



Esquemas de seguridad en archivos

Presentado por:

***Hernández Saldívar Héctor Saúl
Sistemas operativos 6***

Ing. Gunnar Eyal Wolf Iszaevich

Semestre 2025-1

Contenido

- 1. Introducción**
- 2. Esquemas de seguridad en archivos en Linux**
 - a. Modelo permisos POSIX**
 - b. Permisos avanzados (Setuid, Setgid y Sticky bit)**
 - c. Listas de Control de Acceso (ACLs)**
 - d. SELinux y AppArmor**
 - e. Auditoría de seguridad**
- 3. Esquemas de seguridad en archivos en Windows**
 - a. Listas de Control de Acceso (ACLs)**
 - b. Permisos heredados**
 - c. Auditorías (SACLs)**
 - d. Cifrado de Archivos (EFS y Bitlocker)**
 - e. Políticas de Grupo y Active directory**
- 4. Comparaciones**
 - a. Modelos de permisos**
 - b. Control de acceso avanzado**
 - c. Cifrado de Archivos**
 - d. Auditorias**
- 5. Conclusión**
- 6. Referencias**





Introducción



Vivimos en una era digital que trae muchos beneficios, pero también aumenta la vulnerabilidad de la información, que puede ser utilizada indebidamente. Por eso, es crucial priorizar la seguridad de la información al manipular datos en un sistema operativo. Los esquemas de seguridad en archivos son fundamentales para proteger los datos, evitando accesos no autorizados. Aunque estos esquemas varían entre sistemas operativos, se enfocará en comparar cómo se aplican en los dos más utilizados: Windows y Linux.



Esquemas de seguridad en archivos en Linux

Linux es un sistema operativo de código abierto con un enfoque en la seguridad gracias a su arquitectura robusta y un modelo de permisos estricto. La comunidad puede identificar y corregir problemas rápidamente. Aunque tiene menos virus y malware por su menor popularidad, no es completamente seguro.



Modelo permisos POSIX

(Portable Operating System Interface). Es un estándar basado en UNIX que define convenciones y estructuras para la interacción entre programas-SO

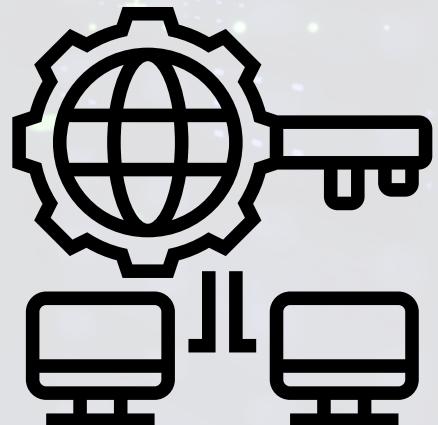
Inodes: Estructuras que contienen propietario (UID), grupo (GID), permisos y más.

El acceso se gestiona mediante una máscara de 9 bits (rwx) para el propietario, grupo y otros.

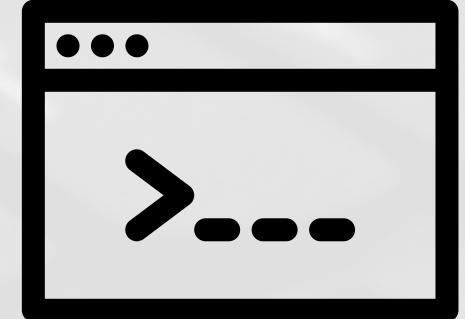
- Para el propietario 1-3
- Para el grupo 4-6
- Para otros del 7-9



rwxr-xr--
111101100



Comandos



chmod

Cambia permisos en un archivo

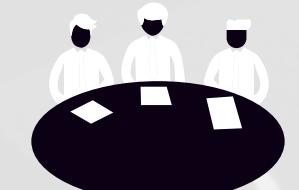
`chmod 754 archivo.txt`



chown

Cambia el propietario del archivo

`chown usuario archivo.txt`

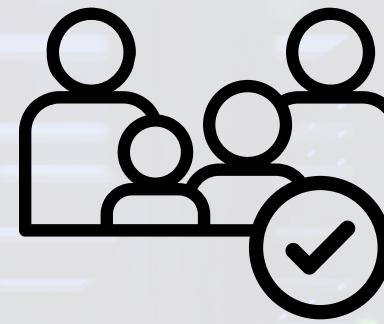


chgrp

Cambia el grupo propietario del archivo

`chgrp grupo archivo.txt`

Permisos avanzados



Setuid

Permite que un archivo ejecutable se ejecute con los privilegios del propietario en lugar del usuario que lo ejecuta

Está representado en el inode por el valor 4xxx

rwsr-xr-x

Setgid

Asegura que los archivos creados en un directorio hereden el grupo del directorio padre, útil para entornos colaborativos

Está representado en el inode por el valor 2xxx

rwxr-sr-x

Sticky bit

En directorios garantiza que solo el propietario de un archivo pueda eliminarlo o renombrarlo, incluso si otros tienen permisos de escritura en el directorio

Está representado en el inode por el valor 1xxx

rwxrwxrwt

Listas de Control de Acceso (ACLs)

Permiten extender los permisos más allá del modelo básico de POSIX, ofreciendo control detallado a múltiples usuarios y grupos

Cada ACL está definida por una serie de Entradas de Control de Acceso(ACEs), estas definen:

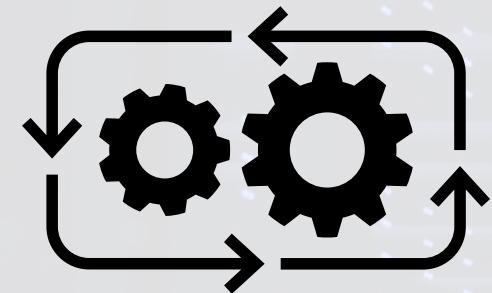
- Tipo de entrada
- Permisos
- Identificador(UID y GID)



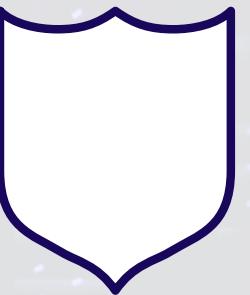
Comando setfacl

- *setfacl -m u:jose:rwx archivo.txt*
- *getfacl archivo.txt*
- *setfacl -x u:jose archivo.txt-*
- *setfacl -m m::rwx archivo.txt*
- *setfacl -b archivo.txt-*
- *getfacl archivo_origen.txt|setfacl --set-file=- archivo_destino.txt*





SELinux y AppArmor



SELinux (Security-Enhanced Linux)

Implementa un modelo MAC (Control de Acceso Obligatorio), donde los permisos son definidos por políticas de seguridad que no pueden ser modificadas por los usuarios.

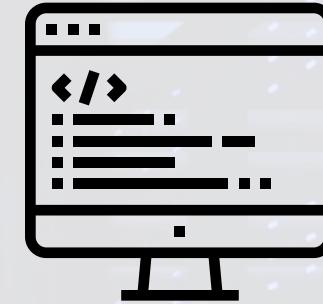
user_u:object_r:httpd_sys_content_t:s0

AppArmor

Es una alternativa que usa perfiles predefinidos en lugar de etiquetas para controlar los accesos de los procesos.

1. Perfiles estrictos
2. Complain mode

Comandos



- **comando getenforce:** Muestra el modo en el que está operando SELinux (enforcing, permissive o disabled).
- **comando semanage:** Permite gestionar las políticas de SELinux, modificar las etiquetas de seguridad de archivos y cambiar los permisos de acceso.

- **comando aa-status:** Muestra el estado de AppArmor y los perfiles que están cargados y activos.
- **comando aa-complain:** Cambia un perfil al modo de queja, permitiendo registrar los intentos de acceso sin bloquearlos.
- **comando aa-enforce:** Cambia un perfil al modo estricto, bloqueando los accesos no

Auditoria de seguridad

Componente clave para garantizar la integridad del sistema y detectar accesos no autorizados o actividades sospechosas en tiempo real

Audtid(Audit Daemon): Es el demonio responsable de capturar y registrar eventos de seguridad en el sistema.

- Acceso a archivos:
- Cambios de permisos
- Modificación de configuraciones críticas
- Accesos fallidos
- Ejecuciones de comandos.



Comandos principales:



auditctl

Se encarga de crear, modificar o eliminar las reglas de auditoría.

`auditctl -w /etc/passwd -p wa -k
passwd_change`

`auditctl -W /etc/passwd`

ausearch

Se encarga de realizar la búsqueda de los eventos creados y registrados

logs:

`ausearch -f /etc/passwd`

Filtrar por clave:

`ausearch -k passwd_change`

Buscar eventos por usuarios:

`ausearch -ua <user_ID>`

auditd:

Inicia, detiene o reinicia el demonio de auditoría en el sistema.

Esquemas de seguridad en Archivos en Windows

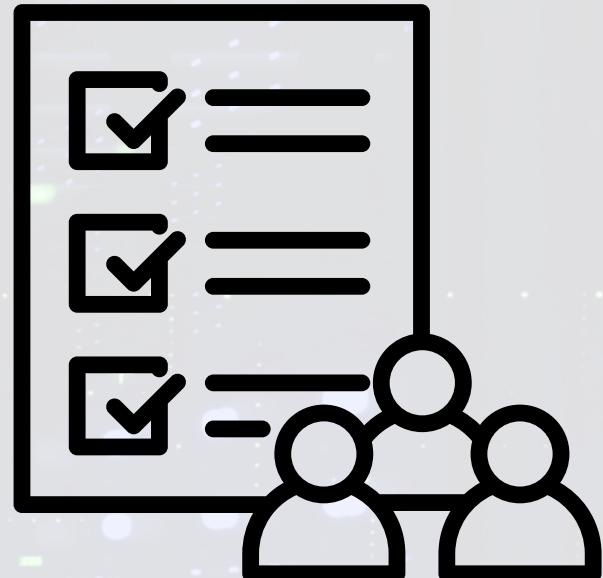
Windows es el sistema operativo más popular y usado, conocido por su facilidad de uso y amplia disponibilidad de software. A diferencia de Linux, es un software propietario, lo que significa que su código no es accesible para modificaciones. Su popularidad lo hace más vulnerable a ataques, virus y malware.

Listas de Control de Acceso (ACLs)

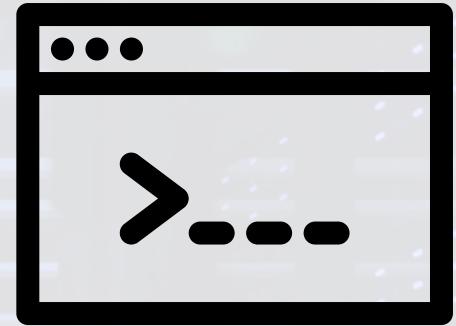
Son un mecanismo avanzado para gestionar permisos de archivos y carpetas de forma granular. Contienen entradas de control de acceso (ACEs)

Componentes clave del ACE:

- SID (Security Identifier)
- Máscara de acceso (Access Mask):
 - 0x001: Lectura (READ_DATA o FILE_READ_DATA para archivos, FILE_LIST_DIRECTORY para directorios).
 - 0x002: Escritura (WRITE_DATA o FILE_WRITE_DATA para archivos, FILE_ADD_FILE para directorios).
 - 0x004: Ejecución (EXECUTE o FILE_EXECUTE para archivos, FILE_TRAVERSE para directorios).
 - Además también incluye permisos avanzados como DELETE, WRITE_ATTRIBUTES y TAKE_OWNERSHIP.
- Permisos
- Tipo de ACE



Comandos



icacls:

para ver: *icacls archivo.txt*

para modificar:

icacls archivo.txt /grant usuario1:(RX)

para eliminar:

icacls archivo.txt /remove usuario1

Powershell

para ver: *Get-ACL archivo.txt*

para modificar:

```
$acl = Get-ACL archivo.txt
```

```
$perm = "usuario1", "Write", "Allow"
```

```
$ace = New-Object
```

```
System.Security.AccessControl.FileSystemAccessRule
```

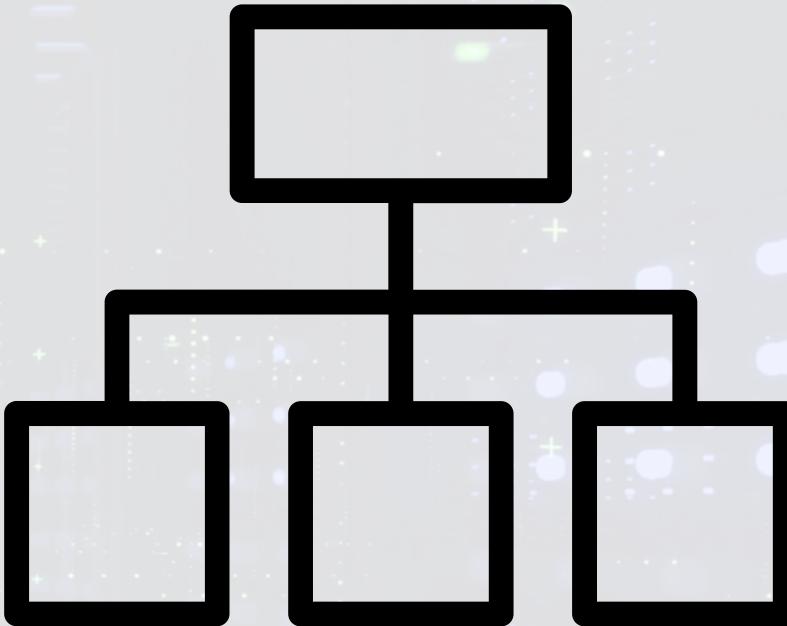
```
$perm
```

```
$acl.SetAccessRule($ace) Set-ACL archivo.txt $acl
```

Permisos heredados

Es un mecanismo que permite que los permisos asignados a una carpeta o directorio padre se transmitan automáticamente a sus subdirectorios y archivos

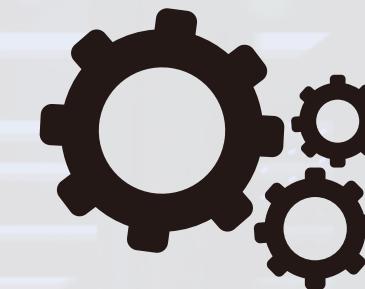
- Herencia explícita
- Herencia implícita



Pasos para detener herencia:

1. Haz clic derecho sobre el archivo o carpeta, selecciona Propiedades.
2. En la pestaña de Seguridad, haz clic en Opciones avanzadas.
3. Haz clic en Deshabilitar herencia.
4. Elige entre Convertir permisos heredados en permisos explícitos o Eliminar todos los permisos heredados.

Herencia con ACEs



- OI (Object Inherit):
- CI (Container Inherit):
- IO (Inherited Only):

Comandos

- ***icacls*** puede gestionar los permisos heredados desde la línea de comandos

Ver herencia: ***icacls "C:\Carpeta"***

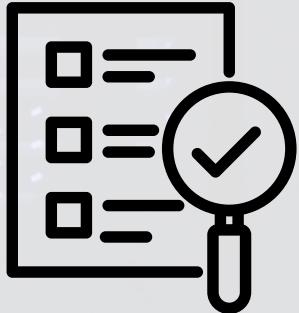
Eliminar herencia:

icacls "C:\Carpeta" /inheritance:d

- **PowerShell** también ofrece una forma avanzada de ver y modificar los permisos heredados

Get-ACL "C:\Carpeta"

Auditorías (SACLs)



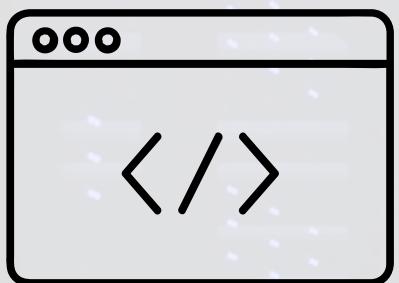
Las Listas de Control de Acceso del Sistema permiten definir auditorías sobre archivos y carpetas, registrando eventos como accesos fallidos o cambios en los permisos.

Cada entrada puede definir: Quién debe ser auditado, tipo de acceso o acciones y si el acceso es exitoso o fallido.

Los logs de auditoría del sistema almacenan los eventos, estos se pueden ver mediante el visor de eventos Windows:

- Acceso a archivos
- Intentos de modificación
- Accesos fallidos





Comandos



auditpol

Para ver la configuración actual:

*auditpol /get /category:**

Habilitar la auditoría de acceso a archivos y carpetas

*auditpol /set /subcategory:"Object Access"
/success:enable /failure:enable*

Habilitar auditoría para cambios de permisos
*auditpol /set /subcategory:"File System"
/success:enable /failure:enable*

Wevtutil

Ver los eventos registrados para un log en específico:

Wevtutil qe Security /f:text

Eliminar registros antiguos:

Wevtutil cl Security

Exportar registros de eventos:

Wevtutil epl Security C:\path\logs.evtx

Cifrado de Archivos (EFS y Bitlocker)

EFS(Encrypting File System):

Permite cifrar archivos individuales en sistemas de archivos NTFS.

Se cifran archivos y carpetas utilizando una clave única conocida como FEK (File Encryption Key)

FEK se gestionan de la siguiente manera:

- Clave del archivo (FEK)
- Cifrado de la FEK
- Haz clic derecho sobre el archivo o carpeta.
- Selecciona Propiedades.
- En la pestaña General, haz clic en Opciones avanzadas.
- Marca la casilla Cifrar contenido para proteger los datos.
- Haz clic en Aceptar.



Bitlocker:

Cifra todo el contenido de una partición o disco, proporcionando seguridad integral para todos los datos almacenados en esa unidad.

Puede usar un TPM (Trusted Platform Module)

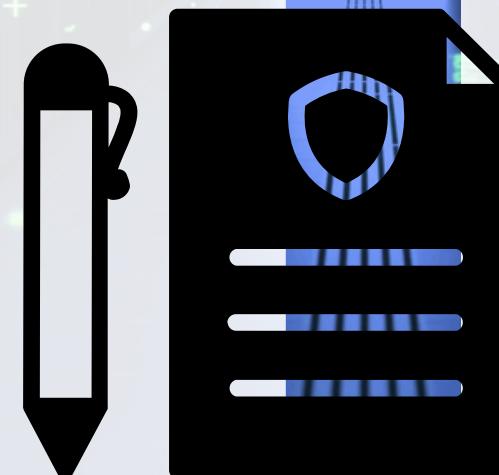
- Abre el Panel de control y selecciona Sistema y seguridad.
- Haz clic en Cifrado de unidad BitLocker.
- Selecciona la unidad que deseas cifrar y haz clic en Activar BitLocker.
- Sigue las instrucciones para elegir un método de autenticación (por ejemplo, usar un TPM, una contraseña o una unidad flash USB).
- BitLocker comenzará a cifrar la unidad.

Políticas de uso y Active directory.

Active directory

Es una base de datos estructurada que almacena información sobre los usuarios, grupos, equipos y otros recursos dentro de una red

- Usuarios:
- Grupos:
- Equipos:
- Unidades Organizativas (OUs)



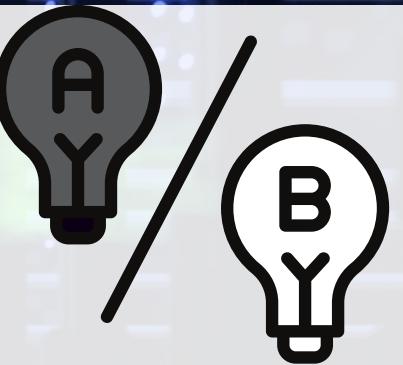
Políticas de uso

Son una característica de Active Directory que permite a los administradores aplicar configuraciones y restricciones centralizadas en todos los equipos y usuarios de una organización.

- Permisos de archivos.
- Configuraciones de seguridad.
- Configuraciones de software.

Centralización: Permite a las organizaciones implementar reglas y configuraciones de seguridad en una escala masiva.

Comparaciones:



Modelos de permisos

Linux: Utiliza un modelo básico y directo basado en permisos POSIX, expandible con ACLs y herramientas avanzadas como SELinux para mayor control.

Windows: Ofrece un enfoque granular con ACLs y SACLs, permitiendo una gestión muy específica de permisos.

Control de acceso avanzado

Linux: SELinux y AppArmor ofrecen control de acceso obligatorio (MAC), que añade una capa de seguridad más estricta sobre los permisos discrecionales de POSIX.

Windows: El control de acceso se gestiona a través de ACLs y SACLs, pero no incluye un sistema MAC como SELinux.

Cifrado de Archivos

Linux: Requiere herramientas externas como LUKS para el cifrado de discos y GPG para archivos individuales.

Windows: Integra soluciones nativas como EFS y BitLocker

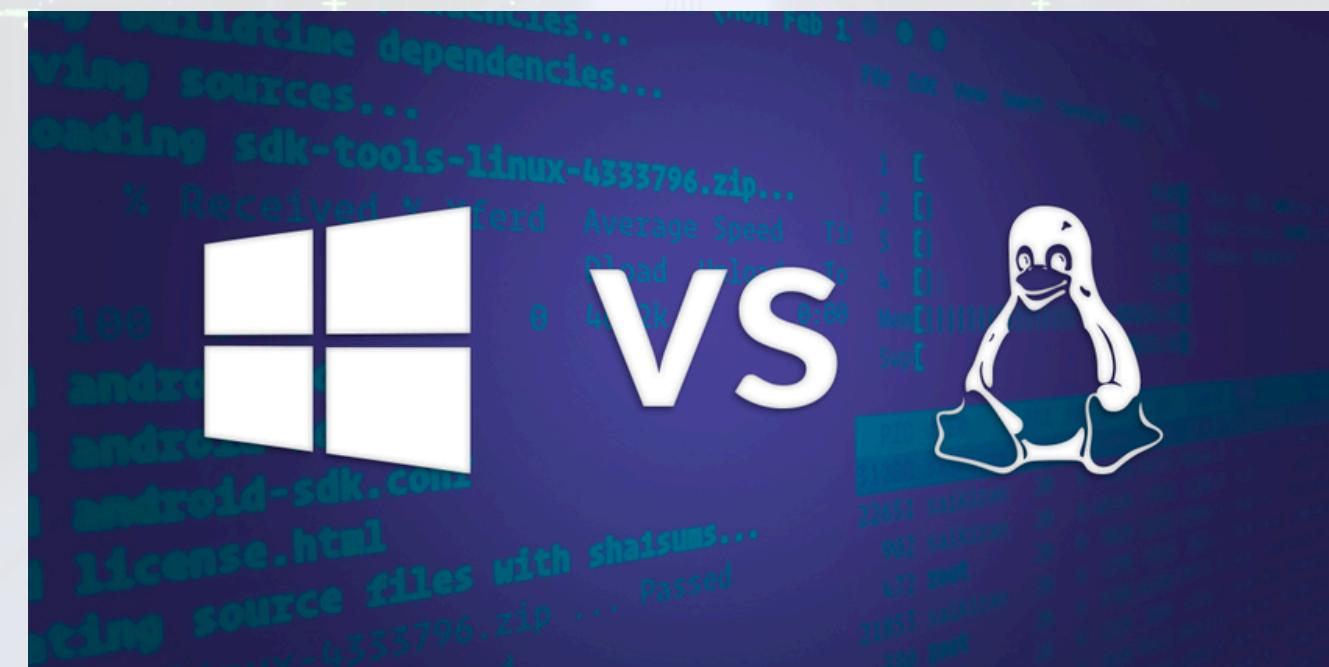
Auditorias

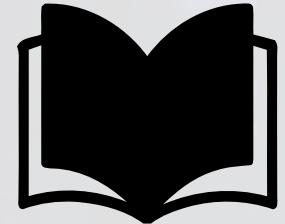
Linux: Herramientas como `auditd` permiten registrar eventos de seguridad de archivos

Windows: Ofrece un sistema más amigable con las SACLs y el Visor de eventos

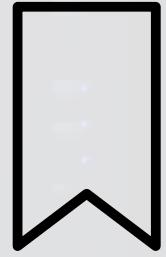
Conclusion:

Esta comparación entre Windows y Linux no busca determinar cuál es mejor en el aspecto de seguridad, sino cómo protegen sus archivos según su diseño. Linux ofrece seguridad sólida con permisos potentes, ideal para entornos críticos, mientras que Windows es flexible y eficiente para redes con muchos usuarios. Ambos son seguros dependiendo del entorno, con fortalezas y debilidades, por lo que no tiene sentido afirmar que uno es superior al otro.

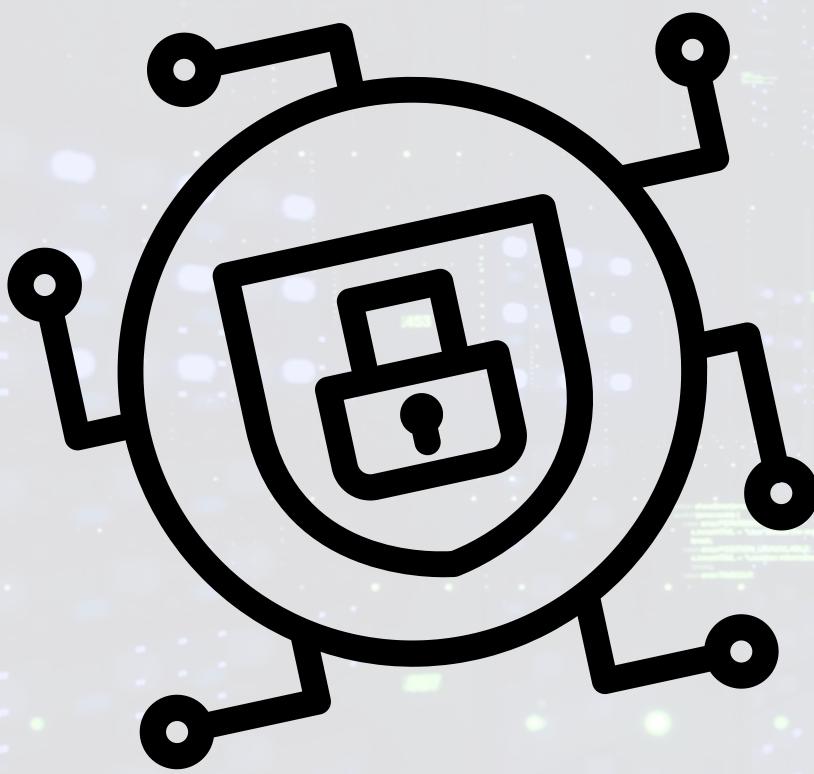




Referencias



- [1] J. Herrera, "Estándar POSIX, ¿qué es y para qué sirve?," *Guía Hardware*, 29 de junio de 2023. <https://www.guiahardware.es/estandar-posix/> (Accedido el 27 de septiembre de 2024).
- [2] "The Linux Documentation Project: Recent Updates," *Tldp.org*, 2015. https://tldp.org/sorted_howtos.html (Accedido el 27 de septiembre de 2024).
- [3] R. Love and Addison-Wesley, *Linux Kernel Development*, 3rd ed. Upper Saddle River: Addison-Wesley, 2015. (Accedido el 28 de septiembre de 2024).
- [4] "Documentation for Red Hat Products," *Red Hat Customer Portal*. <https://access.redhat.com/documentation/en-us/> (Accedido el 28 de septiembre de 2024).
- [5] Wibjorn, "Microsoft Learn: adquirir conocimientos que le abran las puertas en su carrera profesional", *learn.microsoft.com*. <https://learn.microsoft.com/es-es/> (Accedido el 1 de octubre de 2024).
- [6] Pavel Yosifovich, D. A. Solomon, and A. Ionescu, *Windows Internals, Part 1*. Microsoft Press, 2017. (Accedido el 3 de octubre de 2024).
- [7] V. Alonso, "¿Qué es más seguro Linux o Windows?," *sistemasoperativos.info*, 22 de mayo de 2023. [¿Qué es más seguro Linux o Windows? | Análisis 2024 \(sistemasoperativos.info\)](https://sistemasoperativos.info/que-es-mas-seguro-linux-o-windows-analisis-2024/) (Accedido el 3 de octubre de 2024).



**Gracias por
su atención**