Gestione Utenti & Permessi

Corsi Linux 2014

POuL

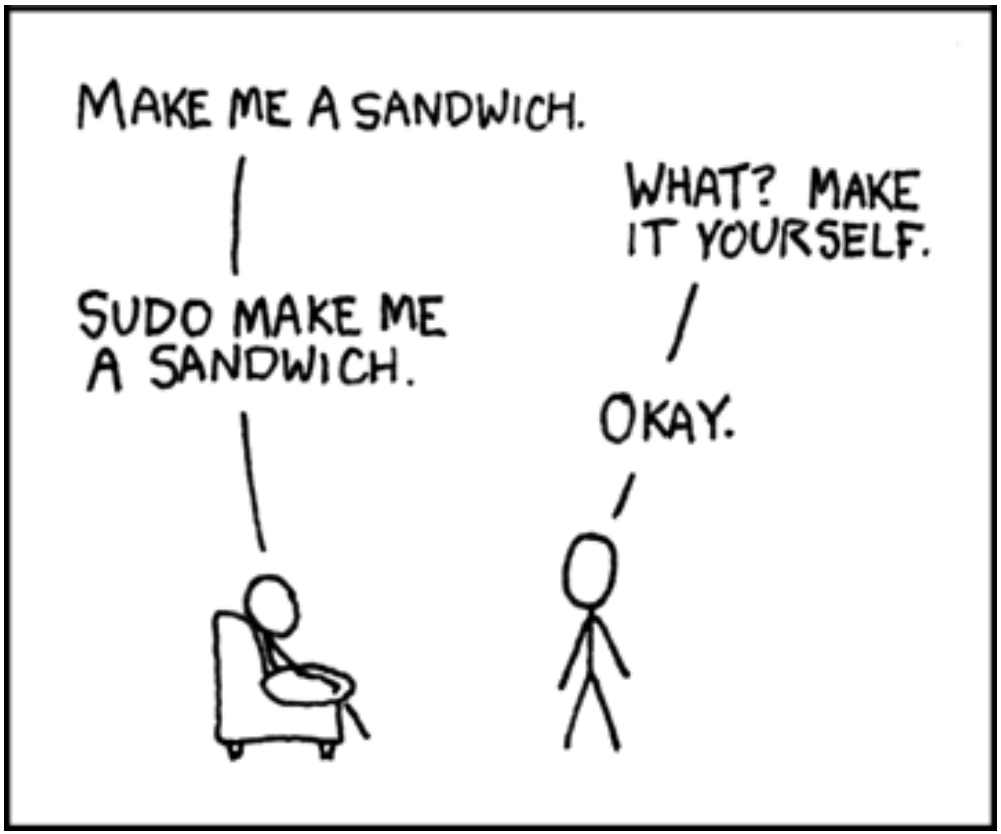
have a lot of fun...

1. Utenti
   - Ci sono utenti e l'Utente...
   - Creare un utente

2. I permessi
   - U can't touch this
   - Assegnare la proprietà di un file
   - Le capabilities

- La maggior parte dei moderni sistemi operativi supporta nativamente la multiutenza, ovvero la possibilità che più utenti siano collegati simultaneamente alla stessa macchina.
- L'esistenza di più utenti su una macchina comporta la necessità di gestirne il comportamento ed i permessi.

Utenti & Root


MAKE ME A SANDWICH.
SUDO MAKE ME A SANDWICH.
WHAT? MAKE IT YOURSELF.
OKAY.

- Nei sistemi GNU/Linux esistono due tipi di utenti: gli **utenti “normali”** e l’**utente root**;
- L’utente root è l’amministratore che gestice il sistema, ha poteri “infiniti” e viene usato unicamente con questo scopo (non è un utente con cui loggarsi in libertà);
- Gli utenti “normali” sono tutti gli altri, solitamente hanno permessi ristretti ma possono ricevere temporaneamente permessi di amministrazione;
- L’amministratore può delegare alcuni permessi a determinati utenti utilizzando i gruppi.

- Gli utenti “normali” possono acquisire permessi di root usando i comandi: **su** e **sudo**;
- Il comando *su* permette di loggarsi come un altro utente inserendo la sua password;
- Il comando *sudo* permette di eseguire un comando con permessi di root, usando la password dell’utente.

su

Sintassi su

```
su [opzioni][utente]
```

- Permette di loggarsi come un altro utente, utilizzando però la sua password;
- Se l'argomento *utente* viene omesso, si logga come l'utente root (è consigliato l'uso di sudo per questo);
- Con l'opzione *-c* viene eseguito un comando con i permessi dell'utente specificato;
- L'utente root può loggarsi come qualunque altro utente senza che gli sia chiesta la password.

sudo

Sintassi sudo

```
sudo [opzioni] [utente] [comando]
```

- Permette di eseguire un comando con i permessi di root, utilizzando la propria password;
- Per far ciò l'utente deve essere abilitato in */etc/sudoers* (o far parte di un gruppo abilitato);
- Per aprire una shell di root è consigliato l'uso dell'opzione *-s* al posto del comando *su*.

/etc/sudoers

Sintassi /etc/sudoers

```
utente host=(utenti_ipersonificabili) [NOPASSWD] comandi
```

- *host*: indica l'host di provenienza, generalmete è settato su *ALL*, ossia accetta comandi provenienti da ogni sorgente;
- *utenti_impersonificabili*: indica quali utenti possono essere impersonificati, se settato su ALL non pone limitazioni. Possono essere impostati più utenti separandoli con una virgola;
- *NOPASSWD*: è opzionale e fa in modo che non sia richiesta la password;
- *comandi*: indica i comandi che possono essere eseguiti, separati da una virgola. Nel caso fosse ALL, significa che può eseguire qualunque comando;
- Si può consentire l'uso di sudo anche ad un intero gruppo, basta mettere a posto del *utente* il nome del gruppo, preceduto da %.

useradd

Sintassi useradd

```
useradd [opzioni] utente
```

- *-d </home/directory/>* assegna il percorso per l'home dell'utente, ma non la crea. Per crearla bisogna usare l'opzione *-m* ;
- *-g <gruppo>* specifica il gruppo principale dell'utente, l'opzione *-G < gruppo1, gruppo2 >* aggiunge l'utente ad una lista di gruppi supplementari separati da una virgola;
- *-s </path/shell>* imposta una determinata shell per l'utente;
- *-e <data>* imposta la data in cui l'utente viene disabilitato. La data è nel formato AAAA-MM-GG;
- Se non sono specificate, vengono usate le opzioni di default. Per vedere quali sono si può usare il comando *useradd -D*;
- Per eliminare un utente si usa *userdel [opzioni] utente*, con l'opzione *-r* vengono eliminati anche i file nella home.

passwd

Sintatssi passwd

```
passwd [opzioni] [nome_utente]
```

- Per impostare la password di un utente (o per cambiarla) si usa il comando *passwd nome_utente;*
- L'opzione *-d* elimina la password dell'utente, mentre *-e* ne forza la scadenza (la password viene richiesta all'utente al succesivo login);
- Le informazioni dell'utente vengono salvate in */etc/passwd* mentre le password in */etc/shadow.*

/etc/passwd

Esempio di /etc/passwd

```
root:x:0:0:root:/root:/bin/bash
pippo1:x2:1000:10003:Pippo,,,4:/home/pippo5:/bin/bash6
```

1. nome utente, tutto scritto in minuscolo e senza spazi;
2. la x sta ad indicare che la password è salvata in /etc/shadow;
3. User ID (UID) e Group ID (GID). Il valore 0 è assegnato sempre a root, per gli utenti normali vengono assegnati valori da 1000 in poi;
4. GECOS, sono i dati personali dell'utente come nome, cognome, numero di telefono, ecc;
5. Directory home, indica dove è posizionata l'home dell'utente;
6. Inidica la shell che viene utilizzata dall'utente.

/etc/shadow

Esempio di /etc/shadow

`root`$_1$`:$6$hashdellapassword`$_2$`:16132`$_3$`:0:99999::::`$_4$

1. sempre il nome utente, scritto in minuscolo e senza spazi;
2. l'hash della password, il valore indicato tra i simboli $ indica il tipo di funzione di hashing utilizzata (SHA-512 in questo caso). Se l'hash della password è preceduta dal simbolo **'*'** vuol dire che il login è disabilitato, mentre se è preceduta da **'!'** che non è ancora stata settata;
3. Questo valore indica quanto è stata modificata l'ultima volta, il valore è il numero di giorni da *epoch* (1 Gennaio 1970);
4. Questi valori indicano alcuni limiti temporali sulla password, come i giorni di scadenza o il tempo minimo prima di poterla cambiare.

Per questioni di sicurezza, il file /etc/shadow è accessibile unicamente da root.

Gruppi

- Come già detto, un utente può appartenere a dei gruppi. Questi gruppi sono un modo efficente per delegare i permessi a certi utenti;
- Per visualizzare in che gruppo un utente si trova, si usa *groups [utente]*. Se omesso prende come utente quello corrente;
- In maniera analoga a quella per gli utenti, si può creare un gruppo con *groupadd*, eliminare uno con *groupdel* e *groupmod* per modificarne le informazioni.
- Sepre in maniera analoga, le informazioni sui gruppi vengono salvati in /etc/groups ed in */etc/gshadow;*

Demo

1. Utenti
   - Ci sono utenti e l'Utente...
   - Creare un utente

2. I permessi
   - U can't touch this
   - Assegnare la proprietà di un file
   - Le capabilities

U can't touch this

- Un utente può avere permesso di scrittura, lettura e/o esecuzione su un determinato file;
- Il file può concedere permessi diversi a seconda di chi sta tentando di accedervi;
- I permessi vengono assegnati secondo la sequenza **UGO**: **user** (U), **group** (G), **others** (O);
- Un utente che non ha nessun permesso su un file non può ne leggerlo, ne modificarlo, ne tantomeno eseguirlo;
- L'unica eccezione è l'utente root, può fare qualsiasi cosa anche se non esplicitamente autorizzato;
- Se il file in questione è una cartella, allora il permesso di esecuzione prende il significato di permesso di attraversamento;
- Per visualizzare i permessi di un file si può usare il comando *ls -l file.*

Oltre ai semplici permessi di lettura, scrittura ed esecuzione esistono alcuni permessi speciali: *sticky bit, setuid, setgid*

- **Sticky bit** fa in modo che un file che possa essere modificato unicamente dal proprietario;
- **Setuid** fa in modo che un eseguibile sia avviato con i permessi assegnati al proprietario e non di chi lo lancia;
- **Setgid** come setuid, ma invece che con i permessi del utente proprietario, con i permessi del gruppo del proprietario.

chmod in forma simbolica

Sintassi chmod

```
chmod [permessi][opzioni] file
```

- Nella notazione simbolica i permessi sono rappresentati come: **x** per l'**esecuzione**, **w** per la **scrittura**, **r** per la **lettura**, **s** per **setuid/setgid** e **t** per **sticky bit**;
- I permessi vengono **assegnati** con il simbolo + e **rimossi** con il simbolo - ;
- Per specificare quali campi della notazione UGO si vuole modificare, si pone **prima** dei permessi **u** per l'**utente**, **g** per il **gruppo** ed **o** per tutti gli altri. Si può usare **a** per indicare tutti e tre assieme.

chmod in notazione ottale

Sintassi chmod

```
chmod [permessi][opzioni] file
```

- Nella notazione ottale i permessi sono dati da alcuni numeri: **1** per l'**esecuzione**, **2** per **scrittura**, **4** per la **lettura**;
- I permessi speciali sono rappresentati da **1** per lo **sticky bit**, da **2** per **setgid** e **4** per **setuid**;
- I permessi vegono assegnati secondo una posizione di 4 cifre che rappresenta nell'ordine permessi speciali + UGO. Il valore dei numeri si somma nel caso bisogna assegnare più valori ad un campo.

Umask

- Serve a visualizzare e/o settare i permessi di default dei file;
- Per visualizzare i permessi di default in orma 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**.

Esempio

Permesso di lettura, scrittura, ed escuzione per l'utente, solo lettura per il gruppo, niente per gli altri in notazione simbolica

`chmod u+wrx file`
`chmod g+r-wx file`
`chmod o-rwx file`
`chmod u+wrx,g+r-wx,o-rwx file` (combinazione in un solo comando dei tre precedenti)

Permesso di lettura, scrittura, ed escuzione per l'utente, solo lettura per il gruppo, niente per gli altri in notazione ottale

`chmod 0700 file` (1 esecuzione + 2 scrittura + 4 lettura = 7)
`chmod 0040 file`
`chmod 0000 file`
`chmod 0740 file` (combinazione dei precedenti)

Esempio

Permesso di lettura, scrittura, ed escuzione per l'utente, solo lettura per il gruppo, niente per gli altri in notazione simbolica

`chmod u+wrx file`
`chmod g+r-wx file`
`chmod o-rwx file`
`chmod u+wrx,g+r-wx,o-rwx file` (combinazione in un solo comando dei tre precedenti)

Permesso di lettura, scrittura, ed escuzione per l'utente, solo lettura per il gruppo, niente per gli altri in notazione ottale

`chmod 0700 file` (1 esecuzione + 2 scrittura + 4 lettura = 7)
`chmod 0040 file`
`chmod 0000 file`
`chmod 0740 file` (combinazione dei precedenti)

MALE ASSOLUTO

Cosa da **NON** fare

chmod 0777 -R /
chown -R utente:gruppo .*

- Nella prima state dando i permessi di lettura, scrittura ed esecuzione a tutti dalla radice del sistema in poi (l'opzione -R indica la ricorsione);
- Nella seconda state dando la proprietà di tutto il sistema all'utente specificato e/o il gruppo.

Da non fare **mai** nella vita!

chown & chgrp

Sintassi di chown e chgrp

```
chown [opzioni] utente:gruppo file
chgrp [opzioni] gruppo file
```

- Per cambiare la proprietà di appartenenza di un file si usa chown per cambiare l'appartenza ad un utente e chgrp per l'appartenenza ad un gruppo;
- Si può usare chown anche per cambiare solo il gruppo, pasta che il nome del gruppo venga preceduto da '**:**';
- Le opzioni sono le medesime, nel caso di una cartella la più significativa è *-R* che cambia l'appartenenza in modo ricorsivo (ovvero a tutto l'albero delle sottocartelle);
- Entrambi i comandi possono essere usati esclusivamente da root.

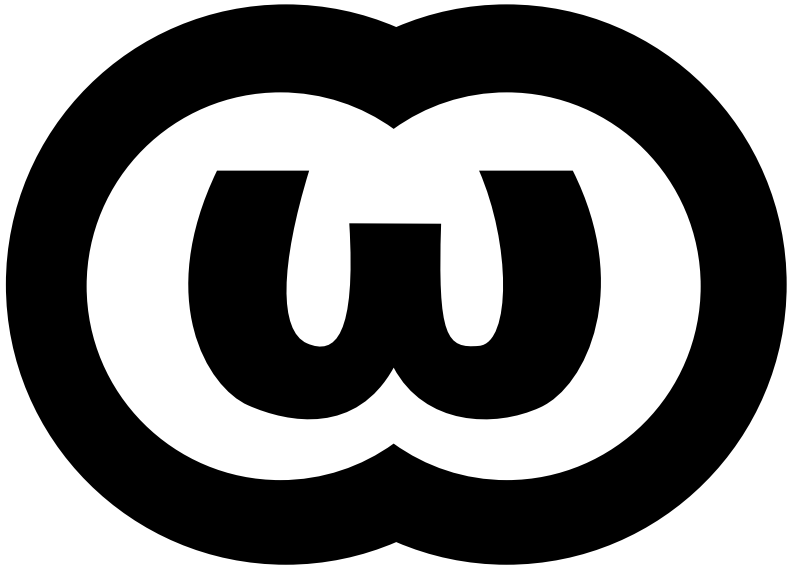
Le capabilities

- A volte è necessario che l'utente esegua un programma che richieda alcuni permessi di root;
- Far girare questo programma con setuid di root non è una buona idea (può essere pericoloso);
- Si può concedere alcuni permessi ad un eseguibile senza dargli i permessi di root;
- Un esempio è il comando ping. Ping ha bisogno di accedere ad i raw socket, accesso riservato solo a root. Nella maggior parte delle distro GNU/Linux viene assegnata la capabilities *cap_net_raw*, in modo che possa essere utilizzato da tutti gli utenti senza i permessi di root;
- Il comando *setcap* permette di assegnare e modificare le capabilities di un eseguibile;
- Il comando *getcap* permette di esaminare le capabilities assegnate ad un eseguibile.

Volete saperne di più?

- Il vostro computer (con su GNU/Linux ;)) ha tutte le risposte: usate le **manpage**
- Cercate su internet (Wiki di Gentoo, Wiki di Arch, Wiki di Debian, Google...)
- Venite a trovarci in Sede :)

May the Schwartz be with you!



Queste slide sono licenziate sotto licenza *Do What The Fuck You Want To Public License (WTFPL).*