

Utenza

# Utenza

- Riferimenti:
  - appunti, capp. 79 – 82
  - APUE, cap. 9.2
  - Michael H. Jackson, Linux Shadow Password HOWTO
  - The Linux-PAM System Administrators' Guide, Andrew G. Morgan,  
<http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html>

# Login

- procedura di login

1. getty in attesa su tty di uno username

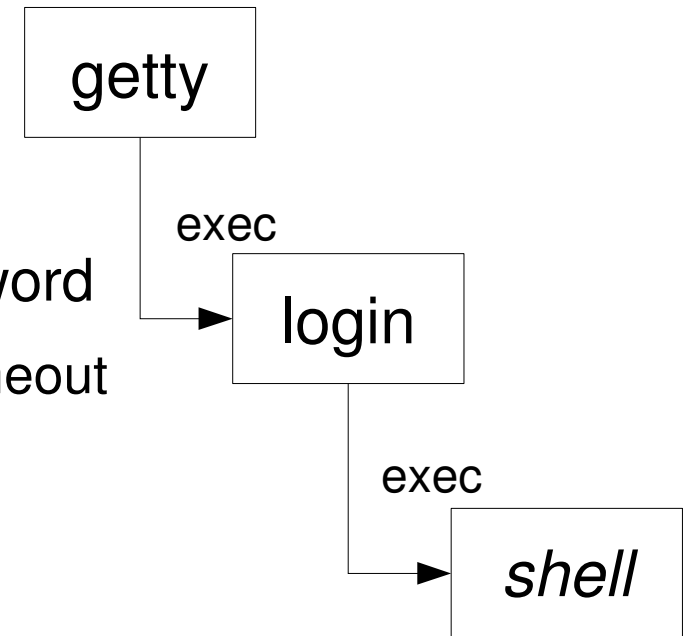
2. getty esegue login che verifica la password

- se la password è errata ritenta dopo timeout
- se l'utente è root
  - verifica che tty sia in /etc/securetty
- se l'utente non è root
  - verifica che non esista /etc/nologin

3. login stampa le informazioni di login (lastlog, mail, /etc/motd)

4. login registra il successo del login nel registro di sistema (log)

5. login passa alla home directory dell'utente (oppure /) ed esegue la sua shell (oppure /bin/sh)



# Il registro degli account

- risiede nel file `/etc/passwd`
- è un database testuale di record a 7 campi, separati da “:”
  1. nome utente
  2. password (crittata)
  3. uid
  4. gid
  5. nome completo
  6. home directory
  7. shell
- esempio di record (1 riga di `/etc/passwd`)

```
zack:bGqudB41a4a:3148:3148:Stefano Zacchioli,,,:/home/zack:/bin/bash
```

- riferimenti
  - `man 5 passwd`
- comandi: `passwd`, `su`, `id`

# Il registro degli account

1. molte applicazioni hanno necessità di accedervi
    - e.g. gli inode contengono l'uid, ma non lo user name, la necessità di `/etc/passwd` per mostrarlo!
  2. in caso di reti che condividono utenti, tutte le macchine necessitano di accesso a `/etc/passwd`
  3. le password sono crittate con crypt (man 3 crypt), è un algoritmo basato su DES, suscettibile a brute force attack
- l'azione combinata di 1, 2 e 3 comporta problematiche di sicurezza
  - soluzione: shadow password

# Shadow password

- osservazione: il campo password di `/etc/passwd` è riservato e necessario durante il login, gli altri campi non sono riservati e sono necessari anche dopo il login
- utilizzando le shadow password il campo password viene spostato da `/etc/passwd` in `/etc/shadow`
- `/etc/shadow` non è accessibile agli utenti di sistema
- quando le shadow password sono in uso, il campo password di `/etc/passwd` viene settato ad “x”
- altri vantaggi delle shadow password
  - gestione degli expire time

# Il registro dei gruppi

- risiede nel file `/etc/group`
- il formato è analogo a quello di `/etc/passwd`, con i campi seguenti
  1. group name
  2. password
  3. group id
  4. utenti membri
- le password dei gruppi sono utilizzate raramente, nel caso siano utilizzate il file `/etc/gshadow` fornisce funzionalità analoghe a quelle di `/etc/shadow`
- comandi: `newgrp`

# Creazione utenti

- procedura di creazione di nuovi utenti
  1. aggiunta utente a `/etc/passwd`
  2. aggiunta gruppo a `/etc/group`
  3. aggiunta password a `/etc/shadow`, `/etc/gshadow`
  4. creazione home directory
- `adduser`
  - automatizza la procedura di creazione nuovi utenti
  - gestisce scheletri di home directory (`/etc/skel/`)
  - gestisce la creazione di utenti di sistema
  - `/etc/adduser.conf`
- comandi: `adduser`, `addgroup`



# Eliminazione utenti

- procedura di eliminazione di utenti
  - è inversa alla procedura di creazione
  - in aggiunta può essere necessario cancellare tutti i file dell'utente (che non risiedono necessariamente nella sua home)
    - `find / -user <username>`
- `deluser`
  - automatizza la procedura di eliminazione di utenti
  - supporta la rimozione di tutti i file dell'utente
  - `/etc/deluser.conf`
- comandi: `deluser`, `delgroup`

# Pluggable Authentication Modules

- inizialmente, login era l'unico programma di autenticazione e la password era l'unico meccanismo disponibile
- nei moderni sistemi GNU/Linux altri programmi hanno necessità di autenticare gli utenti ...
  - e.g.: sshd, pppd, gdm/xdm/kdm, xlock, ...
- ... e meccanismi diversi dalla password sono disponibili
  - e.g.: autenticazione RSA/DSA di ssh, parametri biometrici ...
- PAM permette di utilizzare molteplici *servizi di autenticazione*
  - configurabili ed associabili alle applicazioni senza dovere ricompilare queste ultime
  - la scelta e la configurazione dei servizi di autenticazione sono effettuate dall'amministratore, non dallo sviluppatore

# PAM – installazione

- PAM è modulare, diviso in librerie dinamiche
- installazione tipica:
  - `/lib/security/` moduli PAM (e.g. `/lib/security/pam_unix.so`)
  - `/etc/pam.d/` configurazione applicazioni che utilizzano PAM (e.g. `/etc/pam.d/login`)
- i file di configurazioni sono file testuali composti da record a 4 campi: `<module-type, control-flag, module-path, arguments>`
- ogni file di configurazione lista i moduli PAM utilizzati da una data applicazione
- riferimenti:
  - `man 7 pam`

# PAM – module type

- module-type
  1. *auth* verifica l'identità dell'utente (e.g. password)
  2. *account* verifica lo stato dell'account dell'utente (e.g. è scaduto?)
  3. *password* controlli aggiuntivi sulla password (se prevista)
  4. *session* login/logout hook

# PAM – control flag

- intuitivamente, ogni modulo PAM corrisponde ad una funzione, il cui valore di ritorno è ternario:
  - SUCCESS, IGNORE, FAILURE
- le funzioni corrispondenti ai moduli listati in un file di configurazione vengono eseguiti sequenzialmente
- il control flag stabilisce la politica a riguardo del valore di ritorno richiesto per un dato modulo
  - *requisite* richiede SUCCESS/IGNORE, termina se FAILURE
  - *required* richiede SUCCESS/IGNORE, ma non termina
  - *sufficient* in caso di SUCCESS, termina la procedura
  - *optional* tratta FAILURE come IGNORE
- in caso di soli valori IGNORE non viene consentito l'accesso

# PAM – esempio

- /etc/pam.d/login

```
auth      requisite pam_securetty.so
auth      required  pam_nologin.so
auth      required  pam_env.so
auth      required  pam_unix.so nullok
account   required  pam_unix.so
session   required  pam_unix.so
session   optional  pam_lastlog.so
session   optional  pam_motd.so
session   optional  pam_mail.so standard noenv
password  required  pam_unix.so nullok obscure min=4 max=8
```