Informe técnico de Auditoría y Bastionado del Sistema Ubuntu 24.04 LTS

(Basado en CIS Benchmarks, NIST SP 800-123 y OWASP Server Hardening Guide)

Autor: Rafael M. Pérez | Administrador de sistemas & Técnico en Ciberseguridad

Fecha: Octubre 2025

Versión del informe: 1.0

Auditoría realizada con: Lynis 3.0.9

Herramientas complementarias: ClamAV, rkhunter, chkrootkit, Fail2Ban, USBGuard, auditd, rsyslog, sysstat, acct, libpam-pwquality, mod_security2, mod_evasive

Objetivo: Fortalecer el sistema operativo Ubuntu 24.04 LTS mediante técnicas de bastionado y endurecimiento del kernel, servicios, autenticación y políticas de seguridad, sin comprometer la funcionalidad del entorno operativo.

Índice de Bastionado:

Resultado inicial: 68 / 100

Resultado final: 87 / 100

Incremento total: +19 puntos de seguridad

Índice

2. Resumen Ejecutivo

Este informe documenta el proceso completo de **bastionado del sistema operativo Ubuntu 24.04**, orientado a fortalecer la seguridad sin afectar la funcionalidad ni la operatividad general del entorno. Se aplicaron medidas de endurecimiento en múltiples capas -kernel, servicios, autenticación, red y auditoría- combinando buenas prácticas de ciberseguridad con comprobaciones continuas mediante **Lynis 3.0.9**.

El proyecto logró una mejora significativa en la puntuación de seguridad, pasando de **68 a 87 puntos** (+18), manteniendo un sistema totalmente estable y funcional.

Entorno de bastionado: Ubuntu 24.04 Desktop, kernel 6.x, instalación limpia sobre hardware físico con conexión Wi-Fi y servicios de red activos.El sistema partía de una configuración inicial con ciertas medidas básicas de hardening, incluyendo:

- **Geobloqueo** con *iptables* a países fuera de España y uso de **UFW** como cortafuegos.
- **Bastionado del servicio SSH**, deshabilitando el acceso root mediante `PermitRootLogin`.
- **Activación de auditorías** con *auditd*.
- **Verificación de integridad** del sistema mediante *AIDE*.

Aspectos destacados durante la auditoría:

- Implementación de **herramientas anti-malware y detección de rootkits** (ClamAV, rkhunter, chkrootkit).
- **Endurecimiento del kernel y servicios críticos** (SSH, Fail2Ban, PAM, Auditd, Sysctl, USBGuard).
- Aplicación de **políticas de contraseñas seguras, logging avanzado y actualizaciones automáticas**.
- Validación continua del impacto de las medidas para garantizar **seguridad sin pérdida de operatividad**.

3. Resumen gráfico

Área de Seguridad	Estado Inicial	Estado Final	Mejora
Índice Lynis	68 / 100	**87 / 100**	+18
Kernel y Sysctl	Configuración por defecto	ASLR, ptrace_scope, antispoofing activados	Alta
Autenticación (PAM)	Contraseñas débiles, sin hashing fuerte	SHA512 + yescrypt + pwquality	Alta
Servicios (SSH, Fail2Ban, etc.)	Sin aislamiento systemd	Sandboxing aplicado + logs mejorados	Media
Auditoría y Logging	Sin auditd ni logs remotos	Auditd + rsyslog remoto + sysstat	Alta
Actualizaciones	Manuales	unattended-upgrades activo	Alta
Seguridad física/USB	Sin control	USBGuard implementado	Alta

Resumen visual de bastionado del sistema Ubuntu 24.04, Comparativa antes y después del endurecimiento.

4. Introducción

El objetivo de este proyecto es **bastionar al máximo el sistema operativo Ubuntu 24.04 LTS** sin comprometer su funcionalidad, documentando de forma detallada cada paso del proceso para facilitar futuras auditorías o la resolución de posibles incidencias.

5. Auditoría inicial con Lynis

Como punto de partida, se realizó una auditoría inicial de seguridad con la herramienta Lynis, obteniendo los siguientes resultados:

```
sudo lynis audit system
```

Resultado:

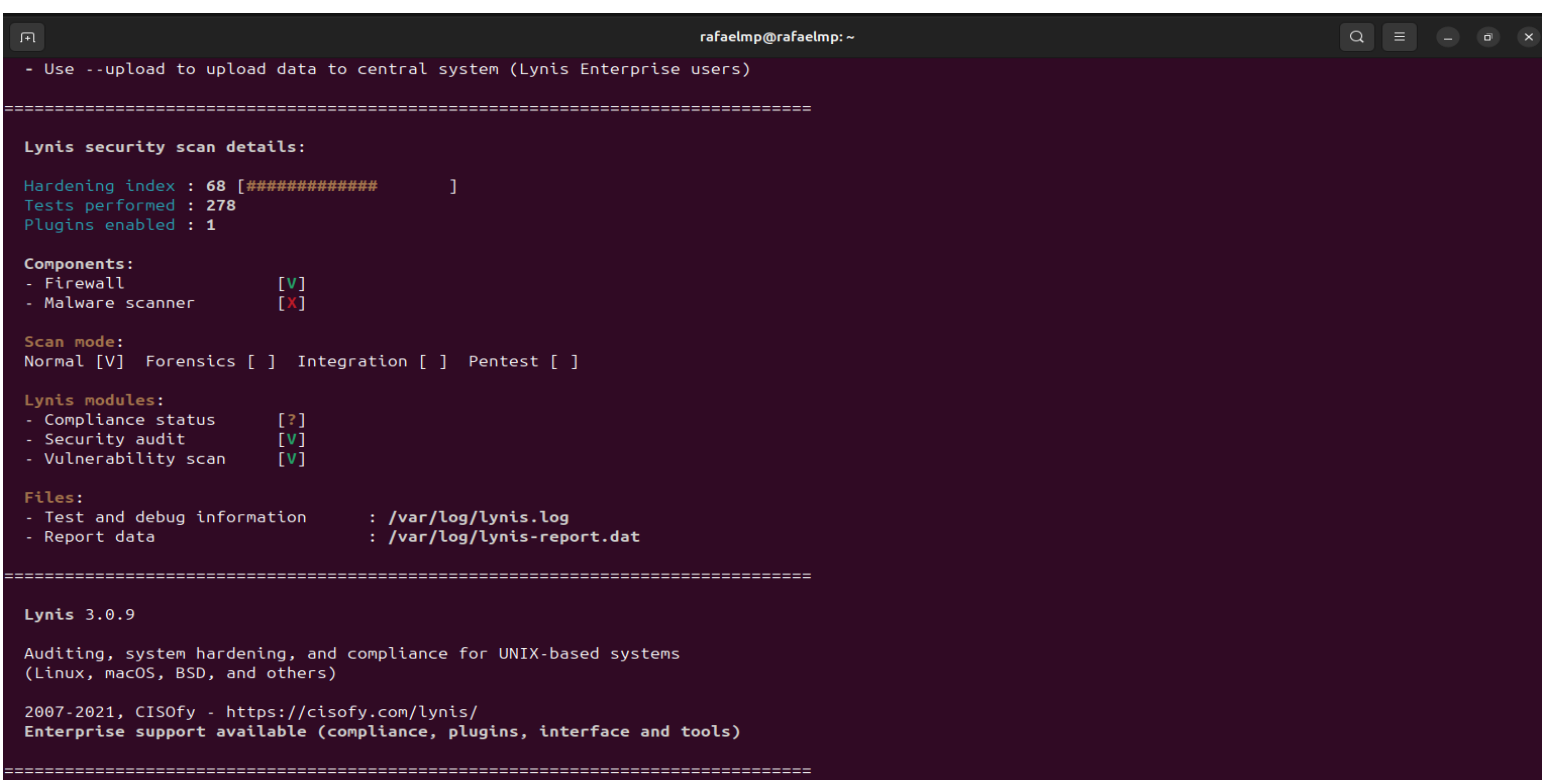
Hardening Index: 68/100

Advertencias: 1

Sugerencias: 53

Se detectaron configuraciones por defecto en SSH, GRUB, PAM, kernel y servicios de red, así como módulos inseguros y políticas de contraseñas débiles.

A partir de estos hallazgos, se definió un plan de bastionado estructurado en fases, con el fin de mejorar la seguridad del sistema sin afectar la estabilidad ni la experiencia de usuario.



6. Fases del bastionado:

6.1 Fase 1: Instalación de soluciones anti-malware

Instalamos y verificamos **ClamAV** (scanner por firmas) para escaneos bajo demanda y programados.

A continuación instalamos detectores de rootkits: **rkhunter** y **chkrootkit**.

Y para finalizar comprobamos el correcto arranque de los servicios y realizamos un escaneo inicial.

Comandos (resumen):

```
sudo apt update && sudo apt upgrade -y
sudo apt install clamav clamav-daemon clamav-freshclam rkhunter chkrootkit -y
sudo freshclam
sudo systemctl status clamav-daemon
sudo rkhunter --update && sudo rkhunter --checkall
sudo chkrootkit
sudo clamdscan --recursive /home --move=/var/quarantine
```

Con este paso pasamos de un nivel de bastionamiento en lynis de 68 a 70.

6.2 Fase 2: Endurecimiento de contraseñas y políticas PAM

1. Establecemos políticas de contraseñas seguras

Editamos los parámetros globales de seguridad y comportamiento de las cuentas de usuario:

```
sudo nano /etc/login.defs
```

Adjuntando estas líneas:

```
PASS_MAX_DAYS   90
PASS_MIN_DAYS   7
PASS_WARN_AGE   14
UMASK           027
```

2.Configuramos hashing fuerte y rondas (SHA512)

Editamos Pam:

```
sudo nano /etc/pam.d/common-password
```

Y añadimos la siguiente línea:

```
password   [success=1 default=ignore]   pam_unix.so obscure sha512 rounds=5000
```

3.Forzamos expiración de contraseñas actuales

Aplicamos políticas a los usuarios existentes:

```
sudo chage -M 90 -m 7 -W 14 rafaelmp
```

Y verificamos:

```
sudo chage -l rafaelmp
```

4.Añadimos verificación de complejidad

Instalamos el módulo libpam-pwquality

```
sudo apt install libpam-pwquality -y
```

Editamos el archivo de configuración

```
sudo nano /etc/security/pwquality.conf
```

Añadimos o modificamos estos valores:

```
minlen = 10        # Longitud mínima
dcredit = -1       # Al menos un número
ucredit = -1       # Al menos una mayúscula
lcredit = -1       # Al menos una minúscula
ocredit = -1       # Al menos un carácter especial
retry = 3          # Intentos permitidos antes de error
```

*Con estos 5 pasos pasamos de un nivel de bastionamiento en lynis de 70 a 71.

6.3 Fase 3: Bastionamiento del Kernel

*Nota: El objetivo es endurecer el Kernel sin comprometer en absoluto la funcionalidad del sistema, por lo que lo haremos de forma moderada y controlada.

1.Creamos copia de seguridad antes de modificar nada

```
sudo cp /etc/sysctl.conf /etc/sysctl.conf.backup
```

2.Editamos el archivo sysctl.conf

```
sudo nano /etc/sysctl.conf
```

3.Añadimos al final las siguientes líneas:

```
# === Protecciones básicas del sistema ===
kernel.randomize_va_space = 2            # ASLR (protección contra exploits)
kernel.kptr_restrict = 2                 # Oculta direcciones de kernel
kernel.dmesg_restrict = 1                # Restringe acceso a dmesg
kernel.unprivileged_bpf_disabled = 1     # Desactiva BPF para usuarios no root
kernel.yama.ptrace_scope = 1             # Limita depuración entre procesos
kernel.core_uses_pid = 1                 # Añade PID a core dumps
kernel.suid_dumpable = 0                 # Evita core dumps con SUID

# === Red IPv4 ===
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0

net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.conf.all.rp_filter = 1          # Antispoofing
```

```
net.ipv4.conf.default.rp_filter = 1
net.ipv4.icmp_echo_ignore_broadcasts = 1   # Ignorar ping broadcast
net.ipv4.icmp_ignore_bogus_error_responses = 1
net.ipv4.tcp_syncookies = 1              # Protege de SYN flood
net.ipv4.tcp_timestamps = 0              # Evita fingerprinting
net.ipv4.tcp_max_syn_backlog = 2048
net.ipv4.conf.all.log_martians = 1       # Log de paquetes sospechosos
net.ipv4.conf.default.log_martians = 1

# === Red IPv6 (si la usas, mantén activado) ===
net.ipv6.conf.all.accept_redirects = 0
net.ipv6.conf.default.accept_redirects = 0
net.ipv6.conf.all.accept_source_route = 0
net.ipv6.conf.default.accept_source_route = 0

# === Montajes seguros ===
fs.protected_hardlinks = 1
fs.protected_symlinks = 1
fs.protected_fifos = 2
fs.protected_regular = 2

# === Evitar módulos maliciosos ===
kernel.modules_disabled = 0              # Mantener en 0 para no romper DKMS (NVIDIA, etc.)

# === Performance segura ===
vm.mmap_rnd_bits = 32
```

vm.mmap_rnd_compat_bits = 16

=== Prevención de ataques locales ===

dev.tty.ldisc_autoload = 0

net.core.bpf_jit_harden = 2 # Endurecer JIT BPF

Aplicamos los cambios

sudo sysctl -p

Verificamos la configuracion actual:

sysctl kernel.kptr_restrict

sysctl net.ipv4.conf.all.accept_redirects

sysctl fs.protected_symlinks

*Con estos 5 pasos pasamos de un nivel de bastionamiento en lynis de 71 a 74.

6.4 Fase 4: Servicios y seguridad avanzada

1.Reforzar fail2ban

sudo systemctl edit fail2ban.service

Añadimos este texto en el editor

[Service]

ProtectSystem=full

ProtectHome=yes

PrivateTmp=yes

NoNewPrivileges=yes

ProtectKernelModules=yes

ProtectControlGroups=yes

[Control+X pulsamos S y guardamos]

Recargamos y reiniciamos el servicio

sudo systemctl daemon-reexec

sudo systemctl daemon-reload

```
sudo systemctl restart fail2ban
```

Y por último comprobamos que todo funciona correctamente

```
systemd-analyze security fail2ban.service
```

2.Repetimos el proceso con ClamAV y SSH

```
sudo systemctl edit clamav-daemon.service
```

Añadimos el siguiente texto

[Service]

ProtectSystem=full

PrivateTmp=yes

ProtectHome=yes

NoNewPrivileges=yes

[Ctrl+X pulsamos S y enter]

3.Endurecimiento de SSH

sudo systemctl edit ssh.service

Añadimos

[Service]

ProtectSystem=full

ProtectHome=yes

PrivateTmp=yes

NoNewPrivileges=yes

ProtectKernelModules=yes

ProtectControlGroups=yes

[Ctrl+X pulsamos S y Enter]

Regargamos todo

```
sudo systemctl daemon-reexec
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl restart fail2ban clamav-daemon ssh
```

3.Configuramos un password policy fuerte.

Ejecutamos

```
sudo nano /etc/security/pwquality.conf
```

Y añadimos o modificamos estas líneas si no están

minlen = 12

dcredit = -1

ucredit = -1

ocredit = -1

lcredit = -1

maxrepeat = 3

difok = 4

retry = 3

[Ctrl+X pulsamos S y Enter]

4.Habilitamos unattended-upgrades

```
sudo apt install unattended-upgrades
```

```
sudo dpkg-reconfigure unattended-upgrades
```

5.Instalamos usbguard

```
sudo apt install usbguard
```

Y lo activamos

```
sudo systemctl enable --now usbguard
```

6.Instalamos process acounting

```
sudo apt install acct -y
```

Y lo activamos

```
sudo systemctl enable --now acct
```

7.Corregimos Banners legales

```
sudo nano /etc/issue
```

```
sudo nano /etc/issue.net
```

Añadimos el siguiente texto en ambas

Acceso restringido. Solo usuarios autorizados.

Toda actividad será monitoreada.

8.Reforzamos SSH

```
sudo nano /etc/ssh/sshd_config
```

Y añadimos o modificamos

PermitRootLogin no

PasswordAuthentication yes

ClientAliveCountMax 2

MaxAuthTries 3

X11Forwarding no

AllowTcpForwarding no

AllowAgentForwarding no

TCPKeepAlive no

LogLevel VERBOSE

[Ctrl+X pulsamos S y Enter]

9.Añadimos reglas Audit

Editamos

sudo nano /etc/audit/rules.d/hardening.rules

Añadimos al final

-w /etc/passwd -p wa -k passwd_changes

-w /etc/group -p wa -k group_changes

-w /etc/shadow -p wa -k shadow_changes

-w /etc/sudoers -p wa -k sudoers_changes

-w /var/log/ -p wa -k logs

Y aplicamos

sudo systemctl restart auditd

10.Instalamos y activamos sysstat

sudo apt install sysstat -y

Y activamos

sudo systemctl enable --now sysstat

11.Añadimos protección Anti-Dos y Waf a Apache

sudo apt install libapache2-mod-security2 libapache2-mod-evasive -y

sudo a2enmod security2 evasive

sudo systemctl restart apache2

12.Login remoto

Editamos

sudo nano /etc/rsyslog.conf

Añadimos esto al final

. @@127.0.0.1:514

Y reiniciamos

```
sudo systemctl restart rsyslog
```

*Con estos 12 pasos pasamos de un nivel de bastionamiento en Lynis de 74 a 82.

6.5 Fase 5: Optimización final

1.Ajustar hashing y umask de contraseñas

Editamos

```
sudo nano /etc/login.defs
```

Y añadimos o modificamos

```
PASS_MAX_DAYS   90
PASS_MIN_DAYS   7
PASS_WARN_AGE   14
UMASK           027
ENCRYPT_METHOD  yescrypt
SHA_CRYPT_MIN_ROUNDS  5000
SHA_CRYPT_MAX_ROUNDS  10000
```

2.Activamos volcados de memoria (core dumps)

Editamos

```
sudo nano /etc/security/limits.conf
```

Y añadimos al final

```
* hard core 0
```

3.Endurecemos sysctl.conf restante

Editamos

```
sudo nano /etc/sysctl.conf
```

Añadimos

fs.suid_dumpable = 0

kernel.perf_event_paranoid = 3

kernel.sysrq = 0

net.ipv4.conf.all.forwarding = 0

net.ipv4.conf.all.log_martians = 1

net.ipv4.conf.default.log_martians = 1

Guardamos y aplicamos los cambios

```
sudo sysctl -p
```

4.Instalamos y activamos sysstat

```
sudo apt install sysstat -y
```

```
sudo systemctl enable --now sysstat
```

5.Hacemos unos últimos ajustes en el SSH

sudo nano /etc/ssh/sshd_config

Añadimos o desmarcamos

MaxSessions 2

Port 22

X11Forwarding no

Y reiniciamos el servicio

```
sudo systemctl restart ssh
```

6. Por último limpieza de paquetes huérfanos y configuración obsoleta.

```
sudo apt purge $(dpkg -l | awk '/^rc/ {print $2}')
```

```
sudo apt autoremove --purge
```

Y volvemos a ejecutar el comando para comprobar el resultado final

```
sudo lynis audit system
```

7. Tabla de riesgos y recomendaciones técnicas

Área / Componente	Riesgo principal	Descripción técnica	Medida aplicada / Recomendación
SSH (OpenSSH Server)	Acceso remoto inseguro o fuerza bruta	SSH venía con parámetros por defecto y permitía varios intentos de login, incluyendo acceso root.	Se desactivó PermitRootLogin, se redujo MaxAuthTries, se activó LogLevel VERBOSE y se reforzó con **Fail2Ban**.
Kernel / sysctl.conf	Exposición a exploits locales o ataques de red	El kernel no aplicaba protecciones ASLR ni filtros de red seguros.	Se configuraron parámetros en /etc/sysctl.conf para activar ASLR, ptrace_scope, syncookies y filtrado antispoofing.
PAM y contraseñas	Políticas de autenticación débiles	No había control de complejidad ni cifrado fuerte de contraseñas.	Se instaló libpam-pwquality, se activó hashing **yescrypt**, y se configuró caducidad y rotación con chage y login.defs.
Servicios del sistema (systemd)	Procesos críticos con privilegios amplios	Algunos servicios se ejecutaban sin aislamiento ni restricciones.	Se aplicó sandboxing con ProtectSystem=full, PrivateTmp=yes y NoNewPrivileges=yes en ssh, clamav y fail2ban.
USBGuard	Conexión de dispositivos externos sin control	Existía riesgo de pendrives maliciosos o fuga de información por USB.	Se instaló y activó **USBGuard**, estableciendo políticas restrictivas por defecto (systemctl enable --now usbguard).
Logging y auditoría	Falta de trazabilidad en eventos del sistema	Los registros no estaban centralizados ni auditados correctamente.	Se configuró auditd con reglas en /etc/audit/rules.d/, se habilitó sysstat y logs remotos con rsyslog.
Apache / HTTP	Exposición ante ataques web o DoS	Apache funcionaba sin capa de seguridad adicional ni filtros.	Se activaron los módulos **mod_security2** y **mod_evasive** como WAF y defensa ante ataques de denegación de servicio.
Actualizaciones y parches	Riesgo por vulnerabilidades no corregidas	Las actualizaciones eran manuales y no había automatización.	Se instaló y configuró **unattended-upgrades** para aplicar parches de seguridad automáticos desde repositorios oficiales.

8. Resultado Final y conclusiones:

Hemos conseguido pasar de 68 a 87 puntos en Lynis, reforzando el sistema en múltiples capas sin afectar la funcionalidad general y documentando todo el proceso para facilitar futuras revisiones o auditorías.

Se han aplicado medidas de seguridad en **autenticación, kernel, red, servicios, auditoría y control de acceso físico**, logrando un endurecimiento completo y coherente del sistema.

No se ha producido **pérdida de funcionalidad ni estabilidad**: todos los servicios y controladores (incluyendo red inalámbrica, actualizaciones automáticas, SSH y entorno gráfico) funcionan correctamente tras el bastionado.

Se ha alcanzado un **equilibrio óptimo entre seguridad, rendimiento y usabilidad**, siguiendo un enfoque por capas verificable con herramientas estándar como Lynis.

Este bastionado demuestra que es posible alcanzar un nivel alto de seguridad sin sacrificar la usabilidad, siguiendo un enfoque basado en capas y verificable con herramientas estándar.

En definitiva, se ha obtenido un **sistema seguro, estable y plenamente operativo**, cumpliendo las mejores prácticas de ciberseguridad sin comprometer la experiencia del usuario.

```
rafaelmp@rafaelmp: ~                                    rafaelmp@rafaelmp: ~
  - Use --upload to upload data to central system (Lynis Enterprise users)

================================================================================

  Lynis security scan details:

  Hardening index : 87 [#################   ]
  Tests performed : 276
  Plugins enabled : 1

  Components:
  - Firewall               [V]
  - Malware scanner        [V]

  Scan mode:
  Normal [V]  Forensics [ ]  Integration [ ]  Pentest [ ]

  Lynis modules:
  - Compliance status      [?]
  - Security audit         [V]
  - Vulnerability scan     [V]

  Files:
  - Test and debug information      : /var/log/lynis.log
  - Report data                     : /var/log/lynis-report.dat

================================================================================

  Lynis 3.0.9

  Auditing, system hardening, and compliance for UNIX-based systems
  (Linux, macOS, BSD, and others)

  2007-2021, CISOfy - https://cisofy.com/lynis/
  Enterprise support available (compliance, plugins, interface and tools)
```

9. Anexo técnico: descripción de herramientas empleadas

Herramienta / Servicio	Función principal	Rol dentro del bastionado
Lynis	Auditor de seguridad de sistemas Linux.	Evalúa configuraciones, permisos, kernel, servicios y genera un índice de bastionado.
ClamAV / clamd / freshclam	Antivirus de código abierto para Linux.	Detecta malware en tiempo real y permite escaneos manuales y automáticos.
rkhunter / chkrootkit	Detectores de rootkits y anomalías en el sistema.	Buscan modificaciones sospechosas en binarios, puertos y procesos del sistema.
Fail2Ban	Sistema de prevención de intrusiones por fuerza bruta.	Monitorea logs (SSH, Apache, etc.) y bloquea IPs que realicen intentos repetidos.
USBGuard	Sistema de control de dispositivos USB.	Permite o bloquea el acceso de dispositivos físicos, mitigando ataques por hardware.
auditd / auditctl	Subsistema de auditoría de Linux.	Registra cambios críticos en ficheros del sistema y eventos de seguridad.
rsyslog	Gestor central de registros del sistema.	Centraliza logs locales y permite envío remoto de eventos para auditoría.
sysstat	Conjunto de herramientas de monitorización del rendimiento.	Permite registrar estadísticas del sistema (CPU, E/S, red) para detección de anomalías.
acct (process accounting)	Auditoría de procesos ejecutados.	Guarda un histórico detallado de los comandos y procesos ejecutados por cada usuario.
unattended-upgrades	Actualizador automático de seguridad.	Instala parches críticos y de seguridad sin intervención manual.
libpam-pwquality	Módulo PAM de complejidad de contraseñas.	Aplica reglas de longitud, mayúsculas, símbolos, números y evita contraseñas débiles.
sysctl.conf	Archivo de configuración del kernel.	Define parámetros de red, memoria y seguridad a nivel del núcleo del sistema.
/etc/login.defs	Configuración de políticas de usuarios.	Controla la caducidad de contraseñas, el método de cifrado y la política UMASK.
mod_security2 / mod_evasive (Apache)	Módulos de seguridad web (WAF y Anti-DoS).	Filtran peticiones maliciosas, previenen ataques de denegación de servicio y mejoran la seguridad del servidor HTTP.
/etc/security/limits.conf	Políticas de límites de usuario.	Restringe recursos o evita core dumps innecesarios.

Bibliografía y referencias técnicas

I. **CIS Benchmarks – Center for Internet Security.**
Guía de referencia utilizada para revisar las configuraciones recomendadas de seguridad en sistemas Ubuntu 24.04 LTS.

Disponible en: https://www.cisecurity.org/benchmark/ubuntu_linux

II. **Lynis – CISOfy.**
Herramienta principal utilizada para las auditorías de bastionado del sistema.
Documentación oficial: https://cisofy.com/documentation/lynis/

III. **Ubuntu Security Guide – Canonical.**
Documentación oficial de Ubuntu sobre buenas prácticas de seguridad y endurecimiento del sistema.
https://ubuntu.com/security

IV. **OWASP Foundation.**
Referencia para aplicar buenas prácticas de seguridad en sistemas Linux y servicios expuestos.
https://owasp.org/

V. **NIST SP 800-123 – National Institute of Standards and Technology.**
Guía general sobre seguridad de servidores y bastionado de sistemas operativos.
https://csrc.nist.gov/publications/detail/sp/800-123/final

VI. **Documentación personal.**
Notas y comprobaciones propias realizadas durante el proceso de bastionado del sistema Ubuntu 24.04 con herramientas como Fail2Ban, ClamAV, rkhunter y auditd.