**Trabajo Práctico – Seguridad en Sistemas Operativos**

Alumnos:

Rodrigo Agüero - [aguerorodrigo91@gmail.com](mailto:aguerorodrigo91@gmail.com)

Federico Villalba Petrelli -

Materia: Sistemas Operativos

Profesor:

Fecha de Entrega: 05 de junio de 2025

**Índice**

1. Introducción

2. Marco Teórico

3. Caso Práctico

4. Metodología Utilizada

5. Resultados Obtenidos

6. Conclusiones

7. Bibliografía

8. Anexos

**Introducción**

La seguridad en sistemas operativos es un aspecto fundamental para proteger la

información y los recursos del sistema. Este trabajo analiza las principales técnicas de

protección y muestra un ejemplo práctico de aplicación de permisos y auditoría en

Linux.

El RBAC es una técnica para gestionar permisos de manera eficiente, asignando roles a los usuarios en lugar de permisos individuales. Esto simplifica la administración y mejora la seguridad del sistema al agrupar permisos en roles específicos según las funciones de los usuarios

**Marco Teórico**

El RBAC consiste en definir roles que contienen permisos específicos y asignar estos roles a los usuarios. Aunque Linux no tiene una implementación nativa de RBAC, se puede simular usando grupos y permisos, donde los roles se gestionan como grupos de usuarios con permisos particulares.

**Caso Práctico**

El ejercicio práctico implica la creación de un entorno Linux con roles de administrador, editor y lector. Se asignarán permisos a archivos y directorios según estos roles, y los usuarios se añadirán a los grupos correspondientes para gestionar el acceso.

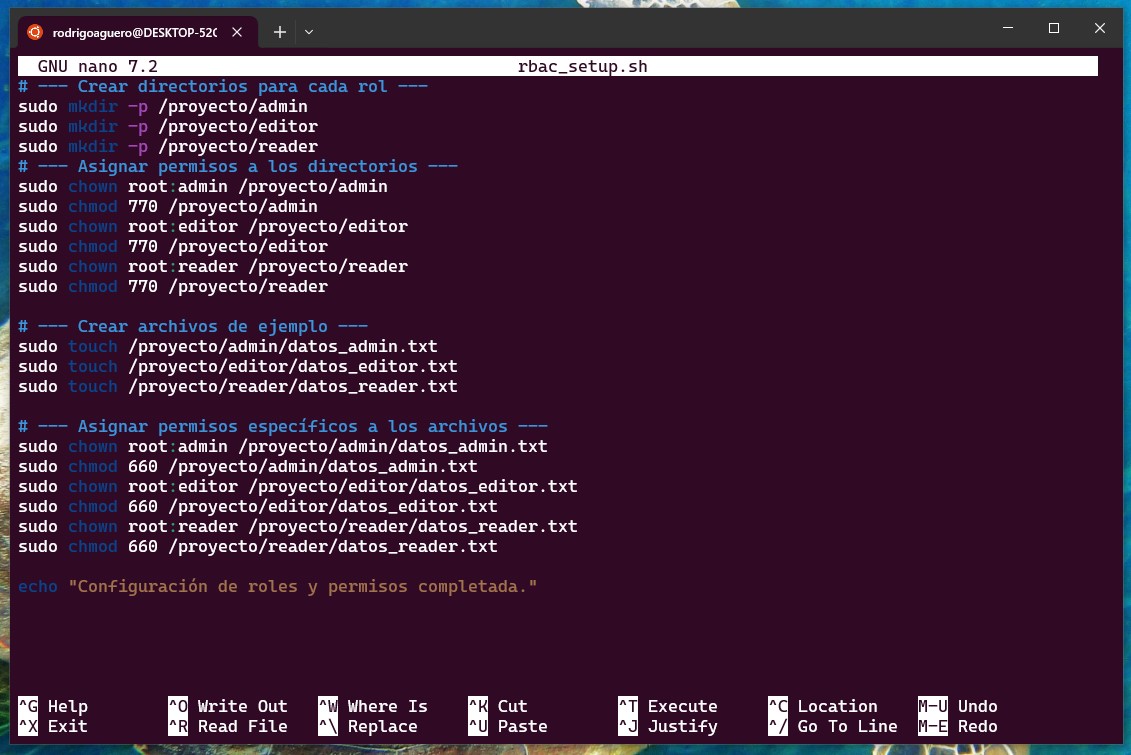
Los pasos para la implementación son:

* Crear grupos que representen los roles.
* Crear usuarios y asignarlos a los grupos.
* Configurar permisos en archivos y directorios.
* Verificar el control de acceso.

**Código Fuente (Script Bash)**

Se proporciona un script Bash (rbac\_setup.sh) para automatizar la configuración:

* **Creación de roles (grupos):** admin, editor, reader.
* **Creación de usuarios y asignación a roles:** admin\_user (al grupo admin), editor\_user (al grupo editor), reader\_user (al grupo reader).
* **Creación de directorios para cada rol:** /proyecto/admin, /proyecto/editor, /proyecto/reader.
* **Asignación de permisos a los directorios:**
  + /proyecto/admin: propietario root:admin, permisos 770.
  + /proyecto/editor: propietario root:editor, permisos 770.
  + /proyecto/reader: propietario root:reader, permisos 770.
* **Creación de archivos de ejemplo:** datos\_admin.txt, datos\_editor.txt, datos\_reader.txt en sus respectivos directorios.
* **Asignación de permisos específicos a los archivos:**
  + datos\_admin.txt: propietario root:admin, permisos 660.
  + datos\_editor.txt: propietario root:editor, permisos 660.
  + datos\_reader.txt: propietario root:reader, permisos 660.

****

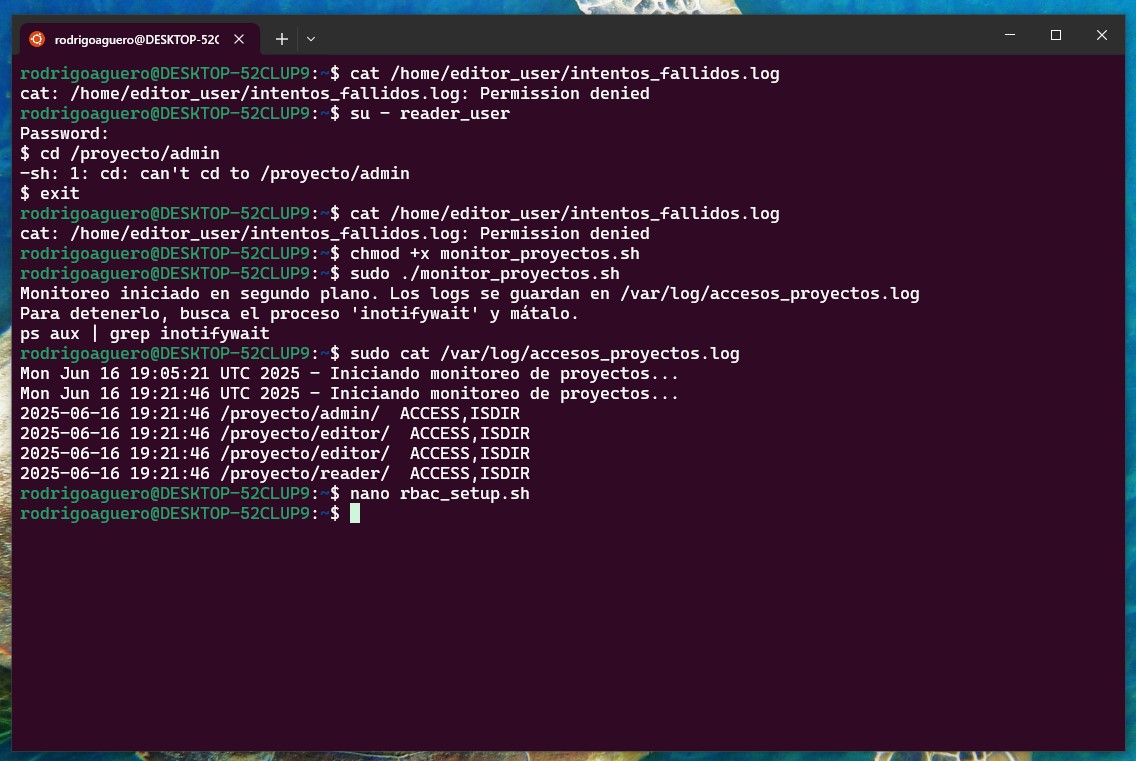
En esta captura se puede ver el archivo creado rbac\_setup.sh con los grupos y usuarios.

**Cómo usar el script**

1. Guardar el código en un archivo llamado rbac\_setup.sh.
2. Dar permisos de ejecución: chmod +x rbac\_setup.sh.
3. Ejecutar con privilegios: sudo ./rbac\_setup.sh.

**Verificación**

Para verificar el control de acceso:

* Iniciar sesión con cada usuario (admin\_user, editor\_user, reader\_user).
* Intentar acceder a los archivos correspondientes.
* Verificar que solo los usuarios con los permisos adecuados puedan acceder.

En esta captura se puede ver los intentos de accesos a proyectos de otros grupos y roles. Para monotiear los accesos no permitidos ultilizamos “inotify”.

Esta es una alternativa mucho más potente y cercana a la auditoría, aunque no es una auditoría de "acceso denegado", sino de "acceso intentado". inotify es una característica del kernel de Linux que permite a las aplicaciones monitorizar eventos del sistema de archivos (como apertura, cierre, modificación, acceso, etc.). inotify-tools es un conjunto de utilidades de línea de comandos que simplifican el uso de inotify.

**Conclusiones**

El uso de grupos en Linux simula eficazmente el control de acceso basado en roles, lo que facilita la gestión de permisos y mejora la seguridad del sistema. La automatización mediante scripts simplifica la implementación y el mantenimiento de estos controles.

**Entregables**

El trabajo debe incluir:

* El informe en formato PDF.
* El script fuente (rbac\_setup.sh).
* Recursos utilizados (manuales, enlaces).
* Bibliografía.