Politécnico de Tomar
Polytechnic University

Bachelor of Science in Computer Engineering

Computer Security (2025/2026)

Practical Assignment 1 – Security management on Linux servers

Objectives

- Install Linux servers in virtual environments
- Configure network interfaces
- Manage user accounts and groups
- Manage file and folder permissions
- Configure runlevels and services
- Manage disk quotas
- Set up scheduling / Task automation
- Update and install packages (yum/rpms dnf)

Theoretical Introduction

Users and Groups

A user is an entity that runs programs or owns files. Examples of users are:

- Other computer systems.
- System functions that run automatically (e.g. accounting systems);
- User groups;
- Users.

There are two elements of user identification:

- Username
- User ID

The UID and GID determine access rights to files and other system resources.

The users are defined in the files: **/etc/passwd** [Figure 1] and **/etc/shadow** [Figure 2].

The function of the /etc/passwd and /etc/shadow files often raises doubts. From a historical point of view, the /etc/passwd file initially contained all the relevant information about the users, and therefore there was no /etc/shadow file. However, there were several attacks that demonstrated that although users' passwords were not stored in clear text, it was relatively easy to recover them through a dictionary attack. That's why it was decided to remove user passwords from the /etc/passwd file and move them to the /etc/shadow file, where they are stored more securely. After the /etc/shadow file, the password field in /etc/passwd always has an x in all users. The /etc/shadow file can only be accessed by root and has been expanded to contain other information about users, such as password expiration. The way the password is stored in /etc/shadow will be studied in detail in TP classes.

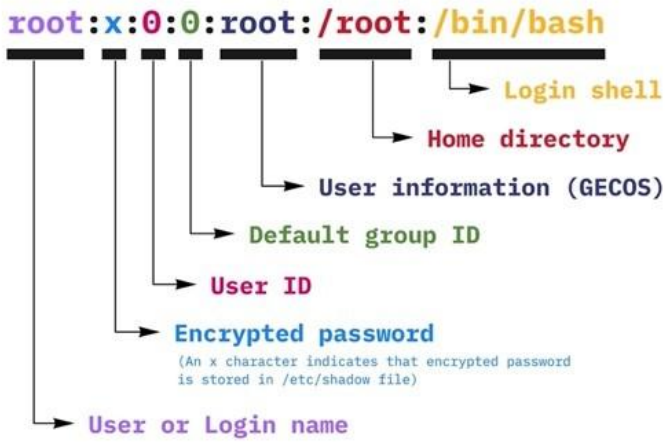


Figure 1 - /etc/password

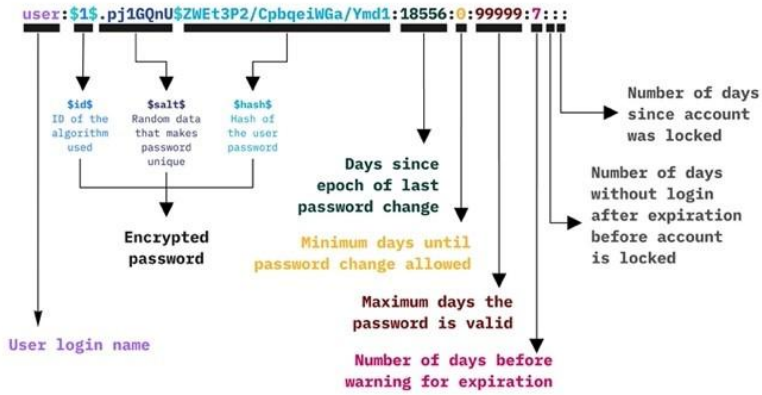


Figure 2 - /etc/shadow

User groups are defined in the files: /etc/group and /etc/gshadow.

The file /etc/passwd – contains one line for each user account in the system, each line consists of 7 fields separated by ":": *username:passwd:UID:GID:name:directory:shell*

The /etc/shadow – file contains the encrypted passwords, each line consists of 8 fields separated by ":".

The fields are:

- Username
- Encript Password
- Last password change (in days since January 1, 1970)
- Minimum number of days between password changes
- Maximum number of days between password changes
- Number of days before the password expires that the user is warned
- Number of days after the password expires until the account is deactivated.

The /etc/group file – Information about groups, each line consists of 4 fields:

username:password:GID:lista_de_usernames

Other relevant files:

- /etc/login.defs
- /etc/default/useradd

Although graphical tools exist for user management, it is sometimes necessary to perform these operations through the command line, for example when administering a remote server by SSH, or when you want to create large number of users using scripts.

All operations on users and groups can be done using the following commands: **`id, useradd, usermod, userdel, groupadd, groupdel, groupmod, passwd.`**

To add users you can use edit the /etc/passwd file or the adduser command. Important note: to edit the passwd and group files the following commands must be used: **`vipw`** and **`vigr`**. Using these commands allows you to prevent passwd and group files from becoming corrupted.

Runlevels and Services

Runlevels are used to define different states of the system. For each state only certain processes are active. The existing states are:

- 0 - Halt (do NOT set initdefault to this!)
- 1 - Single user mode (text mode)
- 2 - Multiuser mode, without NFS (the same as 3, if you do not have networking)
- 3 - Full multiuser mode (text mode)
- 4 - Unused / user definable
- 5 - X11 / full multiuser graphical mode
- 6 - Reboot (do NOT set initdefault to this!)

The initial runlevel is defined in the /etc/inittab file via the "initdefault" entry, the first process to be executed (init) is responsible for starting all the services configured for the initial runlevel of the server.

The most relevant commands related to runlevels are:

1. `runlevel` – Returns the current system runlevel.
2. `Shutdown, Halt, Reboot, Poweroff` – Shut down, stop, and restart the system
3. `/sbin/chkconfig` – application to view and manage runlevels

The services (daemons) work in two distinct modes: standalone – mode in which they are permanently active listening for connections; xinetd – in this mode there is a service (xinetd) that waits for connections to several other services, when receiving a connection it tries to launch the corresponding service.

Modo standalone

1. Starting/stopping a service is accomplished using the scripts in the /etc/rc.d/init.d/ directory
    1. Exemplo: /etc/rc.d/init.d/ntpd start
2. The active services at each runlevel are defined in the respective directory as links to the service startup scripts. For example, /etc/rc3.d contains the links for runlevel **3**.
    1. K15httpd → /etc/rc.d/init.d/httpd

In the constitution of the names of the links, the letters S and K indicate that the services are started or ended, respectively. The number that appears next indicates the order in which each of the services is started/deactivated.

eXtended InterNET Daemon

1. The xinetd mode settings can be found in the **/etc/xinetd.conf file**
2. Each of the services controlled by xinetd has an individual configuration file in the **/etc/xinetd.d/ directory**
3. For changes to the settings of these services to be active, it is necessary to restart xinetd, using the commands:

1. `/etc/rc.d/init.d/xinetd reload`
2. `/sbin/service xinetd reload`

Permissões sobre Ficheiros

Files are central and fundamental entities in Unix systems ("**In Linux, everything is a file**"):

1. Commands are executable files;
2. Privileges and permissions are controlled through file access;
3. I/O operations on devices and files are identical;
4. Inter-process communication is also based on entities that behave like files.

Each file or directory has permissions for its owner (UID), **group** (GID) and anyone else (**others**). These permissions can be of 3 types: **read**, **write** and **execute**, being represented by the symbols "**r**", "**w**" and "**x**" respectively. This information can be seen when listing the contents of a directory using the "`ls -l`" command [Figure 3].

To modify these values, use the command "**`chmod`**" with a symbolic or numerical notation.

5. `chmod ug+x file`
6. `chmod 644 file`

To change the owner and group of a file/directory, the "**`chown`**" and "**`chgrp`**" commands are available.

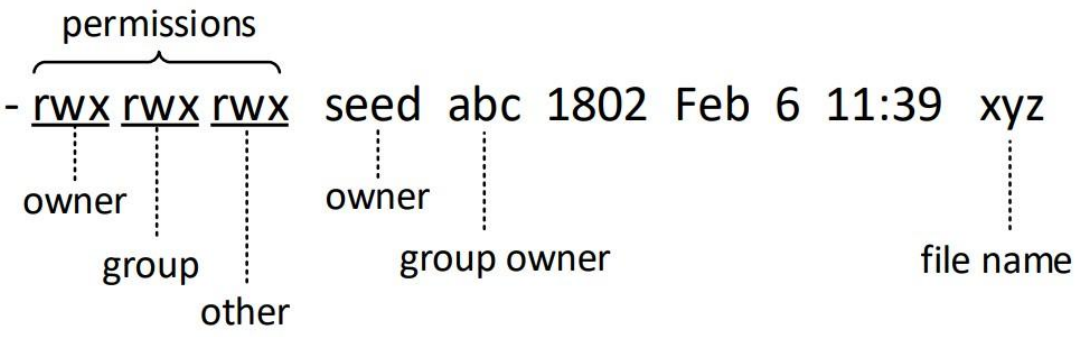


Figure 3 - File Permissions

Special Permissions

At the file level, there are two types of special permissions on Unix systems: Set User ID (**setuid**) and Set Group ID (**setgid**). These permissions allow a file to be executed using the permissions for the owner of the file or their group. A practical example would be a file belonging to root, which in this way could be executed with similar permissions by any other user (e.g. **passwd**).

Regarding directories there are also two different types of permissions: **sticky bit** and Set **Group ID** (**setgid**). When active, the sticky bit makes it so that the directory files can only be deleted by the root, the user who owns the directory or the user to whom the file belongs. When you enable setgid for a given folder, all files created in that folder will inherit the group from that same folder.

Configuration of network interfaces

The TCP/IP configuration of a machine consists of setting the following parameters:

- Your **IP address**
- Its **netmask**
- Outbound address (**gateway**)
- **DNS server** addresses

It is still possible to configure network and broadcast addresses, however these are usually determined by the IP address and subnet mask.

In Unix systems, Ethernet interfaces use names in the format **enp0s3**, the lo interface represents the loopback interface, and is used to communicate with processes running on the machine itself.

The essential commands for this configuration are: `ip`

– used to configure a network interface

1. **ip addr show** or **nmcli device show**
2. **ip addr add 10.10.1.1/24 dev enp0s3**

Note: Try command **nmcli**

The **nmcli** (NetworkManager Command Line Interface) command-line utility is used for controlling NetworkManager and reporting network status. It can be utilized as a replacement for nm-applet or other graphical clients. nmcli is used to create, display, edit, delete, activate, and deactivate network connections, as well as control and display network device status (ex:

nmcli con show,

nmcli dev status,

nmcli con mod enps03 ipv4.addresses 192.168.2.20/24 → *to set up the IP address*

nmcli con mod enps03 ipv4.gateway 192.168.2.1 → *to configure the default gateway*

nmcli con mod enps03 ipv4.dns "8.8.8.8" → *to set up the DNS server*

nmcli con mod enps03 ipv4.method manual → *to change the addressing from DHCP to static*.

)

`IP route` – is used to view and edit the routing table

3. **ip route add 192.168.98.0/24 via 10.10.10.1 dev enp0s3** - Adds a static route for packets bound for network 192.168.98.0 where the egress interface is enp0s3. Note: instead of the output interface, you can specify the address of the next hop.

To restart network interfaces: `systemctl restart NetworkManager.service`

Other relevant commands: `ping, netstat, hostname, traceroute, tcpdump.`

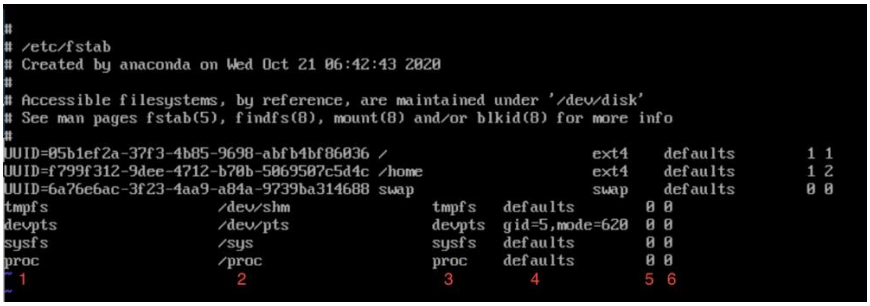
nmtui – graphical interface for network settings

Configuration files:

4. /etc/host.conf – Indicates the order in which DNS resolutions are performed
5. /etc/hosts – file used to map / resolve names to IP addresses (usually queried before the DNS server)
6. /etc/resolv.conf – indicates the IP addresses of the DNS servers

Quotas

Quotas are used to limit inodes and blocks (i.e. disk space) for a particular user or group. To enable quotas, the **/etc/fstab** file must be modified by adding **usrquota** or **grpquota**, as desired (the XFS files of CentOS 9 have some distinct features from ext4fs).



The meaning of each field being:

1. File system label or the Universal Unique Identifier (UUID).
2. Mounting point
3. File system
4. Mounting options
5. Automatic backup option
1. Indicates the sequence of checks that must be performed when the system is restarted
    1. – There is no verification
    2. – Root Directory
    3. – All files

Changes to the /etc/fstab file are only applied after a reboot, or by running the command `mount -o remount [partition]`

Then it is necessary to create the **files uquota, and gquota** (or **aquota.user** and **aquota.group** using the `quotacheck -vagum commands)` and establish the desired quota values through the `edquota` command that will invoke vi to edit the respective file. The quota files are organized as follows:

Filesystem	*blocks*	*soft*	*hard*	*inodes*	*soft*	*hard*
/dev/hda3	*24*	*0*	*0*	*7*	*0*	*0*

The meaning of each field being:

1. Blocks: Amount of space, in 1KB blocks currently occupied by .
2. Inodes: Number of files that the user is currently using.
3. Soft Limit: Maximum number of blocks/inodes that the user can have. A value of 0 indicates that there is no limit. If "grace periods" are defined, the soft limit indicates a limit from which the user will be warned. If the user does not solve the problem (free up space / files) during the "grace period" he will be prevented from using more disk space.
4. Hard Limit: The maximum amount of blocks/inodes that a user can have if grace periods are defined. The "hard limit" can never be exceeded.

Outros comandos relevantes: `quota`, `xfs_quota`, `quotacheck`, `repquota`, `quotaon` e `quotaoff`.

Cron – Task scheduling

Cron is a service (daemon) in Unix that allows you to schedule tasks to run (in the background) at regular time intervals. These scheduled tasks are commonly referred to as "cron jobs". The **crontab** (CRON TABle) is the file that contains the scheduled tasks. The system maintains a crontab file for each user.

```
*     *     *   *    *        command to be executed
-     -     -   -    -
|     |     |   |    |
|     |     |   |    +----- day of week (0 - 6) (Sunday=0)
|     |     |   +------- month (1 - 12)
|     |     +--------- day of        month (1 - 31)
|     +----------- hour (0 - 23)
+------------- min (0 - 59)
```

Restrictions on using Cron

Restrictions can be imposed on the use of task scheduling, authorizing or denying service to a particular user. For this purpose, the files **/etc/cron.allow** and **/etc/cron.deny are used**.

1. If the cron.allow file exists, only the users listed there can use cron (and the cron.deny file is ignored).
2. If only the cron.deny file exists, all users can use cron except those listed in the file.
3. Regardless of the content of the two files, root can always use cron.

The "`crontab`" command allows you to edit, list, and remove existing rules. Each

row in the crontab file represents a task and follows the following format:

minute hour day month dayofweek command

In this list, the "*" can be used to represent all possible values. Intervals can also be used, separated with "-" or lists of values, using ",".

To restart the machine every Sunday at 3:30 a.m. we can use the rule:

*"30 3**0 reboot"*

The rules for a given user are saved in the "**/var/spool/cron/username**" file.

Software management in CentOS

Package management in CentOS is usually done using yum (Yellowdog Updated, Modified), or dnf, command-line applications for RPM package management. These can be used to automate software updates using services such as yum-updated, yumupdateonboot and yum-cron.

Yum/dnf Configuration

The yum configuration is saved in the /etc/yum.conf file and the dnf configuration in /etc/dnf/dnf.conf. Although it is possible to add new repositories to this file, it is now common for them to be organized into separate files in the /etc/yum.repos.d/ folder, for example in /etc/yum.repos.d/CentOS-Base.repo.

Using YUM or DNF

The yum or dnf allows you to manage (install, update, remove, search, etc.) software in an extremely easy and secure way. Some of the most common operations are performed using the following commands:

1. `yum list updates` – lists all existing updates for installed software
2. `yum update` – installs all updates
3. `yum update openssh` – updates a certain package (in this case **openssh**)
4. `yum list installed` – lista todas as packages instaladas
5. `yum list installed openssh` – indica se a package openssh está instalada
6. `yum list` – lists all packages available in the repositories
7. `yum list *ssh*` – indicates all packages whose name has "ssh"
8. `yum install wget net-tools` – instala as packages wget e net-tools
9. `yum remove httpd` – remove a package httpd
10. `yum search net-tools` – search for net-tools in the name and description of packages
11. `yum grouplist` – lists the available package groups
12. `yum groupinstall "FTP Server"` – installs all packages from the "FTP Server" group
13. `yum groupinfo "Java Platform"` – indicates packages that make up the "Java Platform" group
14. `yum list extras` – indicates all packages that have not been installed through the repositories
15. `yum whatprovides /etc/passwd` – indicates which package contains a particular file/value.
16. `yum help` – indicates what a particular command does, in this case what the "help" command does.

In addition to the yum command, it is possible to install packages that are not in the repositories. To do this, you need to download the package-name.rpm file and use the rpm `–i package command`.

Laboratory work

In the report, **keep the numbering and sub-numbering** of the jobs/configuration to be performed. **Present evidence/testing of the configurations performed on how they are operational (or not).**

1. Installing CentOS in the Virtual Machine

Install CentOS 9.x minimal, using the XFS filesytem, configuring the partitions:

/boot (1GiB)

swap (1GiB)

/usr (3GiB)

/log (2GiB)

/var (2GiB)

/home (2GiB, encrypted)

/tmp (2GiB)

/ (rest of space)

And a partition for Apache with the following options: noexec, nodev, and nosuid. Explain why these options are recommended for Apache.

2. Configuration of network interfaces

These instructions are essential throughout the semester. In each class it will be necessary to configure the TCP/IP settings of the VM so that there is connectivity in the lab and to the outside. Use **nmcli** to configure network interfaces. Outside the lab, you will of course have to change the IP addresses. For this to work, it is necessary to configure the network interfaces in bridge mode in VirtualBox.

1. To name the machine vm_numaluno1_numaluno2 (e.g. vm_24001_24002), use the hostnamectl set-hostname command...
2. Run the hostnamectl status command
3. Assign an IP address to the VM considering the network address where it's connected and its group. In the network lab: 192.168.1.0/24.
4. Configure the address 192.168.1.1 as the outbound gateway.
5. Check for connectivity within the internal network and to the outside.
6. Add DNS servers using the IPT server IP and/or others of your choice (Google:

8.8.8.8; 8.8.4.4; OpenDNS: 208.67.222.222, 208.67.220.220)

7. Redo the same configuration, this time so that it persists after a reboot (edit /etc/NetworkManager/system-connections/enp0s3.nmconnection file (if it is a virtualbox vm).
   the. Try using dhclient to get the IP address, default gateway address, and DNS automatically via DHCP.

8. Run the commands
   date; hostnamectl; echo ----------
   rpm -qi basesystem
   and present the results.

3-Users and Groups

To better understand what is happening, check the changes in the user and group files for each paragraph. For example, doing "tail /etc/passwd".

1. Create a user group called CYBERSEC2025.
2. Create a user with their username using the useradd command.
3. Create another user for your groupmate with a UID = 666 and with information about the name (in the comment field) also using the adduser (or useradd) command. If necessary, consult the help (adduser -h; explain the difference between useradd and adduser).
4. Add the users you created to the CYBERSEC2025 group (as an additional group, do not change each user's primary group!).
5. Change the new user's password and check if you can log in.

   As you may have noticed, when creating a user with the adduser command:

   a) an entry is created in the user's file.
   b) a group with the same name is created and becomes the primary group of the created user.
   c) A personal folder is created in the /home directory with some configuration files.

6. Now manually create a new user "leiserver", with the name "SI Central Server 2024". You should use the commands vipw, vigr, etc. Don't forget to:
   a) Create a group with the same name (same as the username) to serve as the primary group.
   b) Create *home folder* and assign permissions and *ownership* of it to the created user.
   c) Set password "leiserver2025" and try to log in. When logging in with manually created users, the command line will look "different." This is because the useradd command copies

the 3 (hidden) Shell configuration files in /etc/skel/ to the /home/<username> folder. They should do the same in this case.

7. Using the chage command, change your colleague's account so that:
   a) Account expires within 1 year
   b) Password expires within 90 days
   c) The user is notified 7 days before the password expires.
8. Manually create (using **vigr**) a new group called **TMR2025LEI**.
9. Add your colleague's user to the group by using the usermod command. For solidarity, also add your *user* to this group, this time editing the necessary files. Attention: the primary groups should not be changed.
10. Replace the shell of your colleague's account with a script that sends an informative message ("Bachelor of Computer Engineering – IPT 2025/2026").
11. Use the id, groups, and lid commands to verify that the preceding paragraphs have been performed correctly.
12. User passwords must comply with rules that ensure levels of complexity that make them robust to attacks.
13. Build a script that checks for accounts without a defined password.
14. Build a script that verifies that there is no account, other than root, that has a UID equal to zero. Explain why this check is important.
15. Disable login through the root account. Explain how after logging in users can perform operations with root permissions.
16. Old passwords cannot be reused.
17. Lock the account after 5 failed login attempts per day. Test the set up.
18. After setting up the password policy, use the John the ripper tool to identify weak passwords.

4- Permissões de ficheiros

1. Create a text file named "wall" in the /home folder.
2. Change the file's permissions so that all users can read the file.
3. Change the owner of the file to its user (**chown**).
4. Altere as permissões do ficheiro para que todos os utilizadores do grupo “TMRLEI24” possam escrever no ficheiro.
5. Construa um script que identifique ficheiros:
   a. sem owner e que os remova caso existam.
   b. Que identifique ficheiros que podem ser modificados por todos (aka world-writable).

5. Task scheduling

1. Create a script (executable) in the /home folder that adds the text "Cybersecurity (at)LEI.ipt!" to the end of the "wall" file.
2. Schedule a task for each of the two TMR2025LEI users to run the script in the previous paragraph every 10 minutes (your user) and at half an hour (your colleague).
3. **Challenge –** create a script to back up the wall file and schedule it to run:
   a. Create a compressed file (tar) with the name backup2024_dd-mm-aaaa.tar, in the
   b. /root/backups
      i. Get the date: date +”%d_%m_%Y”
      ii. Compress file: tar -cvfz (...)
   c. Schedule the task in the root user's crontab so that it runs at 11 p.m. on even-numbered weekdays. (0 = Sunday)

4. Runlevels

1. Verify that the crond service is active at runlevel 3.
2. Configure machine startup at that runlevel (if necessary).
3. Configure sshd service to accept **only 6** concurrent connections. The /etc/ssh/sshd_config file only exists if the sshd service is installed (use yum if necessary).

5. Quotas

1. Enable the use of quotas, **using the XFS file,** for users and groups on the local machine (see slides) on the /home partition.
2. Set quotas in the /home folder for the two users you created (yours and your colleague's, use **xfs_quota** tools):
   a) With a value of 25MB (soft) and 40MB (hard)
   b) Set the grace period to be equal to 5 days for inodes and 10 days for blocks.
3. You can check if they are working by using the quota and repquota command and creating files (e.g. creating a 50MB file for a given user using "dd if=/dev/zero of=~/testing_quotas bs=1MB count=50")
   a) If you receive errors when using any command regarding quotas, it means that the package quota is not yet installed either.

6. Package update and management

1. Use the CentOS package manager to update the installed software (yum –help or dnf or rpm). Note: This task can take time because it depends on the amount of data to be transferred and the throughput of the link.
2. Look for information about the wget package (yum info) to determine its function.
   a) Download without installing the wget package.
   b) Install the wget package.

9. Run the commands:

   date; hostnamectl; echo ----------
   rpm -qi basesystem

   and present the results.