### **Elevazione dei privilegi**

***Elevazione manuale***: l’utente digita i comandi per diventare amministratore (*su -*) dopodichè esegue il comando in questione

Possibili problemi

* *Diffusione delle credenziali*: l’utente conosce le credenziali dell’utente privilegiato.
* *Scomodità d’uso*: l’utente deve digitare il comando per l’elevazione dei privilegi (su -)

***Elevazione automatica***: l’utente esegue il comando in questione (per esempio “*passwd user*”). Il SO esegue l’elevazione dei privilegi opportuno.

**Possibili problemi**

* ***Ottenimento di un privilegio enorme:*** un processo SETUID/SETGID ottiene pieni i poteri dell’utente privilegiato. Se l’utente è root il processo può fare tutto.
* ***Ottenimento per l’intera esecuzione:*** un processo SETUID/SETGID mantiene il privilegio elevato per l’intera esecuzione. Una qualunque debolezza nel programma può essere attaccata per ottenere un privilegio.

*I sistemi UNIX moderni sono configurati per l'elevazione automatica dei privilegi di alcuni programmi di sistemi.*

### **SETUID & SETGID**

Un processo in esecuzione assume le credenziali, UserID e GroupID, dell’ utente che ha lanciato il comando se i bit SETUID e SETGID sono disattivati.

Se il bit SETUID, UserID=UserID utente creatore del file.

Se il bit SEGID, GroupID=GroupID gruppo creatore del file.

### **Utente e gruppo reale**

*Vale sempre la relazione:*

UID reale = UID utente che ha *lanciato* il comando

GID reale = GID gruppo che ha *lanciato* il comando

### **Utente e gruppo effettivo**

*Se i bit SEUID e SETGID sono disattivati:*

UID effettivo = UID utente che ha *lanciato* il comando

GID effettivo = GID gruppo che ha *lanciato* il comando

*Se i bit SEUID e SETGID sono attivati:*

UID effettivo = UID utente che ha *creato* il file

GID effettivo = GID gruppo che ha *creato* il file

### **Abbassamento e ripristino dei privilegi**

Se l'applicazione non svolge operazioni critiche può decidere di abbassare i propri privilegi a quelli dell'utente che ha eseguito il comando (**privilege drop**).

Quando l'applicazione svolge operazioni critiche, ripristina nuovamente i privilegi ottenuti tramite l'elevazione automatica (**privilege restore**).

#### Gli User/Group ID disponibili

sono previste tre tipologie di user/group ID:

* user/group ID reale
* user/group ID effettivo
* user/group ID salvato

Quando parte un processo, le credenziali “salvate” contengono una copia delle credenziali “effettive”:

* UID effettivo -> UID salvato
* GID effettivo -> GID salvato

#### Chiamate di sistema getuid(), geteuid(), setuid(), seteuid()...

* **getuid()** ritorna l’UID reale del processo invocante
* **geteuid()** ritorna l’UID effettivo del processo invocante
* **setuid(uid)** imposta l’UID reale di un processo al valore *uid*
* **seteuid(uid)** imposta l’UID effettivo di un processo al valore *uid*
* **getresuid()** restituisce tutti gli UID del processo invocante (anche l’UID salvato)
* **setresuid(u, e, s)** imposta tutti gli UID del processo invocante. Se u/e/s = -1 allora UID/EID/SID non viene modificato. Se e = 0 allora UID/EID/SID = u/e/s.

L’*abbassamento permanente* dei privilegi avviene impostando tutti gli UID a un valore non privilegiato: **setresuid(uid, uid, uid)** gli UID reale, effettivo e salvato assumeranno lo stesso valore **uid**.

L’*abbassamento temporaneo* dei privilegi si ottiene impostando lo UID effettivo a un valore non privilegiato: **setresuid(-1, getuid(), -1)** in queso modo si preserva lo UID salvato e si può effettuare il rispristino. ***setresuid(-1, getuid(), -1) = seteuid(uid)***

Il *rispristino temporaneo* dei privilegi si ottiene impostando lo UID effettivo ad un valore privilegiato: **setresuid(-1, *privileged\_id*, -1)** dove **privileged**\_id è l’UID dell’utente privilegiato ottenuto in partenza (tipicamente 0 per l’utente root).