El repte de Permisos, Propietaris i Seguretat

Sistemes Operatius Monolloc 1r B SMIX
8 d'Abril del 2026
Puertas Amaral, Erik

ÍNDEX

PART I - Cerca d'informació

1. Què és un UID i un GID a Linux i com es relacionen amb el nom d'usuari i de grup?

El **UID** User ID és el número únic que identifica un usuari, mentre que el **GID** Group ID identifica el seu grup principal. El sistema utilitza aquests números per gestionar permisos interns, relacionant-los amb els noms d'usuari i de grup mitjançant fitxers de configuració perquè siguin recognoscibles per les persones.

2. A Windows, què és el SID (Security Identifier) i què passa amb ell quan eliminem un usuari i el tornem a crear amb el mateix nom?

El **SID** Security Identifier és un codi únic que identifica cada compte d'usuari o grup a Windows. Si s'elimina un usuari i es torna a crear amb el mateix nom, el sistema li assigna un **SID nou**, per tant, Windows el considera un usuari diferent i no recuperarà automàticament els permisos que tenia l'anterior

3. Explica la diferència entre un permís explícit i un permís heretat a Windows.

Un permís explícit a Windows és aquell que s'assigna directament a un fitxer o directori concret per part d'un usuari o administrador. En canvi, un permís heretat és el que rep un objecte de manera automàtica des de la seva carpeta pare, facilitant que la configuració de seguretat es propagui cap avall en l'arbre de directoris.

4. A Linux, per a què serveix l'opció -R en les ordres chmod i chown?

A Linux, l'opció **-R** en les ordres `chmod` i `chown` s'utilitza per aplicar els canvis de permisos o de propietat de forma recursiva. Això permet que l'acció afecti no només el directori seleccionat, sinó també a tots els fitxers i subcarpetes que hi hagi al seu interior de cop

PART II - Tasca pràctica

BLOC A: Laboratori Avançat Ubuntu (Terminal)

1. Canvi de Propietat. Crea el fitxer proves.txt. Fes que el propietari sigui tecnic i el grup G_RRHH.

```
erik@erik-virtualbox:~$ sudo touch /tmp/MASTER_LINUX/proves.txt
erik@erik-virtualbox:~$ sudo chown tecnic:G_RRHH /tmp/MASTER_LINUX/proves.txt
erik@erik-virtualbox:~$
```

Imatge 1: Fitxer proves

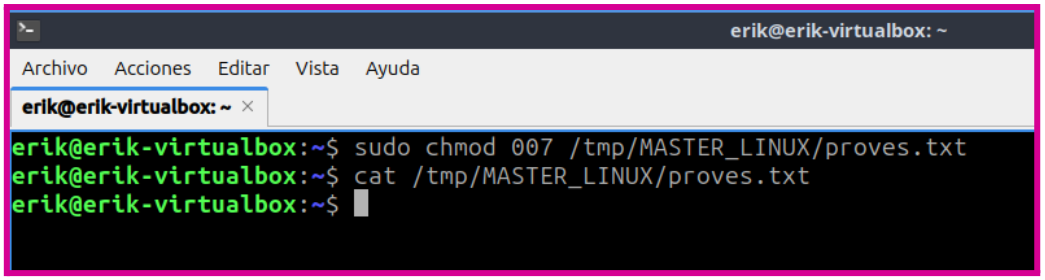
2. Restricció de Grup: Aplica permís 640. Pot el gestor llegir-lo? Pot el tecnic modificar-lo? Comprava-ho.

```
erik@erik-virtualbox:~$ sudo chmod 640 /tmp/MASTER_LINUX/proves.txt
erik@erik-virtualbox:~$
```

Imatge 2: Permis 640

3. El Propietari Impotent. Canvia el permís a 007. Pot el propietari (tecnic) llegir el seu propi fitxer? Per què?

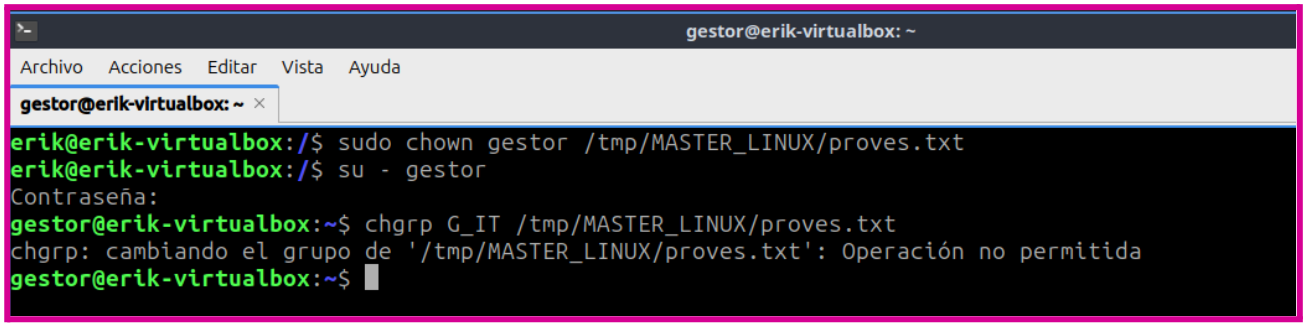
No. Si se intenta fer un `cat` al fitxer, el sistema et donarà un error de "Permís denegat". A Linux, els permisos es llegeixen d'esquerra a dreta en l'ordre: **Propietari**, **Grup** i **Altres**. Amb el permís **007**, el primer dígit (0) indica que el propietari no té cap permís (ni lectura, ni escriptura, ni execució). Tot i que el fitxer és seu i els "Altres" tenen control total (7), el sistema s'atura en comprovar que el propietari té un 0 i li bloqueja l'accés immediatament.



Imatge 3: Permis 007

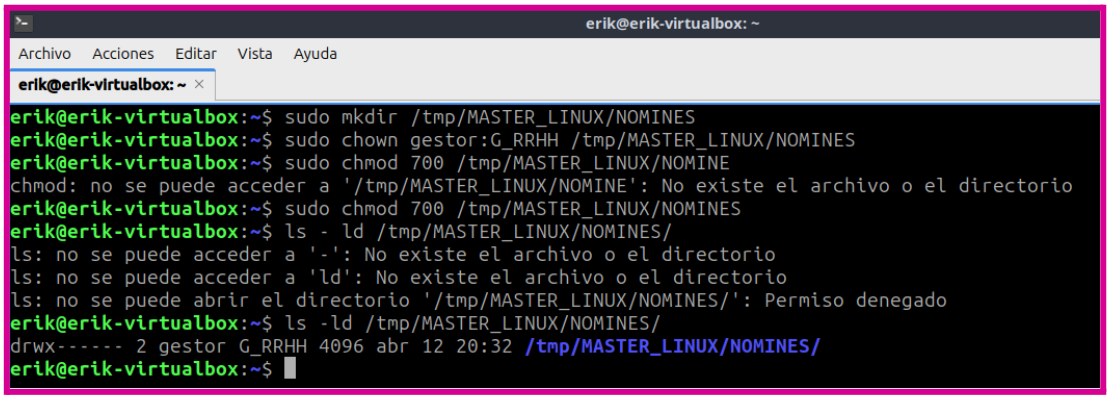
4. Traspàs de Fitxers. Com a root, canvia la propietat del fitxer proves.txt a l'usuari gestor. Ara intenta canviar-li el grup a G_IT des de la sessió de gestor. Et deixa?

No. El sistema et mostrarà un error tipus "operació no permesa" *operation not permitted*. A Linux, encara que siguis el propietari d'un fitxer en aquest cas, gestor, només pots canviar el grup del fitxer a un grup al qual **jo mateix pertany**. Com que l'usuari gestor pertany al grup G_RRHH i no al grup G_IT, el sistema li prohibeix fer aquest canvi per seguretat. Només el superusuari root té el poder de canviar un fitxer a qualsevol grup del sistema sense restriccions



Imatge 4: Traspàs de fitxer

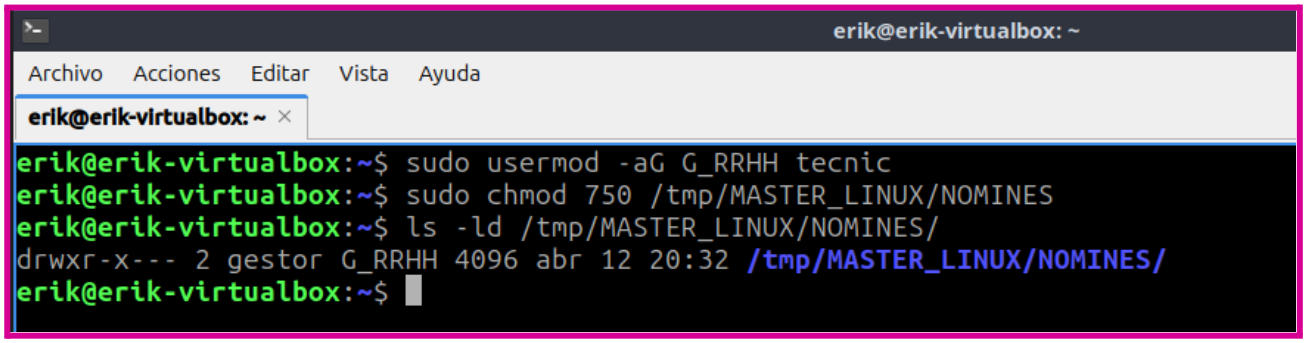
5. Permís de Directori. Crea la carpeta /tmp/MASTER_LINUX/NOMINES. Fes propietari a l'usuari gestor i grup G_RRHH amb permís 700.



Imatge 5:Permis 700

6. Escalada de Grup. Afegeix l'usuari tecnic al grup G_RRHH temporalment (usermod). Canvia el permís de la carpeta anterior a 750. Pot el tecnic entrar-hi ara?

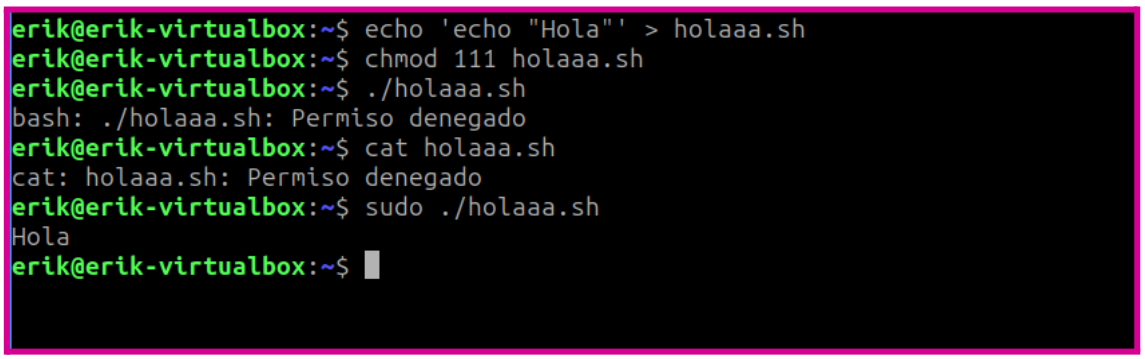
Sí. Un cop el tecnic tanqui la sessió i la torni a obrir (perquè s'actualitzin els seus grups), podrà entrar-hi. Amb el permís **750**, el segon dígit (**5**) indica que el grup G_RRHH té permisos de lectura i execució. Com que ara el tecnic forma part d'aquest grup, hereta aquests permisos i pot accedir al directori.



Imatge 6: Escalada de grup

7. L'execució cega. Crea un script hola.sh que faci un echo "Hola". Dona-li permís 111. Intenta executar-lo. Intenta llegir-ne el codi amb cat. Què passa?

En intentar executar l'script amb ./hola.sh, aquest funciona correctament i mostra el missatge "Hola" a la terminal, ja que el permís d'execució (1) està actiu per a l'usuari. Tanmateix, en intentar llegir el contingut del fitxer amb l'ordre cat hola.sh, el sistema retorna un error de "Permís denegat".



Imatge 7: Execució cega

8. Seguretat de Carpeta Pare. Crea un fitxer de root (permís 600) dins d'una carpeta on el tecnic té control total (777). Pot el tecnic esborrar el fitxer que no és seu?

Sí. El fitxer és de root i el tecnic no té permís ni per llegir-lo ni per escriure-hi (permís 600), el sistema li permetrà eliminar-lo.

```
erik@erik-virtualbox:~$ sudo chmod 777 /tmp/MASTER_LINUX
erik@erik-virtualbox:~$ sudo touch /tmp/MASTER_LINUX/root_secret.txt
erik@erik-virtualbox:~$ sudo chmod 600 /tmp/MASTER_LINUX/root_secret.txt
erik@erik-virtualbox:~$ rm /tmp/MASTER_LINUX/root_secret.txt
rm: ¿borrar el fichero regular vacío '/tmp/MASTER_LINUX/root_secret.txt'  protegido contra escritura? (s/n) n
erik@erik-virtualbox:~$ su - tecnic
Contraseña:
tecnic@erik-virtualbox:~$ rm /tmp/MASTER_LINUX/root_secret.txt
rm: ¿borrar el fichero regular vacío '/tmp/MASTER_LINUX/root_secret.txt'  protegido contra escritura? (s/n)
```

Imatge 8:Seguretat carpeta pare

9. Mascarada (Umask). Mira el valor de umask. Crea un fitxer nou i explica quins permisos ha rebut per defecte i per què.

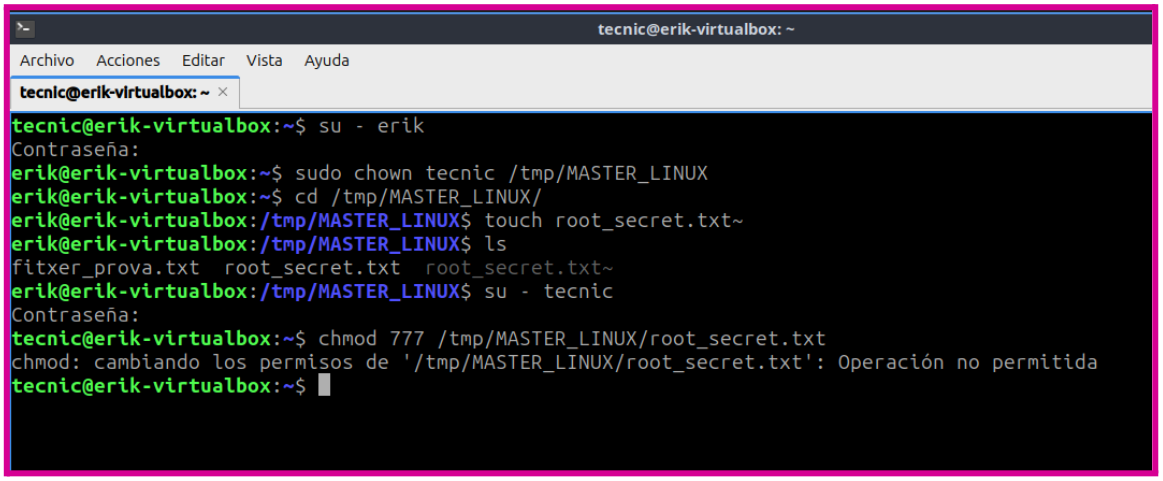
El valor de la **umask** determina els permisos per defecte en crear fitxers, actuant com un filtre que "resta" permisos a la base inicial de 666 per a fitxers lectura i escriptura. Per exemple, si la meva umask és 0022, el fitxer rep el permís 644 perquè es treu el permís d'escriptura al grup i als altres (666 - 022 = 644), deixant al propietari amb control de lectura i escriptura mentre que els altres només poden llegir-lo.

```
tecnic@erik-virtualbox:~$ cd /tmp/MASTER_LINUX/
tecnic@erik-virtualbox:/tmp/MASTER_LINUX$ touch fitxer_prova.txt
tecnic@erik-virtualbox:/tmp/MASTER_LINUX$ ls -l fitxer_prova.txt
-rw-rw-r-- 1 tecnic tecnic 0 abr 12 21:47 fitxer_prova.txt
tecnic@erik-virtualbox:/tmp/MASTER_LINUX$
```

Imatge 9: Umask

10. Propietari de Directori. Canvia el propietari de /tmp/MASTER_LINUX a tecnic. Pot ara el tecnic canviar els permisos de tots els fitxers de dins encara que siguin de root?

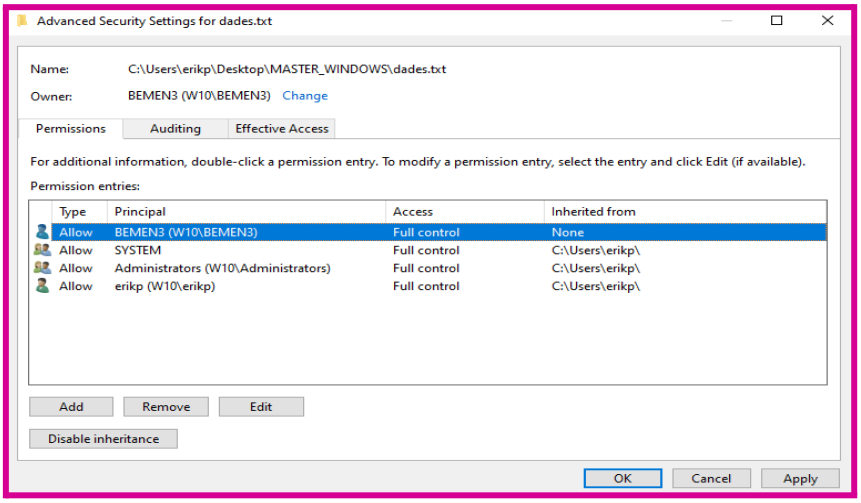
No, el sistema dona un error de "Operació no permesa"



Imatge 10: canv de propietari

BLOC B: Laboratori Avançat Windows 10 (Interfície i CLI)

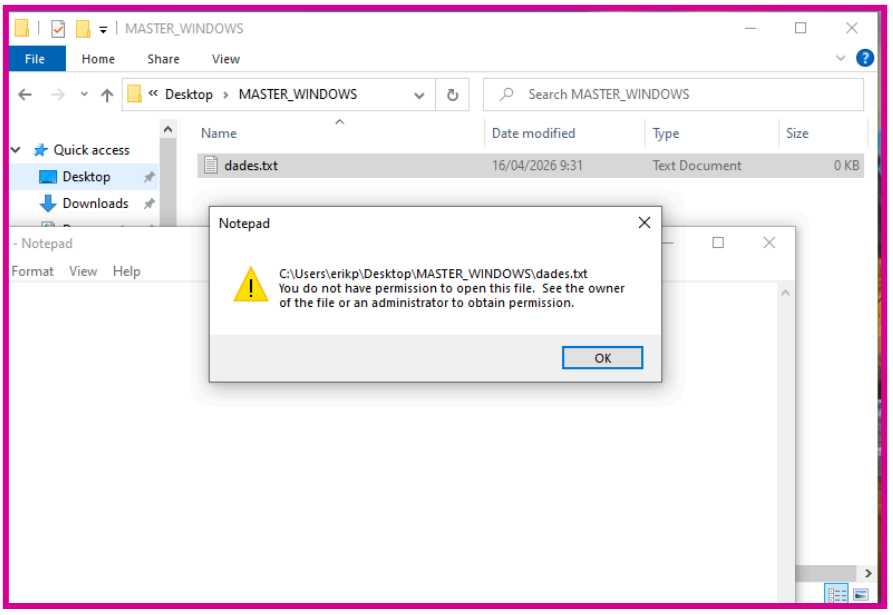
11. Canvi de Propietari (Owner). Crea dades.txt. A Seguridad → Opciones avanzadas, canvia el "Propietario" de Administradores a l'usuari tecnic.



Imatge : 11 Cambi de propietari

12. Auto-bloqueig. Com a tecnic (propietari), treu-te a tu mateix de la llista de permisos. Intenta obrir el fitxer. Com a propietari, pots tornar-te a donar permís?

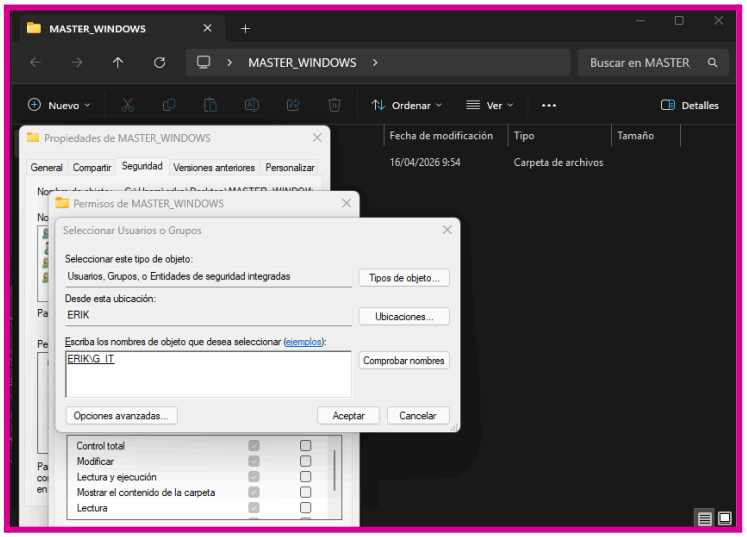
Com a propietari, he intentat eliminar-me de la llista de permisos, però primer he hagut de **deshabilitar l'herència** a les opcions avançades per poder fer canvis explícits al fitxer. Un cop eliminat l'usuari, en intentar obrir **dades.txt**, el sistema m'ha donat un error d'**accés denegat** perquè ja no figurava a la llista d'accessos. Tot i així, com que sóc el **propietari** , he pogut entrar de nou a la configuració de seguretat i **tornar-me a donar permisos** de "Control total", recuperant així l'accés complet al fitxer



Imatge 12: Auto bloqueig

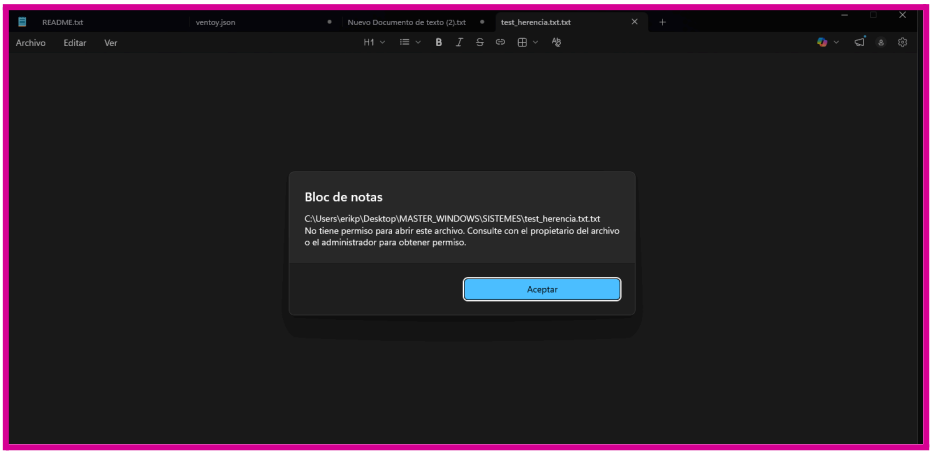
13. Herència de Grups. Crea la carpeta SISTEMES. Dona "Control Total" al grup G_IT. Comprova si el fitxer que creis a dins hereta aquest permís.

El fitxer ha rebut automàticament el permís de **"Control Total"** per al grup **G_IT** sense que jo hagi hagut de configurar-lo manualment. Això demostra el funcionament de l'**herència** (Inheritance), on els objectes fills reben la configuració de seguretat del directori pare per defecte



Imatge 13: Herencia de grups

14. Prevalença de Denegació. Afegeix el grup Usuarios amb "Lectura". Afegeix l'usuari tecnic amb "Denegar Lectura". Què passa quan el tecnic intenta llegir-lo?



Imatge 14: Prevaleça de degenació

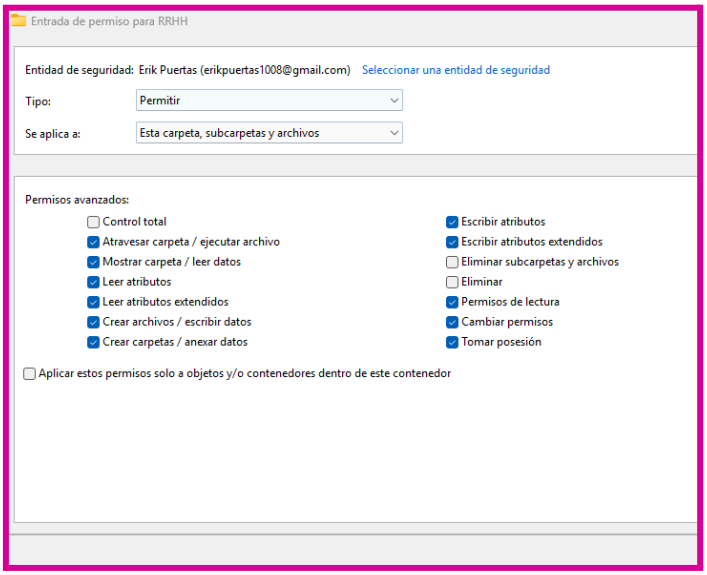
15. Moure vs Copiar. Copia un fitxer d'una carpeta amb permisos restrictius a una carpeta pública. Mira si els permisos es mantenen o canvien. Ara fes el mateix però movent el fitxer.

Copiar crea un objecte nou que rep l'herència de la carpeta de destí, mentre que moure simplement canvia la ubicació de l'objecte dins de la mateixa partició, mantenint la seva seguretat original per integritat.

Imatge 15: Moure vs copiar

16. Permisos de Creació. Configura la carpeta RRHH perquè l'usuari gestor pugui crear fitxers però no pugui esborrar-ne cap (permisos avançats: Eliminar desmarcat).

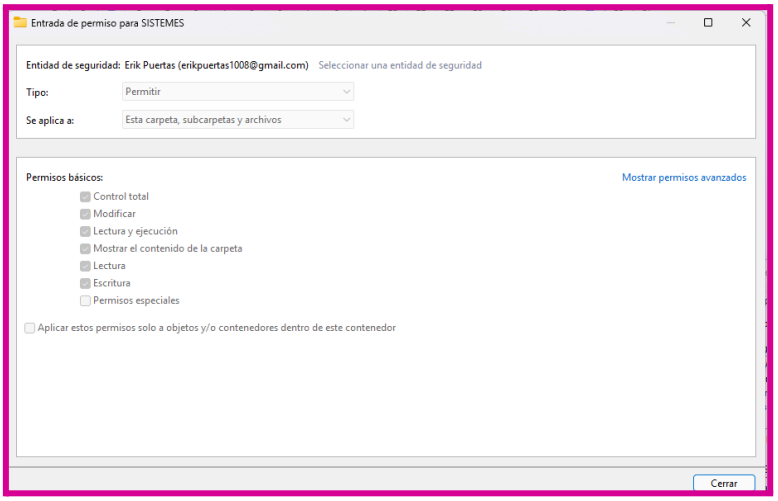
El sistema m'ha permès crear el document sense problemes, però en intentar eliminar-lo, Windows ha llançat un missatge de **"Acceso denegado a la carpeta"**, indicant que es necessiten permisos per a realitzar aquesta acció. Això confirma que la configuració de permisos avançats s'ha aplicat correctament



Imatge 16: Permisos de creació

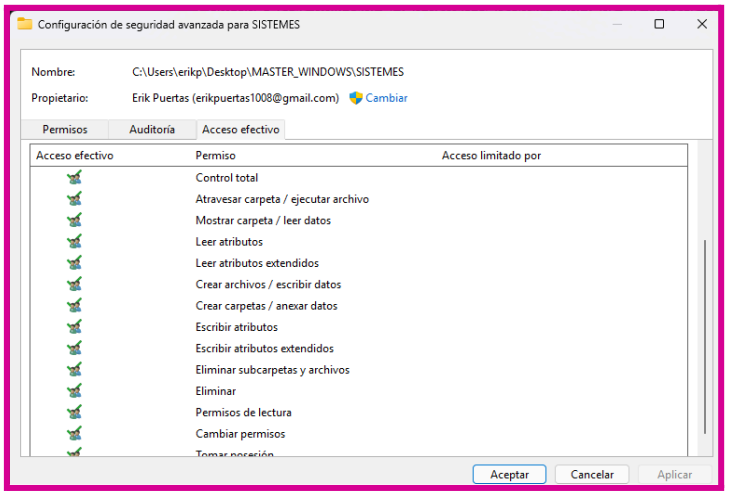
17. L'usuari d'IT tafaner. Intenta entrar a la carpeta del gestor amb l'usuari tecnic. Quan surti l'avís d'UAC (Escalada de privilegis), nega't. Pots veure el contingut?

Encara que el meu usuari tingui privilegis d'administrador, el sistema NTFS protegeix la privacitat de les carpetes d'altres usuaris. Si em nego a l'escalada de privilegis (UAC), el sistema manté la restricció d'accés i no em permet llistar ni llegir els fitxers que hi ha a l'interior



Imatge 17: Escalada de privilegis

18. Permisos Efectius. Utilitza la pestanya "Acceso efectivo" per veure quins permisos reals té el gestor sobre la carpeta de SISTEMES.



Imatge 18: Permisos efectius

19. Atribut de Sistema. Des del CMD, posa el fitxer dades.txt com a fitxer de sistema i ocult (attrib +s +h). Apareix a l'explorador de fitxers normal?

```
PS C:\Users\erikp\Desktop\MASTER_WINDOWS> attrib +s +h dades.txt
PS C:\Users\erikp\Desktop\MASTER_WINDOWS>
```

Imatge 19: Atributs del sistema

20. Audit de Seguretat. Exporta amb icacls tots els permisos de C:\MASTER_WINDOWS a un fitxer i cerca on apareix el SID de l'usuari tecnic.

He realitzat l'exportació dels permisos de la carpeta C:\MASTER_WINDOWS utilitzant l'eina icacls. Tot i que inicialment he tingut errors d'accés denegat per intentar executar-ho sobre carpetes del sistema, la forma correcta de fer-ho és situant el terminal directament dins del directori de la pràctica.

En executar l'ordre icacls . /save permisos_auditoria.txt /t, el sistema ha generat correctament el fitxer amb totes les llistes de control d'accés. En obrir aquest fitxer amb el Bloc de notes, he pogut localitzar el **SID** del meu usuari, que és el codi que comença per **S-1-5-21-**.... Això demostra que Windows no assigna els permisos basant-se en el nom de l'usuari , sinó en aquest identificador numèric únic, la qual cosa garanteix que la seguretat es mantingui encara que es canviï el nom del compte.

PART III - Reflexió final

1. Després de la prova 8 de Linux, per què creus que és tan perillós donar permís d'escriptura (w) a tothom en una carpeta, encara que els fitxers de dins siguin de root?

És extremadament perillós perquè, com has comprovat a la prova 8, el permís d'escriptura sobre una carpeta atorga la capacitat d'afegir o eliminar els objectes que hi ha dins. Això implica que qualsevol usuari podria esborrar fitxers crítics del sistema o de l'administrador (root), encara que no tingués permisos per llegir-los o modificar-ne el contingut. En essència, qui té control sobre la carpeta té el poder de decidir si un fitxer existeix o no.

2. Quina diferència has notat entre el concepte de "Propietari" a Linux i a Windows a l'hora de recuperar l'accés a un fitxer bloquejat?

La principal diferència rau en l'autonomia per gestionar permisos:

- **A Linux:** El propietari té el dret legal de canviar els permisos d'un objecte, però si ell mateix es treu els permisos , encara pot recuperar l'accés tornant a executar **chmod** perquè el sistema reconeix la seva autoritat sobre l'atribut.

- **A Windows:** El concepte de propietari, és més potent per recuperar accessos. Si un propietari es bloqueja a si mateix en les ACL, pot utilitzar el seu dret de propietat per prendre possessió de nou o modificar la llista de seguretat avançada per restaurar els seus privilegis, cosa que permet una recuperació d'accés més visual i estructural.

3. Quins avantatges té utilitzar grups (G_IT, G_RRHH) en lloc d'assignar permisos usuari per usuari en una empresa de 50 treballadors?

Utilitzar grups en lloc d'assignar permisos usuari per usuari permet una gestió molt més àgil i eficaç, ja que en una empresa de 50 treballadors seria extremadament lent i propens a errors configurar cada accés individualment