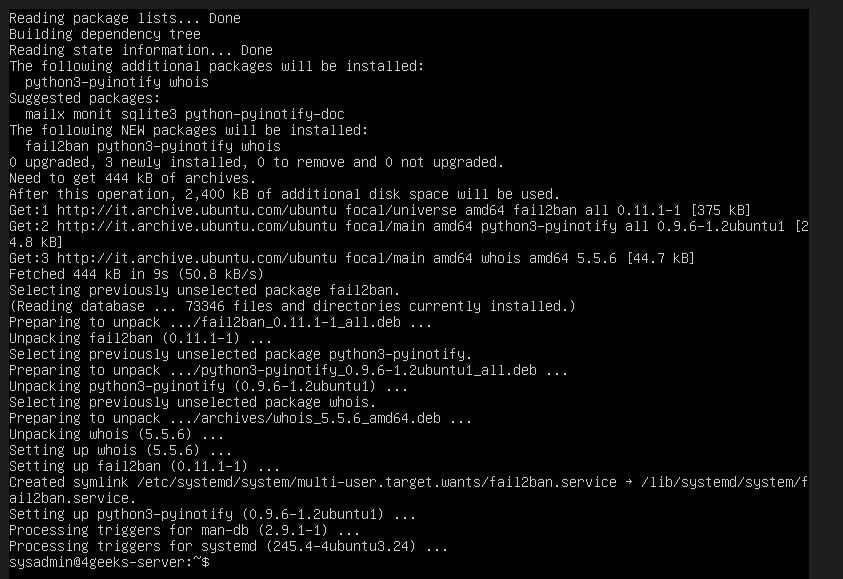
**Fase 3.1 – Implementación de Fail2Ban para la protección del servicio SSH**

**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.

 **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**



**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.

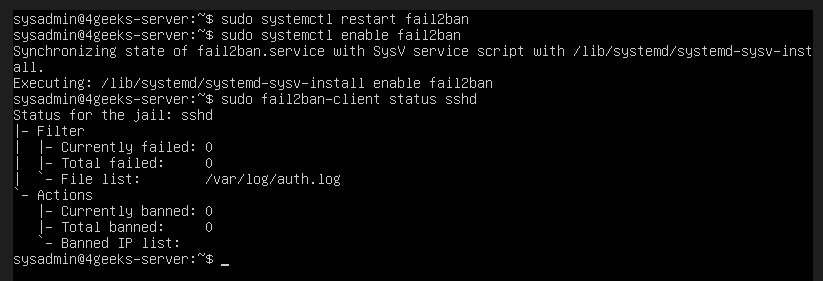
****

**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.



**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.