

# RESPUESTA EN VIVO A INCIDENTES

**Daniel jose javier Ramirez**

FASE 1

# RECONOCIMIENTO Y RECOLECCIÓN DE EVIDENCIAS

## RESUMEN:

Lo que se observa con suficiente consistencia es que en el sistema hay repetidos intentos fallidos de acceso por SSH (múltiples “Failed password”), puertos de red escuchando (SSH, Apache), y se ha generado un conjunto de archivos de recolección (phase 1\_\*) que contienen listados de puertos, logs y hashes. Con los datos actuales hay indicios de **fuerza bruta / intentos de acceso por SSH** desde IPs internas (IP visibles en logs), pero a esta etapa no puedo confirmar con absoluta certeza que exista una escalada de privilegios o backdoor persistente — para eso haría falta imagen forense, volcado de memoria y más logs completos.

# COMPROBACIÓN DE TIEMPO DE ACTIVIDAD

**Evidencia:** 1)uptime\_para\_ver\_el\_tiempo\_de\_actividad\_y\_cargag.png

**Comando:** `uptime`

**Salida:** `14:35:37 up 3:36, 1 user, load average: 0.11, 0.04, 0.01`

**Interpretación:** El sistema lleva encendido 3 horas y 36 minutos, hay 1 usuario conectado y la carga de CPU es baja.

**Relevancia:** El reinicio reciente puede haber eliminado evidencia volátil.

**Acciones:** Revisar los comandos `last` y `dmesg` para confirmar la causa del reinicio y, si se sospecha intrusión, evitar más reinicios y guardar memoria/logs.

```
sysadmin@4geeks-server:~$ uptime
14:15:37 up 3:36, 1 user, load average: 0.11, 0.04, 0.01
sysadmin@4geeks-server:~$ _
```

# USO DE MEMORIA Y SWAP

**Evidencia:** 2)uso\_de\_memoria\_y\_swap.png

**Comando:** `free -h`

**Explicación del comando:**

Muestra el uso de memoria RAM y swap en el sistema de forma legible para humanos (-h).

**Salida:**

- Memoria total: 3.8 GiB
- Memoria usada: 231 MiB
- Memoria libre: 2.8 GiB
- Memoria en caché: 865 MiB
- Swap total: 3.1 GiB
- Swap usada: 0 B

```
ysadmin@4geeks-server:~$ free -h
              total        used        free      shared  buff/cache   available
mem:          3.8Gi         231Mi         2.8Gi          1.0Mi          865Mi          3.4Gi
swap:          3.1Gi           0B          3.1Gi
ysadmin@4geeks-server:~$
```

**Interpretación:** El sistema tiene bajo consumo de memoria, la mayor parte está libre o en caché. No hay uso de swap.

**Relevancia:** No se observan signos de sobrecarga de memoria ni intercambio en disco, lo que indica un estado estable.

**Acciones:** Continuar monitoreando si el uso aumenta con procesos sospechosos.

# PROCESOS ACTIVOS

**Evidencia:** 3)top\_para\_ver\_procesos\_activos\_en\_tiempo\_real.png

**Comando:** top

## Explicación del comando:

Muestra en tiempo real los procesos activos, su consumo de CPU, memoria y el estado general del sistema. Es útil para identificar procesos sospechosos o con alto consumo de recursos.

## Salida:

- Tiempo encendido: 3:41 horas
- Usuarios conectados: 1
- Carga del sistema: 0.00 (sin carga)
- Tareas: 114 (113 en espera, 1 en ejecución, 0 detenidas, 0 zombis)
- CPU: 100% inactiva, sin procesos consumiendo recursos
- Memoria: 3919.9 MiB total, 228.8 MiB usada, 2824.4 MiB libre, 866.8 MiB en caché
- Swap: 3167.0 MiB total, 0 usada

**Interpretación:** El sistema está prácticamente inactivo, con solo procesos de sistema ejecutándose (ejemplo: systemd, kthreadd, rcu\_sched). No hay procesos sospechosos ni consumo anómalo de CPU o memoria.

**Relevancia:** Permite confirmar que el servidor no está bajo carga ni ejecutando malware visible en primer plano.

**Acciones:** Continuar monitoreando para detectar si aparecen procesos desconocidos o con alto consumo de recursos.

```
top - 14:20:41 up 3:41, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 114 total, 1 running, 113 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3919.9 total, 2824.4 free, 228.8 used, 866.8 buff/cache
MiB Swap: 3167.0 total, 3167.0 free, 0.0 used, 3456.1 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	169408	12972	8488	S	0.0	0.3	0:02.54	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.03	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kuworker/0:0H-kblockd
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
9	root	20	0	0	0	0	S	0.0	0.0	0:00.25	ksoftirqd/0
10	root	20	0	0	0	0	I	0.0	0.0	0:07.55	rcu_sched
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.30	migration/0
12	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
15	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
16	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/1
17	root	rt	0	0	0	0	S	0.0	0.0	0:00.70	migration/1
18	root	20	0	0	0	0	S	0.0	0.0	0:02.55	ksoftirqd/1
20	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kuworker/1:0H-kblockd
21	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kdevtmpfs
22	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
23	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_kthre
24	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kauditd
25	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khungtaskd
26	root	20	0	0	0	0	S	0.0	0.0	0:00.00	oom_reaper
27	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	urlliback
28	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kcompactd0
29	root	25	5	0	0	0	S	0.0	0.0	0:00.00	ksmd
30	root	39	19	0	0	0	S	0.0	0.0	0:00.00	khugenaged
77	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kintegrityd
78	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kblockd
79	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	blkcg_punt_bio
80	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	tpm_dev_wq

# PROCESOS INICIADOS RECIENTEMENTE

**Evidencia:** 5)que\_se\_lanzo\_recientemente.png y 5.1.png

## Comando utilizado:

Se utilizó el comando `sudo ps aux --sort=start_time` para listar todos los procesos y ordenarlos según la hora en que fueron iniciados.

## Explicación del comando:

Este comando muestra todos los procesos en ejecución y al ordenarlos por el tiempo de inicio permite identificar qué programas se lanzaron más recientemente.

## Salida:

- Procesos de sistema: systemd, ModemManager, atd, sshd, apache2.
- Procesos de seguridad: wazuh-agentd, wazuh-execd, wazuh-syscheckd, wazuh-logcollector, wazuh-modulesd.
- Sesiones de usuario: sysadmin en tty1.
- El proceso más reciente fue el propio `ps aux` usado para la verificación.

## Interpretación:

No se observan procesos maliciosos o inesperados. El sistema ejecuta servicios habituales.

## Relevancia:

Permite confirmar que en las últimas horas no se activaron programas sospechosos.

```
root      737  0.0  0.3 393276 12272 ?        Ssl  10:39  0:00 /usr/lib/udisks2/udisksd
daemon    742  0.0  0.0   3796  2216 ?        Ss   10:39  0:00 /usr/sbin/atd -f
root      749  0.0  0.0   6808  3120 ?        Ss   10:39  0:00 /usr/sbin/vsftpd /etc/vsftpd.conf
root      766  0.0  0.1  12168  7312 ?        Ss   10:39  0:00 sshd: /usr/sbin/sshd -D [listener
] 0 of 10-100 startups
root      785  0.0  0.2 315112 11648 ?        Ssl  10:39  0:00 /usr/sbin/ModemManager
root      825  0.0  0.5 107924 20868 ?        Ssl  10:39  0:00 /usr/bin/python3 /usr/share/unatt
ended-upgrades/unattended-upgrade-shutdown --wait-for-signal
root      843  0.0  0.1   6532  4488 ?        Ss   10:39  0:01 /usr/sbin/apache2 -k start
www-data  844  0.0  0.1 1212444 6344 ?        Sl   10:39  0:00 /usr/sbin/apache2 -k start
www-data  845  0.0  0.1 1212332 6216 ?        Sl   10:39  0:00 /usr/sbin/apache2 -k start
root      927  0.0  0.0   25880  3524 ?        Sl   10:39  0:00 /var/ossec/bin/wazuh-execd
wazuh     937  0.0  0.1 173620  7080 ?        Sl   10:39  0:03 /var/ossec/bin/wazuh-agentd
root     1014  0.0  0.2 124264  8140 ?        SNI  10:39  0:00 /var/ossec/bin/wazuh-syscheckd
root     1027  0.0  0.2 528084 11932 ?        Sl   10:39  0:04 /var/ossec/bin/wazuh-logcollector
root     1067  0.0  0.4 535264 16280 ?        Sl   10:39  0:01 /var/ossec/bin/wazuh-modulesd
root     1621  0.0  0.0   6000  3992 tty1    Ss   10:44  0:00 /bin/login -p --
root     1775  0.0  0.0     0     0 ?        S<   10:44  0:00 [loop3]
root     1810  0.0  0.3 1923104 39776 ?        Ssl  10:45  0:03 /usr/lib/snapd/snapd
sysadmin  2181  0.0  0.2  19060  9740 ?        Ss   10:45  0:00 /lib/systemd/systemd --user
sysadmin  2182  0.0  0.1 170748  4492 ?        S    10:45  0:00 (sd-pam)
sysadmin  2195  0.0  0.1   8496  5412 tty1    S    10:45  0:00 -bash
root     2323  0.0  0.0     0     0 ?        S<   10:45  0:00 [loop4]
root     2444  0.0  0.0     0     0 ?        S<   10:45  0:00 [loop5]
root     3432  0.0  0.2 249520  9748 ?        Ssl  12:31  0:00 /usr/lib/upower/upowerd
root     3476  0.0  0.0     0     0 ?        I    12:31  0:02 [kworker/0:0-events]
root     3482  0.1  0.0     0     0 ?        I    12:31  0:17 [kworker/1:2-mm_percpu_wq]
root     4142  0.0  0.0     0     0 ?        I    14:31  0:00 [kworker/1:1-events]
root     4163  0.0  0.0     0     0 ?        I    14:31  0:00 [kworker/0:2-events]
root     4233  0.0  0.0     0     0 ?        I    14:42  0:00 [kworker/u4:0-events_power_effici
ent]
root     4278  0.0  0.0     0     0 ?        I    14:53  0:00 [kworker/u4:2-events_unbound]
root     4350  0.0  0.0     0     0 ?        I    14:58  0:00 [kworker/u4:1-events_power_effici
ent]
root     4544  0.0  0.1   9264  4556 tty1    S+   15:07  0:00 sudo ps aux --sort=start_time
root     4545  0.0  0.0   9040  3368 tty1    R+   15:07  0:00 ps aux --sort=start_time
sysadmin@geeks-server:~/tmp$ _
```

# RELACIÓN JERÁRQUICA DE PROCESOS

**Evidencia:** 5.2)para\_ver\_quien\_lanzo\_a\_quien.png y 5.3.png

## Comando utilizado:

Se utilizó el comando `sudo pstree -p` para mostrar los procesos en forma de árbol junto con sus identificadores (PID).

## Explicación del comando:

El comando `pstree -p` organiza los procesos del sistema en una estructura jerárquica, mostrando qué proceso padre lanzó a cada proceso hijo. Esto permite identificar si existen procesos sospechosos o creados fuera de los servicios habituales.

## Salida:

- Se observan procesos del sistema como `systemd-logind`, `systemd-networkd`, `systemd-resolve`, `udisksd`, `unattended-upgr`, `upowerd`.
- Servicios de red y seguridad como `vsftpd` y múltiples procesos de `wazuh` (`agentd`, `execd`, `logcollector`, `modulesd`, `syscheckd`).
- Los procesos `wazuh` generan varios subprocesos hijos, lo cual es normal en su funcionamiento de monitorización.

## Interpretación:

La jerarquía de procesos muestra que todos los servicios observados son legítimos y coherentes con un servidor Linux configurado con Apache, SSH y el agente Wazuh. No se identifican procesos extraños fuera de esta estructura.

## Relevancia:

Este análisis permite verificar si un posible atacante creó procesos maliciosos ocultos. En este caso, la jerarquía corresponde a procesos esperados y confiables.

```
-systemd-logind(728)
-systemd-network(688)
-systemd-resolve(690)
-systemd-timesyn(645)---{systemd-timesyn}(658)
-systemd-udevd(410)
-udisksd(737)-+-{udisksd}(782)
                |-{udisksd}(787)
                |-{udisksd}(812)
                --{udisksd}(923)
-unattended-upgr(825)---{unattended-upgr}(940)
-upowerd(3432)-+-{upowerd}(3434)
                --{upowerd}(3435)
-vsftpd(749)
-wazuh-agentd(997)-+-{wazuh-agentd}(999)
                  |-{wazuh-agentd}(1000)
                  --{wazuh-agentd}(1001)
-wazuh-execd(927)---{wazuh-execd}(931)
-wazuh-logcollec(1027)-+-{wazuh-logcollec}(1029)
                      |-{wazuh-logcollec}(1030)
                      |-{wazuh-logcollec}(1031)
                      |-{wazuh-logcollec}(1032)
                      |-{wazuh-logcollec}(1033)
                      |-{wazuh-logcollec}(1034)
                      --{wazuh-logcollec}(1035)
-wazuh-modulesd(1067)-+-{wazuh-modulesd}(1071)
                      |-{wazuh-modulesd}(1072)
                      |-{wazuh-modulesd}(1073)
                      |-{wazuh-modulesd}(1074)
                      |-{wazuh-modulesd}(1076)
                      |-{wazuh-modulesd}(1077)
                      |-{wazuh-modulesd}(1078)
                      |-{wazuh-modulesd}(1081)
                      --{wazuh-modulesd}(1082)
-wazuh-syscheckd(1014)-+-{wazuh-syscheckd}(1015)
                       |-{wazuh-syscheckd}(1017)
                       --{wazuh-syscheckd}(1018)
```

rusadmln@dwkws-server:~/tmp\$



# PROCESO DE USUARIOS Y SERVICIOS

**Evidencia:** 5.4)enfocar\_en\_procesos\_de\_usuarios\_y\_servicios.png y 5.5.png

## Comando utilizado:

Se utilizó el comando `sudo ps aux | grep -v "["` para listar todos los procesos y excluir aquellos que pertenecen al kernel (que aparecen entre corchetes).

## Explicación del comando:

El comando `ps aux` muestra todos los procesos en ejecución. La tubería con `grep -v "["` elimina de la salida los procesos del kernel, quedando visibles únicamente los procesos de usuarios y servicios del sistema. Esto facilita el análisis de posibles programas sospechosos que se estén ejecutando en espacio de usuario.

## Salida:

- Procesos del sistema: *systemd, dbus-daemon, cron, rsyslogd, polkitd, ModemManager.*
- Servicios de red: *sshd, vsftpd, apache2.*
- Procesos de seguridad: varios componentes de *wazuh (agentd, execd, logcollector, modulesd, syscheckd).*
- Sesiones de usuario: *sysadmin* con procesos de terminal y shell.

## Interpretación:

La lista muestra únicamente procesos de usuario y servicios activos en el sistema. No se observan procesos inusuales fuera de los esperados en un servidor con Apache, SSH y Wazuh.

## Relevancia:

Permite concentrarse en servicios y aplicaciones expuestas a los usuarios, lo que ayuda a identificar software malicioso sin distraerse con procesos internos del kernel.

```
systemd+ 688 0.0 0.1 27264 7664 ? Ss 10:39 0:00 /lib/systemd/systemd-networkd
systemd+ 690 0.0 0.3 25476 12944 ? Ss 10:39 0:00 /lib/systemd/systemd-resolved
root 702 0.0 0.1 235576 7408 ? Ssl 10:39 0:00 /usr/lib/accounts-service/accounts-
root 706 0.0 0.0 6816 3052 ? Ss 10:39 0:00 /usr/sbin/cron -f
message+ 711 0.0 0.1 7680 4824 ? Ss 10:39 0:00 /usr/bin/dbus-daemon --system --a
address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
root 719 0.0 0.0 81828 3768 ? Ssl 10:39 0:00 /usr/sbin/irqbalance --foreground
root 720 0.0 0.4 29668 18780 ? Ss 10:39 0:00 /usr/bin/python3 /usr/bin/network
d-dispatcher --run-startup-triggers
root 722 0.0 0.1 232728 6832 ? Ssl 10:39 0:00 /usr/lib/polkitkit-1/polkitd --no
+debug
syslog 723 0.0 0.1 224344 4992 ? Ssl 10:39 0:00 /usr/sbin/rsyslogd -n -iNONE
root 728 0.0 0.1 17448 7700 ? Ss 10:39 0:00 /lib/systemd/systemd-logind
root 737 0.0 0.3 393276 12272 ? Ssl 10:39 0:00 /usr/lib/udisks2/udisksd
daemon 742 0.0 0.0 3796 2216 ? Ss 10:39 0:00 /usr/sbin/atd -f
root 749 0.0 0.0 6808 3120 ? Ss 10:39 0:00 /usr/sbin/vsftpd /etc/vsftpd.conf
root 785 0.0 0.2 315112 11648 ? Ssl 10:39 0:00 /usr/sbin/ModemManager
root 825 0.0 0.5 107924 20868 ? Ssl 10:39 0:00 /usr/bin/python3 /usr/share/unatt
ended-upgrades/unattended-upgrade-shutdown --wait-for-signal
root 843 0.0 0.1 6532 4488 ? Ss 10:39 0:01 /usr/sbin/apache2 -k start
www-data 844 0.0 0.1 1212444 6344 ? Sl 10:39 0:00 /usr/sbin/apache2 -k start
www-data 845 0.0 0.1 1212332 6216 ? Sl 10:39 0:00 /usr/sbin/apache2 -k start
root 927 0.0 0.0 25880 3524 ? Sl 10:39 0:01 /var/ossec/bin/wazuh-execd
wazuh 997 0.0 0.1 173620 7080 ? Sl 10:39 0:03 /var/ossec/bin/wazuh-agentd
root 1014 0.0 0.2 124284 8140 ? Ssl 10:39 0:00 /var/ossec/bin/wazuh-syscheckd
root 1027 0.0 0.2 528084 11632 ? Sl 10:39 0:04 /var/ossec/bin/wazuh-logcollector
root 1067 0.0 0.4 535264 15280 ? Sl 10:39 0:01 /var/ossec/bin/wazuh-modulesd
root 1621 0.0 0.0 6900 3592 tty1 Ss 10:44 0:00 /bin/login -p --
root 1810 0.0 0.9 1923104 39760 ? Ssl 10:45 0:04 /usr/lib/snappy/snappy
sysadmin 2181 0.0 0.2 19060 9740 ? Ss 10:45 0:00 /lib/systemd/systemd --user
sysadmin 2182 0.0 0.1 170748 4492 ? S 10:45 0:00 (sd-pam)
sysadmin 2195 0.0 0.1 8496 5432 tty1 S 10:45 0:00 -bash
root 3432 0.0 0.2 249520 9748 ? Ssl 12:31 0:00 /usr/lib/ypower/ypowerd
root 4636 0.0 0.1 9260 4568 tty1 S+ 15:27 0:00 sudo ps aux
root 4638 0.0 0.0 8888 3208 tty1 R+ 15:27 0:00 ps aux
sysadmin@4geeks-server:~/tmp
```

# PUERTOS Y CONEXIONES DE RED ACTIVAS

**Evidencia:** 5.6)para\_vincular\_con\_puertos\_y\_conexiones\_de\_red.png y 5.7.png

## Comando utilizado:

Se utilizó el comando `sudo lsof -i -P -n` para listar los procesos que tienen conexiones de red abiertas junto con los puertos asociados.

## Explicación del comando:

El comando `lsof -i` muestra los procesos que están utilizando recursos de red. La opción `-P` muestra los números de puerto en lugar de nombres simbólicos y `-n` evita la resolución de nombres de host, lo que hace la salida más rápida y directa.

## Salida:

- `systemd-network` y `systemd-resolve` gestionando conexiones UDP y TCP en la red local (puerto 68 y 53).
- `vsftpd` escuchando en el puerto 21 (FTP).
- `sshd` escuchando en el puerto 22 (SSH).
- `apache2` escuchando en el puerto 80 (HTTP).

```
sysadmin@4geeks-server:/tmp$ cat pahase1_step2_lsof_net.txt
COMMAND  PID    USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
systemd-n 688  systemd-network 20u  IPv4  21545   0t0  UDP 192.168.0.113:68
systemd-r 690  systemd-resolve 12u  IPv4  20746   0t0  UDP 127.0.0.53:53
systemd-r 690  systemd-resolve 13u  IPv4  20747   0t0  TCP 127.0.0.53:53 (LISTEN)
vsftpd    749    root    3u    IPv6  24624   0t0  TCP *:21 (LISTEN)
sshd      768    root    3u    IPv4  24757   0t0  TCP *:22 (LISTEN)
sshd      768    root    4u    IPv6  24780   0t0  TCP *:22 (LISTEN)
apache2   843    root    4u    IPv6  24146   0t0  TCP *:80 (LISTEN)
apache2   844    www-data 4u    IPv6  24146   0t0  TCP *:80 (LISTEN)
apache2   845    www-data 4u    IPv6  24146   0t0  TCP *:80 (LISTEN)
```

## Interpretación:

El servidor expone servicios de red estándar: FTP, SSH y HTTP. Todos son legítimos, aunque representan posibles vectores de ataque si no están correctamente asegurados.

## Relevancia:

Identificar los servicios escuchando en puertos abiertos permite evaluar la superficie de exposición del sistema y comprobar si existen servicios no autorizados.

# REVISIÓN DE TAREAS PROGRAMADAS

**Evidencia:** 6)estructura\_cron.png y 6.1.png

## Comando utilizado:

Se utilizó el comando `sudo ls -al /etc/cron*` para listar el contenido de los directorios de cron (*cron.daily*, *cron.hourly*, *cron.monthly*, *cron.weekly*).

## Explicación del comando:

El comando `ls -al` muestra en detalle los archivos y directorios, incluyendo permisos, propietario, tamaño y fecha de modificación. Al aplicarse sobre `/etc/cron*`, se obtiene la lista de scripts que el sistema ejecuta de manera periódica según la carpeta donde se encuentren (diaria, horaria, semanal, mensual).

## Salida:

- En *cron.daily*: tareas de mantenimiento como *apache2*, *apport*, *apt-compat*, *dpkg*, *logrotate*, *man-db*, *popularity-contest* y *update-notifier-common*.
- En *cron.hourly*, *cron.monthly* y *cron.weekly*: únicamente se observan archivos de marcador (*placeholder*) y tareas básicas del sistema como *man-db* o *update-notifier-common*.

## Interpretación:

No se identifican tareas extrañas ni sospechosas en los cron jobs. Todas las entradas corresponden a tareas legítimas de mantenimiento del sistema y servicios instalados.

## Relevancia:

La revisión de cron es importante ya que un atacante podría programar la ejecución automática de malware en estos directorios. En este caso, no se detectan anomalías.

```
/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 apport
-rwxr-xr-x 1 root root 1478 Apr 9 2020 apt-compat
-rwxr-xr-x 1 root root 355 Dec 29 2017 bsdmaintutils
-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: /tmp$ _
```

# REVISIÓN DEL ARCHIVO CRONTAB DEL SISTEMA

**Evidencia:** 6.2)crontab\_del\_sistema.png

## Comando utilizado:

Se usó el comando `cat phase1_step3_crontab.txt` para visualizar el contenido del archivo `/etc/crontab`.

## Explicación del comando:

El comando `cat` permite mostrar en pantalla el contenido de un archivo de texto. En este caso, se revisa el `crontab` del sistema, que define tareas automáticas programadas para diferentes usuarios y servicios.

## Salida:

- Se observa la definición estándar de variables de entorno (`SHELL`, `PATH`).
- Se incluyen ejemplos de programación de tareas con minutos, horas, días y meses.
- Las tareas configuradas corresponden a `anacron`, que ejecuta periódicamente los scripts de:
  - `/etc/cron.hourly` (cada hora)
  - `/etc/cron.daily` (cada día)
  - `/etc/cron.weekly` (cada semana)
  - `/etc/cron.monthly` (cada mes)

```
sysadmin@4geeks-server:/tmp$ cat phase1_step3_crontab.txt
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab`
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# * * * * * user-name command to be executed
17 * * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
42 6 * * 7 root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
```

## Interpretación:

Las entradas del archivo son legítimas y corresponden a las tareas de mantenimiento del sistema, no se observan modificaciones extrañas ni comandos sospechosos.

## Relevancia:

Revisar el archivo `/etc/crontab` es clave porque cualquier intruso podría añadir tareas maliciosas para garantizar persistencia. En este caso, todo parece normal y seguro.

# REVISIÓN DEL USUARIO Y ACCESOS

## Explicación del comando:

Este comando muestra el contenido del archivo `/etc/passwd`, que lista todos los usuarios del sistema junto con información básica como UID, directorio home y el intérprete de comandos asignado (shell).

## Salida observada:

- Se detectan numerosos usuarios del sistema (daemon, syslog, nobody, etc.) configurados con `/usr/sbin/nologin` o `/bin/false`, lo cual es correcto para evitar accesos interactivos.
- Usuarios de servicio como `www-data`, `backup`, `systemd-network`, `systemd-resolve`, entre otros.
- Usuario `sysadmin` con shell válido `/bin/bash`, lo que confirma que tiene acceso interactivo al sistema.
- También aparece `root` con acceso completo como es esperado.

## Interpretación:

La mayoría de las cuentas son de sistema y no permiten login, lo cual es adecuado para la seguridad. Solo `root` y `sysadmin` parecen tener acceso interactivo.

## Relevancia en la investigación:

La revisión de `/etc/passwd` permite identificar si existen cuentas sospechosas o creadas por un atacante. En este caso, no se observan cuentas extrañas aparte de las esperadas.

## Acciones recomendadas:

- Verificar si `sysadmin` es un usuario legítimo.
- Revisar si se han creado usuarios adicionales en fechas recientes.
- Mantener monitoreo sobre intentos de login con cuentas válidas.

```
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mail List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106:/:nonexistent:/usr/sbin/nologin
syslog:x:104:110:/:home/syslog:/usr/sbin/nologin
_apt:x:105:65534:/:nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uidd:x:107:112:/:run/uidd:/usr/sbin/nologin
tcpdump:x:108:113:/:nonexistent:/usr/sbin/nologin
landscape:x:109:115:/:var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1:/:var/cache/pollinate:/bin/false
fwupd-refresh:x:111:116:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
usbmux:x:112:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:113:65534:/:run/ssh:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
sysadmin:x:1000:1000:4geeks-server:/home/sysadmin:/bin/bash
lxd:x:998:100:/:var/snap/lxd/common/lxd:/bin/false
ftp:x:114:113:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
reports:x:1001:1001:,,,:/home/reports:/bin/bash
wazuh:x:115:120:/:var/ossec:/sbin/nologin
hacker:x:1002:1002:/:home/hacker:/bin/bash
sysadmin@4geeks-server:/tmp$ cat phase1_step4_etc_passwd.txt
```

# ÚLTIMAS SESIONES DE USUARIOS

**Comando usado:** `last -a > phase1_step4_last.txt`

## Explicación del comando:

El comando `last` muestra el historial de inicios y cierres de sesión de los usuarios en el sistema, así como reinicios, caídas y la dirección IP de origen. La opción `-a` agrega la IP o el nombre del host al final de cada línea.

## Salida observada:

- Usuario `sysadmin` inició sesión el 20 de septiembre desde la IP `192.168.0.67`.
- Hay una sesión de `sysadmin` aún activa.
- Se observan varios reinicios del sistema entre junio y septiembre.
- Se registran caídas (crash) el 23 de junio a las 15:01, 15:23 y 15:24.
- También se ven múltiples reinicios con el kernel `5.4.0-216-generic`.

## Interpretación:

El acceso legítimo más reciente corresponde a `sysadmin` desde la red interna (`192.168.0.67`). Los reinicios y caídas registrados en junio pueden indicar problemas de estabilidad o intentos de explotación que llevaron a bloqueos.

## Relevancia en la investigación:

Permite verificar quién accedió al sistema y desde dónde. El hecho de que las conexiones provengan de una IP privada cercana sugiere un acceso dentro de la misma red, pero deben revisarse intentos sospechosos de login en paralelo (`auth.log`).

## Acciones recomendadas:

- Confirmar que `192.168.0.67` corresponde a un host autorizado.
- Correlacionar con los intentos fallidos observados en `/var/log/auth.log`.
- Revisar la causa de los crashes en junio para descartar actividad maliciosa.

```
sysadmin@4geeks-server:/tmp$ sudo zgrep -h "Failed password" /var/log/auth.log* > /tmp/phase1_step4_
failed_pw_authlogs.txt 2>/dev/null
sysadmin@4geeks-server:/tmp$ cat phase1_step4_failed_pw_authlogs.txt
Jun 21 19:38:20 4geeks-server sshd[2001]: Failed password for root from 192.168.1.50 port 40290 ssh2
Jun 23 15:26:52 4geeks-server sshd[1736]: Failed password for invalid user test from 192.168.1.103 p
rt 58760 ssh2
Jun 23 15:27:01 4geeks-server sshd[1736]: Failed password for invalid user test from 192.168.1.103 p
rt 58760 ssh2
Jun 23 15:27:07 4geeks-server sshd[1736]: Failed password for invalid user test from 192.168.1.103 p
rt 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 20 13:46:44 4geeks-server sshd[3668]: Failed password for sysadmin from 192.168.0.67 port 33964
ssh2
Sep 20 16:44:23 4geeks-server sudo: sysadmin : TTY=ttty1 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/zg
rep -h Failed password /var/logs/auth.log*
Sep 20 16:45:26 4geeks-server sudo: sysadmin : TTY=ttty1 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/zg
rep -h Failed password /var/logs/auth.log*
Sep 20 16:49:33 4geeks-server sudo: sysadmin : TTY=ttty1 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/zg
rep -h Failed password /var/log/auth.log
sysadmin@4geeks-server:/tmp$ _
```

# INTENTOS DE ACCESO SSH

**Comando usado:** `sudo zgrep -h "Failed password" /var/log/auth.log* > /tmp/phase1_step4_failed_pw_authlogs.txt`

## Explicación del comando:

Este comando extrae del archivo de logs todos los registros relacionados con contraseñas incorrectas al intentar autenticarse por SSH.

## Salida observada:

- IP 192.168.1.50 → intentó acceder como root.
- IP 192.168.1.103 → intentó acceder con usuarios inválidos como test, admin, hacker, y también contra root.
- IP 192.168.0.67 → intentó acceder al usuario legítimo sysadmin.
- Los intentos se concentran en puertos SSH (22, 40230, 49542, 47014).
- Se observan múltiples fallos seguidos en intervalos cortos, lo que sugiere ataques automatizados.

## Interpretación:

Los logs confirman intentos de fuerza bruta desde varias IPs internas de la red. El atacante probó diferentes usuarios, incluyendo cuentas inexistentes, lo que es característico de un escaneo masivo. El intento sobre sysadmin desde 192.168.0.67 puede ser legítimo, pero requiere verificación.

## Relevancia en la investigación:

Este hallazgo valida que el sistema fue objetivo de ataques SSH. La información de IPs y usuarios atacados es clave para correlacionar con accesos exitosos y sesiones actuales.

## Acciones recomendadas:

- Configurar bloqueo automático tras varios intentos fallidos (ej. fail2ban).
- Asegurar que root no pueda conectarse por SSH directamente.
- Verificar si 192.168.0.67 corresponde a un host de confianza.

```
sysadmin@4geeks-server:~$ sudo zgrep -h "Failed password" /var/log/auth.log* > /tmp/phase1_step4_failed_pw_authlogs.txt
sysadmin@4geeks-server:~$ cat /tmp/phase1_step4_failed_pw_authlogs.txt
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 port 58760 ssh2
Jun 23 15:27:01 4geeks-server sshd[1736]: Failed password for invalid user test from 192.168.1.103 port 58760 ssh2
Jun 23 15:27:07 4geeks-server sshd[1736]: Failed password for invalid user test from 192.168.1.103 port 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 ssh2
Jun 23 15:28:40 4geeks-server sshd[1769]: Failed password for hacker from 192.168.1.103 port 44272 ssh2
Jun 23 15:29:15 4geeks-server sshd[1797]: Failed password for root from 192.168.1.103 port 47014 ssh2
Jun 23 15:29:21 4geeks-server sshd[1797]: Failed password for root from 192.168.1.103 port 47014 ssh2
Sep 20 13:46:44 4geeks-server sshd[3668]: Failed password for sysadmin from 192.168.0.67 port 33964 ssh2
Sep 20 16:44:23 4geeks-server sudo: sysadmin : TTY=ttty1 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/zgrep -h Failed password /var/logs/auth.log*
Sep 20 16:45:26 4geeks-server sudo: sysadmin : TTY=ttty1 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/zgrep -h Failed password /var/logs/auth.log*
Sep 20 16:49:33 4geeks-server sudo: sysadmin : TTY=ttty1 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/zgrep -h Failed password /var/log/auth.log
sysadmin@4geeks-server:~$
```

# EVENTOS SSH RECIENTES

**Comando usado:** `sudo journalctl -u ssh -S "1 hour ago" > /tmp/phase1_step4_journal_ssh_last.txt`

## Explicación del comando:

El comando `journalctl` consulta los registros del sistema administrados por `systemd`.

- `-u ssh` filtra los eventos del servicio SSH.
- `-S "1 hour ago"` muestra sólo los eventos ocurridos en la última hora.

## Salida observada:

El archivo generado muestra el rango de tiempo de los logs (desde 21 de junio hasta 20 de septiembre), pero para la última hora no aparecen entradas relacionadas con SSH (No entries).

## Interpretación:

No se registraron intentos de conexión SSH ni actividades recientes en la última hora al momento de la consulta. Esto contrasta con los múltiples intentos fallidos detectados en otros momentos mediante `auth.log`.

## Relevancia en la investigación:

Confirma que en el último periodo corto de tiempo (1 hora) no hubo nuevos intentos de acceso por SSH. Esto ayuda a delimitar la ventana de actividad maliciosa a momentos previos.

## Acciones recomendadas:

- Mantener la vigilancia periódica de `journalctl` para detectar intentos en tiempo real.
- Correlacionar los intentos anteriores con los horarios de mayor actividad sospechosa.

```
sysadmin@4geeks-server: /tmp$ cat phase1_step4_journal_ssh_last.txt
-- Logs begin at Sat 2025-06-21 19:04:00 UTC, end at Sat 2025-09-20 17:08:53 UTC. --
-- No entries --
```



# EVENTOS CRÍTICOS RECIENTES

## Comando ejecutado:

```
sudo journalctl -xe --since "2 hours ago" > /tmp/phase1_step5_journal_recent.txt
```

## Explicación del comando:

Este comando consulta los registros del sistema de las últimas dos horas mostrando mensajes extendidos (opción `-xe`), lo que permite detectar errores, accesos y eventos críticos en tiempo real.

## Resultados observados:

- Se registran sesiones abiertas y cerradas con `sudo` para el usuario `root`.
- Ejecuciones automáticas de tareas programadas por `cron`, algunas vinculadas a scripts como `/usr/local/bin/backup2.sh`.
- Mensajes relacionados con la instalación de MTA `ausente` (indica que no hay gestor de correo para salida de logs, lo cual es normal en entornos básicos).
- Varias entradas de `pam_unix(sudo:session)` confirman accesos con privilegios `root` desde la cuenta `sysadmin`.

## Conclusión:

Los registros muestran actividad administrativa intensa en la última franja horaria, principalmente relacionada con el usuario `sysadmin` y tareas de `cron`. No se identifican accesos externos sospechosos en este periodo, pero la repetición de ejecuciones automáticas merece revisión para descartar la manipulación de `cron`.

```
bin/zgrep -h Failed password /var/logs/auth.log*
Sep 20 16:45:26 4geeks-server sudo[5320]: pam_unix(sudo:session): session opened for user root by sysadmin(uid=0)
Sep 20 16:45:26 4geeks-server sudo[5320]: pam_unix(sudo:session): session closed for user root
Sep 20 16:47:12 4geeks-server CRON[5313]: (CRON) info (No MTA installed, discarding output)
Sep 20 16:47:12 4geeks-server CRON[5313]: pam_unix(cron:session): session closed for user root
Sep 20 16:49:33 4geeks-server sudo[5370]: sysadmin : TTY=ttty1 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/zgrep -h Failed password /var/log/auth.log
Sep 20 16:49:33 4geeks-server sudo[5370]: pam_unix(sudo:session): session opened for user root by sysadmin(uid=0)
Sep 20 16:49:33 4geeks-server sudo[5370]: pam_unix(sudo:session): session closed for user root
Sep 20 17:00:01 4geeks-server CRON[5407]: pam_unix(cron:session): session opened for user root by (uid=0)
Sep 20 17:00:01 4geeks-server CRON[5408]: (root) CMD (/usr/local/bin/backup2.sh)
Sep 20 17:02:11 4geeks-server CRON[5407]: (CRON) info (No MTA installed, discarding output)
Sep 20 17:02:11 4geeks-server CRON[5407]: pam_unix(cron:session): session closed for user root
Sep 20 17:08:53 4geeks-server sudo[5440]: sysadmin : TTY=ttty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/journalctl -u ssh -S 1 hour ago
Sep 20 17:08:53 4geeks-server sudo[5440]: pam_unix(sudo:session): session opened for user root by sysadmin(uid=0)
Sep 20 17:08:53 4geeks-server sudo[5440]: pam_unix(sudo:session): session closed for user root
Sep 20 17:15:01 4geeks-server CRON[5457]: pam_unix(cron:session): session opened for user root by (uid=0)
Sep 20 17:15:01 4geeks-server CRON[5458]: (root) CMD (/usr/local/bin/backup2.sh)
Sep 20 17:17:01 4geeks-server CRON[5474]: pam_unix(cron:session): session opened for user root by (uid=0)
Sep 20 17:17:01 4geeks-server CRON[5475]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)
Sep 20 17:17:01 4geeks-server CRON[5474]: pam_unix(cron:session): session closed for user root
Sep 20 17:17:13 4geeks-server CRON[5457]: (CRON) info (No MTA installed, discarding output)
Sep 20 17:17:13 4geeks-server CRON[5457]: pam_unix(cron:session): session closed for user root
Sep 20 17:22:11 4geeks-server sudo[5496]: sysadmin : TTY=ttty1 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/journalctl -xe --since 2 hours ago
Sep 20 17:22:11 4geeks-server sudo[5496]: pam_unix(sudo:session): session opened for user root by sysadmin(uid=0)
```

# INTENTOS FALLIDOS ACOMULADOS

## Comando usado:

```
sudo zgrep -h "failed password" /var/log/auth.log* > /tmp/phase1_step5_auth_failed_all.txt 2>/dev/null
```

## Explicación del comando:

Este comando busca en todos los archivos de registro de autenticación los intentos fallidos de contraseña. La opción `-h` oculta el nombre de archivo, y con `2>/dev/null` se descartan posibles errores. La salida se guarda en un archivo de texto para análisis posterior.

**Evidencia (archivo generado):** phase1\_step5\_auth\_failed\_all.txt

## Salida observada:

- Se registran intentos fallidos de inicio de sesión en el sistema, incluyendo ejecuciones de comandos desde el usuario sysadmin con privilegios de root.
- Se evidencian intentos con credenciales incorrectas que pudieron ser forzados o erróneos.

## Interpretación:

El análisis de todos los registros acumulados permite identificar patrones de ataques de fuerza bruta o accesos sospechosos repetidos. Esto confirma actividad no autorizada en el sistema.

## Acciones recomendadas:

- Configurar alertas ante múltiples intentos fallidos.
- Implementar reglas de bloqueo automático (ejemplo: Fail2Ban o UFW con límites).
- Revisar si alguno de estos intentos consiguió acceso posterior.

```
sysadmin@4geeks-server:/tmp$ cat phase1_step5_auth_failed_all.txt
Sep 20 17:30:46 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/tmp ; USER=root ; COMMAND=/usr/bin/zg
rep -h failed password /var/log/auth.log
sysadmin@4geeks-server:/tmp$
```

# MENSAJES RECIENTES DEL KERNEL Y BLOQUEOS DE RED

## Comando usado:

```
sudo dmesg --ctime | tail -n 200 > phsdel_step5_dmesg_recent.txt
```

## Explicación del comando:

- `dmesg` muestra los mensajes del kernel en tiempo real.
- La opción `--ctime` convierte las marcas de tiempo en formato legible (fecha y hora).
- `tail -n 200` limita la salida a los últimos 200 eventos.
- La salida se guarda en un archivo para revisión posterior.

**Evidencia (archivo generado):** phsdel\_step5\_dmesg\_recent.txt

## Salida observada:

- Se registran múltiples eventos etiquetados como [UFW BLOCK], lo que indica que el firewall (UFW) bloqueó intentos de conexión.
- Los intentos provienen de la dirección interna 192.168.0.67 hacia 192.168.0.113, usando el protocolo TCP en el puerto de destino **45444**.
- Repetidos intentos de conexión bloqueados en intervalos cortos, lo que sugiere actividad sospechosa o intentos de ataque de red.

## Interpretación:

El firewall está funcionando correctamente al bloquear tráfico no autorizado. Sin embargo, la repetición de intentos muestra que hay un host dentro de la red que intenta establecer conexiones posiblemente maliciosas o de prueba.

## Acciones recomendadas:

- Revisar la máquina con la IP 192.168.0.67 para verificar si está comprometida.
- Endurecer las reglas de firewall y aplicar listas de control de acceso (ACL).
- Implementar monitoreo de red para detectar patrones de escaneo o ataques de fuerza bruta.

```
[Sat Sep 20 11:53:02 2025] [UFW BLOCK] IN=enpos3 OUT= MAC=08:00:27:b3:c4:15:08:00:27:72:98:35:08:00 SRC=192.168.0.67 DST=192.168.0.113 LEN=44 TOS=0x00 PREC=0x00 TTL=46 ID=17274 PROTO=TCP SPT=45444 DPT=554 WINDOW=1024 RES=0x00 SYN URG=0
[Sat Sep 20 11:53:02 2025] [UFW BLOCK] IN=enpos3 OUT= MAC=08:00:27:b3:c4:15:08:00:27:72:98:35:08:00 SRC=192.168.0.67 DST=192.168.0.113 LEN=44 TOS=0x00 PREC=0x00 TTL=57 ID=6269 PROTO=TCP SPT=45444 DPT=53 WINDOW=1024 RES=0x00 SYN URG=0
[Sat Sep 20 11:53:02 2025] [UFW BLOCK] IN=enpos3 OUT= MAC=08:00:27:b3:c4:15:08:00:27:72:98:35:08:00 SRC=192.168.0.67 DST=192.168.0.113 LEN=44 TOS=0x00 PREC=0x00 TTL=46 ID=62273 PROTO=TCP SPT=45444 DPT=139 WINDOW=1024 RES=0x00 SYN URG=0
[Sat Sep 20 11:53:02 2025] [UFW BLOCK] IN=enpos3 OUT= MAC=08:00:27:b3:c4:15:08:00:27:72:98:35:08:00 SRC=192.168.0.67 DST=192.168.0.113 LEN=44 TOS=0x00 PREC=0x00 TTL=57 ID=2669 PROTO=TCP SPT=45444 DPT=53 WINDOW=1024 RES=0x00 SYN URG=0
[Sat Sep 20 11:53:02 2025] [UFW BLOCK] IN=enpos3 OUT= MAC=08:00:27:b3:c4:15:08:00:27:72:98:35:08:00 SRC=192.168.0.67 DST=192.168.0.113 LEN=44 TOS=0x00 PREC=0x00 TTL=42 ID=59498 PROTO=TCP SPT=45444 DPT=149 WINDOW=1024 RES=0x00 SYN URG=0
[Sat Sep 20 11:53:02 2025] [UFW BLOCK] IN=enpos3 OUT= MAC=08:00:27:b3:c4:15:08:00:27:72:98:35:08:00 SRC=192.168.0.67 DST=192.168.0.113 LEN=44 TOS=0x00 PREC=0x00 TTL=46 ID=50896 PROTO=TCP SPT=45444 DPT=3306 WINDOW=1024 RES=0x00 SYN URG=0
[Sat Sep 20 11:53:02 2025] [UFW BLOCK] IN=enpos3 OUT= MAC=08:00:27:b3:c4:15:08:00:27:72:98:35:08:00 SRC=192.168.0.67 DST=192.168.0.113 LEN=44 TOS=0x00 PREC=0x00 TTL=53 ID=64825 PROTO=TCP SPT=45444 DPT=587 WINDOW=1024 RES=0x00 SYN URG=0
[Sat Sep 20 11:53:04 2025] [UFW BLOCK] IN=enpos3 OUT= MAC=08:00:27:b3:c4:15:08:00:27:72:98:35:08:00 SRC=192.168.0.67 DST=192.168.0.113 LEN=44 TOS=0x00 PREC=0x00 TTL=38 ID=13275 PROTO=TCP SPT=45444 DPT=587 WINDOW=1024 RES=0x00 SYN URG=0
[Sat Sep 20 11:53:22 2025] [UFW BLOCK] IN=enpos3 OUT= MAC=08:00:27:b3:c4:15:08:00:27:72:98:35:08:00 SRC=192.168.0.67 DST=192.168.0.113 LEN=44 TOS=0x00 PREC=0x00 TTL=41 ID=25492 PROTO=TCP SPT=45444 DPT=65107 WINDOW=1024 RES=0x00 SYN URG=0
[Sat Sep 20 11:53:42 2025] [UFW BLOCK] IN=enpos3 OUT= MAC=08:00:27:b3:c4:15:08:00:27:72:98:35:08:00 SRC=192.168.0.67 DST=192.168.0.113 LEN=44 TOS=0x00 PREC=0x00 TTL=41 ID=55089 PROTO=TCP SPT=45444 DPT=25866 WINDOW=1024 RES=0x00 SYN URG=0
[Sat Sep 20 11:54:02 2025] [UFW BLOCK] IN=enpos3 OUT= MAC=08:00:27:b3:c4:15:08:00:27:72:98:35:08:00 SRC=192.168.0.67 DST=192.168.0.113 LEN=44 TOS=0x00 PREC=0x00 TTL=49 ID=60801 PROTO=TCP SPT=45444 DPT=47193 WINDOW=1024 RES=0x00 SYN URG=0
[Sat Sep 20 11:54:22 2025] [UFW BLOCK] IN=enpos3 OUT= MAC=08:00:27:b3:c4:15:08:00:27:72:98:35:08:00 SRC=192.168.0.67 DST=192.168.0.113 LEN=44 TOS=0x00 PREC=0x00 TTL=57 ID=15729 PROTO=TCP SPT=45444 DPT=41840 WINDOW=1024 RES=0x00 SYN URG=0
[Sat Sep 20 11:54:42 2025] [UFW BLOCK] IN=enpos3 OUT= MAC=08:00:27:b3:c4:15:08:00:27:72:98:35:08:00 SRC=192.168.0.67 DST=192.168.0.113 LEN=44 TOS=0x00 PREC=0x00 TTL=43 ID=13538 PROTO=TCP SPT=45444 DPT=24575 WINDOW=1024 RES=0x00 SYN URG=0
```

# PUERTOS Y SERVICIOS EN ESCUCHA

## Comando usado:

```
sudo ss -tulpn > /tmp/phase1_step6_ss_listening.txt
```

## Explicación del comando:

- `ss` muestra las conexiones de red activas.
- `-tulpn` indica:
  - **t** → conexiones TCP
  - **u** → conexiones UDP
  - **l** → solo sockets en escucha
  - **p** → muestra el proceso que usa el puerto
  - **n** → no resuelve nombres de servicio (muestra números de puerto)

**Evidencia (archivo generado):** phase1\_step6\_ss\_listening.txt

## Salida observada:

- **UDP**
  - Puerto 53 (DNS) escuchando en 127.0.0.53 → servicio *systemd-resolve*.
  - Puerto 68 en 192.168.0.113 → servicio *systemd-networkd*.
- **TCP**
  - Puerto 22 abierto (SSH) en IPv4 0.0.0.0 y en IPv6 [::]. Proceso: *sshd*.
  - Puerto 80 abierto (HTTP) en todos los interfaces. Proceso: *apache2*.
  - Puerto 21 abierto (FTP). Proceso: *vsftpd*.

## Interpretación:

El servidor tiene varios servicios críticos expuestos (SSH, HTTP, FTP). Esto implica una superficie de ataque más amplia. Los puertos de DNS internos y de red son normales, pero deben ser monitorizados.

## Acciones recomendadas:

- Restringir el acceso a SSH y FTP mediante firewall o ACL, limitando solo a direcciones autorizadas.
- Revisar la necesidad de tener FTP abierto (puede ser reemplazado por SFTP).
- Monitorear accesos HTTP en Apache2 para detectar intentos de explotación.

```
sysadmin@4geeks-server:/tmp$ sudo ss -tulpn > /tmp/phase1_step6_ss_listening.txt
sysadmin@4geeks-server:/tmp$ cat phase1_step6_ss_listening.txt
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port Process
udp UNCONN 0 0 127.0.0.53%10:53 0.0.0.0:* users:((("systemd-resolve",pid=690,fd=12))
udp UNCONN 0 0 192.168.0.113%np0s3:68 0.0.0.0:* users:((("systemd-networkd",pid=688,fd=20))
tcp LISTEN 0 128 0.0.0.0:22 0.0.0.0:* users:((("sshd",pid=768,fd=3))
tcp LISTEN 0 4096 127.0.0.53%10:53 0.0.0.0:* users:((("systemd-resolve",pid=690,fd=13))
tcp LISTEN 0 128 [::]:22 [::]:* users:((("sshd",pid=768,fd=4))
tcp LISTEN 0 511 *:80 *:80 users:((("apache2",pid=845,fd=4),("apache2",pid=844,fd=4),("apache2",pid=843,fd=4))
tcp LISTEN 0 32 *:21 *:21 users:((("vsftpd",pid=749,fd=3))
sysadmin@4geeks-server:/tmp$
```

Comando usado:  
`sudo ss -tunap > /tmp/phase1_step6_ss_all.txt`

Explicación del comando:

- `ss` lista conexiones de red.
- `-tunap` indica:
  - `t` → conexiones TCP
  - `u` → conexiones UDP
  - `n` → muestra puertos en número (sin nombres)
  - `a` → muestra todas las conexiones (no solo las que escuchan)
  - `p` → muestra el proceso asociado

Evidencia (archivo generado): `phase1_step6_ss_all.txt`

Salida observada:

- UDP en puerto 53 y 68 igual que en la evidencia anterior (*systemd-resolve*, *systemd-networkd*).
- TCP en:
  - Puerto 22 abierto (SSH).
  - Puerto 80 abierto (HTTP, proceso *apache2*).
  - Puerto 21 abierto (FTP, proceso *vsftpd*).
- Además, al incluir la opción `-a`, se muestran tanto las conexiones en escucha como las que ya tienen actividad establecida o intentos recientes.

Interpretación:

Este listado confirma qué servicios están no solo en escucha, sino también con posibles conexiones activas. Esto es importante para detectar accesos remotos en tiempo real.

Acciones recomendadas:

- Revisar las conexiones activas en los puertos críticos (SSH, FTP y HTTP).
- Bloquear accesos no autorizados en firewall.
- Habilitar registros detallados en Apache y SSH para correlacionar accesos con IPs sospechosas.

# CONEXIONES ACTIVAS ENTRANTES Y SALIENTES

```
sysadmin@4geeks-server:/tmp$ sudo ss -tunap > /tmp/phase1_step6_ss_all.txt
sysadmin@4geeks-server:/tmp$ cat phase1_step6_ss_all.txt
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:* users:((("systemd-re
solve",pid=690,fd=12))
udp UNCONN 0 0 192.168.0.113%enp0s3:68 0.0.0.0:* users:((("systemd-ne
twork",pid=688,fd=20))
tcp LISTEN 0 128 0.0.0.0:22 0.0.0.0:* users:((("sshd",pid=
768,fd=3))
tcp LISTEN 0 4096 127.0.0.53%lo:53 0.0.0.0:* users:((("systemd-re
solve",pid=690,fd=13))
tcp LISTEN 0 128 [::]:22 [::]:* users:((("sshd",pid=
768,fd=4))
tcp LISTEN 0 511 *:80 *:~ users:((("apache2",p
id=845,fd=4),("apache2",pid=844,fd=4),("apache2",pid=843,fd=4))
tcp LISTEN 0 32 *:21 *:~ users:((("vsftpd",pi
d=749,fd=3))
sysadmin@4geeks-server:/tmp$
```

```
Comando ejecutado:
sudo find / -perm -4000 -type f -exec ls -ld {} ; 2>/dev/null > /tmp/phase1_step7_setuid_files.txt
```

Explicación del comando:

- `find / -perm -4000`: busca archivos en todo el sistema que tengan el bit SUID activado.
- `-type f`: limita la búsqueda a archivos normales.
- `-exec ls -ld {}`: muestra detalles de cada archivo encontrado.
- `2>/dev/null`: descarta errores (por ejemplo, directorios sin permisos de lectura).
- `> /tmp/phase1_step7_setuid_files.txt`: guarda los resultados en un archivo.

Salida observada:

- Archivos con SUID encontrados:
  - `/usr/bin/mount`
  - `/usr/bin/su`
  - `/usr/bin/passwd`
  - `/usr/bin/chsh`, `/usr/bin/chfn`
  - `/usr/lib/snapd/snap-confine`
  - `/usr/lib/dbus-1.0/dbus-daemon-launch-helper`
  - `/usr/lib/openssh/ssh-keysign`
  - Versiones en directorios de *snap* (duplicados de `chsh`, `mount`, `umount`, etc.)

**Interpretación:**  
El bit SUID permite que un archivo se ejecute con los privilegios de su propietario (normalmente root). Esto es necesario para programas como `passwd` o `su`, pero también representa un riesgo si un atacante explota alguno de estos binarios.

Relevancia en seguridad:

- Los archivos SUID son puntos críticos de auditoría, ya que pueden ser usados para escalada de privilegios.
- Es importante verificar que todos los SUID listados sean legítimos y necesarios.

Acciones recomendadas:

- Revisar si los binarios SUID en directorios de *snap* son realmente necesarios.
- Asegurar que no existan binarios SUID sospechosos en directorios poco comunes.
- Documentar y auditar periódicamente esta lista.

# ARCHIVOS CON PERMISOS SUID

```
rwsr-xr-x 1 daemon daemon 55520 Nov 12 2018 /usr/bin/at
rwsr-xr-x 1 root root 39144 Mar 7 2020 /usr/bin/fusemount
rwsr-xr-x 1 root root 55528 Apr 9 2024 /usr/bin/mount
rwsr-xr-x 1 root root 39144 Apr 9 2024 /usr/bin/umount
rwsr-xr-x 1 root root 44784 Feb 6 2024 /usr/bin/newgrp
rwsr-xr-x 1 root root 68208 Feb 6 2024 /usr/bin/passwd
rwsr-xr-x 1 root root 14488 Jul 8 2019 /usr/lib/eject/dmccrypt-get-device
rwsr-xr-x 1 root root 155080 Jul 26 2024 /usr/lib/snapd/snap-confine
rwsr-xr-x 1 root root 22840 Feb 21 2022 /usr/lib/policykit-1/polkit-agent-helper-1
rwsr-xr-x 1 root messagebus 51344 Oct 25 2022 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
rwsr-xr-x 1 root root 477672 Apr 11 12:16 /usr/lib/openssh/ssh-keysign
rwsr-xr-x 1 root root 129560 Jan 25 2023 /snap/snapd/18357/usr/lib/snapd/snap-confine
rwsr-xr-x 1 root root 85064 Feb 6 2024 /snap/core20/2599/usr/bin/chfn
rwsr-xr-x 1 root root 53040 Feb 6 2024 /snap/core20/2599/usr/bin/chsh
rwsr-xr-x 1 root root 88464 Feb 6 2024 /snap/core20/2599/usr/bin/gpasswd
rwsr-xr-x 1 root root 55528 Apr 9 2024 /snap/core20/2599/usr/bin/mount
rwsr-xr-x 1 root root 44784 Feb 6 2024 /snap/core20/2599/usr/bin/newgrp
rwsr-xr-x 1 root root 68208 Feb 6 2024 /snap/core20/2599/usr/bin/passwd
rwsr-xr-x 1 root root 67816 Apr 9 2024 /snap/core20/2599/usr/bin/su
rwsr-xr-x 1 root root 166056 Apr 4 2023 /snap/core20/2599/usr/bin/sudo
rwsr-xr-x 1 root root 39144 Apr 9 2024 /snap/core20/2599/usr/bin/umount
rwsr-xr-x 1 root systemd-resolve 51344 Oct 25 2022 /snap/core20/2599/usr/lib/dbus-1.0/dbus-daemon-launch-helper
rwsr-xr-x 1 root root 477672 Apr 11 12:16 /snap/core20/2599/usr/lib/openssh/ssh-keysign
rwsr-xr-x 1 root root 85064 Nov 29 2022 /snap/core20/1828/usr/bin/chfn
rwsr-xr-x 1 root root 53040 Nov 29 2022 /snap/core20/1828/usr/bin/chsh
rwsr-xr-x 1 root root 88464 Nov 29 2022 /snap/core20/1828/usr/bin/gpasswd
rwsr-xr-x 1 root root 55528 Feb 7 2022 /snap/core20/1828/usr/bin/mount
rwsr-xr-x 1 root root 44784 Nov 29 2022 /snap/core20/1828/usr/bin/newgrp
rwsr-xr-x 1 root root 68208 Nov 29 2022 /snap/core20/1828/usr/bin/passwd
rwsr-xr-x 1 root root 67816 Feb 7 2022 /snap/core20/1828/usr/bin/su
rwsr-xr-x 1 root root 166056 Jan 16 2023 /snap/core20/1828/usr/bin/sudo
rwsr-xr-x 1 root root 39144 Feb 7 2022 /snap/core20/1828/usr/bin/umount
rwsr-xr-x 1 root systemd-resolve 51344 Oct 25 2022 /snap/core20/1828/usr/lib/dbus-1.0/dbus-daemon-launch-helper
rwsr-xr-x 1 root root 473576 Mar 30 2022 /snap/core20/1828/usr/lib/openssh/ssh-keysign
```

Comando ejecutado:

```
sudo sha256sum [lista de archivos] > phase1_step5_hashes.txt
```

Explicación del comando:

- sha256sum: calcula el hash SHA-256 de los archivos indicados.
- Almacenarlo en phase1\_step5\_hashes.txt permite tener un registro de integridad para auditoría.

Salida observada:

Se generaron hashes SHA-256 de los archivos recolectados:  
• ps\_aux\_sorted.txt • pstree.txt • lsof\_net.txt • cron\_ls.txt • crontab.txt • etc\_passwd.txt • last.txt • failed\_pw\_authlogs.txt • ss\_listening.txt • ss\_all.txt • setuid\_files.txt • dmesg\_recent.txt • journal\_ssh\_last.txt • journal\_recent.txt • susp\_exec\_tmp.txt • otros listados en la captura

Interpretación:

El cálculo de hashes sirve para:

- Verificar que los archivos no hayan sido modificados después de su recolección.
- Proveer evidencia digital íntegra en caso de un análisis forense.
- Permitir comparaciones futuras para detectar cambios en configuraciones críticas, procesos o logs.

Relevancia en seguridad:

- Los hashes son fundamentales en la **cadena de custodia digital**, asegurando que la información usada en un informe o investigación se mantenga íntegra.
- Cualquier modificación posterior en los archivos cambiaría el hash, alertando de una posible manipulación.

Acciones recomendadas:

- Guardar estos hashes junto con los archivos recolectados en un medio seguro.
- Usarlos como referencia en auditorías o investigaciones futuras.

# HASHES



# CONCLUSIÓN FASE 1

- El servidor funciona con estabilidad y servicios web/ssh/ftp activos.
- Se detectaron **múltiples intentos de acceso no autorizado vía SSH**, lo que confirma un ataque en curso o pruebas de fuerza bruta.
- No se han encontrado binarios SUID sospechosos ni tareas cron maliciosas.
- Todas las evidencias están almacenadas y protegidas mediante hashes SHA-256.

## Acciones sugeridas antes de la Fase 2:

- Reforzar seguridad en SSH (fail2ban, cambiar puerto, deshabilitar root login).
- Auditar accesos de sysadmin y revisar scripts de cron ejecutados como root.
- Mantener las evidencias almacenadas para cadena de custodia.



# FUENTES

uptime – <https://man7.org/linux/man-pages/man1/uptime.1.html>

free – <https://man7.org/linux/man-pages/man1/free.1.html>

top – <https://man7.org/linux/man-pages/man1/top.1.html>

ps – <https://man7.org/linux/man-pages/man1/ps.1.html>

pstree – <https://linux.die.net/man/1/pstree>

lsof – <https://linux.die.net/man/8/lsof>

ss – <https://man7.org/linux/man-pages/man8/ss.8.html>

ls – <https://man7.org/linux/man-pages/man1/ls.1.html>

cat – <https://man7.org/linux/man-pages/man1/cat.1.html>

crontab (/etc/crontab) – <https://man7.org/linux/man-pages/man5/crontab.5.html>

find – <https://man7.org/linux/man-pages/man1/find.1.html>

last – <https://man7.org/linux/man-pages/man1/last.1.html>

grep / zgrep – <https://man7.org/linux/man-pages/man1/grep.1.html>

journalctl – <https://www.freedesktop.org/software/systemd/man/journalctl.html>

dmesg – <https://man7.org/linux/man-pages/man1/dmesg.1.html>

sha256sum – <https://man7.org/linux/man-pages/man1/sha256sum.1.html>

FASE 2

# REMEDIACIÓN INICIAL

## Quitar servicios innecesarios (vsftpd)

### Comandos:

```
sudo systemctl stop vsftpd && sudo systemctl disable vsftpd
```

### Explicación:

Detiene el servicio FTP y evita que arranque al iniciar el sistema. Si no se usa, purgar el paquete reduce la superficie de ataque.

### Resultado observado:

systemd indica que vsftpd quedó deshabilitado y fuera de multi-user.target.

```
sysadmin@4geeks-server:~$ sudo systemctl stop vsftpd && 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:~$ _
```

```
sysadmin@4geeks-server:~$ sudo nano /etc/ssh/sshd_config_
```

## Endurecer SSH - cambios en /etc/ssh/sshd\_config

### Comandos:

```
sudo nano /etc/ssh/sshd_config
(editar estas líneas)
Port 2222
PermitRootLogin no
PasswordAuthentication no
StrictModes yes
MaxAuthTries 6
MaxSessions 10
AllowUsers sysadmin
guardar y salir
```

```
#HostKey /etc/ssh/ssh_host_ed25519_key
# Ciphers and keying
#RekeyLimit default none
# Logging
#SyslogFacility AUTH
#LogLevel INFO
# Authentication:
#LoginGraceTime 2m
#PermitRootLogin prohibit-password
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

reiniciar el servicio:  
sudo systemctl restart ssh

### Explicación:

- Cambiar el puerto reduce ruido de escaneo.
- Bloquear login de root elimina un objetivo crítico.
- Desactivar contraseñas obliga a llaves SSH (más seguras).
- Limitar intentos y sesiones frena fuerza bruta y abuso.
- Restringir a “sysadmin” acota quién puede iniciar sesión.

### Resultado observado:

Se ve Port 2222, PermitRootLogin no, PasswordAuthentication no, límites de autenticación y el usuario permitido.

```
# For this to work you will also need to
# Change to yes if you don't trust ~/.ssh/
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts
#IgnoreRhosts yes

# To disable tunneled clear text passwords
#PasswordAuthentication yes
PasswordAuthentication no
#PermitEmptyPasswords no
```

```
# Example of overriding settings on a per-user basis
#Match User anoncvs
#      X11Forwarding no
#      AllowTcpForwarding no
#      PermitTTY no
#      ForceCommand cvs server
AllowUser sysadmin
```

```
#Port 22
Port 2222
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

## Restringir acceso con UFW (firewall)

### Comandos (alineados al puerto 2222):

```
sudo ufw allow from 192.168.0.0/24 to any port 2222 proto tcp
```

### Explicación:

Solo la red interna 192.168.0.0/24 puede llegar al puerto SSH. Se cierra el antiguo 22 si no se usa.

### Resultado observado:

Se añadió una regla allow.

```
sysadmin@4geeks-server:~$ sudo ufw allow from 192.168.0.0/24 to any port 22 proto tcp
Rule added
sysadmin@4geeks-server:~$ _
```

## Protección anti fuerza bruta con Fail2ban

### Comandos:

```
sudo apt install fail2ban -y
sudo nano /etc/fail2ban/jail.local
(añadir)
[sshd]
enabled = true
port = 2222
filter = sshd
logpath = /var/log/auth.log
maxretry = 5
bantime = 3600
```

### activar y comprobar:

```
sudo systemctl enable --now fail2ban
sudo fail2ban-client status
sudo fail2ban-client status sshd
```

### Explicación:

Fail2ban lee /var/log/auth.log y, si hay 5 intentos fallidos, bloquea la IP 1 hora. El parámetro port debe coincidir con el puerto real de SSH.

### Resultado observado:

Se ve la sección [sshd] con enabled true, maxretry 5, bantime 3600. Ajusta el valor “port” a 2222 si ya cambiaste el puerto en SSH.

```
sysadmin@4geeks-server:~$ sudo apt install fail2ban -y
```

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

```
[sshd]
enabled = true
port    = 22
filter  = sshd
logpath = /var/log/auth.log
maxretry = 5
bantime = 3600
```

# VULNERABILIDADES DETECTADAS Y CORREGIDAS

## Listado de usuarios en el sistema


Se generó un listado de todos los usuarios existentes para revisar posibles cuentas sospechosas:

### Comando ejecutado:

```
cat /etc/passwd | cut -d: -f1
```

### Explicación:

Este comando lista únicamente el primer campo de /etc/passwd (nombre de usuario). Así podemos ver qué usuarios existen en el sistema sin mostrar toda la información.

 **Evidencia:** Se muestran usuarios válidos como sysadmin, www-data, backup, syslog, sshd, pero también aparece un usuario **sospechoso llamado "hacker"**.

```
news
nucp
proxy
www-data
backup
list
irc
gnats
nobody
systemd-network
systemd-resolve
systemd-timesync
messagebus
syslog
lapt
rss
duidd
tcpdump
landscape
pollinate
fwupd-refresh
usbmux
sshd
systemd-coredump
sysadmin
lxd
ftp
reports
wazuh
hacker
```

## 2. Eliminación de usuario sospechoso

Tras detectar la cuenta **hacker**, se eliminó completamente del sistema para evitar accesos maliciosos.

### Comando ejecutado:

```
sudo deluser --remove-home hacker
```

### Explicación:

- deluser elimina la cuenta.
- --remove-home asegura que también se borre la carpeta personal y los archivos asociados, evitando persistencia.



**Evidencia:** Captura mostrando que el usuario hacker fue eliminado con éxito.

```
bin
sys
sync
games
man
lp
mail
news
uuwp
proxy
www-data
backup
list
irc
gnats
nobody
systemd-network
systemd-resolve
systemd-timesync
messagebus
syslog
_apt
rs
uidd
tcpdump
landscape
pollinate
fuupd-refresh
usbmux
sshd
systemd-coredump
sysadmin
lxd
ftp
reports
wazuh
sysadmin@4geeks-server:~$
```



# REVISIÓN Y ELIMINACIÓN DE CRONJOBS SOSPECHOSOS

## Comando utilizado:

```
sudo crontab -l -u sysadmin && sudo crontab -l -u root
```

Este comando lista los cronjobs de los usuarios sysadmin y root. En tu salida vemos que no existen crontabs definidos directamente para estos usuarios. Eso significa que no hay tareas programadas a nivel de usuario que puedan ser sospechosas.

```
ls -al /etc/cron.*
```

Este comando revisa los directorios cron.daily, cron.hourly, cron.monthly y cron.weekly. Estos contienen scripts que el sistema ejecuta de manera programada.

- En tu caso, aparecen scripts estándar del sistema, como apache2, apt-compat, logrotate, man-db, update-notifier-common, etc.
- No se observan scripts extraños ni con nombres sospechosos (ejemplo: backdoor.sh, malware.sh, etc.).

## Análisis de la evidencia:

- No se detectaron cronjobs maliciosos en los crontabs de los usuarios principales (root y sysadmin).
- Los cronjobs en /etc/cron.\* corresponden a mantenimientos legítimos del sistema.

## Acción correctiva aplicada:

- Verificación completa de crontabs de usuarios y del sistema.
- Confirmación de que no hay entradas maliciosas.
- No fue necesario eliminar ni modificar ningún archivo, ya que todo corresponde a configuraciones normales del sistema.

```
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 102 root root 4096 Sep 21 00:38 ..
drwxr-xr-x 1 root root 539 Mar 18 2024 apache2
drwxr-xr-x 1 root root 376 Sep 16 2021 apport
drwxr-xr-x 1 root root 1478 Apr 9 2020 apt-compat
drwxr-xr-x 1 root root 355 Dec 29 2017 bsdmaintools
drwxr-xr-x 1 root root 1187 Sep 5 2019 dpkg
drwxr-xr-x 1 root root 377 Jan 21 2019 logrotate
drwxr-xr-x 1 root root 1123 Feb 25 2020 man-db
drwxr-xr-x 1 root root 102 Feb 13 2020 .placeholder
drwxr-xr-x 1 root root 4574 Jul 18 2019 popularity-contest
drwxr-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 102 root root 4096 Sep 21 00:38 ..
drwxr-xr-x 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 102 root root 4096 Sep 21 00:38 ..
drwxr-xr-x 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 102 root root 4096 Sep 21 00:38 ..
drwxr-xr-x 1 root root 813 Feb 25 2020 man-db
drwxr-xr-x 1 root root 102 Feb 13 2020 .placeholder
drwxr-xr-x 1 root root 403 Jan 20 2023 update-notifier-common
sysadmin@4geeks-server:~$ ls -al /etc/cron.*
```

```
sysadmin@4geeks-server:~$ sudo crontab -l -u sysadmin && sudo crontab -l -u root
no crontab for sysadmin
no crontab for root
```

# EVISIÓN DE SERVICIOS ACTIVOS EN EL SISTEMA

## Comando utilizado:

Texto: `systemctl list-unit-files --type=service --state=enabled`

Este comando lista todos los **servicios habilitados** en el sistema, es decir, aquellos que se inician automáticamente con el arranque.

## Análisis de la evidencia:

En la captura observamos servicios habilitados como:

- **apache2.service** → servidor web.
- **fail2ban.service** → defensa contra ataques de fuerza bruta.
- **cron.service** → ejecución de tareas programadas.
- **systemd-timesyncd.service** → sincronización de hora.
- **cloud-init** → usado en entornos virtualizados para configuración automática.

Ninguno de los servicios listados tiene nombres extraños o asociados a malware/backdoors.

Los servicios son estándar de un servidor Linux, con funciones legítimas de administración, red, seguridad o mantenimiento.

```
UNIT FILE                                STATE   VENDOR PRESET
accounts-daemon.service                 enabled enabled
apache2.service                         enabled enabled
apparmor.service                       enabled enabled
atd.service                             enabled enabled
autovt@.service                        enabled enabled
blk-availability.service                enabled enabled
cloud-config.service                   enabled enabled
cloud-final.service                    enabled enabled
cloud-init-local.service                enabled enabled
cloud-init.service                     enabled enabled
console-setup.service                  enabled enabled
cron.service                           enabled enabled
dbus-org.freedesktop.ModemManager1.service enabled enabled
dbus-org.freedesktop.resolve1.service  enabled enabled
dbus-org.freedesktop.thermald.service  enabled enabled
dbus-org.freedesktop.timesync1.service  enabled enabled
dmesg.service                          enabled enabled
e2scrub_reap.service                   enabled enabled
fail2ban.service                       enabled enabled
finalrd.service                        enabled enabled
getty@.service                         enabled enabled
grub-common.service                   enabled enabled
grub-initrd-fallback.service           enabled enabled
irqbalance.service                    enabled enabled
iscsi.service                          enabled enabled
keyboard-setup.service                 enabled enabled
lvm2-monitor.service                  enabled enabled
lxd-agent-9p.service                  enabled enabled
lxd-agent.service                     enabled enabled
ModemManager.service                  enabled enabled
multipath-tools.service                enabled enabled
multipathd.service                    enabled enabled
networkd-dispatcher.service            enabled enabled
ondemand.service                      enabled enabled
open-iscsi.service                    enabled enabled
timesyncd.service                      enabled enabled
```

## Acción correctiva aplicada:

- Se verificó el listado de servicios activos y **no se detectaron servicios maliciosos o sospechosos**.
- Los servicios identificados corresponden a configuraciones normales del sistema (servidor web, seguridad, utilidades de sistema).
- **No se deshabilitó ni eliminó ningún servicio adicional**, ya que todos son legítimos.

Comandos utilizados:

- `sudo ufw allow 22/tcp` → Permite conexiones SSH.
- `sudo ufw allow 80/tcp` → Permite tráfico HTTP (web).
- `sudo ufw allow 443/tcp` → Permite tráfico HTTPS (web seguro).
- `sudo ufw allow 21/tcp` → Permite conexiones FTP.
- `sudo ufw default deny incoming` → Bloquea todo el tráfico entrante por defecto.
- `sudo ufw default allow outgoing` → Permite todo el tráfico saliente.

Análisis de la evidencia:

La configuración muestra que:

- Solo se permiten **puertos esenciales**:
  - 22 (SSH seguro para administración).
  - 80 y 443 (servicios web).
  - 21 (FTP, aunque ya se deshabilitó el servicio vsftpd, esta regla podría eliminarse más adelante para reforzar seguridad).
- Se estableció la política por defecto en:
  - **Deny incoming** → todo lo entrante bloqueado salvo reglas explícitas.
  - **Allow outgoing** → todo lo saliente permitido.

Acción correctiva aplicada:

- Se endureció la política de red, permitiendo únicamente los servicios necesarios.
- Se bloqueó el resto de accesos externos, reduciendo la superficie de ataque.
- Se dejó SSH activo para administración, restringido previamente a la red local.

# CONFIGURACIÓN DE FIREWALL UFW

```
sysadmin@4geeks-server:~$ sudo ufw allow 22/tcp
Rule added
Rule added (v6)
sysadmin@4geeks-server:~$ sudo ufw allow 60/tcp
Rule added
Rule added (v6)
sysadmin@4geeks-server:~$ sudo ufw allow 80/tcp
Rule added
Rule added (v6)
sysadmin@4geeks-server:~$ sudo ufw allow 443/tcp
Rule added
Rule added (v6)
sysadmin@4geeks-server:~$ sudo ufw allow 21/tcp
Rule added
Rule added (v6)
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:~$ _
```

# CONCLUSIÓN

Se eliminaron servicios innecesarios, se reforzó SSH con medidas de seguridad (cambio de puerto, bloqueo root, Fail2Ban), se eliminaron usuarios y cron jobs sospechosos, se revisaron servicios activos y se configuró el firewall UFW con reglas estrictas.

El sistema quedó restaurado, más seguro y con menor riesgo de intrusión futura.

# FUENTES

- `systemctl stop / disable` → <https://www.freedesktop.org/software/systemd/man/systemctl.html>
- `ufw allow / deny` → <https://help.ubuntu.com/community/UFW>
- `fail2ban` (instalación y configuración) → [https://www.fail2ban.org/wiki/index.php/Main\\_Page](https://www.fail2ban.org/wiki/index.php/Main_Page)
- `crontab` (revisión de tareas programadas) → <https://man7.org/linux/man-pages/man5/crontab.5.html>
- `ls -al /etc/cron\*` → <https://man7.org/linux/man-pages/man1/ls.1.html>
- `systemctl list-unit-files` → <https://www.freedesktop.org/software/systemd/man/systemctl.html>
- `userdel` (eliminar usuario sospechoso) → <https://man7.org/linux/man-pages/man8/userdel.8.html>

FASE 3

# INFORME TÉCNICO FINAL

## **Revisión del incidente (Live Incident Response)**

- Se realizó un análisis en tiempo real del servidor para identificar procesos sospechosos, servicios activos, conexiones de red y usuarios.
- Se documentaron intentos de acceso fallidos, principalmente a través de SSH desde direcciones no autorizadas.
- Se detectaron cuentas sospechosas (ej. *hacker*) y configuraciones de servicios que no eran necesarias para la operación.

## Vulnerabilidades detectadas y corregidas

- **Puerto FTP (vsftpd):** activo sin necesidad → corregido deteniendo y deshabilitando el servicio.
- **SSH inseguro:** expuesto en puerto 22 con intentos de fuerza bruta → corregido endureciendo configuración en `sshd_config` (cambio de puerto, deshabilitar root login, limitar usuarios y deshabilitar autenticación por contraseña).
- **Cuentas sospechosas:** presencia de usuario *hacker* → eliminado.
- **Cronjobs sospechosos:** verificados, ninguno encontrado malicioso.
- **Servicios habilitados innecesarios:** identificados y removidos.
- **Firewall (UFW)** no estaba configurado → se aplicaron reglas restrictivas (deny incoming, allow outgoing, solo servicios esenciales).



## **Acciones de contención, erradicación y recuperación**

- Contención inicial: detención inmediata de servicios no requeridos (ej. FTP).
- Erradicación: eliminación de cuentas sospechosas y revisión de crontabs.
- Recuperación: refuerzo de SSH con fail2ban para proteger contra futuros intentos de fuerza bruta.
- Validación final: verificación de logs recientes, asegurando que no existían accesos no autorizados tras los cambios.

## Recomendaciones para fortalecer el sistema y evitar futuras intrusiones

- Mantener actualizado el sistema operativo y paquetes con **apt upgrade** regular.
- Usar **claves SSH** en lugar de contraseñas.
- Monitorear con herramientas como **Wazuh** para alertas en tiempo real.
- Revisar periódicamente **usuarios, crontabs y servicios habilitados**.
- Establecer **políticas de contraseñas fuertes** y rotación periódica.
- Respaldar la configuración y mantener un plan de recuperación ante incidentes.

## **Capturas justificadas de comandos clave y hallazgos críticos**

- Se incluyeron capturas de:
  - Procesos activos (ps, pstree).
  - Servicios en red (lsof, ss).
  - Usuarios del sistema (/etc/passwd).
  - Últimos accesos (last).
  - Intentos fallidos de autenticación (auth.log).
  - Configuración de SSH endurecida.
  - Configuración de firewall UFW.
- Cada captura fue acompañada de explicación técnica y acción correctiva aplicada.