Seguridad básica en Linux: Gestión de usuarios, permisos y hardening inicial

Alumnos

Maximo Perrotta,Camilo Quiroga.

Tecnicatura Universitaria en Programación - Universidad Tecnológica Nacional.

Índice:

1. Introducción

La seguridad en los sistemas operativos representa uno de los pilares fundamentales para el funcionamiento confiable de cualquier infraestructura tecnológica. A medida que los sistemas crecen en complejidad y se conectan en red, la exposición a riesgos también aumenta. Por eso, asegurar el control sobre los accesos, los permisos y los servicios activos en un sistema se vuelve esencial, especialmente en entornos multiusuario como los basados en Linux.

En este trabajo se aborda el concepto de seguridad básica en sistemas

operativos Linux, enfocándonos en las herramientas que el sistema ofrece para proteger datos y restringir el acceso no autorizado. A través del uso de comandos como adduser, chmod, chown, ufw, y la gestión de logs, se busca demostrar cómo aplicar técnicas de hardening básico para reforzar un sistema recién instalado.

El objetivo principal del trabajo es mostrar, de forma práctica, cómo aplicar configuraciones simples pero efectivas que permiten aumentar la seguridad del sistema, sin recurrir a herramientas externas. El caso práctico se basa en un script Bash que automatiza algunas de estas configuraciones y puede ser ejecutado en distribuciones como Ubuntu o Debian.

2. Marco Teórico

2.1 ¿Qué es la seguridad en sistemas operativos?

La seguridad en un sistema operativo se refiere al conjunto de políticas, herramientas y mecanismos que protegen la integridad, confidencialidad y disponibilidad de los recursos del sistema. Estos recursos incluyen datos, procesos, archivos y la propia estructura del sistema. Las amenazas pueden incluir accesos no autorizados, modificaciones accidentales o maliciosas, pérdida de información y explotación de vulnerabilidades.

2.2 Modelo de seguridad CIA

La tríada CIA (Confidentiality, Integrity, Availability) es un modelo estándar que resume los principios fundamentales de la seguridad informática:

- **Confidencialidad:** garantizar que solo las personas autorizadas accedan a la información.
- **Integridad:** asegurar que los datos no sean modificados de forma incorrecta o maliciosa.
- **Disponibilidad:** asegurar que los recursos estén accesibles cuando sean requeridos.

2.3 Gestión de usuarios y permisos en Linux

Linux es un sistema operativo multiusuario. Cada usuario puede tener permisos distintos sobre los archivos y servicios del sistema. Estos permisos se manejan mediante:

- **Usuarios y grupos:** creados con comandos como adduser o usermod.

- **Permisos de archivo:** definidos mediante chmod (read, write, execute).

- **Propiedad de archivos:** gestionada con chown (usuario y grupo propietario).

Los archivos sensibles como /etc/passwd y /etc/shadow almacenan información crítica sobre los usuarios y sus contraseñas, protegida por el sistema de permisos.

2.4 Hardening básico del sistema

El **hardening** o "endurecimiento" de un sistema implica aplicar medidas para reducir su superficie de ataque. En sistemas Linux, esto incluye:

- Desactivar servicios innecesarios (systemctl disable)

- Configurar un firewall básico (ufw, iptables)

- Limitar accesos mediante el archivo /etc/sudoers o bloqueos de SSH

- Establecer políticas de contraseñas seguras y caducidad (chage)

- Monitorear accesos mediante logs (/var/log/auth.log, journalctl)

2.5 Logs del sistema

Los logs son archivos que registran eventos en el sistema. En términos de seguridad, son esenciales para detectar accesos indebidos, errores de autenticación y actividad inusual. Linux almacena estos registros en /var/log, especialmente en archivos como auth.log, syslog y dmesg.

3. Caso práctico

Como parte de la aplicación práctica de los conceptos estudiados, se desarrolló un pequeño script en Bash llamado seguridad_basica.sh, que automatiza una serie de tareas orientadas al refuerzo inicial de seguridad en un sistema Linux. Estas tareas incluyen la creación de usuarios con acceso restringido, la configuración de permisos en directorios críticos, la activación de un firewall básico, la desactivación de servicios innecesarios y la revisión de registros de seguridad del sistema.

El script fue diseñado, ejecutado y validado en el entorno Google Cloud Shell, el cual ofrece una terminal Linux basada en Debian accesible desde navegador web, sin necesidad de instalar software adicional.Para la creación de usuarios, se utilizó el comando useradd en lugar de adduser. Esta decisión se tomó para evitar la solicitud de datos personales innecesarios como nombre completo, número de habitación o

teléfono, los cuales no aportan valor a la demostración del caso práctico y pueden generar confusión en el entorno elegido. Complementariamente, se usó el comando passwd para asignar una contraseña segura al nuevo usuario.

El script también crea una carpeta protegida en el directorio raíz (/seguridad), cuya propiedad y permisos están asignados exclusivamente al usuario root, impidiendo el acceso de cualquier otro usuario del sistema. Esto demuestra la capacidad del sistema de gestionar accesos y proteger recursos.

En cuanto a la seguridad en red, se utilizó el firewall ufw para permitir únicamente conexiones SSH y bloquear el puerto 23 (Telnet), un protocolo considerado inseguro. Además, se intentó desactivar un servicio no esencial (bluetooth) como ejemplo de hardening, aunque dicho servicio no estaba disponible en el entorno de Cloud Shell.

Finalmente, se analizaron los logs del sistema ubicados en /var/log/auth.log, mediante el comando grep, para detectar intentos de acceso fallidos al sistema. Estos registros permiten al administrador identificar posibles ataques o errores de autenticación.

Todos estos pasos fueron documentados con capturas de pantalla, y el script completo fue subido a un repositorio público de GitHub. Su diseño apunta a ser sencillo, reutilizable y fácil de entender.

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to stoked-castle-463721-u5.
Use `gcloud config set project [PROJECT_ID]` to change to a different project.
maxperrotta2011@cloudshell:~ (stoked-castle-463721-u5)$ ./seguridad_basica.sh
== Seguridad básica en Linux ==
Ingrese el nombre del nuevo usuario seguro:
```

```
== Seguridad básica en Linux ==
Ingrese el nombre del nuevo usuario seguro: maximo
useradd: user 'maximo' already exists
Usuario 'maximo' creado con shell restringido.
New password:
```

```
== Seguridad básica en Linux ==
Ingrese el nombre del nuevo usuario seguro: maximo
useradd: user 'maximo' already exists
Usuario 'maximo' creado con shell restringido.
New password:
Retype new password:
passwd: password updated successfully
mkdir: cannot create directory '/seguridad': File exists
Carpeta /seguridad creada con acceso exclusivo para root.
Activando el firewall y bloqueando el puerto 23 (telnet)...
sudo: ufw: command not found
sudo: ufw: command not found
sudo: ufw: command not found
sudo: ufw: command not found
Desactivando servicio innecesario (bluetooth)...
Servicio bluetooth no disponible en este entorno.
Mostrando últimos intentos de acceso fallidos:
2025-06-22T22:02:05.679461+00:00 cs-299520461966-default sudo: maxperrotta2011 : TTY=pts/14 ; PWD=/home/maxperrotta2011 ; USER=ro
2025-06-22T22:02:21.681992+00:00 cs-299520461966-default sudo: maxperrotta2011 : TTY=pts/14 ; PWD=/home/maxperrotta2011 ; USER=ro
2025-06-22T22:30:12.509474+00:00 cs-299520461966-default sudo: maxperrotta2011 : TTY=pts/15 ; PWD=/home/maxperrotta2011 ; USER=ro
2025-06-22T22:30:36.283341+00:00 cs-299520461966-default sudo: maxperrotta2011 : TTY=pts/15 ; PWD=/home/maxperrotta2011 ; USER=ro
2025-06-22T22:33:39.648339+00:00 cs-299520461966-default sudo: maxperrotta2011 : TTY=pts/18 ; PWD=/home/maxperrotta2011 ; USER=ro
== Proceso de configuración finalizado ==
maxperrotta2011@cloudshell:~ (stoked-castle-463721-u5)$ ls -ld /seguridad
drwx------ 2 root root 4096 Jun 22 21:35 /seguridad
maxperrotta2011@cloudshell:~ (stoked-castle-463721-u5)$
```

4. Metodología Utilizada

Para llevar a cabo este trabajo se adoptó un enfoque práctico y progresivo, basado en la implementación directa de conceptos teóricos vistos a lo largo de la cursada. La metodología se centró en el uso de herramientas reales del sistema operativo Linux para aplicar medidas de seguridad básicas.

En primer lugar, se realizó una investigación sobre los comandos y archivos más relevantes para la gestión de usuarios, permisos y servicios en entornos basados en Debian/Ubuntu. Se consultaron fuentes oficiales como la documentación de Ubuntu, artículos técnicos y recursos comunitarios.

Luego, se diseñó un script en Bash que automatiza ciertas tareas de “hardening” o refuerzo básico del sistema. Este script fue desarrollado y probado en el entorno Google Cloud Shell, el cual ofrece una terminal Linux accesible desde el navegador, sin necesidad de instalaciones adicionales.

Las pruebas consistieron en:

- Ejecutar el script paso a paso para verificar el comportamiento de cada comando.
- Comprobar la creación de usuarios y la asignación de permisos.
- Observar el resultado de las configuraciones del firewall.
- Visualizar logs reales del sistema para detectar intentos de acceso fallidos.

Se tomaron capturas de pantalla del proceso, que se incluyen como anexos. El código fuente del script fue subido a un repositorio público de GitHub para su revisión.

5. Resultados Obtenidos

La ejecución del script `seguridad_basica.sh` permitió verificar el correcto funcionamiento de cada una de las tareas propuestas en el caso práctico. A continuación se detallan los resultados obtenidos en cada sección:

- **Creación de usuario seguro**: Se solicitó el ingreso de un nombre de usuario, el cual fue creado exitosamente mediante el comando `useradd`, asignándole un shell restringido (`/usr/sbin/nologin`). Luego, se definió una contraseña segura mediante `passwd`, sin requerir datos personales adicionales, lo que permitió una creación simple y directa.

- **Configuración de carpeta protegida**: Se generó el directorio `/seguridad`, cuya propiedad se asignó a `root` y se limitaron sus permisos exclusivamente al superusuario. Esta configuración fue verificada mediante el comando `ls -ld /seguridad`, mostrando permisos `drwx------` como esperado.
- **Activación de firewall (UFW)**: Luego de instalar el paquete `ufw`, se habilitó el firewall con éxito. Se permitió únicamente el puerto SSH (22) y se bloqueó el puerto Telnet (23), considerado obsoleto e inseguro. El comando `ufw status verbose` mostró las reglas correctamente aplicadas.

- **Desactivación de servicios innecesarios**: Se intentó desactivar el servicio `bluetooth` como ejemplo de hardening. Aunque este servicio no estaba presente en Google Cloud Shell, el script reaccionó adecuadamente, mostrando un mensaje informativo sin provocar errores.

- **Revisión de logs de seguridad**: Se utilizaron los comandos `grep` y `tail` para obtener las últimas entradas del archivo `/var/log/auth.log` relacionadas con intentos de acceso fallidos. El resultado mostró eventos reales registrados en el sistema, lo que confirma la utilidad de los logs para tareas de monitoreo.

En todos los casos, el script mostró mensajes claros al usuario y se comportó de forma estable. El entorno elegido (Google Cloud Shell) resultó adecuado para pruebas simples de configuración, permitiendo trabajar con privilegios elevados (`sudo`) y ejecutar comandos administrativos sin inconvenientes.

6. Conclusiones

A través de este trabajo integrador fue posible aplicar de forma práctica varios conceptos clave relacionados con la seguridad en sistemas operativos Linux. La experiencia permitió no solo profundizar el conocimiento sobre comandos como useradd, chmod, ufw y grep, sino también comprender su propósito real en el contexto de la administración del sistema.

Una de las conclusiones más relevantes es que muchas medidas básicas de seguridad pueden implementarse fácilmente mediante scripts, lo que facilita la gestión de sistemas y reduce la posibilidad de errores humanos. El hecho de automatizar tareas como la creación de usuarios, la configuración de permisos o la activación de un firewall demuestra cómo los conocimientos teóricos se pueden convertir en acciones concretas.

Asimismo, trabajar en un entorno real como Google Cloud Shell ofreció la posibilidad de probar configuraciones sin comprometer un sistema local, permitiendo experimentar, equivocarse y corregir en un entorno seguro. Esto resultó fundamental para entender el comportamiento del sistema y reforzar la confianza en el uso de la terminal.

Por último, el desarrollo del script permitió ejercitar la lógica, la resolución de problemas y el razonamiento técnico, habilidades clave en la carrera de programación. También se reafirmó la importancia de los registros del sistema como herramienta de auditoría, y se reconoció la utilidad de medidas simples como restringir shells, limitar servicios y proteger directorios sensibles.

7. Bibliografía

- Ubuntu Documentation. (2024). *Security* – https://ubuntu.com/security
- Arch Wiki. (2024). *Security* – https://wiki.archlinux.org/title/Security
- Linuxize. (2023). *How to Use UFW Firewall on Ubuntu* – https://linuxize.com/post/ufw-essentials-common-firewall-rules-and-commands/
- GNU Project. (2022). *Bash Manual* – https://www.gnu.org/software/bash/manual/bash.html
- Silberschatz, A., Galvin, P., & Gagne, G. (2018). *Operating System Concepts* (9th ed.). Wiley.
- DigitalOcean. (2024). *How To Manage Users and Groups on Linux* – https://www.digitalocean.com/community/tutorials/how-to-manage-users-and-groups-on-linux
- Fecha de acceso general a los recursos: 19 al 22 de junio de 2025.

8.Anexos

8.1 Link Repositorio:

https://github.com/MaximoPerrotta/seguridad-linux-ayso/tree/main

8.2 Link Video:

https://youtu.be/SdtN-jHVP_c