

Advanced Linux System Administration

Topic 5. User management



Pablo Abad Fidalgo
José Ángel Herrero Velasco

Departamento de Ingeniería Informática y Electrónica

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

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 those 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.
- **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 permissions 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 assignment: **mkdir + chown + chgrp**.
 - 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:
 - 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 from starting a new session.
 - Not safe, some services not using shell still available (ftp).
 - Only to temporarily 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 or 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...).

Top 10 passwords

123456 = 1666 (0.38%)
password = 780 (0.18%)
welcome = 436 (0.1%)
ninja = 333 (0.08%)
abc123 = 250 (0.06%)
123456789 = 222 (0.05%)
12345678 = 208 (0.05%)
sunshine = 205 (0.05%)
princess = 202 (0.05%)
qwerty = 172 (0.04%)

Top 10 base words

password = 1373 (0.31%)
welcome = 534 (0.12%)
qwerty = 464 (0.1%)
monkey = 430 (0.1%)
jesus = 429 (0.1%)
love = 421 (0.1%)
money = 407 (0.09%)
freedom = 385 (0.09%)
ninja = 380 (0.09%)
writer = 367 (0.08%)

Security

- Minister of Home Affairs (*Ministro del Interior*), Greek Government.

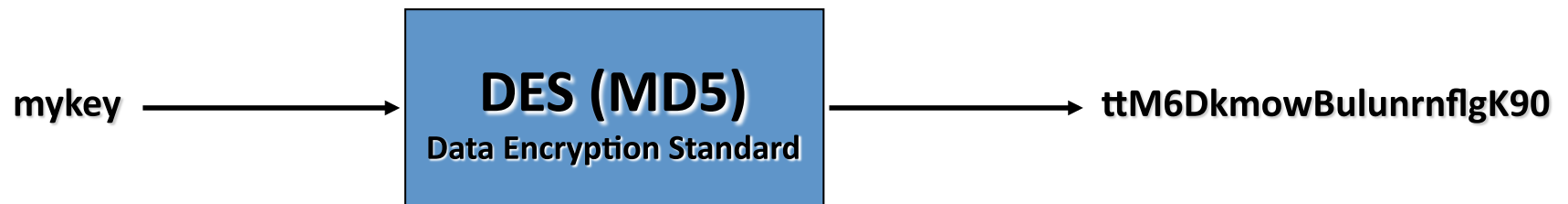
User + passwd
on a post-it !!!!



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

Security

- Password protection in the system: **Encryption:**
 - If we ask the user to avoid writing down their passwd, the system should not do it either. How can we 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 he also modify the rest of the users' passwords?

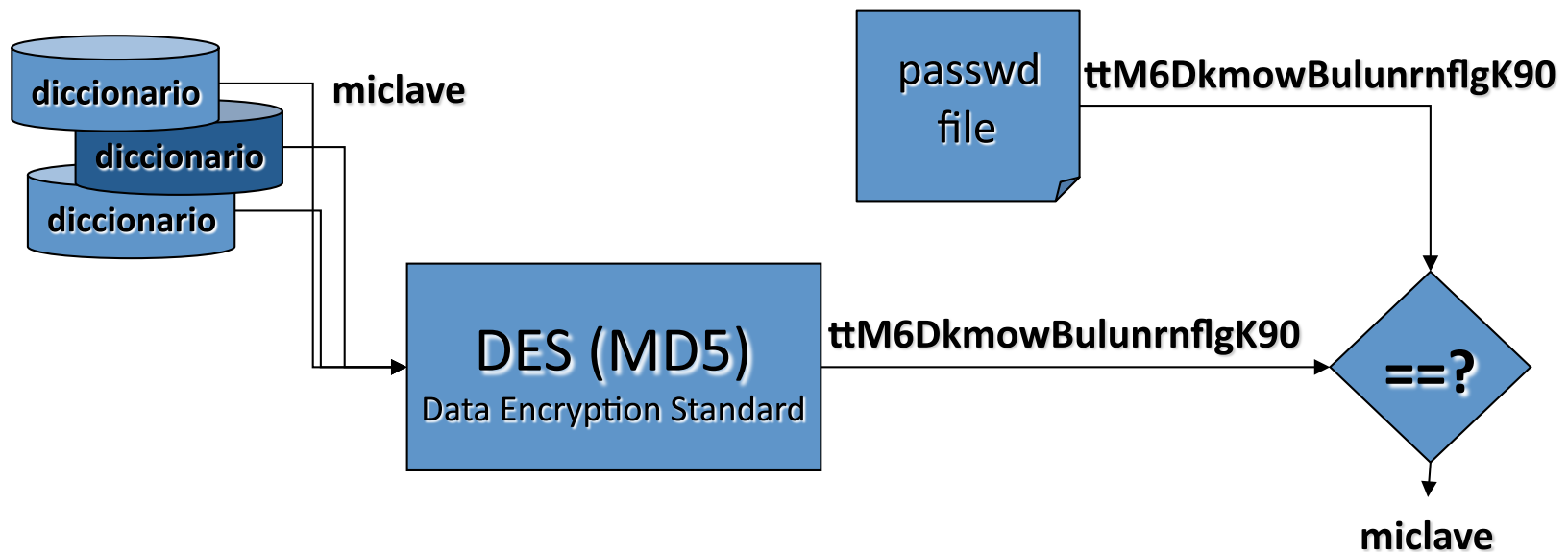
Assign permission to the command, not the file (chmod 4700).

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 contents 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**:

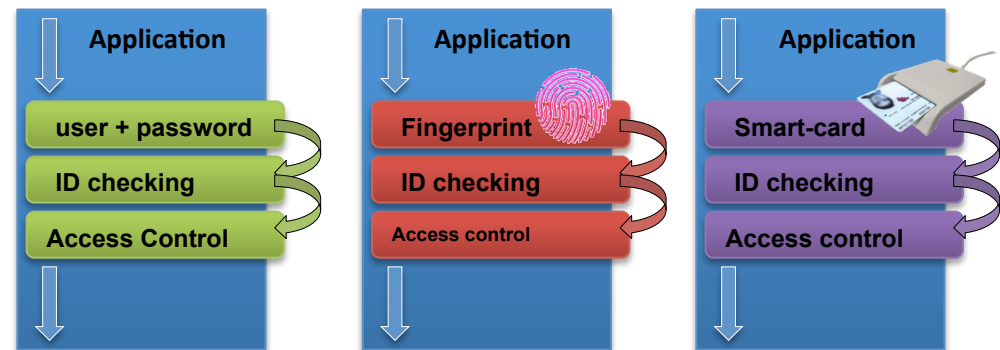
```
root:$1$h.OyoX6l$85/.BIV6ziyVW2G.8Gfyr0:11547:0:99999:7::-1:-1:-1073744240
bin:*:11547:0:99999:7:::
daemon:*:11547:0:99999:7:::
adm:*:11547:0:99999:7:::
lp:*:11547:0:99999:7:::
...
named:*:11547:0:99999:7:::
test:$1$decPVavl$ttM6DkmowBulunrnflgK90:11547:0:99999:7:::
...
```

test:	\$1\$decPVavl\$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 prior to password expiration.
- **Inactive:** inactivity days after expiration (then locking).
- **Expire:** account expiration date.

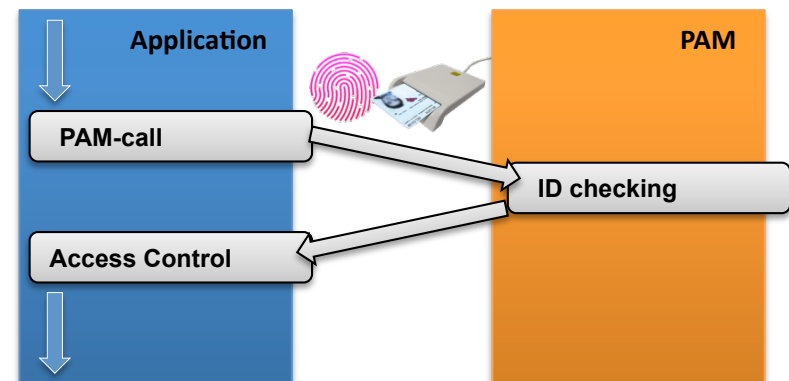
Security

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

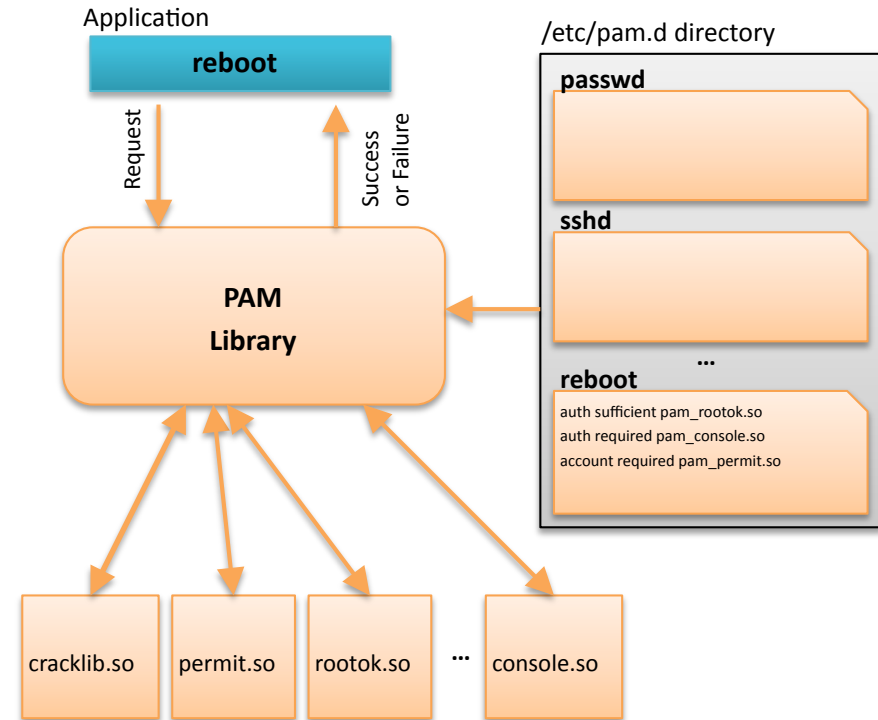


- **PAM: Pluggable Authentication Module:**

- 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 be dealt with.
 - **auth:** authentication. Example: verify password validity.
 - **account:** verify access permissions. Example: check if a user 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 verifying password strength:
 - Objective: to forbid the use 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-password`:
 - 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 give normal users permission to perform some tasks reserved for 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: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, comma 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.