

Permessi, utenti e gruppi

Daniele Venzano

9 novembre 2003

Indice

1	Introduzione	1
2	Concetti generali	2
2.1	Esempio	2
3	File importanti	2
3.1	/etc/group	2
3.2	/etc/passwd	3
4	Panoramica sui permessi	3
4.1	Permessi per le directory	4
4.2	Permessi in formato ottale	4
5	Comandi fondamentali	4
5.1	adduser, deluser	4
5.2	addgroup, delgroup	4
5.3	chown e chgrp	5
5.4	chmod	5
6	Uso sicuro dei permessi	5
7	Dove trovare altre informazioni	6
8	Copyright e altro	6

1 Introduzione

É molto importante verificare regolarmente il livello di sicurezza del proprio sistema, questa pratica consente sia di essere al riparo da eventuali visitatori indesiderati (o quanto meno a rendergli la vita un po' piú difficile), sia di tenere confinati gli errori di programmi ancora in sviluppo, che a causa di *bugs* possono causare, ad esempio, la cancellazione di qualche file di importanza fondamentale per il sistema.

In tutto il documento si fa riferimento a Linux, ma molti (quasi tutti, in effetti) i concetti sono applicabili con pochissime differenze ad altri sistemi Unix.

2 Concetti generali

In Linux gli accessi a file, directory e device vengono gestiti tramite dei bit, memorizzati in aree speciali del disco fisso, che indicano la possibilità di lettura (r = read), di scrittura (w = write) e di esecuzione (x = execute). Anche le zone di memoria sono protette con un sistema molto simile, questo per fare in modo che un programma non possa accedere (intenzionalmente o per errore) ad aree di memoria occupate da un programma di un altro utente¹

Inoltre sempre a file, directory e device sono associati un *owner* (proprietario) e un *group* (gruppo) che permettono di dare privilegi diversi ad utenti diversi.

2.1 Esempio

Molte distribuzioni (Debian e Slackware, tra le altre) permettono l'accesso in lettura e scrittura al disco fisso (/dev/hd?) solo ad un gruppo di utenti (di solito chiamato, con scarsa fantasia, disk). A questo modo si può restringere solo agli amministratori della macchina l'accesso diretto al disco (utile per riparare danni da black-out), accesso che é invece negato a tutti gli altri.

3 File importanti

I file che permettono di gestire gli utenti e gruppi sono:

```
/etc/group  
/etc/passwd
```

3.1 /etc/group

Il primo contiene una lista di tutti i gruppi presenti nel sistema e i nomi degli utenti che appartengono a ciascun gruppo con questo formato:

nome gruppo:x:GID²:lista di utenti, separati da virgole; Ad esempio:

```
dialout:x:20:mario,filippo
```

significa che nel gruppo *dialout* ci stanno gli utenti mario e filippo.

¹Linux infatti é un sistema multiutente, cioè può essere usato contemporaneamente da piú utenti (attraverso terminali dedicati o connessioni di rete con telnet, ssh, ftp, ecc.) e deve garantire la separazione completa dei processi di utenti diversi.

²Identificatore unico del gruppo, é un numero intero positivo e l'importante é che non ci siano 2 gruppi con lo stesso numero

3.2 /etc/passwd

Il secondo file, un po' piú complicato, contiene una lista di tutti gli utenti registrati sul sistema, molti degli utenti che si trovano nel file sono fittizi e vengono utilizzati solo da programmi³. Il formato di questo file é:

¡nome utente¡:x⁴:¡UID¡:¡GID¡:¡GECOS¡:¡percorso alla home directory¡:¡shell utilizzata¡
Il campo ¡UID¡ é l'identificatore (numerico) unico dell'utente, il ¡GID¡ é l'identificatore del gruppo principale (deve essere preso da /etc/group), il ¡GECOS¡ contiene diverse informazioni, separate da virgole, come il nome e il cognome, numero di telefono, ecc. Un esempio:

```
mario:x:1000:100:Mario Rossi,,,:/home/mario:/bin/bash
```

4 Panoramica sui permessi

I permessi per i file, i device e le directory sono memorizzati utilizzando 9 bit, divisi in tre gruppi da tre bit ciascuno, il primo gruppo identifica i permessi per il proprietario, il secondo per il gruppo e il terzo per tutti gli altri, un esempio dovrebbe chiarire parecchie cose:

```
$ ls -l
totale 28
-rw-r----- 1 venza users 488 mag 3 18:42 gruppi_e_utenti.aux
-rw-r----- 1 venza users 5476 mag 3 18:42 gruppi_e_utenti.dvi
-rw-r----- 1 venza users 4462 mag 3 18:42 gruppi_e_utenti.log
-rw-r----- 1 venza users 3453 mag 3 18:42 gruppi_e_utenti.tex
-rw-r----- 1 venza users 259 mag 3 18:42 gruppi_e_utenti.toc
$
```

La prima colonna del comando `ls -l` sono proprio i nove bit dei permessi⁵. I primi tre (`rw-`) indicano i permessi di lettura e scrittura, ma non di esecuzione, del proprietario (venza, nella terza colonna), il secondo gruppo (`r--`) indica permesso solo di lettura per il gruppo (users, quarta colonna) e il terzo gruppo (`---`) nessun permesso per tutti gli altri utenti che non fanno parte del gruppo users. Un altro esempio, da decifrare per esercizio, é:

```
-rwxr-x--- 1 venza users 126 mag 3 18:48 creadoc*
```

³Ad esempio é utile avere un utente `www` per il server web, in modo da limitare i file accessibili attraverso lo stesso server.

⁴Qui stavano le password, ma per ulteriore sicurezza sono state spostate nel file `/etc/shadow`

⁵il primo trattino sulla sinistra serve ad identificare il tipo di file, '-' per i file normali, 'd' per le directory, 'l' per i *link* e cosí via.

4.1 Permessi per le directory

Le directory si comportano in maniera leggermente differente rispetto ai file, infatti i permessi hanno un significato diverso.

Il bit **r** concede il permesso di leggere la lista dei file contenuti in quella directory, il bit **w** concede il permesso di poter creare o cancellare dei file e il bit **x** permette di attraversare la directory per accedere ad un file o ad un'altra directory.

4.2 Permessi in formato ottale

I permessi hanno anche una rappresentazione numerica, infatti é possibile scrivere tutte le possibili combinazioni di r, w e x, vediamo come:

u	g	o		u	g	o													
---	---	---		rw	r-x	r--	---	>	4+2+1	4+1	4	---	>	754					
421	421	421		421	401	400													

Bisogna sommare i numeri corrispondenti ai bit attivati separatamente per ciascun gruppo e poi metterli insieme per ottenere un numero di 3 cifre. Per chi ha qualche esperienza nel lavorare con basi diverse da 10 é utile sapere che, nell'esempio di sopra, 754 é la rappresentazione in ottale del numero binario 111101100 che corrisponde alla nonupla `rwxr-xr-`.

5 Comandi fondamentali

I comandi fondamentali per la gestione degli utenti sono (accanto al nomi trovate i numeri delle sezioni di *man* a cui trovate le istruzioni dettagliate):

```
adduser, deluser (8)
addgroup, delgroup (8)
chmod, chown, chgrp (1)
```

5.1 adduser, deluser

`adduser` e `deluser` servono, rispettivamente a creare e cancellare un utente, le diverse distribuzioni tendono a personalizzare parecchio questi programmi, ne esistono almeno due versioni, una é interattiva e pone all'amministratore domande sull'utente da creare per riempire i campi di `passwd`, l'altra funziona completamente da linea di comando, assumendo dei valori arbitrari per i campi non specificati.

5.2 addgroup, delgroup

`addgroup` e `delgroup` possono essere utilizzati per creare e cancellare un gruppo, di solito essendo il formato di `/etc/group` molto semplice, si preferisce modificarlo a mano.

5.3 chown e chgrp

`chown` permette di modificare il proprietario di uno o piú file, mentre `chgrp` cambia il gruppo. La sintassi é:

```
chown [-R] <nome utente del nuovo proprietario> <file o directory>
chgrp [-R] <nome del gruppo> <file o directory>
```

l'opzione `-R` attiva la modalitá ricorsiva, modificando contemporaneamente piú file in una sola volta.

5.4 chmod

`chmod` permette di modificare i permessi per uno o piú file o directory, si può utilizzare sia il formato numerico, sia un altro formato, spiegato in dettaglio nella *man page* che consente di modificare i singoli bit.

6 Uso sicuro dei permessi

Alcuni file e alcune directory devono avere dei permessi particolari per poter garantire un livello base di sicurezza, qui sotto c'è un elenco⁶, vi consiglio vivamente di verificare le impostazioni delle vostre installazioni, perché spesso molte distribuzioni si dimenticano qualcosa di fondamentale qua e là.

- `/etc/passwd` deve essere `rw-r---`
- `/etc/shadow` deve essere `rw-----`
- `/etc/group` deve essere `rw-r---`
- `/home/*` (tutte le directory degli utenti) devono essere `rwx---`
- `/sbin` e `/usr/sbin` contengono comandi per l'amministrazione del sistema e quindi non dovrebbero poter essere eseguiti dai normali utenti (quindi qualcosa del tipo `rwxr-x---`)
- `/var/log` contiene tutti i log del sistema, questi file contengono informazioni preziosissime sia per l'amministratore (che attraverso di essi può verificare se ci sono state intrusioni), sia per chi volesse informazioni sul sistema e sui programmi che ci girano, quindi devono essere protetti al massimo. Bisogna fare attenzione, però, perché i programmi che scrivono nei log devono avere i permessi per farlo.
- `/root` deve essere `rwX-----`

⁶Questo elenco non é sicuramente completo, inoltre ci grandi differenze di pensiero tra sistemisti paranoici e sistemisti pigri

7 Dove trovare altre informazioni

Le *man pages* dei comandi descritti qui sono sicuramente delle fonti molto ricche, inoltre anche il comando *info* può dare utili consigli.

Su Internet ci sono molte risorse dedicate alla sicurezza di sistemi Linux, Unix e Windows, il principale é sicuramente:

<http://www.securityfocus.com/archive/>

dove vengono messi avvisi e annunci su nuovi buchi di sicurezza relativi a programmi e sistemi operativi.

8 Copyright e altro

Questo documento é stato scritto da Daniele Venzano, ©2002, 2003.

This work is licensed under the Creative Commons Attribution-NoDerivs-NonCommercial License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nd-nc/1.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

L'ultima versione del documento é disponibile presso:

<http://teg.homeunix.org/documentation.html>