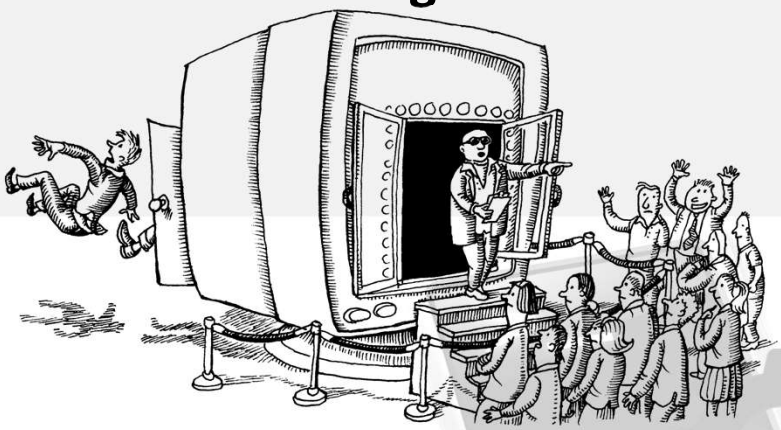
User Management



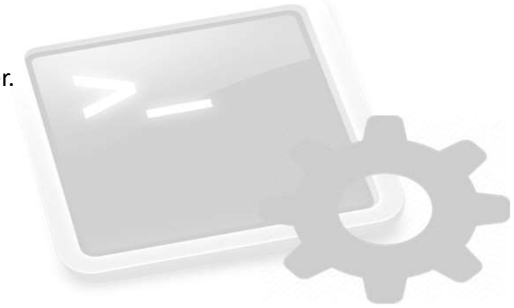
Chapter 8, Unix and Linux System Administration Handbook

Index

- **User Management**
  - Definition and Generation
  - Environment configuration
  - Group management
  - Elimination
- **Security and access control**
  - password management
- **Privilege delegation**

User Management

- Steps to create a new user:
  1. Decide some **basic configuration parameters** for the user.
     - username, identification (UID), user group (GID), user root directory ($HOME) location, kind of shell, …
  2. Add that parameters to the system **database**.
     - /etc/passwd, /etc/shadow, /etc/group
  3. Configure **security aspects**.
     - Password, special privileges, account expiration date,…
  4. Create and configure **$HOME directory** for the user.
     - Shell, X environ, owner and group, …
  5. Check that new account works correctly.



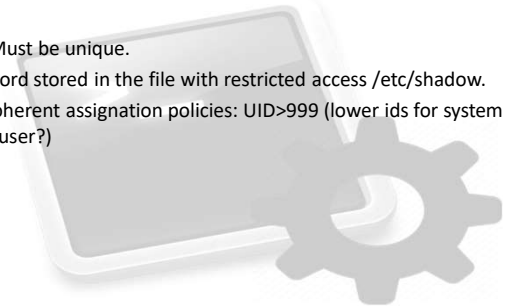
User Management

- **Steps 1 & 2**: Definition and Creation
  - The file **/etc/passwd**:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
…
test:x:500:500:Usuario test:/home/test:/bin/bash
apache:x:48:48:Apache:/var/www:/bin/false
…
```

test:x:500:500:Usuario test:/home/test:/bin/bash:

- **Name:** all the resources employed by the user identified with that label. Must be unique.
- **Password**: the x indicates the account is protected by an encrypted password stored in the file with restricted access /etc/shadow.
- **UID:GID**: numerical identifiers for user and group. UID must be unique. Coherent assignation policies: UID>999 (lower ids for system accounts). Do not reuse UIDs (¿What happens if we want to restore an old user?)
- **GECOS**: Personal information about the user. Limitless, comma separated.
- **$HOME directory**: system directory where the user stores its files.
- **Shell**: binary associated to the shell employed by the user.



User Management

- **Steps 1 & 2**: Definition and Creation
  - UNIX manages users through **groups**:
    - Allows grouping users to share files/resources.
    - Files have specific access permisions for the group.
    - The group assigned to a user can be found in: /etc/group or in the GID of /etc/passwd
  - The file **/etc/group**:

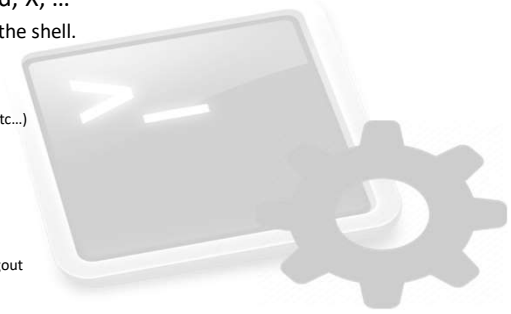
```
test:*:500:
cdrom:*:24:test
```

- **Name:** group name
- **Password**: in general, not employed.
- **GID**: group identifier.
- **Additional users**: a user can belong to more than one group.

  - Command **groups**: identifies the groups for a specific user.
  - Command **newgrp**: allows a user to change its group.
  - Command **id**: prints IDs (numerical) of user and group.

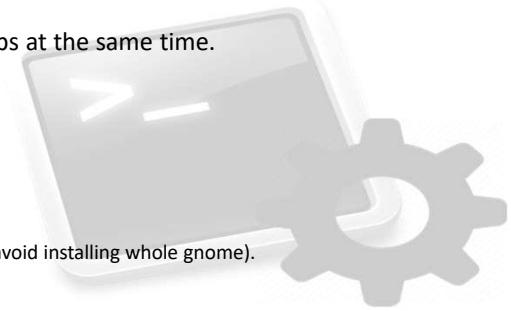
User Management

- **Step 3**… Next section
- **Step 4**: Creation and configuration of $HOME
  - Usually under /home directory, with the same name as the username.
  - Creation and permission assignation: **mkdir + chown + chg**rp.
  - Configuration file generation: Shell, history command, X, …
    - The directory **/etc/skel/**: Contains the configuration files for the shell.
    - Its content must be copied (/etc/skel/*) to $HOME
    - Common files for shell configuration:
      - [tcsh,csh]: /etc/csh.login (environment), /etc/csh.cshrc (functions, alias, etc…)
      - [bash]: /etc/bashrc, /etc/profile
      - [sh]: /etc/profile
    - Files that can be customized by the user
      - [tcsh]: $HOME/.login, $HOME/.cshrc, $HOME/.tcshrc
      - [csh]: $HOME/.login $HOME/.cshrc
      - [bash]: $HOME/.bash, $HOME/.login, $HOME/.bashrc, $HOME/.bash_logout
      - [sh]: $HOME/.profile



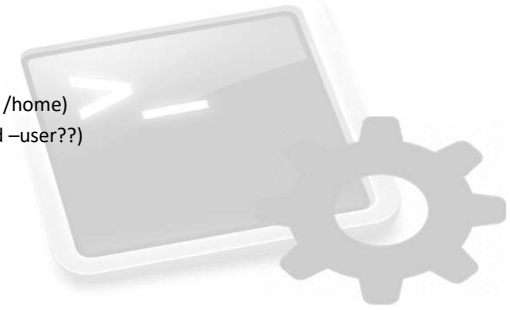
User Management

- **Step 5**: Account checking:
  - su – newuser / ssh localhost –l newuser / Ctrl+Alt+F1
- More pending tasks (Lesson 5, Resource management)
  - Quota assignation(disk), resource limits (limits), mail alias, …
- Automatization tools:
  - The command **adduser** allows to perform all the steps at the same time.
    - Default definition of users: /etc/adduser.conf
  - Command **usermod** to modify a user account.
  - Command **addgroup**: group management.
  - GUI "**users-admin**"
    - Part of "gnome-system-tools"
    - "apt-get –no-install-recommends gnome-system-tools" (to avoid installing whole gnome).
    - Not very useful (How to create 100 new users?)



User Management

- **Removing** system users:
  - Change the **shell** to **/bin/false**
    - Modify the line in /etc/passwd forbids a user to start a new session.
    - Not safe, some services not using shell still available (ftp).
    - Only to temporary block a user account.
  - Account Blocking
    - Command: **passwd –l** (blocks the password of an account).
  - Account elimination
    - Command: **userdel** [-r] username (-r removes user files from /home)
    - Some work still pending: mail or print queues, cron, etc. (find –user??)



Index

- User Management
  - Definition and Generation
  - Environment configuration
  - Group management
  - Elimination
- **Security and access control**
  - password management
- Privilege delegation

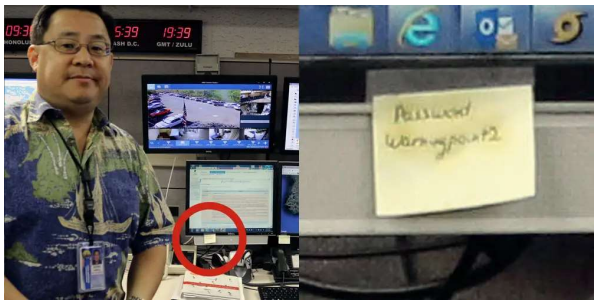
Security

- Key element for security: user **password**
  - Extremely sensitive information, any intruder with that information could attack the system.
  - **NEVER write it down** and change it regularly.
  - Some basic rules to choose a "safe" password:
    - No username neither variations (root/root ??)
    - No dictionary words or familiar names (test/temporal ??)
    - Combine upper case/ lower case letters and numbers.
    - Do not repeat patterns with each change (root1, root2, root3, root4,…)

https://www.teamsid.com/splashdatas-top-100-worst-passwords-of-2018/

Security

- Hawaii's Emergency Management Agency's operation officer



https://www.businessinsider.com/hawaii-emergency-agency-password-discovered-in-photo-sparks-security-criticism-2018-1

- Tv5Monde revealed his own passwords in an interview

https://arstechnica.com/information-technology/2015/04/hacked-french-network-exposed-its-own-passwords-during-tv-interview/

Security

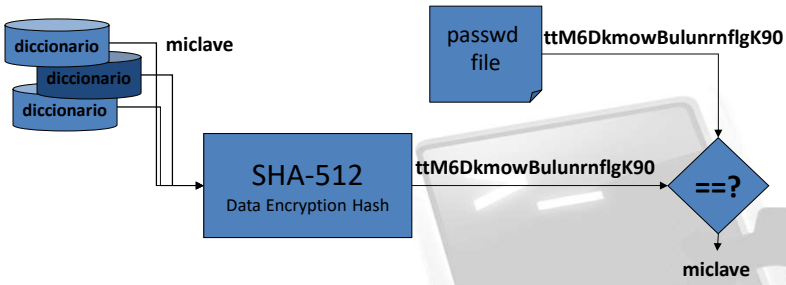
- Password protection in the system: **Encryption**
  - **If we ask the user to avoid writing down its passwd, system should not do it neither. How to compare?**
  - Solution: One direction Encryption algorithm.

  - Command passwd: assignation/modification of user password.
    - The encrypted password generated is saved in file /etc/shadow
    - But… **if the user can modify this file, couldn't also modify the rest of the users passwd?**

Assign permission to the command, not the file (chmod 4700).
Command passwd with SETUID activated (the process is executed with owner UID)
-rwsr-xr-x 1 root root 31704 nov 14 2009 /usr/bin/passwd

Security

- ¿Is Encryption enough?
  - Even encrypted, the content of /etc/shadow file can still be useful for an attacker.



  - This file must be protected against reading.
  - Additional rules to reduce brute-force security risks in password definition: **Aging and Cracklib.**

Security

- How long does it take to crack a password? (bruteforce)
  - bcrypt hashes cracked by 8 A100 GPUs

Number of Characters	Numbers Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	7 secs	14 secs	23 secs
5	Instantly	11 secs	6 mins	15 mins	27 mins
6	1 sec	5 mins	5 hours	15 hours	1 day
7	10 secs	2 hours	2 weeks	1 month	3 months
8	2 mins	2 days	2 years	7 years	17 years
9	16 mins	2 months	84 years	411 years	1k years
10	3 hours	4 years	4k years	25k years	85k years
11	1 day	111 years	228k years	1m years	5m years
12	2 weeks	2k years	11m years	97m years	419m years
13	4 months	75k years	616m years	6bn years	29bn years
14	3 years	1m years	32bn years	376bn years	2tn years
15	30 years	50m years	1tn years	23tn years	144tn years
16	303 years	1bn years	86tn years	1qd years	10qd years
17	3k years	34bn years	4qd years	89qd years	705qd years
18	30k years	894bn years	234qd years	5qn years	49qn years

https://hivesystems.io/password

Security

- Password protection in the system: **Password Aging**
  - Security mechanism that forces the users to change their password regularly.
  - It is also useful to maintain an updated list of valid users.
  - Command **chage**
    - Can be interactive (chage user) or through options (chage -<opt> user)
    - Option **–m**: minimal number of days between changes. If 0, user can change its passwd at any time.
    - Option **–d**: Number of days since January 1, 1970 when the passwd was last changed.
    - Option **–M**: validity period for the password (with –W activates previous warnings)
      - **When MAX_DAYS (-M) + LAST_DAYS (-d) is less than current day, user required to change password.**
    - Option **–E**: number of days since 1970 on which the user account will no longer be accessible (expiration date).
    - Option **–I**: number of days of inactivity after a password has expired before the account is locked.
  - All this information, as well as encrypted password is stored in /etc/shadow.

Security

- The file **/etc/shadow**:

```
root:$1$h.OyoX6l$85/.BIV6ziyVW2G.8Gfyr0:11547:0:99999:7:-1:-1:-1073744240
bin:*:11547:0:99999:7:::
daemon:*:11547:0:99999:7:::
adm:*:11547:0:99999:7:::
lp:*:11547:0:99999:7:::
…
named:*:11547:0:99999:7:::
test:$1$decPVavl$ttM6DkmowBulunrnflgK90:11547:0:99999:7:::
…
```

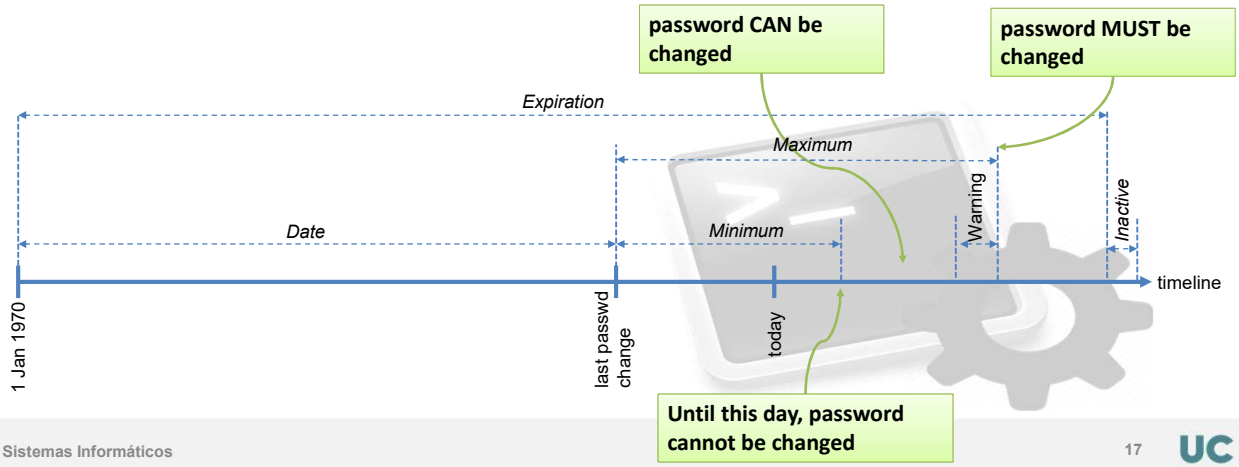
test:1decPVavl$ttM6DkmowBulunrnflgK90:11547:0:99999:7: : :

  - **Name**: username (same as passwd file).
  - **Password**: encrypted password ($algorithm$salt$hashed).
  - **Date**: last change date (days since January 1, 1970)
  - **Minimum**: number of days before next passwd change (if set to zero, no limits between changes).
  - **Maximum**: upper limit for password modification.
  - **Warning**: number of warning days previous to password caducity.
  - **Inactive**: inactivity days after caducity(then locking)
  - **Expire**: Account expiration date.

Security

- The file **/etc/shadow**:

```
root:$1$h.OyoX6l$85/.BIV6ziyVW2G.8Gfyr0:11547:0:99999:7:-1:-1:-1073744240
bin:*:11547:0:99999:7:::
daemon:*:11547:0:99999:7:::
adm:*:11547:0:99999:7:::
lp:*:11547:0:99999:7:::
…
named:*:11547:0:99999:7:::
test:$1$decPVavl$ttM6DkmowBulunrnflgK90:11547:0:99999:7:::
…
```



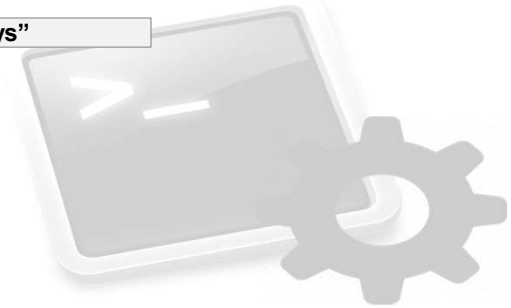
(aside) 1 January 1970

- Known as epoch. used as starting date or day in calculations (command or config file)
  - Calculate the number of days from 1 January 1970 to current date

```
# expr $(date +%s) / 86400
```

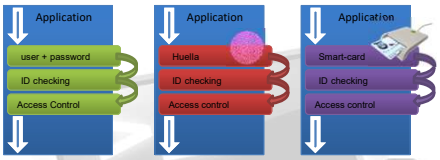
  - Calculate the date from supplied days

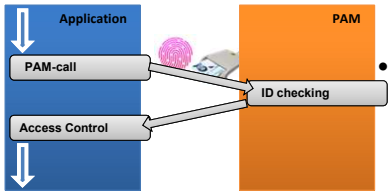
```
# date –d “1970-01-01 <num-days> days”
```



Security

- Unification of Authentication Mechanisms:
  - Basic authentication mechanism: login + password. Currently, safer alternatives: Smart-cards (DNIe), Biometry (touchID).
  - Is it necessary a different sw version for each authentication mechanism?

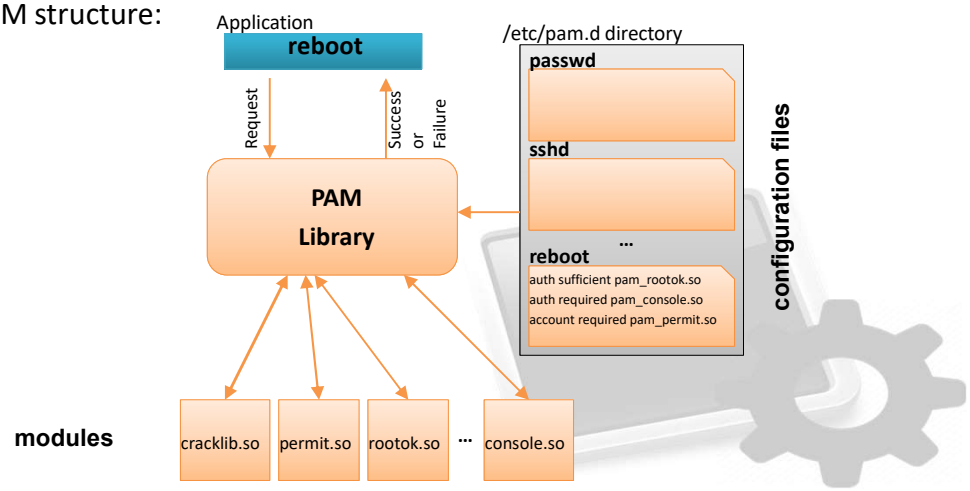




- **PAM: Pluggable Authentication Module**
  - Common framework for authentication.
  - Its library allows developers to abstract from authentication process.

Security

- Internal PAM structure:



Security

reboot

```
auth sufficient pam_rootok.so
auth required pam_console.so
account required pam_permit.so
```

- PAM configuration files:
  - Each service/application making use of PAM has its own file in **/etc/pam.d**
  - List of rules with the format: [module_type] [control_flag] [PAM_module] [arguments]
  - **Module types** (depend on authentication process aspect to deal with):
    - **auth**: authentication. Example: verify password validity.
    - **account**: verify access permissions. Example: check if a use account has expired.
    - **password**: interface for user password change.
    - **session**: configuration and administration of user sessions.
  - **control flags** (simple syntax)
    - **required**: the module result must be successful for auth to continue (remaining stacked modules are still invoked)
    - **requisite**: same as required but no more modules invoked
    - **sufficient**: ignored if fails. If success (and no prior required module failed) following modules ignored
    - **optional**: only relevant if no other modules associated with this service+type
    - **include**: pull in all lines in the config file reserved
  - *control flags (complicated syntax): [val1=action1 val2=action2 …] (see man pam.d)*

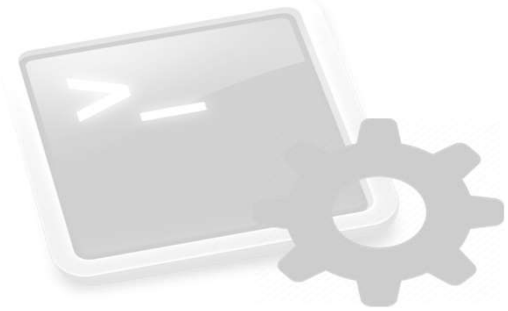
Security

- Password protection in the system: **Cracklib module**
  - PAM module in charge of the verification of password strength.
    - Target: forbid the utilization of weak passwords (1234).
  - Password strength is checked in the following way:
    - Dictionary comparison (default English). Could be linked to different dictionaries.
    - Comparison with previous password (also available in pam_unix): Upper / Lower case, number of different characters.
      - /etc/security/opasswd (warning, same protection level as shadow file)
    - Length & Strength: is it too short? mix of different character types?
  - Required package: **libpam_cracklib**
    - #apt-get install libpam-cracklib
  - Configuration file: /etc/pam.d/common-password.
    - Include the following line: password required pam_cracklib.so retry=3 minlen=6

Warning: be carefull with the interactions between pam_cracklib and pam_unix modules, take a look at man pam_cracklib for further details.

Index

- User Management
  - Definition and Generation
  - Environment configuration
  - Group management
  - Elimination
- Security and access control
  - password management
- **Privilege delegation**



Privilege Delegation

- Sometimes it could be useful to provide normal users permission to perform some tasks reserved to root (halfway between raw user and admin).
- Command **sudo**:
  - Allows a user to execute a command as if it were root.
  - The file **/etc/sudoers** contains the commands each user can execute
    - This file can only be edited with command visudo.
    - Format: [user] [SYSTEMS=(privileges)] [allowed actions]
    - Example: %admin ALL=(ALL:ALL) NOPASSWD: /usr/bin/apt-get
      - %admin: all the users in the admin group (single user without %)
      - ALL: from any HOST/IP direction
      - (ALL:ALL): can execute command as any user:group (not only root)
      - NOPASSWD: no password required
      - /usr/bin/apt-get: list, comm separated, of allowed commands.
    - Users/hosts/commands grouping is usually managed through alias.
    - Ubuntu, OSX, make use of sudo for administration tasks (root account not exposed to the user). In the sudoers file: user (ALL)=ALL.