GESTIÓN DE LOS PERMISOS DE ACCESO DE LOS SISTEMAS OPERATIVOS

Índice

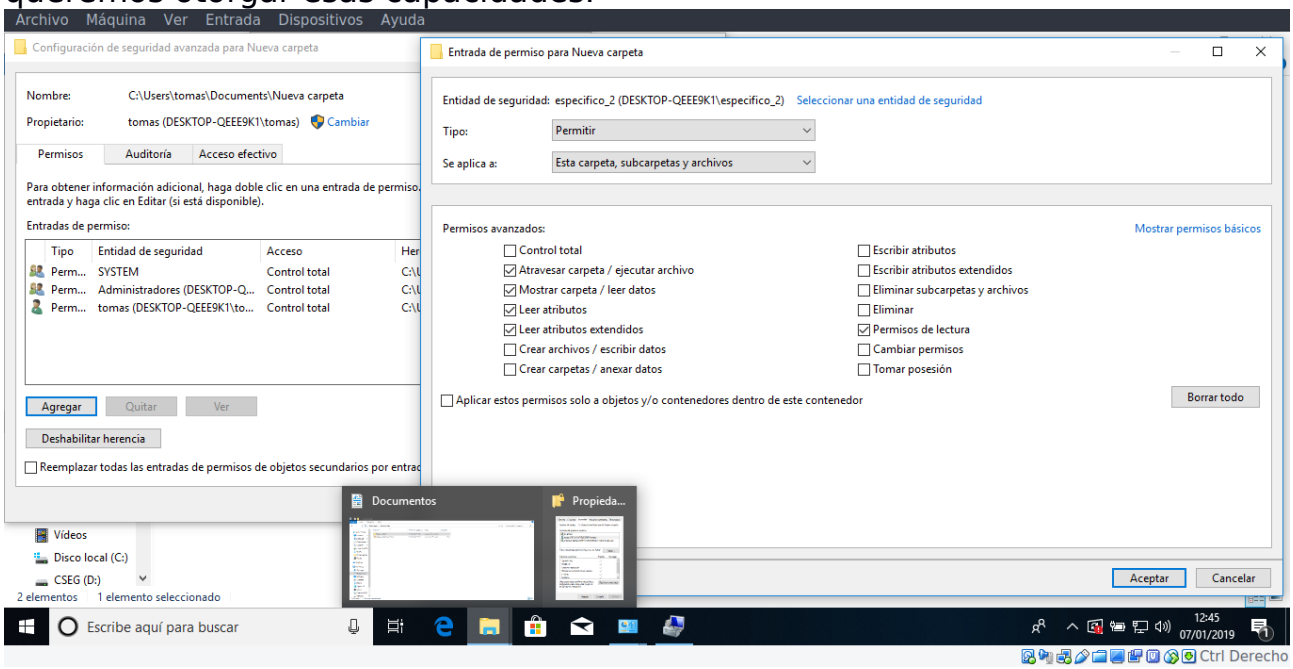
1. INTRODUCCIÓN

En este documento vamos a repasar las opciones que tenemos en cuanto a la configuracíón de los permisos de acceso que podemos asignar a los ficheros y carpetas de nuestros equipos.
Repasaremos las herramientas que, tanto Windows, como Linux, nos ofrecen el respecto.

> ***Trabajaremos con los permisos de acceso denominados DAC (Control de acceso discrecional). Puedes ver el punto 2.3 de los apuntes del tema para recordar a qué se refiere este termino.***

2. WINDOWS

Windows nos proporciona una funcionalidad muy completa para asignar permisos, permitiendo filtrar muy bien a qué entidad (usuario, equipo, grupo..) queremos otorgar esas capacidades.



Para poder realizar esta asignación tan concreta de tanta cantidad de permisos, los sistemas Windows hacen uso de listas de control de acceso.

3. **LINUX**

En este apartado vamos a repasar algunos conceptos que ya visteis o estáis viendo en otros módulos.

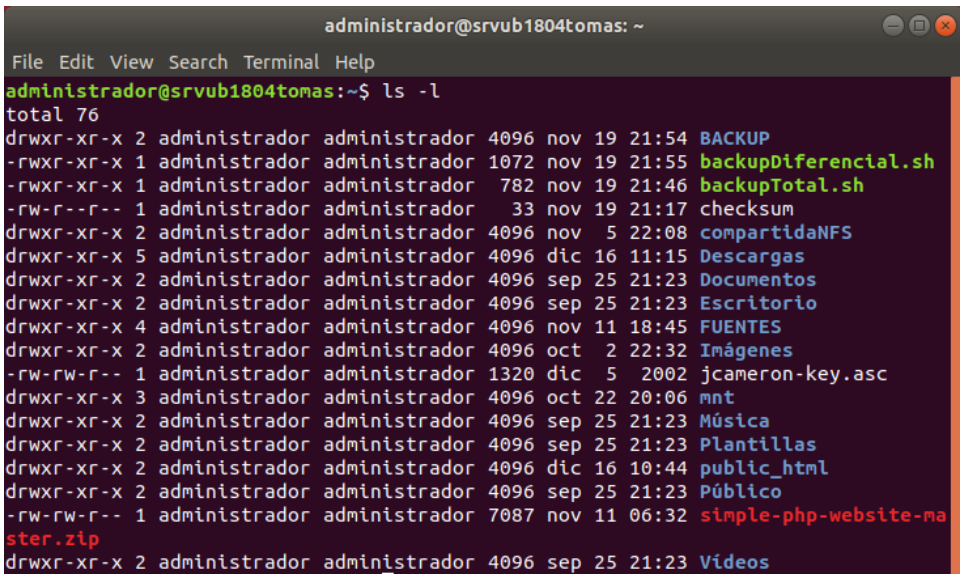
A) Permisos básicos

Manual en la web

Se trata de los 3 grupos de 3 bits, que ya conocemos, para asignar los permisos de lectura, escritura y ejecución, entendiendo que la ejecución significa cosas distintas dependiendo del tipo de fichero. Por ejemplo ejecución para un directorio es poder acceder a su contenido con un cd.
Tenemos, además, un primer bit que nos indica qué tipo de fichero es. La estructura de estos permisos es:

Podemos consultarlos con el comando:

```
$ls -l
```



Tenemos los siguientes comandos para gestionar estos permisos:

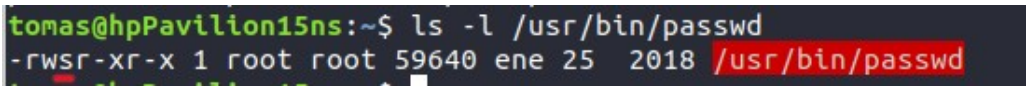
- ***CHMOD: Cambiar los permisos. Se pueden pasar en formato octal, decimal y otros.***
- ***CHOWN: Cambia el propietario, y si queremos el grupo también, de un fichero***
- ***CHGRP: Cambia el grupo propietario únicamente.***

B) Permisos extendidos

Es posible establecer algunos permisos especiales, que nos permiten modificar el comportamiento del sistema. Vamos a conocer los siguientes:

- **SUID**: El bit SUID es una extensión del permiso de ejecución. Se utiliza en escasas ocasiones y sirve para que cuando un usuario ejecute una aplicación, ésta se ejecute con permisos del usuario propietario en lugar de hacerlo con los del usuario que ejecuta la aplicación, es decir, es equivalente a que sea ejecutada por el propietario.
  Para activar el bit SUID, **se puede ejecutar el comando chmod u+s nombre_archivo o sumar 4000 al número en octal si utilizamos dicho sistema**. También se puede hacer lo mismo para el grupo, es el denominado bit SGID sumando 2000 al número en octal. Activar los bits SUID ó SGID puede ocasionar problemas de seguridad sobre todo si el propietario es root.

El ejecutable passwd es un ejemplo de este bit



- **SGID**: Si aplicamos el bit SGID a una carpeta, todas las subcarpetas y archivos creados dentro de dicha carpeta tendrán como grupo propietario el grupo propietario de la carpeta en lugar del grupo primario del usuario que ha creado el archivo. Es una ventaja cuando varias personas pertenecientes a un mismo grupo, trabajan juntas con archivos almacenados en una misma carpeta. Si otorgamos permisos de lectura y escritura al grupo, los archivos podrán ser modificados por todos los miembros del grupo y cuando cualquiera de ellos cree un archivo, éste pertenecerá al grupo. Para activarlo **se puede ejecutar el comando chmod g+s directorio o sumar 2000 al número en octal si utilizamos dicho sistema**.

- **Sticky Bit**: Hoy en día, el sticky bit se utiliza con directorios, anteriormente se utilizaba con ejecutables, para dejarlos en la memoria swap. Cuando se le asigna a un directorio, significa que los elementos que hay en ese directorio sólo pueden ser renombrados o borrados por el propietario del elemento, el propietario del directorio o el usuario root, aunque el resto de usuarios tenga permisos de escritura y, por tanto, puedan modificar el contenido de esos elementos.

El sticky bit está a menudo configurado para el directorio /tmp.

Para activarlo|desactivarlo, se usa el comando chmod:

```
#chmod +t|-t directorio
```

```
drwxr-xr-x   2 root root  4096 jul 25 07:17 srv
dr-xr-xr-x  13 root root     0 ene  7 10:29 sys
drwxrwxrwt  19 root root  4096 ene  7 13:10 tmp
drwxr-xr-x  10 root root  4096 jul 25 07:17 usr
```

C) ACL

Manual en la web

Lo que vamos a poder hacer con las Listas de Control de Acceso es asignar permisos de una manera más específica. Por ejemplo tenemos un fichero XXX al que queremos dar acceso total a un usuario en concreto. Con lo visto hasta ahora deberíamos incluirlo en nuestro grupo, y darle los permisos a TODO el grupo. Este puede ser algo que no queramos que sea así. Veamos como nos ayudan las ACL.

Para poder trabajar necesitamos:

1. Tener el soporte para acl instalado:

```
$sudo apt-get install acl
```

2. Montar nuestra unidad con la opción de acl activa. Muchas veces ya viene así por defecto.
   Para ver si tengo las acl activas:

```
#tune2fs -l /dev/sda2
```

```
tomas@hpPavilion15ns:~$ sudo tune2fs -l /dev/sda2
tune2fs 1.44.1 (24-Mar-2018)
Filesystem volume name:   <none>
Last mounted on:          /
Filesystem UUID:          7f4728cc-d6be-4625-b4bb-20b1
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_
eeds_recovery extent 64bit flex_bg sparse_super large_
xtra_isize metadata_csum
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
```

Si no las tuviera..

```
# mount -o remount -o acl /dev/sda2
```

3. Usar los comandos getfacl y setfacl para modificar los permisos según nuestras necesidades.

Establecer ACL

Para agregar permiso a un usuario (user es el nombre o el ID):

```
# setfacl -m "u:user:permissions" <file/dir>
```

Agregar permisos para un grupo (group es el nombre o el ID):

```
# setfacl -m "g:group:permissions" <file/dir>
```

Para borrar todas las entradas ACL:

```
# setfacl -b <file/dir>
```

Mostrar ACL

Para mostrar los permisos use:

```
# getfacl <nombre archivo>
```

Ejemplo

Establecer todos lo permisos para el usuario Jhonny en el archivo con nombre “abc”:

```
# setfacl -m "u:johny:rwx" abc
```

Check permissions

```
# getfacl abc
```

```
# file: abc
# owner: someone
# group: someone
user::rw-
user:johny:rwx
group::r--
mask::rwx
other::r--
```

4. La salida del comando ls nos muestra información sobre los ficheros/directorios que tienen ACL incluidas

```
# ls -l  <nombre archivo>
```

Notará que hay un ACL para un archivo dado porque se mostrará un + (signo más) después del permiso Unix en la salida de ls -l.

```
$ ls -l /dev/audio
crw-rw----+ 1 root audio 14, 4 nov.   9 12:49 /dev/audio
```