Permisos

El que fa que Linux sigui un sistema operatiu segur són els privilegis d'accés granular a fitxers i directoris. Una definició precisa de qui té permís per llegir, modificar dades o executar programes específics proporciona una protecció excel·lent contra els ulls de qualsevol tafaner o d'una mala configuració intencionada. L'administrador, root, no està subjecte a restriccions, cosa que inclou assignació de lectura, escriptura i l'execució de permisos a altres usuaris a través del sistema. Si s'és el propietari d'un fitxer o directori, es pot garantir l'accés a aquests recursos per a altres comptes. Si també s'és un membre d'un grup específic, es pot modificar la propietat del grup de fitxers i directoris per a les assignacions de permisos granulars a fitxers.

Drets i Obligacions

Per a cada fitxer (és a dir, per directoris, fitxers de dispositiu, etc.), Linux defineix de manera precisa qui pot llegir, escriure i executar aquest fitxer. A més, cada fitxer pertany a un usuari ia un grup. Els tres permisos s'assignen separadament per aquestes tres categories i per a usuaris que no pertanyen a cap d'elles:

- Permís de Lectura: Els usuaris poden veure el contingut d'un fitxer o carpeta a la pantalla, copiar-lo o fer altres quantes coses.
- Permís d'Escriptura : Els usuaris poden canviar fitxers i directoris i desar els canvis. Això inclou l'habilitat d'esborrar.
- Permís d'execució: Per als programes, permisos d'execució significa que l'usuari té permís per executar-lo. Executar per a un directori significa que l'usuari té permís per canviar al directori (ordre cd) (addicionalment, l'usuari necessita permís de lectura per poder veure el contingut de la carpeta).

Descobrir Permisos

Per descobrir els permisos d'un fitxer podem veure la carpeta detallada a un administrador de fitxers gràfic com Konqueror o Nautilus, o simplement establir el paràmetre -l (sortida llarga) per a la comanda ls.

En ambdós casos els permisos s'indiquen mitjançant la lletra *r* (de "Read" o lectura), *w* (de "Write" o escriptura) i *x* (de "Execute" o executar). El primer bloc dels tres mostra els permisos per al propietari, el segon es refereix al grup i el tercer a tots els altres usuaris. Les carpetes s'indiquen amb una *d* (de "directory" o directori) al començament de la llista (veure Figura 1).

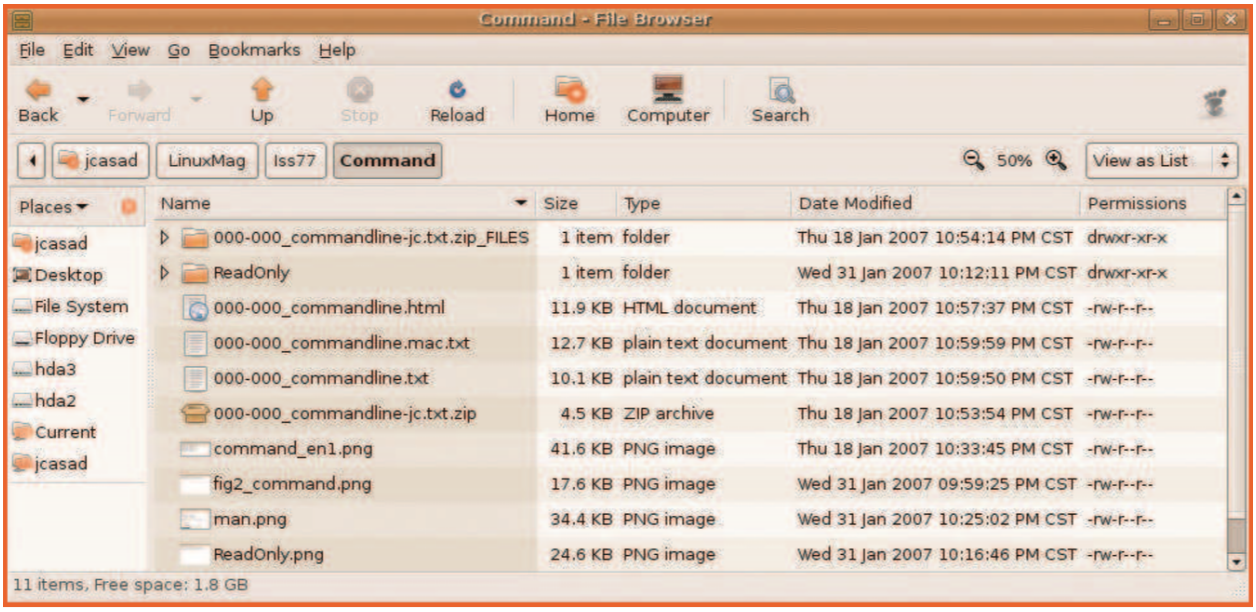

Figura 1

Permisos especials

Linux posseeix també dos permisos especials: el bit *s* (conegut també com el bit setuid/setgid) i el bit *t* (o "sticky" bit). Tots dos substitueixen la *x* en el bloc dels tres rwx.

La *s* es troba comunment en fitxers executables, mentre que el bit *t* és més comú amb directoris.

Tal com suggereix el nom, el bit setuid/setgid (configuració de l'usuari ID i configuració del ID del grup, respectivament) executa un programa amb els permisos d'un usuari o grup sense importar qui executi el programa. D'aquesta manera, els usuaris sense privilegis poden accedir als recursos als que normalment no tindrien accés.

Encara que això pugui suposar un risc potencial de seguretat, però, el bit *s* té els seus usos. Molts programes, incloent su, sudo, mount o passwd compten amb el bit *s:*

```
$ ls -l /usr/bin/passwd
-rwsr-xr-x root root 27132 Jul 11 20:06 /usr/bin/passwd*
```

El programa passwd modifica les contrasenyes, accedint en el procés al fitxer /etc/shadow per introduir la nova contrasenya. Per defecte, el fitxer es troba protegit contra l'accés d'escriptura mitjançant usuaris sense privilegis, i reservat per al seu ús per l'administrador per evitar d'aquesta manera que qualsevol pugui manipular les contrasenyes. El bit *s* executa el programa passwd com l'usuari root i introdueix la nova a /etc/shadow en nom de root.

L'altre permís especial, el bit *t* passa normalment en directoris compartits (llegeix, escriu i executa permisos per a tots) , en lloc del paràmetre d'execució, per assegurar que els usuaris

només han permès modificar, i per tant esborrar, les seves pròpies dades.

El bit sticky està configurat típicament a /tmp, tal com veiem aquí:

```
$ ls -ld / tmp
drwxrwxrwt 16 root root 4096 Jan 28 19:51 /tmp/
```

La carpeta /tmp emmagatzema temporalment fitxers per a múltiples usuaris.

Si tothom tingués el dret de lectura, escriptura i execució d'aquests fitxers, en teoria, tothom podria eliminar el sistema i esborrar dades arbitràriament.

Tanmateix, el bit *t* evita que això passi, assegurant-se que els usuaris només puguin esborrar el seu propis fitxers (o aquells sobre els quals posseeixen permís d'escriptura). L'excepció a aquesta regla és que el propietari de la carpeta amb el bit sticky també té permís per esborrar dins d'aquesta carpeta.

Modificació de Permisos

El programa chmod permet modificar els permisos de fitxers i directoris assumint que ets el propietari o l'administrador del sistema, i comprenent dos tipus diferents d'ordres.

En una manera, poden usar lletres per definir permisos. En aquest cas, u per "user" (propietari), g per a "group" , o per "others" (tots els altres usuaris) i a (all) per a tots els usuaris; r per "read", w per "write", x per "execute", s per el bit setuid/setgid i t per al bit sticky.

Una combinació d'aquestes lletres amb els signes més, menys i igual li diu a chmod que afegeixi, tregui o assigni, respectivament, amb precisió aquests permisos. Per exemple, per donar permisos de lectura i escriptura a un grup per a un fitxer hem d'escriure

```
$ chmod g+rw fitxer.
```

L'eliminació de permisos segueix el següent mètode: l'ordre

```
$ chmod o-rw fitxer
```

elimina tots els permisos per a tots els usuaris que no són propietaris ni membres del grup propietari.

També és possible combinar tots dos comandaments de la manera següent:

```
$ chmod g+rw,o-rwx fitxer
```

Com ja s'ha dit abans, un signe igual permet assignar amb precisió tots els permisos especificats

en la línia de ordres. Per exemple, la comanda:

```
$ chmod ugo=rwx directori
o equivalent
$ chmod a=rwx directori
```

dóna al propietari, membres del grup i a tots els altres usuaris permisos de lectura, escriptura i execució per al directori especificat en qüestió.

El programa chmod també comprèn xifres. Quan s'executa l'eina, és possible passar números octals[1] de quatre dígits o de tres, en lloc de lletres.

Podem calcular el nombre de la manera següent: 4 per permisos de lectura, 2 per a permisos d'escriptura i 1 per a permisos d'execució. El primer número es refereix al propietari, el segon al grup i el tercer a tots els altres restants.

Sobre aquesta base, pots veure que, per exemple, 644 significa u=rw, go=r (resultant rw-r--r--), o 777 seria igual a=rwx (resultant rwxrwxrwx). La taula "Permisos" dóna més detalls.

Número	Letras Octal
0	- - -
1	- - x
2	- w -
3 (= 2+1)	- w x
4	r - -
5 (= 4+1)	r - x
6 (= 4+2)	r w -
7 (= 4+2+1)	r w x

Taula Permisos

Per establir el bit *s* o el *t* cal afegir un quart número al començament del bloc de tres.

El número 4 representa el bit *s* per al propietari (setuid), 2 estableix el bit *s* per al grup (setgid), i 1 estableix el bit t. Un exemple.

```
$ ls-l script.sh
-rw-r--r-- 1 Huhn Huhn 3.191.789 6 octubre 05:01 script.sh
$ chmod 4755 script.sh
$ ls-l script.sh
-rwsr-xr-x 1 Huhn Huhn 3.191.789 6 octubre 05:01 script.sh
```

[1] Números octals: El sistema octal fa servir la base 8, és a dir, inclou vuit números entre el 0 i el 7. El següent nombre després del 7 és el 10, 20 segueix a 17, etc. Cada número en un nombre octal està representat per tres bits, en el cas dels permisos, els tres bits especifiquen el que li està permès fer a cada classe d'usuaris.

Canvi dels Membres del Grup

Com a usuari "normal" podem assignar nostres fitxers propis a grups específics, però, això ha d'implicar que ets un membre del grup en qüestió. La comanda següent ens diu els membres del grup:

```
$ groups
Huhn dialout cdrom floppy audio video
```

Per assignar un fitxer al grup audio cal escriure:

```
$ chgrp audio fitxer
```

Canvi de Propietaris i Grups

En un sistema Linux, l'administrador de la mateixa pot assignar nous grups a fitxers i directoris. Imaginem que volem configurar un nou compte anomenada mike i que hem establert un nou directori d'inici per a Mike i copiats els fitxers de configuració crítics des de /etc/skel.

L'últim pas seria donar a Mike els permisos que necessita per establir-se i utilitzar el seu directori d'inici i els subdirectoris sota ell.

La comanda següent lliura el directori d'inici i tots els fitxers en ell (incloent els fitxers de configuració ocults) a l'usuari mike:

```
$ chown -R mike /home/mike
```

L'opció -R usada aquí li diu a chown que actua de manera recursiva (això s'explicarà més tard). També és útil per poder definir un nou grup propietari per a les dades a la vegada:

```
$ chown -R mike:mike /home/mike
```

En altres paraules, afegim el nom del grup (algunes distribucions tenen un grup per defecte anomenat users, mentre que altres usen el nom del compte com el grup per defecte), amb una coma per separar-lo del nom del compte.

Recursió

Les tres eines, chmod, chgrp i chown, suporten un paràmetre -R per accions recursives. Per exemple, si volem permetre que els membres del grup vídeo tinguin accés al directori, a tots els subdirectoris i als fitxers que conté, escriurem:

```
$ chgrp -R vídeo directori
```

L'opció -R també pot evitar-nos haver d'escriure massa quan la fem servir en combinació amb la comanda chmod.

Per eliminar els permisos de lectura, escriptura i execució d'aquesta carpeta per a tots els usuaris que no són propietaris o membres del grup vídeo, escrivim:

```
$ chmod -R o-rwx directori
```

Avís

Cal anar amb compte quan executem ordres recursives que eliminen el paràmetre d'execució. Si erròniament escrivim a-x en comptes de o-x, descobrirem que ens hem deixat fora a nosaltres mateixos: chmod elimina els permisos d'execució del directori pare i la nostra habilitat per canviar al directori i modificar els fitxers. L'ús de find pot ajudar-nos a evitar aquest tipus de dilemes:

```
$ find directori -type f -exec chmod a-x "{}" ","
```

La comanda find descobreix primer els fitxers (-type f) i després executa chmod contra ells, ignorant el directori.

Des del Principi

umask especifica els permisos per defecte assignats als fitxers i directoris creats recentment. Si escrivim la comanda umask sense cap paràmetre, ens presentarà la configuració actual:

```
$ umask
0022
```

Aquest valor ens indica per a cada tipus d'usuari els permisos per defecte, seguint la seguent taula:

Octal	Permís
0	**rwx**
1	**rw-**
2	**r-x**
3	**r--**

4	**-wx**
5	**-w-**
6	**--x**
7	**---**

Aquest valors, per a un calcul més precís corresponen de restar 777 al valor que posis, tenint em compte que el permis "x" per defecte mai es pot activar en fitxers (i per tant, encara que surti no s'activarà).

Per canviar la umask introduïm el fitxer i el nou valor en la línia d'ordres:

```
$ umask 0077
```

Aquesta entrada significa que els nous fitxers i directoris només estan disponibles per als seus propietaris. umask és vàlid per a la shell actual, encara podem afegir una entrada al nostre fitxer de configuració bash ~/.bashrc per fer que el canvi sigui permanent. Treballant com a root, també podríem afegir una entrada global a /etc/profile per modificar el umask per al sistema.

El primer dígit fa referència als permisos especials: SUID, SGID i Sticky bit

4000 = SUID
2000 = SGID
1000 = sticky bit

L'ordre umask també permet especificar aquests permisos de forma alfanumèrica:

ACL. Acces Control List

Els ACL permeten afegir una capa de seguretat més al nostre sistema de fitxers, que pot complementar als permisos clàssics de linux. Afegeix més flexibilitat poden afegir permisos diferents a usuaris i grups particulars.

Per exemple podem fer que uns determinats usuaris puguin escriure sobre una carpeta, i uns altres no. És un sistema molt més flexible i semblant a Windows.

Per a poder emprar ACL cal que el sistema de fitxers estigui muntat amb l'opció ACL. Normalment , i per defecte, ja tenen aquesta opció activada (sense necessitat d'especificar-ho com una opció al fstab o amb l'ordre mount). Ho podem comprovar mb l'ordre tune2fs juntament amb grep:

```
Dispositiu Arrencada    Start    Final  Sectors  Size Id Tipus
/dev/sda1  *             2048 60817407 60815360   29G 83 Linux
/dev/sda2           60819454 62912511  2093058 1022M  5 Estesa
/dev/sda5           60819456 62912511  2093056 1022M 82 Intercanvi Linux / Sola
armand@DESKTOP:~$ sudo tune2fs -l /dev/sda1 | grep "Default mount options:"
sudo: no s'ha pogut resoldre l'amfitrió DESKTOP
Default mount options:    user_xattr acl
```

Si observem que apareix “acl”, ja no ens hem de preocupar. Cal, això si, que a les opcions de muntatge de fstab aparegui “defaults”.

Crear ACL

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

Exemples:

```
armand@DESKTOP:~$ sudo setfacl -m "u:armand:rwx" /root
```

Executada aquesta ordre l’usuari indicat tindrà permisos “rwx” sobre la carpeta /root. És a dir, podrà fer en aquesta carpeta el que vulgui sense ser l’usuari root. (Observa que no s’empra sudo)

```
armand@DESKTOP:~$
armand@DESKTOP:~$ touch /root/fitxerot
armand@DESKTOP:~$
armand@DESKTOP:~$ ls /root
A100  A103  A106  A201  commuta  execaula  hoster    neteja.sh
A102  A104  A110  A206  Desktop  fitxerot  macs.csv  passwdaula
```

Els permisos estandard segueixen essent els mateixos que abans, però observa que s’ha afegit un símbol “+”, indicant que hi ha permisos addicionals.

```
armand@DESKTOP:~$ ls -ld /root
drwxrwx---+ 19 root root 4096 feb  6 17:08 /root
```

Si intento entrar amb un altre usuari diferent a l’anterior, observareu que no podem accedir a la carpeta /root.

```
linux@DESKTOP:~$ ls /root
ls: no s’ha pogut obrir el directori '/root': S’ha denegat el permís
```

Els permisos només s’han establert per a un usuari i per a una carpeta concreta, Si volem afegir fitxer o carpetes addicionals ho podem fer sense problemes.

Igualment si volem assignar un permis sobre una carpeta o fitxer a un grup d’usuaris:

```
Fet.
armand@DESKTOP:~$ sudo setfacl -m "g:alumnes:rwx" /var/www/html
armand@DESKTOP:~$
```

```
linux@DESKTOP:~$
linux@DESKTOP:~$ id
uid=1000(linux) gid=1000(linux) grups=1000(linux),1002(alumnes)
linux@DESKTOP:~$ ls -ld /var/www/html
drwxrwxr-x+ 2 root root 4096 feb  6 17:21 /var/www/html
linux@DESKTOP:~$ touch /var/www/html/total.jsp
linux@DESKTOP:~$ ls -l /var/www/html
total 12
-rw-r--r-- 1 root  root  11321 set 19 23:08 index.html
-rw-rw-r-- 1 linux linux     0 feb  6 17:21 total.jsp
linux@DESKTOP:~$
```

En aquesta imatge creem una ACL per al grup alumnes (al qual pertany l'usuari linux) per a que qualsevol membre d'aquest grup tingui accés complert sobre la carpeta /var/www/html (on el servidor web apache2 llegeix les pàgines, per defecte)

Comprovar ACL

L'ordre getfacl ens permet comprar les ACL per a una carpeta o fitxer:

```
armand@DESKTOP:~$ sudo getfacl /var/www/html
getfacl: Removing leading '/' from absolute path names
# file: var/www/html
# owner: root
# group: root
user::rwx
group::r-x
group:alumnes:rwx
mask::rwx
other::r-x

armand@DESKTOP:~$ sudo getfacl /root
getfacl: Removing leading '/' from absolute path names
# file: root
# owner: root
# group: root
user::rwx
user:armand:rwx
group::---
mask::rwx
other::---

armand@DESKTOP:~$
```

Observa que les línies user, group i other indiquen els permisos per a l'usuari i grup propietari, però s'afegeix una o més línies amb els usuaris i grups que volem amb permisos diferents.

Esborrar ACL

Amb l'opció –x de setfacl

```
armand@DESKTOP:~$
armand@DESKTOP:~$ sudo setfacl -x armand /root
[sudo] contrasenya per a armand:
armand@DESKTOP:~$ touch /root/fitxeron2
touch: no s'han pogut canviar les dates de '/root/fitxeron2': S'ha denegat el permís
armand@DESKTOP:~$
```

Si volem esborrar totes les ACL d'un determinat fitxer o carpeta

```
touch: no s'han pogut canviar les dates de '/root/fitxeron2': S'ha denegat el permís
armand@DESKTOP:~$ sudo setfacl -b /var/www/html
armand@DESKTOP:~$ su linux
Contrasenya:
linux@DESKTOP:/home/armand$ touch /var/www/html/fitxeron2
touch: no s'han pogut canviar les dates de '/var/www/html/fitxeron2': S'ha denegat el permís
linux@DESKTOP:/home/armand$
```