Gestione Utenti & Permessi

Mattia Rizzolo
mattia@mapreri.org

POuL
have a lot of fun...

Corso Linux Avanzato 2015

Multiuser?

Multiuser

- **root**
- Tutti gli altri utenti normali

Multiuser: Root

- Noto anche come *AMMINISTRATORE DI SISTEMA*
  - ancora meglio come **Super User**
- È quell'utente che possiede tutti i diritti di fare quello che vuole, incluso:
  - installare applicazioni system-wide
  - gestire gli utenti

Multiuser: Root

Non è un utente con cui ci si logga
per fare un giro sul web

Multiuser: Normal Users

Tutti gli altri utenti sono utenti normali
Questo vuol dire che non possono:

- Installare nuovi programmi
- modificare i file che non appartengono a loro (more on this later on)
  - Tra cui anche i file *di sistema*

Multiuser: Normal Users

L'amministratore di sistema può delegare permessi agli utenti normali

Multiuser: Groups

- Un utente può appartenere a 1 o più gruppi
- L'appartenenza a un gruppo può dare poteri ulteriori a quell'utente:
  - *adm* ti permette di leggere i log di sistema
  - *cdrom* ti permette di accedere al lettore CD
  - *audio* ti permette di accedere ai dispositivi audio-related
  - *scanner* ti permette di accedere agli scanner
  - ...
- Molto di questo ↑↑↑ non è più necessario oggi giorno

Multiuser: su

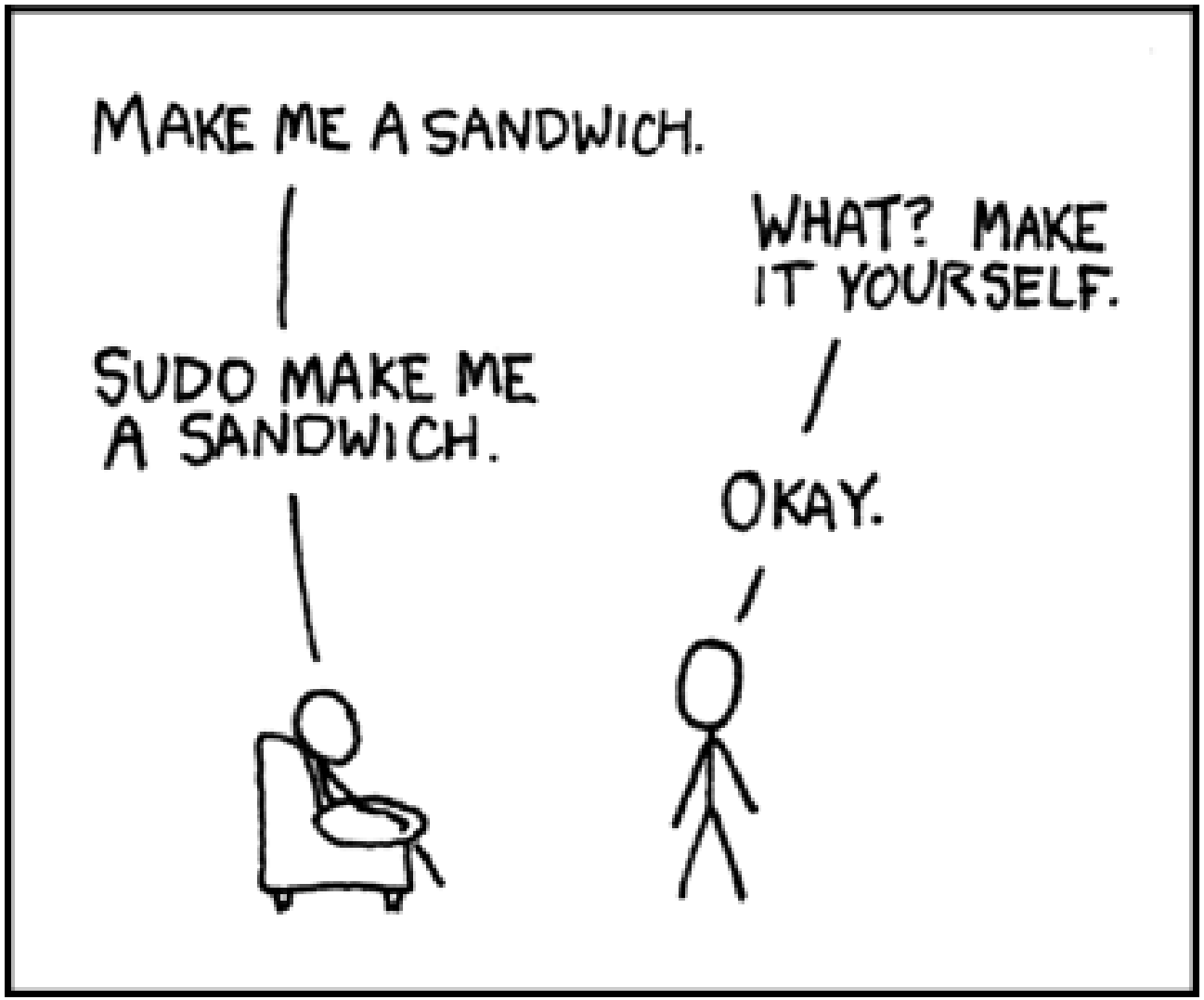
```
NAME
       su - change user ID or become superuser

SYNOPSIS
       su [options] [username]
```

permette di passare da un utente ad un altro.

Viene chiesta la password dell'utente di destinazione.

Multiuser: Root



Multiuser: sudo

`sudo` è il "nuovo" modo di eseguire comandi come altri utenti.

Ovviamente è usatissimo per eseguire comandi come root. Esempio:

```
sudo vim /etc/default/grub
```

Questo apre l'editor di testo `vim` per modificare il file di configurazione di grub.

Le impostazioni di sudo sono in `/etc/sudoers`.

Di solito gli utenti abilitati ad usare sudo sono quelli nel gruppo `sudo` o `wheel`.

Viene prima chiesta la *propria* password.

Multiuser: Create New Users

useradd

Multiuser: useradd

useradd(8)

```
NAME
       useradd - create a new user or update
                 default new user information

SYNOPSIS
       useradd [options] LOGIN

       useradd -D

       useradd -D [options]
```

Multiuser: useradd

Ok, ora alcune opzioni che possono tornare utili:

- **-c** il contenuto del campo GECOS. Di solito il nome completo dell'utente
- **-d** la directory home. Di default */home/login*
- **-g** il *GID* del gruppo principale dell'utente, che deve già esistere
- **-G** lista di gruppi aggiuntivi
- **–system** crea un utente di sistema
- **-s** specifica la shell predefinita dell'utente
- **-u** l'*UID* dell'utente

Multiuser: useradd

Tutte le opzioni sono opzionali, e di default prendono i valori dal file /etc/login.defs

Multiuser: passwd

passwd(8)

```
NAME
       passwd - change user password

SYNOPSIS
       passwd [options] [LOGIN]
```

Multiuser: passwd

- **-d** delete a password: non serve più una password per accedere: **BAD**
- **-e** expire a password: l'utente è costretto a cambiare la password all'accesso successivo
- **-i** inactive a password: l'utente non può più loggarsi

Multiuser: passwd

Più semplicemente *passwd login* cambia la password dell'utente.

L'utente root può cambiare (ma non conoscere!) le password di tutti gli utenti.

Ogni utente può cambiare felicemente la propria password quando vuole.

Multiuser: groupadd

Questo è semplice:

`groupadd groupname`

ci sono un po' di opzioni, ma groupadd(8) è bello e conciso.

Multiuser: /etc/passwd

File che contiene informazioni sugli utenti

```
Debian-exim:x:104:109::/var/spool/exim4:/bin/fal
messagebus:x:105:110::/var/run/dbus:/bin/false
statd:x:106:65534::/var/lib/nfs:/bin/false
mattia:x:1000:1000:Mattia Rizzolo:/home/mattia:/
sshd:x:107:65534::/var/run/sshd:/usr/sbin/nologi
geoclue:x:108:117::/var/lib/geoclue:/bin/false
saned:x:109:118::/var/lib/saned:/bin/false
alessandro:x:1001:1001:Alessandro Cecchin:/home/
dnsmasq:x:113:65534:dnsmasq,,,:/var/lib/misc:/bi
avahi:x:116:129:Avahi mDNS daemon,,,:/var/run/av
hplip:x:117:7:HPLIP system user,,,:/var/run/hpli
```

Multiuser: /etc/passwd

- Una riga = un utente
- formato della riga: 7 colon separated values:
  - login
  - password
  - UID
  - GID
  - GECOS
  - home directory
  - shell

Multiuser: /etc/shadow

File che continue le password degli utenti

```
Debian-exim:!:16460:0:99999:7:::
messagebus:*:16460:0:99999:7:::
statd:*:16460:0:99999:7:::
mattia:$6$CzYs8XWn$SSwdA6Mf4MXAYUiJJ78PAhaaAELfK
sshd:*:16460:0:99999:7:::
geoclue:*:16460:0:99999:7:::
saned:*:16460:0:99999:7:::
alessandro:!:16460:0:99999:7:::
dnsmasq:*:16460:0:99999:7:::
avahi:*:16474:0:99999:7:::
hplip:*:16495:0:99999:7:::
```

Multiuser: /etc/shadow

- solo root può leggere questo file
- stesso formato di /etc/passwd, 9 campi separati da due-punti:
  - login name
  - encryped password
  - date of last password change
  - minimum password age
  - maximum password age
  - password warning period
  - password inactivity period
  - account expiration date
  - reserved field because the future is not yet written, and another field is always handy, and another field in this file is what people always dreamt for, it is well known.

Multiuser: /etc/group

```
Debian-exim:x:109:
messagebus:x:110:
mlocate:x:111:
ssh:x:112:
mattia:x:1000:
sambashare:x:113:
scanner:x:115:saned
vboxusers:x:116:
geoclue:x:117:
saned:x:118:
alessandro:x:1001:
ssl-cert:x:122:
pulse:x:123:
```

Multiuser: /etc/group

- Reinventare la ruota è brutto, quindi il formato è uguale
- 4 valori:
  - group_name
  - password
  - GID
  - user_list (comma-separated)

Multiuser

DEMO

Multiuser

Questions?!?

Permissions?

Permissions: Files

Ogni file ha:

- owner
- group
- permission
- extended attributes
- capabilities

Permissions: Classes

I permessi regolari di ogni file sono divisi in *classi*:

- owner (**u**)
- group (**g**)
- others (**o**)

Permissions: Modes

Ogni classe permette di specificare un diverso livello di permessi per quel file:

- read (**r**)
- write (**w**)
- execute (**x**)

Permissions: Special Modes

Le implementazioni di questi variano tra i sistemi Unix. Su Linux il comportamento è il seguente:

- sticky bit (**t**, **1**)
  - regular files: ignored
  - directories: "files in that directory may only be unlinked or renamed by root or the directory owner or the file owner"
- setuid (**u+s**, **4**)
  - regular files: quel file, quando eseguito, viene eseguito sotto l'UID dell'utente owner del file
  - directores: ignored
- segid (**g+s**, **2**)
  - regular files: stessa cosa di setuid, ma per i gruppi
  - directories: i file creati in quella directory avranno come gruppo il gruppo che possiede la directory padre

Permissions: Octal Representation

In genere i permessi si assegnano/rappresentano con una rappresentazione ottale, secondo questa tabella:

0	-	none
1	x	execute
2	w	write
4	r	read

Permissions: Octal Representation

Quando si invoca il comando per cambiare permessi si specificano 4 (o meno) numeri in cui il primo specifica lo sticky bit, il secondo i permessi per l'owner, il terzo per il gruppo, e il quarto per tutti gli altri. Il numero finale è dato dalla somma dei valori.

Numeri mancanti sono assunti come 0 a partire da sinistra.

Tralasciando i modi speciali:

3	rw	read & write
5	rx	read & execute
7	rwx	read & write & execute

Permissions: Symbolic Representation

C'è anche un modo più carino di specificare i permessi di un file:

`u+wrx,g+r-wx,o-rwx`

Questo assicura il permesso di lettura, scrittura, ed escuzione per l'utente, solo lettura per il gruppo, niente per gli altri.

La notazione ottale sarebbe

0740

Permissions: umask

- Serve a visualizzare e/o settare i permessi di default dei file;
- Per visualizzare i permessi di default in forma simbolica si usa umask -S, in caso contrario li si ottiene in forma ottale;
- Per rimuovere i permessi, si chiama umask passandogli come argomenti la forma ottale dei permessi da riumovere;
- Per dare dei permessi, utilizzando la stessa sintassi di chmod, si chiama umask passandogli come argomenti la forma simbolica dei permessi da aggiungere.

Permissions: chown

chown(1):

```
NAME
       chown - change file owner and group

SYNOPSIS
       chown [OPTION]... [OWNER][:[GROUP]] FILE.
       chown [OPTION]... --reference=RFILE FILE.
```

Permissions: chown

`chown [OPTION]... [OWNER][:[GROUP]] FILE`

serve per cmabiare owner e/o group di un file:

- `chown mattia file`
  - quel file ora appartiene a me
- `chown mattia: file`
  - quel file ora appartiene a me, e al mio gruppo di default
- `chown :mattia file`
  - quel file ora appartiene al gruppo *mattia*
- `chown mattia:adm file`
  - quel file ora appartiene a me, a al gruppo *adm*
- `chown -R mattia: ~`
  - tutta la mia home directory ora appartiene a me e al mio gruppo

Permissions: chown

- il proprietario di un file può cambiarne solo il gruppo.
  - yet, `chown $USER file` non da errori se *file* è gia mio
- root può cambiare il prorietario di qualsiasi file
- opzioni più utili:
  - **-R**: ricorre tutto l'albero delle directory
  - **-H**/**-L**/**-P**: controllano come `chown` si comporta quando incontra un link simbolico

Permissions: chmod

chmod(1)

```
NAME
       chmod - change file mode bits

SYNOPSIS
       chmod [OPTION]... MODE[,MODE]... FILE...
       chmod [OPTION]... OCTAL-MODE FILE...
       chmod [OPTION]... --reference=RFILE FILE.
```

Permissions: chmod

Una manciata di esempi...

```
chmod u+wrx file
chmod g+r-wx file
chmod o-rwx file
chmod u+wrx,g+r-wx,o-rwx file

chmod 0700 file
chmod 0040 file
chmod 0000 file
chmod 0740 file
```

NON SONO EQUIVALENTI

Permissions

DEMO

Permissions

Questions?!?

Do you want more cookies?

~~RTFM and stop ranting~~

1. Chiedi al tuo computer: `man chmod`, per esempio.
   - se non riesci a usare man: `man man`
2. GIYF[1]
3. wiki della tua distribuzione
   - Hint: anche se non è della distribuzione che stai usando va bene lo stesso.
4. chiedi nei canali di supporto della tua distribuzione
5. chiedi a noi ;)

[1] Google Is Your Friend

Thanks

- POuL, for the organization
- **YOU**, for attending
- Holger Levsen <holger@debian.org> and Jérémy Bobbio <lunar@debian.org> for the template used for these slides
- Linux for being so funny

email	mattia@mapreri.org
GPG key	66AE 2B4A FCCF 3F52 DA18 4D18 4B04 3FCD B944 4540