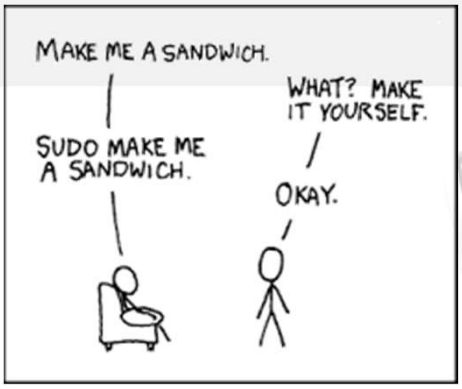
User Management



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.

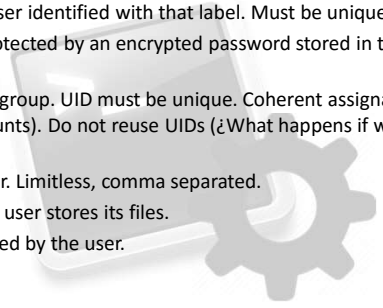
User Management

```
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
…
```

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

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.
- **Root 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 en el GID de /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 ó ssh localhost –l newuser ó Ctrl+Alt+F1
- More pending tasks
  - 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 Manageme
  - Definition and C
  - Environment co
  - Group management
  - Elimination
- **Security and access control**
  - password management
- Privilege delegation

Security

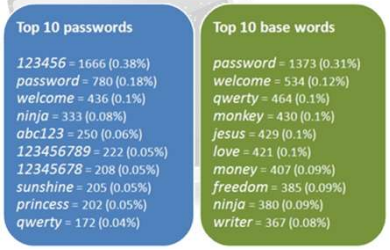
- Minister of Internal Affairs ("Ministro del Interior"), Greek Government

User + passwd in a post-it !!!!

www.microsiervos.com/archivo/seguridad/ministro-griego-contrasena.html

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,…)



Security

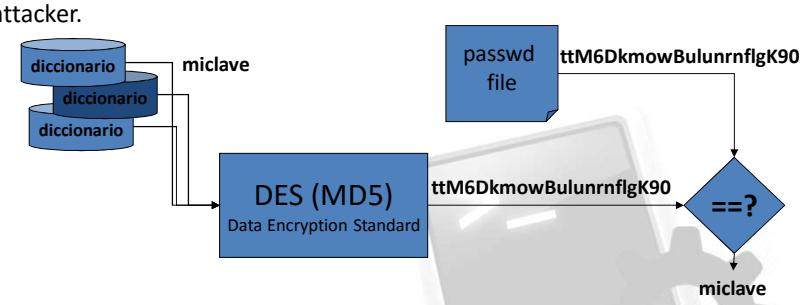
- Password protection in the system: **Encryption**
  - If we ask the user to avoid writing down its passwd, system should not do it neither. How to compare?
  - Solution: One direction Encryption algorithm.

  - 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

- 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 cannot change its passwd.
    - Option **–M**: validity period for the password (with –W activates previous warnings)
    - Option **–d**: Number of days since January 1, 1970 when the passwd was last changed.
    - 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**:

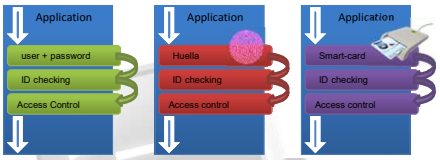
```
root:$1$h.OyoX6l$85/.BIV6ziyVW2G.8Gfyr0:11547:0:99999:7:-1:-1:-
     1073744240
bin:*:11547:0:99999:7:::
daemon:*:11547:0:99999:7:::
adm:*:11547:0:99999:7:::
lp:*:11547:0:99999:7:::
…
named:*:11547:0:99999:7:::
test:$1$decPVavl$ttM6DkmowBulunrnflgK90:11547:0:99999:7:::
…
```

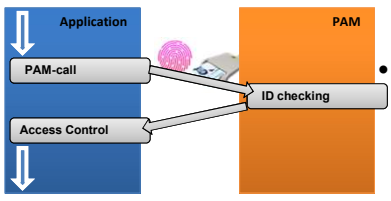
test:1decPVavl$ttM6DkmowBulunrnflgK90:11547:0:99999:7: : :

- **Name:** username.
- **Password**: encrypted password.
- **Date**: last change date (days since January 1, 1970)
- **Minimum**: number of days before next passwd change.
- **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

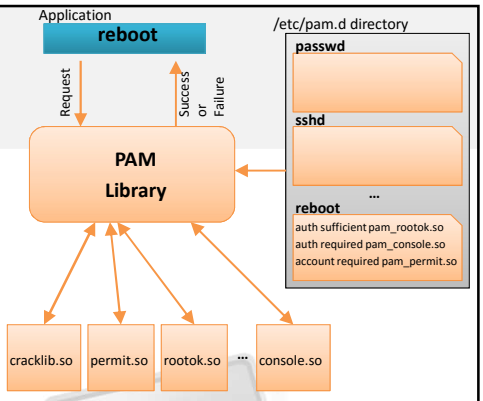
- Unification of Authentication Mechanisms: **PAM**
  - Basic authentication mechanism: login + password. Currently, safer alternatives: Smart-cards (DNIe), Biometry (touchID).
  - Is it necessary a different sw version for each authentication mechanism?





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

Security



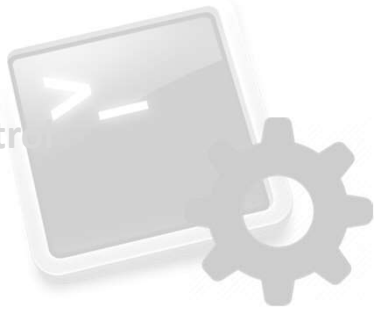
- Internal PAM structure:
  - **Configuration files**:
    - Each service/application making use of PAM has its own file in **/etc/pam.d**
    - File format : [module_type] [control_flag] [PAM_module] [arguments]
  - **PAM Modules**:
    - 4 types, depending on the authentication process aspect to deal 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.

Security

- Password protection in the system: **Cracklib**
  - 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: Upper / Lower case, number of different characters.
    - Length: is it too short?
    - ...
  - Required module: **libpam_cracklib** (installed)
    - #apt-get install libpam-cracklib
  - Configuration file: /etc/pam.d/common-passwd.
    - Include the following line: password required pam_cracklib.so retry=3 minlen=6

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.
- 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)NOPASSWD /usr/bin/apt-get
      - %admin: all the users in the admin group
      - ALL: from any HOST/IP direction
      - (ALL): can execute command as any user (not only root)
      - NOPASSWD: no password required
      - /usr/bin/apt-get: list, comm separated, of allowed commands.
    - Users/hosts/commands grouping is usually managed thorugh alias.
    - Ubuntu, OSX, make use of sudo for administration tasks (root account not exposed to the user). In the sudoers file: user (ALL)=ALL.