***FASE 1:***

# **1. Análisis con Lynis**

Se ha ejecutado la herramienta Lynis con el objetivo de identificar configuraciones inseguras, servicios expuestos y recomendaciones para mejorar la seguridad general del sistema. El análisis arrojó información sobre configuraciones del sistema, servicios activos y módulos del kernel que pueden representar un riesgo si no se gestionan adecuadamente.

# **2. Comprobación de servicios activos con systemd** *sudo systemctl list-units --type=service* Se han listado todos los servicios activos en ejecución. Entre ellos destacan servicios importantes como Apache2, Docker, Dovecot, MySQL, entre otros. También se detectó un fallo en el servicio logrotate.service, el cual aparece como 'failed', lo que puede suponer un riesgo en la gestión de logs si no se corrige.

# **3. Puertos en escucha detectados con netstat** *sudo netstat -tulpn* Para identificar puertos abiertos y los servicios asociados. Se detectaron puertos en escucha relacionados con servicios como dnsmasq, MySQL, squid, dovecot y CUPS. Esta información es crucial para identificar posibles vectores de ataque en la red.

# **4. Módulos del kernel cargados** *lsmod* Se ha comprobado qué módulos están actualmente cargados en el kernel. Destacan varios módulos relacionados con el audio, como snd\_intel8x0, snd\_pcm, snd\_ac97\_codec, los cuales posteriormente fueron añadidos a la blacklist para impedir su carga por razones de seguridad.

# 

# 

# **5. Blacklist de módulos del kernel**

# Se ha editado el archivo: */etc/modprobe.d/blacklist.conf* Y se han añadido los siguientes módulos para evitar su carga automática: *- snd\_intel8x0 - snd\_ac97\_codec - snd\_pcm - snd\_seq\_midi*

# **6. Revisión de usuarios válidos en el sistema**

# *cat /etc/passwd | grep -v nologin sudo cat /etc/shadow | grep '::'* Se han identificado los usuarios con acceso interactivo permitido. Los usuarios relevantes incluyen: - daviniase - desarrolladorx - ana - carlos - elena - prueba1 - empleado1x - empleado2x También se revisó si existían contraseñas en blanco usando el archivo /etc/shadow. No se encontraron usuarios con contraseñas vacías ('::').

# **7. Comprobación del archivo sudoers**

Se comprobó el contenido del archivo:  
  
 */etc/sudoers*  
 Y se confirmó que los grupos 'admin' y 'sudo' tienen permisos para ejecutar comandos como superusuario (root), lo cual es adecuado. Se incluye también la configuración de 'secure\_path' y otras directivas de seguridad recomendadas por defecto.

***FASE 2:***

Esta fase consistía en reforzar la seguridad de los accesos al servidor SSH, mediante la desactivación del acceso root, la configuración de autenticación por clave pública, y la implementación de políticas de seguridad adicionales como la caducidad de contraseñas y bloqueos por intentos fallidos.

**Configuraciones aplicadas (esperadas)**

1. Deshabilitar login SSH como root:

· Editado el archivo */etc/ssh/sshd\_config* con la línea:

· *PermitRootLogin no*

2. Habilitar autenticación con clave pública:

· En */etc/ssh/sshd\_config:*

· *PubkeyAuthentication yes*

· *AuthorizedKeysFile .ssh/authorized\_keys .ssh/authorized\_keys2*

· 3. Copiar clave pública del cliente al servidor:

· *ssh-copy-id -i ~/.ssh/id\_ed25519.pub -p 2222 usuario@IP*

· 4. Cambiar permisos y propietario del archivo authorized\_keys:

·   
 *chmod 700 ~/.ssh  
 chmod 600 ~/.ssh/authorized\_keys  
 chown usuario:usuario ~/.ssh/authorized\_keys*

· 5. Establecer caducidad de contraseñas:

· *sudo chage -M 60 usuario*

· 6. Configurar bloqueo tras varios intentos fallidos con *pam\_tally2 o faillock*

· 7. Mostrar mensaje legal en */etc/issue.net y activar en sshd\_config:*

·  *Banner /etc/issue.net*

# **Problemas encontrados**

Durante la implementación se intentó establecer conexión SSH con clave pública, pero siempre se denegó el acceso. Se realizaron múltiples pruebas:  
 - Verificación del contenido de authorized\_keys.  
 - Revisión de permisos correctos.  
 - Cambios en la contraseña del usuario.  
 - Comprobación del archivo *sshd\_config.* - Análisis del *log /var/log/auth.log.*  
 A pesar de estos esfuerzos, el servidor seguía mostrando: *Permission denied (publickey,password).*

# **Alternativa documentada (funcionamiento esperado)**

En un entorno donde no haya errores, estos pasos deberían ser suficientes para tener acceso SSH sin contraseña:

1. En el cliente:

*ssh-keygen -t ed25519*

*ssh-copy-id -i ~/.ssh/id\_ed25519.pub -p 2222 usuario@IP*

2. En el servidor:

- Asegurarse que el archivo */home/usuario/.ssh/authorized\_keys* tenga los permisos correctos y que el servicio SSH esté reiniciado.

- Verificar que el usuario puede autenticarse por clave pública.

3. Reiniciar servicio SSH:

*sudo systemctl restart ssh*

4. Probar conexión:

*ssh -i ~/.ssh/id\_ed25519 -p 2222 usuario@IP*

***FASE 3:***

## **1. Desactivación de servicios innecesarios**

Se identificaron y desactivaron los siguientes servicios que no son esenciales para el funcionamiento del servidor:  
 - CUPS (servicio de impresión).  
 - Avahi (descubrimiento de servicios en red).  
 - NFS no estaba instalado.

Pasos realizados:

· a) Verificación de los servicios:

·  *systemctl is-enabled cups*

*· systemctl is-enabled avahi-daemon*

*· systemctl is-enabled nfs-server*

· b) Desactivación de los servicios:

·  *sudo systemctl stop cups*

*· sudo systemctl disable cups*

*· sudo systemctl stop avahi-daemon*

*· sudo systemctl disable avahi-daemon*

## **2. Endurecimiento del Kernel con sysctl**

Se modificó el archivo `/etc/sysctl.conf` para proteger el sistema frente a varias amenazas de red.

· Parámetros añadidos o modificados:

*# Protección contra IP Spoofing*

· *net.ipv4.conf.default.rp\_filter = 1*

*# Protección contra IP Spoofing*

*· net.ipv4.tcp\_syncookies = 1*

*# Protección contra ataques SYN Flood*

*· net.ipv4.conf.all.accept\_redirects = 0*

*· net.ipv4.conf.default.accept\_redirects = 0*

*# Evitar redirecciones maliciosas*

· *net.ipv4.conf.all.accept\_source\_route = 0*

*# Evitar rutas de origen falsas*

*· net.ipv4.conf.default.accept\_source\_route = 0*

Una vez editado el archivo, se aplicaron los cambios con:

*sudo sysctl -p*

## **Verificación final**

Se ejecutó `*sudo sysctl -p*` y se confirmó que todos los parámetros estaban activos correctamente.

***FASE 4:***

**1. Establecer permisos correctos para archivos sensibles**

Se han verificado y/o corregido los siguientes permisos:

*• /etc/shadow → 600*

*• /etc/passwd → 644*

*• /var/log → Propietario root, permisos 750*

**2. Activar el sistema de auditoría con auditd**

Comandos ejecutados:

*• sudo apt update*

*• sudo apt install auditd -y*

*• sudo systemctl enable auditd*

*• sudo systemctl start auditd*

*• sudo systemctl status auditd*

3. Configurar monitoreo de integridad con AIDE

Comandos utilizados:

*• sudo apt install aide -y*

*• sudo aideinit*

Luego, se ha movido el archivo generado con el siguiente comando

*sudo mv /var/lib/aide/aide.db.new.gz /var/lib/aide/*[*aide.db.gz*](http://aide.db.gz)

Para verificar la integridad:

*• sudo aide --check*

**Observaciones**

Durante la ejecución se detectó que la base de datos generada por AIDE no estaba en la ruta esperada, por lo que fue necesario localizarla con 'find' y moverla manualmente.

***FASE 5:***

**1. Activar UFW (Uncomplicated Firewall)**

Se intentó habilitar el firewall UFW con el comando `sudo ufw enable`, pero se mostró el error 'orden no encontrada'. Esto indica que el paquete no estaba instalado. Para solucionarlo se deben ejecutar los siguientes comandos:

*sudo apt update*

*sudo apt install ufw*

Una vez instalado, se activa con:

*sudo ufw enable*

Y se permiten servicios como SSH con:

*sudo ufw allow ssh*

**2. Instalar y configurar Fail2Ban**

Se instaló fail2ban y se copió su archivo de configuración por defecto:

*sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local*

Luego se editó con:

*sudo nano /etc/fail2ban/jail.local*

En la sección [sshd] se configuraron los siguientes parámetros:

*- enabled = true*

*- filter = sshd*

*- logpath = /var/log/auth.log*

*- maxretry = 3*

*- bantime = 3600*

Se detectó un error de duplicación del bloque [sshd] que provocaba que fail2ban no arrancara. El error fue identificado con el comando:

*sudo fail2ban-client -d*

Se solucionó eliminando el bloque duplicado.

**3. Detección de conexiones anómalas con portsentry**

Para detectar conexiones sospechosas se puede usar `portsentry`, instalándolo con:

*sudo apt install portsentry*

Después se edita *`/etc/default/portsentry`* y se activan las opciones necesarias (ENABLE).

**4. Registro de actividad sospechosa y alertas**

Fail2ban registra todos los intentos fallidos y acciones en `/var/log/fail2ban.log`. También se pueden configurar alertas por correo modificando el parámetro `action` en jail.local, por ejemplo:

*action = %(action\_mwl)s*