***FASE 1:***

**Analizar logs del sistema** Se revisaron los siguientes archivos de log para detectar actividad sospechosa o patrones anómalos:  
 *- /var/log/syslog  
 - /var/log/auth.log  
 - /var/log/apache2/*  
 Además, se utilizaron comandos como *'journalctl'* para analizar eventos del sistema en tiempo real.

**Buscar Indicadores de Compromiso (IoC)** Se buscaron los siguientes IoC en el sistema:  
 - IPs anómalas repetitivas o conexiones persistentes desde una misma dirección remota.  
 - Ejecución de binarios sospechosos desde directorios no habituales como /tmp o /dev/shm.  
 - Cambios recientes en el archivo /etc/passwd o aparición de usuarios no autorizados.

**Usar who, last, netstat, ss, lsof**

Se ejecutaron comandos como '*who*', '*ltas*', '*netstat*', '*ss*' y '*lsof*' para:

- Identificar sesiones activas de usuarios.

- Ver accesos recientes al sistema.

- Detectar puertos abiertos y conexiones de red.

- Identificar archivos abiertos por procesos en ejecución

***FASE 2:***

**1. Aislamiento de red**

Se ha desconectado el servidor de la red local para evitar la posible propagación del incidente o la pérdida de datos.

Esto se ha hecho desactivando la interfaz de red con el siguiente comando:

*sudo ip link set enp0s3 down*

**2. Simulación de copia forense**

Dado que no se dispone de un dispositivo USB montado correctamente, se ha optado por realizar una copia comprimida

de los elementos más críticos del sistema, como los registros de log y los archivos de usuarios:

sudo tar -cvzf /root/evidencia\_basica.tar.gz /var/log /etc/passwd /etc/shadow

**3. Verificación de integridad (Hash SHA256)**

Para garantizar la integridad de la evidencia recogida, se ha calculado el hash SHA256 del archivo comprimido generado.

Esto permite verificar que el archivo no ha sido modificado posteriormente:

*sha256sum /root/evidencia\_basica.tar.gz*

El valor obtenido debe almacenarse de forma segura para futuras comparaciones.

***FASE 3:***

Herramientas utilizadas

*- chkrootkit*

*- rkhunter*

*- clamav (clamscan, freshclam)*

Análisis con ClamAV

1. Se actualizó la base de datos de virus con el comando:

*sudo freshclam*

2. Se escaneó el sistema con el comando:

*sudo clamscan -r / --infected*

Este comando recorre recursivamente todo el sistema en busca de archivos infectados, mostrando únicamente aquellos que representan una amenaza.

Métodos de persistencia revisados

- Se revisarán los siguientes archivos/directorios para detectar persistencia maliciosa:

*• /etc/crontab*

*• /etc/rc.local*

*• ~/.bashrc*

***FASE 4:***

**1. Eliminar binarios, usuarios y configuraciones comprometidas**

Se han revisado rutas como /tmp y /dev/shm en busca de binarios sospechosos, y se ha inspeccionado /etc/passwd para detectar usuarios no autorizados.

Ejemplo de eliminación de binario sospechoso:

*sudo rm -f /ruta/al/binario\_sospechoso*

Ejemplo de eliminación de usuario sospechoso:

*sudo userdel -r backdoor*

**2. Reinstalar servicios clave desde repositorios confiables**

Se han reinstalado servicios críticos como Apache o SSH para garantizar que estén limpios y actualizados.

*sudo apt install --reinstall apache2 openssh-server*

**3. Aplicar hardening inmediato**

Se han aplicado varias medidas de refuerzo de seguridad para proteger el sistema de futuros ataques.

Reforzar configuración SSH:

*- PermitRootLogin no*

*- PasswordAuthentication no*

*- AllowUsers tu\_usuario*

Reiniciar servicio SSH:

sudo systemctl restart ssh

Activar Fail2ban:

*sudo systemctl enable fail2ban*

*sudo systemctl start fail2ban*

Verificar estado de fail2ban:

*sudo fail2ban-client status sshd*

Cambiar contraseñas del sistema:

passwd

**4. Aplicar parches de seguridad del sistema**

Se han aplicado las últimas actualizaciones de seguridad disponibles para asegurar el sistema.

*sudo apt update && sudo apt upgrade -y*

*sudo apt autoremove --purge*

***FASE 5:***

**1. Activar la autenticación en dos factores (2FA) en Ubuntu Server**

Se utilizará Google Authenticator junto con PAM (Pluggable Authentication Module).

Paso 1: Instalar Google Authenticator

*sudo apt update*

*sudo apt install libpam-google-authenticator -y*

Paso 2: Configurar el 2FA para el usuario

Ejecutar este comando desde el usuario con el que se inicia sesión (por ejemplo, "davinia"):

*google-authenticator*

- Responder "s" a todas las preguntas que aparecen (puede aceptarse la configuración por defecto).

- Se mostrará un código QR que se puede escanear con la app Google Authenticator en el móvil.

Paso 3: Configurar PAM para SSH

Editar el archivo PAM para SSH:

*sudo nano /etc/pam.d/sshd*

Agregar esta línea al final del archivo:

*auth required pam\_google\_authenticator.so*

Paso 4: Habilitar el uso de contraseña y 2FA en el servicio SSH

Editar el archivo de configuración de SSH:

*sudo nano /etc/ssh/sshd\_config*

Asegurarse de tener estas opciones configuradas:

*ChallengeResponseAuthentication yes*

*UsePAM yes*

Reiniciar el servicio SSH:

*sudo systemctl restart ssh*

A partir de ahora, al conectarse por SSH se solicitará el código temporal generado por la app Google Authenticator además de la contraseña.

2. Segmentación de red por niveles de riesgo

Se implementa mediante el uso de reglas de iptables o ufw para separar servicios según su nivel de seguridad.

Paso 1: Activar UFW (Uncomplicated Firewall)

*sudo ufw enable*

Paso 2: Definir las reglas por nivel de riesgo

Zona segura (gestión por SSH):

*sudo ufw allow from 192.168.1.0/24 to any port 22 proto tcp*

Zona de servicios web (acceso público):

*sudo ufw allow 80/tcp*

*sudo ufw allow 443/tcp*

Bloquear accesos innecesarios:

*sudo ufw deny 23/tcp # Bloquear Telnet*

*sudo ufw deny 21/tcp # Bloquear FTP si no se usa*

Paso 3: Ver el estado de las reglas

*sudo ufw status numbered*

De este modo, se limita el acceso según el uso y se refuerza la seguridad por niveles.

**2. Instalar y configurar OSSEC HIDS** como sistema de detección de intrusiones (HIDS) en el servidor Ubuntu, asegurando su funcionamiento correcto y básico para la monitorización del sistema.

1. Se descargó el instalador desde GitHub con el siguiente comando:

*curl -O https://github.com/ossec/ossec-hids/archive/3.7.0.tar.gz*

2. Se descomprimió el archivo y se accedió al directorio:

*tar -xvzf 3.7.0.tar.gz*

*cd ossec-hids-3.7.0*

3. Se ejecutó el script de instalación:

*sudo ./install.sh*

4. Durante el proceso de instalación se eligieron las siguientes opciones:

*- Idioma: Español [es]*

*- Tipo de instalación: Servidor*

*- Ruta de instalación: /var/ossec (por defecto)*

*- Activar notificaciones por correo: No*

*- Activar verificación de integridad del sistema: Sí*

5. Se inició y activó el servicio de OSSEC HIDS con los siguientes comandos:

*sudo systemctl start ossec*

*sudo systemctl enable ossec*

6. Se comprobó el estado de OSSEC con:

*sudo /var/ossec/bin/ossec-control status*

El sistema confirmó que OSSEC está en funcionamiento.

**Conclusión**

La instalación y configuración básica de OSSEC HIDS se ha completado correctamente. El sistema queda protegido mediante un sistema de detección de intrusiones local.

# **3. Fases del Playbook y Checklists**

## Detección y Análisis Inicial

· ☐ Revisar /var/log/syslog, auth.log, apache2, etc.

· ☐ Buscar usuarios sospechosos o IPs anómalas.

· ☐ Analizar actividad reciente con who, last, ss, netstat.

## Aislamiento y Preservación

· ☐ Desconectar la red (ip link set enp0s3 down)

· ☐ Crear imagen forense (dd, dcfldd)

· ☐ Calcular hash (sha256sum)

· ☐ Montar copia en solo lectura

## Análisis Profundo

· ☐ Usar chkrootkit, rkhunter, clamav

· ☐ Revisar crontab, .bashrc, rc.local

· ☐ Generar cronología de eventos

## Erradicación y Reconstrucción

· ☐ Eliminar malware, usuarios sospechosos

· ☐ Reinstalar servicios

· ☐ Cambiar contraseñas y claves

· ☐ Aplicar actualizaciones (apt update && apt upgrade)

## Reforzamiento y Prevención

· ☐ Activar 2FA

· ☐ Configurar IDS (OSSEC/Wazuh)

· ☐ Segmentar red

· ☐ Redactar playbook

# 2. Roles y Responsables

| Rol | Responsable |
| --- | --- |
| Analista de Seguridad | Davinia |
| Técnico de Sistemas | Davinia |
| Coordinador de Respuesta | Davinia |
| Documentación y seguimiento | Davinia |

# 3. Herramientas

Monitoreo: journalctl, tail, grep

Análisis: chkrootkit, clamav, rkhunter

Preservación: dd, sha256sum, mount

Refuerzo: sshd\_config, fail2ban, ufw

IDS: OSSEC

# 4. Tiempos Estimados

| Fase | Tiempo Estimado |
| --- | --- |
| Detección inicial | 30-45 min |
| Aislamiento y copia forense | 1-2 horas |
| Análisis de persistencia | 1 hora |
| Erradicación y reinstalación | 1-2 horas |
| Hardening y documentación | 1 hora |