TOPIC 3
OPERATING SYSTEM SECURITY

Different areas to secure any operating system

JOSÉ MANUEL REDONDO LÓPEZ *"S-81 ISAAC PERAL"* PROJECT V4.0

Source: Bing Chat AI

Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo

Logo: @creative_vanesa (*Instagram*)

General topic goals

- Understand how to manually harden an operating system
  - What are the basic operating system areas to cover to do that
- Learn what areas require control and special handling
- Know about specialized protection software for certain areas of the operating systems
- Evaluate the security of a given operating system
- And relate with other courses
  - Complementing the contents of the ASR course from a security point of view
  - Link with the end of the SO course to see “what’s next” regarding OS security

INDEX

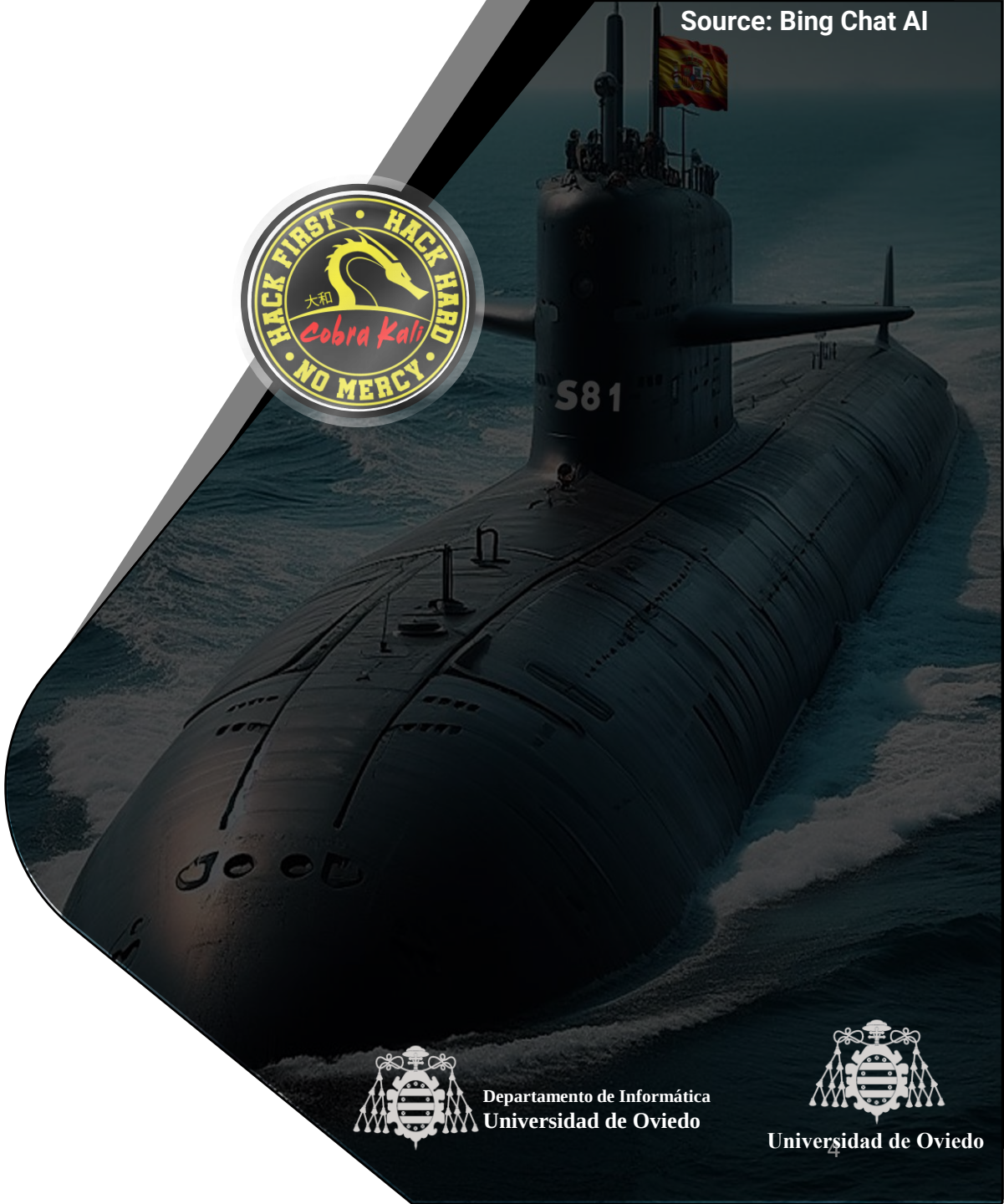
< Go to Index

The "Security chain"

Security as a combined effort

*Icons by Bing Chat AI

Source: Bing Chat AI



What are you going to find in this block?

- **In this block you will learn**
  - The elements involved in a typical "**safety chain**“
  - Where an attack can be initiated **using a system's software as a vehicle**
  - The problem of **insecure device** or feature designs
  - The problem of **over-privilege** of a user or program
  - That users are often the **weakest link** in the chain

Security as a "chain"

- **The security level of a system is not obtained by adding all security measures implemented in an organization…**
- **…but checking the security level of the "weakest" element** of this organization
- **These elements are**
  - Applications in use
  - Software design
  - Operating systems
  - Users (and they are very frequently the weakest link!)
  - The context of the **physical site** where the devices are (**Seminar 2**)

SECURITY DEFECTS IN SOFTWARE

- **Malware takes advantage of defects (security bugs or vulnerabilities) in software**
  - Targeting bugs or loopholes **to exploit them**
  - **Outdated** applications (e.g., Internet Explorer, programs only compatible with Windows XP…)
  - Applications with **known vulnerabilities**
    - E. g.: vulnerable versions of browser plugins such as Acrobat Reader, Java SE…
- **Sometimes installing updated versions does not automatically uninstall old ones**
  - The security problem still exists!
- **We must monitor the security advisories from application vendors**
  - They announce security-related updates that fix problems found in CVE databases (**Topic 1**)
- **Maintaining an adequate control ("having an inventory") of installed applications in any OS and their patches is vital**
  - **ONLY install the applications that are strictly needed** in any OS
  - **Idem with dependencies / frameworks and libraries** used by our applications (SCA tools, **Topic 6**)

OPERATING SYSTEMS

- **Homogeneity in OS usage can also be a vulnerability**
  - When all computers in a network run the same OS, if one is exploited, all of them could fall as well
  - E.g.: EternalBlue + WannaCry attack in 2017
- **But introducing diversity for robustness also has unwanted effects**
  - Produces maintenance, management and authentication complexity, increasing costs…
- **What we could do is to ensure all devices do not depend on the same authentication service**
  - E. g.: Active Directory domain
  - Their diversification could avoid a total shutdown if the service is compromised or taken down
  - Allowing some devices to help recovering disabled nodes
  - Give recovery options for malware or ransomware attacks
- **OS installation must be done with the most secure settings** (Topic 4)
  - ASR covers both Linux-based and Windows OS basic installations

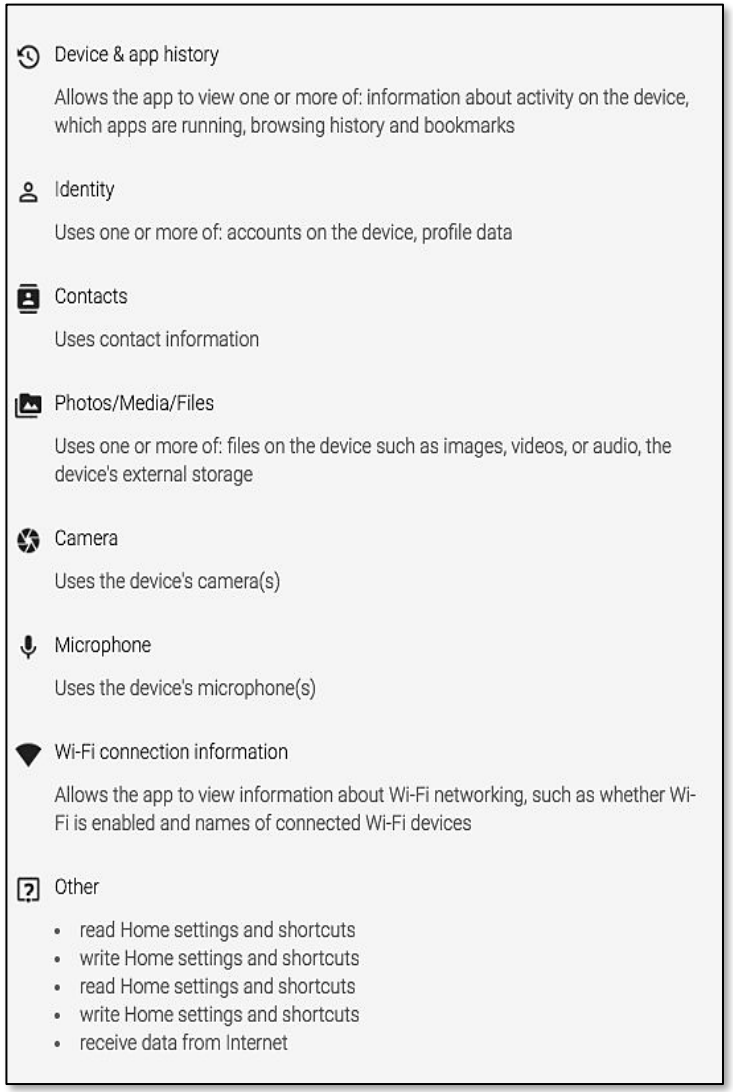
INSECURE FUNCTIONALITY DESIGN OR USER ERRORS

- **Some common legitimate operations were designed with no security in mind**
  - Attackers frequently trick users to perform these operations so these are exploited. E. g.:
    - Microsoft vishing scam and RATs (Remote Administration Tools)
    - The Windows embedded Remote Assistance support ("Microsoft call" scam)
- **A popular example is the external device boot / run problem**
  - Computers boot from an HDD/SSD, but they can be configured to boot from USB devices
    - This also happens when **running software automatically** upon external hardware insertion (USB)
  - These mechanisms were designed as convenient features, but without thinking about security
    - So, malware distributors trick the user into **booting or running software from an infected device**
    - **Any device that plugs into a USB port** can be used to spread malware
      - Even lights, fans, speakers, toys… be VERY careful!
    - Devices can be infected during manufacturing or supply (**supply chain attack**)
  - This can be avoided by setting up OS to boot from the internal hard drive only
    - And **block autorun features from devices**
- **Vulnerable email clients might autorun malicious JavaScript attachments**

Over-privileged users & Over-privileged code

- **Privilege** refers to how much a user or program can modify a system
- In poorly designed computer systems, users and programs **can be assigned more privileges** than needed
  - **Overprivileged users**: Some systems allow all users to modify the internal structures of the OS
    - No or poor distinction between an `Administrator` / `root` and a regular user
  - **Overprivileged code**: systems allow programs to access and use all privileges and permissions of the user that runs them
    - Malware running as over-privileged code can use them to attack or damage the system
    - Almost all currently popular OS and many applications give too many privileges to parts of their code
      - This is due to errors or inadequate software security practices (Topic 6)

Source: Android App permissions



Device & app history

Allows the app to view one or more of: information about activity on the device, which apps are running, browsing history and bookmarks

Identity

Uses one or more of: accounts on the device, profile data

Contacts

Uses contact information

Photos/Media/Files

Uses one or more of: files on the device such as images, videos, or audio, the device's external storage

Camera

Uses the device's camera(s)

Microphone

Uses the device's microphone(s)

Wi-Fi connection information

Allows the app to view information about Wi-Fi networking, such as whether Wi-Fi is enabled and names of connected Wi-Fi devices

Other

- read Home settings and shortcuts
- write Home settings and shortcuts
- read Home settings and shortcuts
- write Home settings and shortcuts
- receive data from Internet

USERS AS POTENTIAL RISK SOURCES

- Users can be **the greatest threat** to the security of an organization, either consciously or unconsciously
  - Defects in information or software operation, handling, and management
  - Open malicious email attachments
  - Place important documents in non properly secured / publicly accessible places
  - Don't block sessions when leaving the computer unattended
  - Loss of important information
  - …
- Some of these attacks are due negligence or lack of training, others may be "**inside jobs**" (deliberate attacks)
  - **Studies determine that half of the attacks to an organization come from actual employees!**

Users as liabilities: Countermeasures

- **Training**
  - Every user must receive **training** about the company resources and how they can or cannot be used
  - **A set of management rules must be developed to create a security policy (Topic 4)**
  - **With also consequences** to apply over users that do not comply with the security policy rules
- **OS Hardening (this topic)**
  - Properly configuring an OS to stop user dangerous activities (either intentional or not)
- **Create an Information Security Management System (ISMS) (Topic 4)**
  - Describing a security policy and how the assets will be hardened against attacks
  - Detailing what to do in case of attacks, and what steps must be performed to recover from it
  - Performing periodic audits to check the compliance with the established rules
  - With full consensus of all organization employees

Think you've understood it all? Evaluate yourself!

- **You can do the following**
  - *Understand that security doesn't depend on just one thing, but on the applications running (including the OS), how they are designed, and the actions of users*
  - *Understand that keeping an up-to-date inventory of all used applications is imperative for proper security*
    - *Including minimizing the installed applications in any scenario*
  - *Be aware of the advantages and disadvantages of having the same OS and/or authentication service across an entire infrastructure*
  - *Understand that some features have been designed at the outset with security in mind, and that we are now suffering the consequences of that*
  - *Be clear that, as a developer, it is very tempting to give "permissions for everything" to avoid execution problems, but that by doing so we are opening the door to serious security issues*
  - *Always think of users as potential threats, either carelessly (mitigable with training) or deliberately (mitigable with an appropriate security policy)*

< Go to Index

Source: Bing Chat AI

Boot >

Logging >

General OS hardening

First steps in securing our OS

*Icons by Bing Chat AI

Departamento de Informática
Universidad de Oviedo

What are you going to find in this block?

- **In this block you will learn**
  - That the **boot process** requires special protection because it can give someone full access to the system
  - That in a "serious" installation, there are usually **partitions assigned** to different parts of the file system
  - The **importance of file locations** and security restrictions at the kernel level
  - The enormous importance of a **system's logs** in terms of security and their correct configuration, interpretation and study for adequate surveillance
  - That to implement surveillance properly you need **HIDS** (Host-Based Intrusion Detection Systems)

Security on the boot process

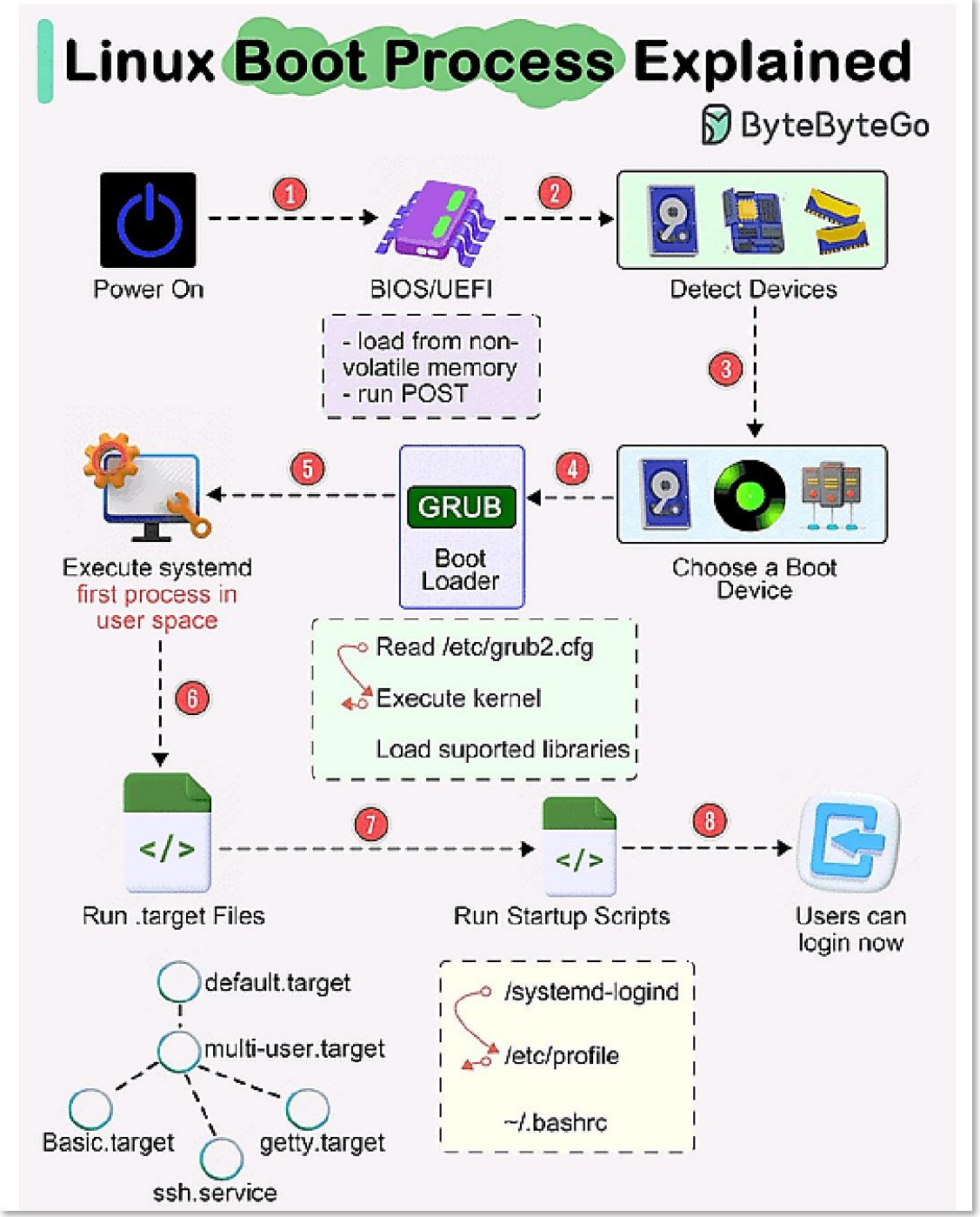
We don’t (want to) start the fire! ☺

*Icons by Bing Chat AI

The OS Boot process

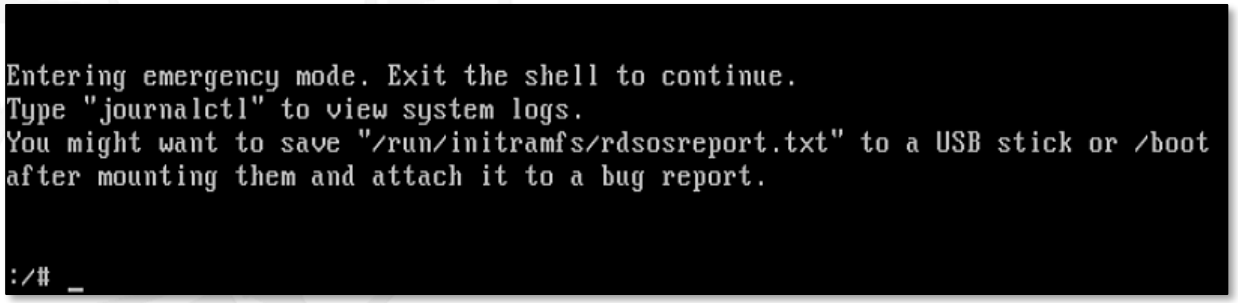
Seguridad de Sistemas Informáticos

- It's a multi-stage process where everything that is required to work is loaded
- If it is interrupted or intervened, there are ways to gain access to the machine with elevated privileges
  - Maintenance or single-user modes
- Their protection is therefore of paramount importance
  - Or we may be compromised before we see the login screen...



Establish a boot / bootloader password

- The boot process of an OS (BIOS/UEFI) is described in detail in ASR
  - They also cover bootloader details
- Boot passwords prevent unauthorized users with physical access to the system obtaining `root` privileges
  - Through the mentioned **single user/maintenance mode**
- These modes usually allows dangerous operations like
  - Changing settings at boot time
  - Access to the bootloader console
  - Reset the root password
  - Access to non-secure OS (Live images)
  - …



- We also don't want users to enable these modes back
  - Proper permissions ensure that only privileged users can modify important boot parameters

Disk partitioning

- **Critical file systems should be separated in different partitions**
  - Partitions and filesystems are explained in ASR
- **Several top-level system directories should be placed on their own physical partition or logical volume**
  - Ensures data separation and protection
  - Prevents partitions overflow
  - Isolates a possible data corruption
  - Provides logical separation of data
  - Decreases the duration of `fsck`
  - Allows using different file systems, suited to different needs
  - Allows using specific SELinux contexts
- **E. g.: these Linux filesystems should be mounted on separate partitions**
  - `/home` (each user's personal data)
  - `/var` (files written while the OS runs)
  - `/var/tmp`
  - `/tmp`
    - Temporal "throw-away" files
  - `/var/log`
  - `/var/log/audit`
    - **Log files**, what happens while the OS runs
  - If a critical partition size is incorrectly calculated, the OS may be inoperable!
- **Windows has an isolated recovery partition**
  - User data should be in another one

RESTRICT MOUNT OPTIONS

- **Some OS allow to flag partitions with security properties. E. g.: in Linux:**
  - `nodev`: specifies that the filesystem cannot contain special devices
    - A world-accessible filesystem should not allow to create character devices or access device hardware
  - `nosuid` - specifies that the filesystem cannot contain files with the `setuid` permission enabled
    - Preventing **SETUID executables** on a world-writable filesystem is convenient
    - They may be used for privilege escalation (explained **topic 7**)

```
-rwsr-xr-x  1 redondo redondo 138208 sep 14 10:50 ls
```

  - `noexec`: useful for partitions without binaries (like `/var`)
    - Or containing binaries that should not be executed…
- **Example**
  - The boot directory of a Linux system contains important kernel-related files
  - It **should be restricted to read-only permissions** and flagged with the three previous properties

```
 GNU nano 2.2.4            Archivo: /etc/fstab

# /etc/fstab: static file system information.
#
# Use 'blkid -o value -s UUID' to print the universally unique identifier
# for a device; this may be used with UUID= as a more robust way to name
# devices that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    nodev,noexec,nosuid 0       0
# / was on /dev/sda1 during installation
/dev/sda1       /               ext4    errors=remount-ro 0       1
# swap was on /dev/sda2 during installation
/dev/sda2       none            swap    sw              0       0
```

TEMPORAL FILE LOCATIONS

- **Temporal file locations are usually world-writable by default**
  - Such as `/tmp` and `/var/tmp` in **Linux**
  - Several daemons / applications use these directories to temporarily store data, log information…
    - Or to share information between their sub-components
  - However, due to its shared nature, several attacks are possible
    - Leaks of confidential data via secrets contained in the temporal files
    - Race-condition attacks on the integrity of processes and data
    - Denial-of-Service (DoS) attacks due to race conditions (TPP), etc.
    - …

```
redondo@redondo-VirtualBox:~/folder$ ls -la /tmp
total 116
drwxrwxrwt 23 root    root    12288 sep 14 10:05 .
drwxr-xr-x 20 root    root     4096 ago 31  2021 ..
drwxrwxrwt  2 root    root     4096 sep 14 09:55 .font-unix
drwxrwxrwt  2 root    root     4096 sep 14 09:56 .ICE-unix
drwx------  3 root    root     4096 sep 14 09:56 snap.firefox
drwx------  3 root    root     4096 sep 14 09:55 snap.snapd-deskt
drwx------  3 root    root     4096 sep 14 09:56 snap.snap-store
drwx------  3 root    root     4096 sep 14 09:55 systemd-private-
fbfe87ecb6f811d90-colord.service-eiX0lC
drwx------  3 root    root     4096 sep 14 09:56 systemd-private-
fbfe87ecb6f811d90-fwupd.service-IHXwhn
```

- **Malicious applications usually write to temporal directories and then attempt to run whatever was written**
  - Therefore, preventing execution of software from temporal directories is advisable
  - **Linux**: mount `/tmp` on a separate partition with the options `nodev`, `nosuid`, and `noexec` enabled
    - This will deny binary execution from `/tmp`, disable any binary to be `suid root`, and disable creation of any block device

OTHER CONSIDERATIONS

- **Swap space**

```
-rw-------   1 root root 2147483648 ago 31  2021 swapfile
```

  - When an OS runs out of RAM, it swaps parts of it to a swap file or partition (SO)
  - This contains partial contents of the process RAM, which **can contain critical information** such as passwords, private tokens or data with restricted access
  - Therefore, it is very advisable to **encrypt the swap file** or partition to avoid leaking critical information
- **Kernel modules**
  - Security vulnerabilities in kernel code are not frequently discovered
    - But their consequences can be dramatic!
  - For this reason, unused kernel modules should not be loaded (E. g.: USB, Bluetooth…)
    - To protect the system against the exploitation of any of their implementation flaws
  - And this again could be done installing **only** the software / drivers you **really** need
- **The network stack (which manages connections) must be configured correctly**
  - Including **disabling support for any unused protocols**
- **General disk management, quotas, etc. are explained in ASR**

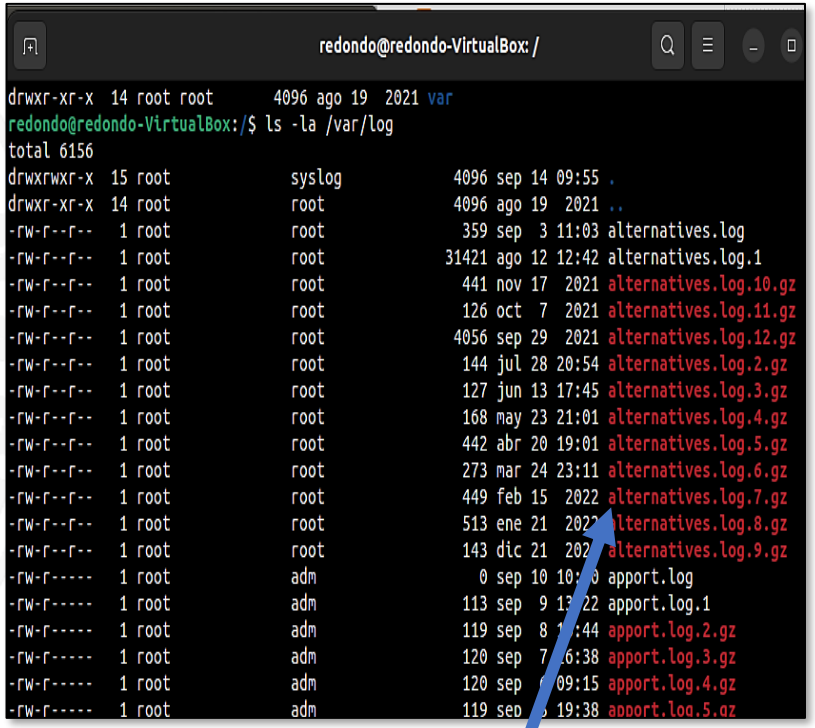
Logging and Auditing

Registering what happened

*Icons by Bing Chat AI

IMPLEMENTING A PROPER AUDIT CONFIGURATION

- Recording system activities and security-relevant events is **VITAL**
- Implementing a proper audit configuration means
  - **General system log management:** The audit daemon/service must never be disabled, and have immutable configuration
    - Malware tend to disable / misconfigure it
    - It should collect information on kernel module loading / unloading
    - And about processes starting prior to the audit daemon/services
  - **Log size management:** Log files have a maximum size
    - **And must be retained over a required period**
    - If log files exhaust disk space, actions / notifications are needed
      - Send alert emails
      - Log rotations (restarting the log after a size is reached, storing the previous data in another file)
      - Or sending logs to external storage systems periodically



Rotated logs (renamed and compressed if needed in the future)

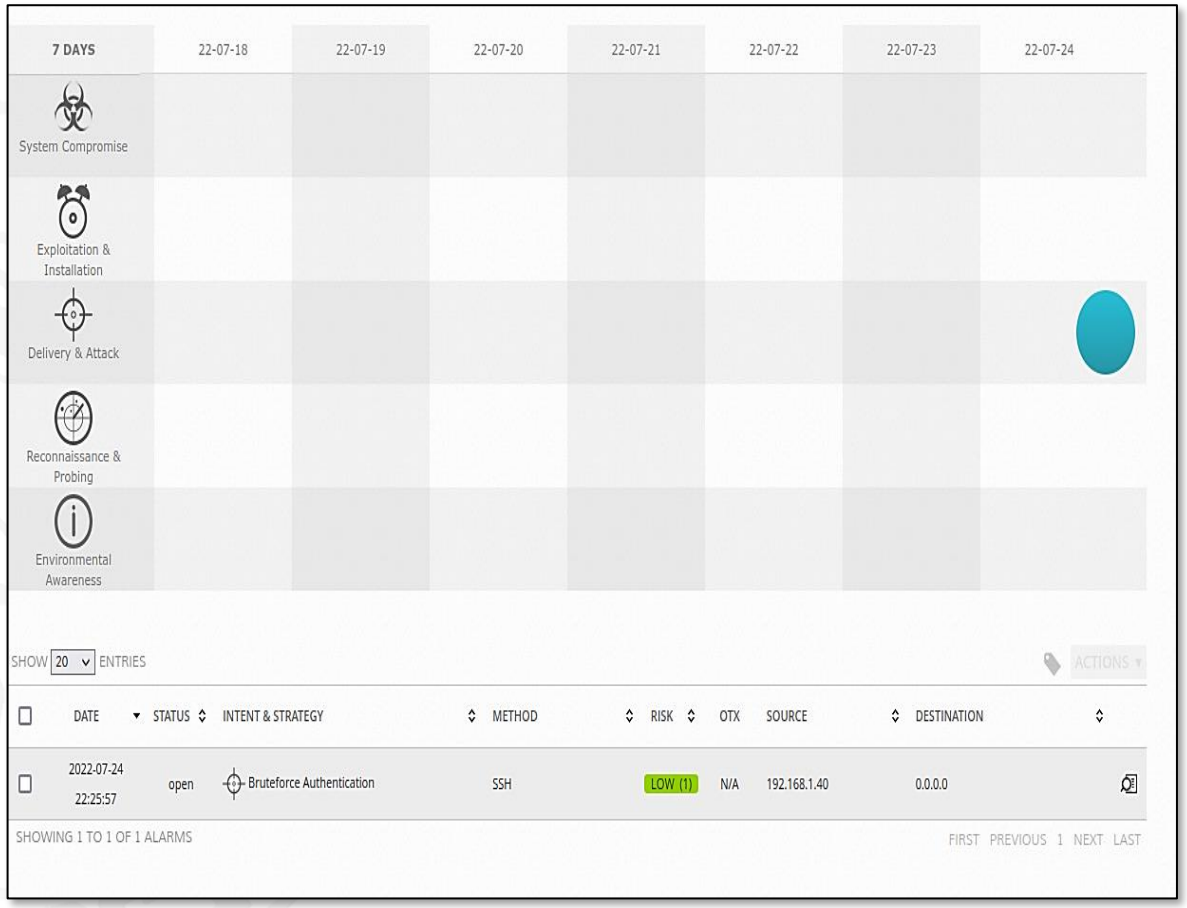
➢ Exercise **L5B4_AUDIT**

Implementing a proper audit configuration

- **User behavior must also be audited to create a trail of every problematic action**
  - **Standard users: logon and logout** events, attempts to **alter time**, change **permissions**, **delete** files, **unauthorized file access** attempts, and **export files** to external media
    - All commands able to do these operations must be audited
    - Attempting to change these log files are indications of **removing intrusion evidences** (**ongoing attack**)
  - **Administrators:** every administration action or command usage requiring **high privileges**, modifications of the **network environment**, the **session initiation** and **user/group information**, and the system's **mandatory access controls (MAC)**
    - Modifying sensitive parts of the OS by an administrator is part of its duties
    - But may also indicate a system compromise if unwanted changes are detected!
  - In Linux, this also includes using process accounting (`pacct`)
- **Log locations per OS will be explained in ASR**
  - The IFA course teaches you how to analyze and interpret log contents
  - ML/IA techniques (SI course) may be also used to analyze log contents and identify threats

IMPLEMENTING A PROPER AUDIT CONFIGURATION

- **Log files of a system usually contains**
  - Date, time, and user that performed an action
  - Details about each important action
- **There are many log files, belonging to major services installed in the system**
  - https://thehackerway.com/2021/06/21/15-ficheros-de-log-interesantes-en-sistemas-linux/
- **Host-based Intrusion Detection Systems (HIDS) uses them to detect threats**
  - Like OSSEC (see image)
    - https://computingforgeeks.com/how-to-install-ossec-hids-on-ubuntu-18-04-16-04-debian-9/
  - They may have other functionalities
  - Like policy monitoring, file integrity checking, real-time alerting, rootkit detection, active response…



OSSEC reporting to an OSSIM SIEM (https://cybersecurity.att.com/products/ossim). OSSEC is also part of other products, such as Wazuh (saw in labs)

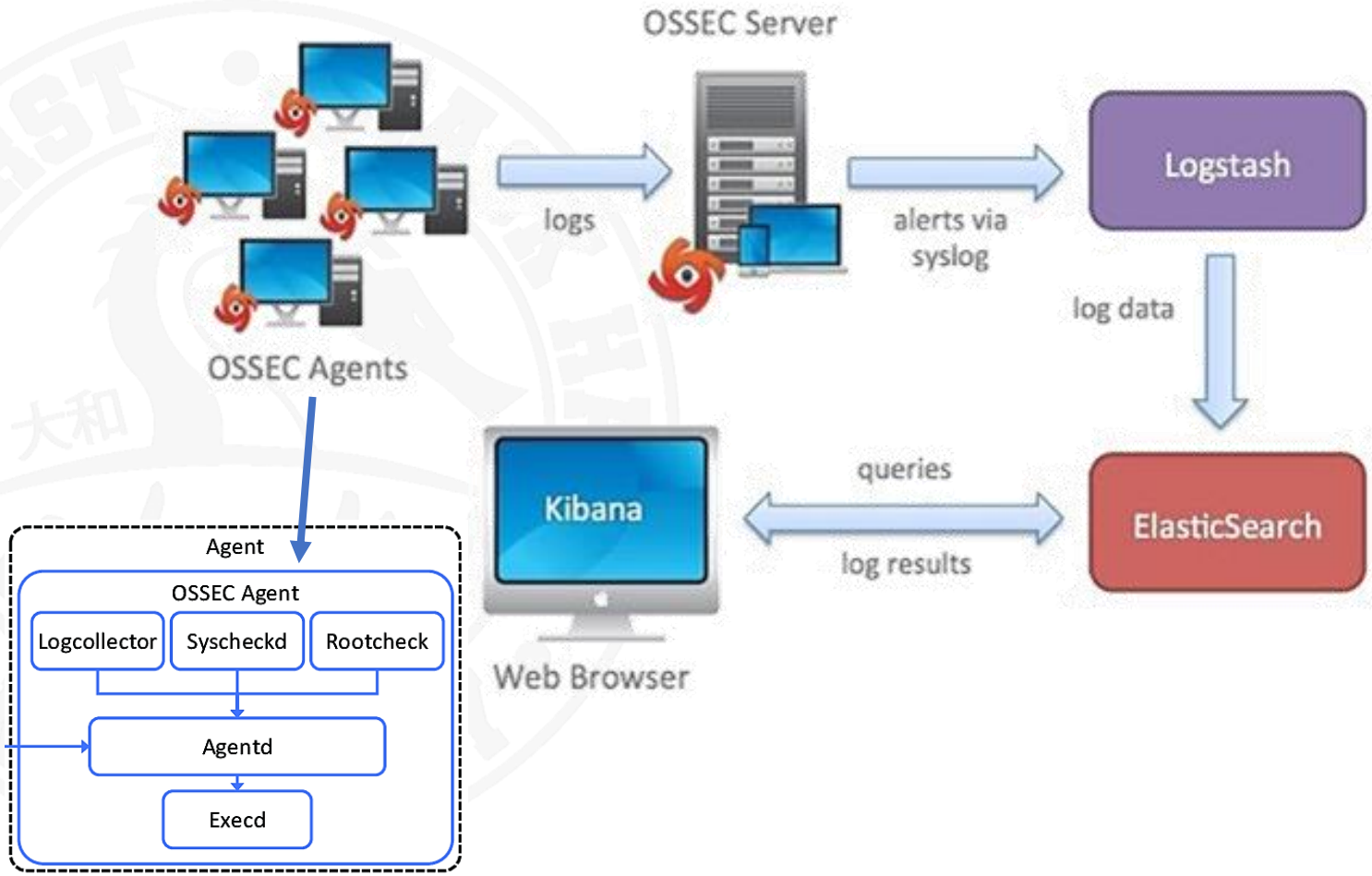
Remote storage of system logs

- **Most computers in highly secure infrastructures do not keep local logs**
  - Sends them to **central log servers** to store and analyze them
  - **Protects log integrity from local attacks**: if an attacker gains `root` access on the local system, they could tamper with or remove its log data
  - Allows to easily have a backup of the logs of all machines in the infrastructure
  - Also, to have a machine with specialized hardware to keep and analyze large text files
- **And, of course, we must protect central log servers at all costs**
  - Their information is vital!
  - Administration LAN, isolated from the rest of the server infrastructure and networks (Topic 5)
  - Properly identified and configured to receive only logs from authorized systems (and don't send logs to fake remote log collection systems)
- **The `rsyslog` service typically implements this in Linux**
- **To know forensic techniques of log analysis, you should enroll the IFA course**

AUDITING COMPUTERS OF AN INFRASTRUCTURE

- **Installing HIDs agents on each machine allows full infrastructure auditing**
  - Based on log contents of each system, unauthorized file changes, suspicious behavior...
- **All this info is aggregated**
  - In a central log server storing logs sent via `rsyslogd` (Logstash)
- **These can be viewed and analyzed globally**
  - Thanks to tolls like Kibana and ElasticSearch (**ELK stack**) (**AS**)

Source: https://www.researchgate.net/figure/Integration-architecture-of-OSSEC-12_fig1_320649753



Source: https://www.researchgate.net/figure/Open-Source-Host-based-Intrusion-Detection-System-OSSEC-components_fig1_335806029

Think you've understood it all? Evaluate yourself!

?

- **You can do the following**
  - *Understand that during the boot process there are options to switch to a maintenance or emergency mode that give you full access to the system*
    - *And that we can only avoid them by putting a password in the right place*
  - *Understand the benefits of splitting the file system into multiple partitions or disks*
    - *And how this division allows you to take a greater advantage of the partition mounting options in **Linux**, for example*
  - *Be aware that temporary files may contain private data*
    - *Or used maliciously to cause a process to no longer provide service*
    - *And that, in that sense, swap files can be another source of private data leaks*
  - *Understand that the best way to opt out of potentially insecure functionality is to not install support for such functionality at the kernel level*
  - ***About the logs***
    - *Understand that everything relevant that happens in the system and with users, erroneous or not, must be reflected in the logs*
    - *Be aware that the volume of information generated is such that without an analysis tool (like a HIDS) we would not be able to see the threats to a system*
    - *And that, due to the tremendous importance of logs, it is very common to store them externally*

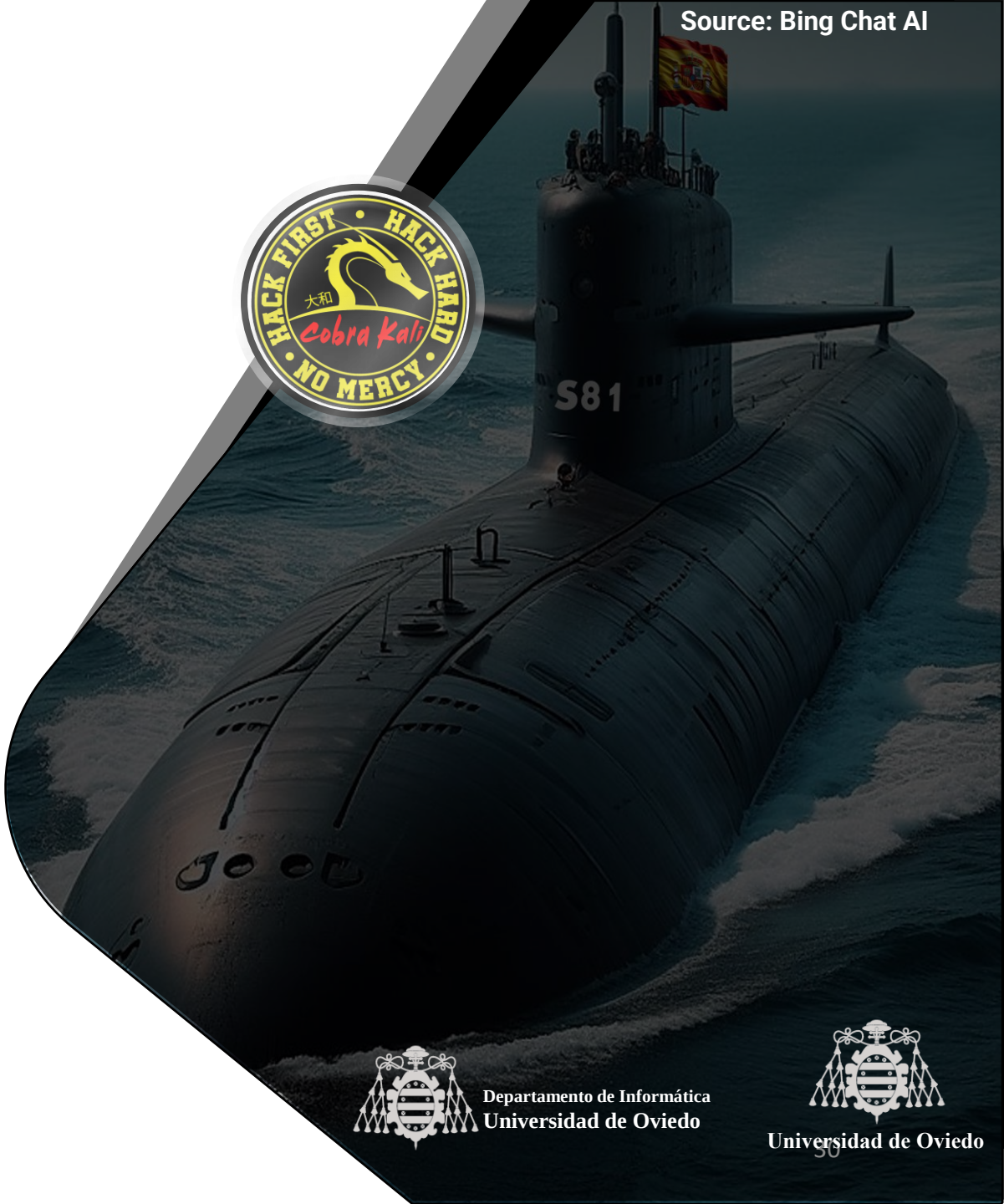
< Go to Index

HARDENING SERVICES / SOFTWARE

Security applied to programs

*Icons by Bing Chat AI

Source: Bing Chat AI



What are you going to find in this block?

- **In this block you will learn**
  - That software should be **updated and always minimized**
  - The most important operations when **managing software**
  - Different access control modes, namely **MAC** and **DAC**
  - The importance of **controlling, monitoring, and scanning** running software

Maintaining and updating software

- **Software maintenance is extremely important to have a secure system**
  - Patch software as soon as new versions become available, in order to prevent attackers from using known vulnerabilities to infiltrate a system
  - Changes to any software components can have significant effects on the overall security of the OS
  - **Companies usually have an update policy adapted to their context**
- **Activating cryptographic package signatures (Topic 2) ensures the software has not been tampered with and that it is provided by a trusted vendor (E. g.: `debsums`)**
- **Update all packages and dependencies regularly, specially when serious vulnerabilities are reported in the software we use**
  - Debian - based systems: `sudo apt update`, `sudo apt full-upgrade`
    - And `sudo snap refresh` in newer Ubuntu versions
  - Red Hat - based systems: `sudo yum upgrade` (see ASR)
- **The best protection against vulnerable software is running less software**
  - How to start and stop OS services is covered in ASR

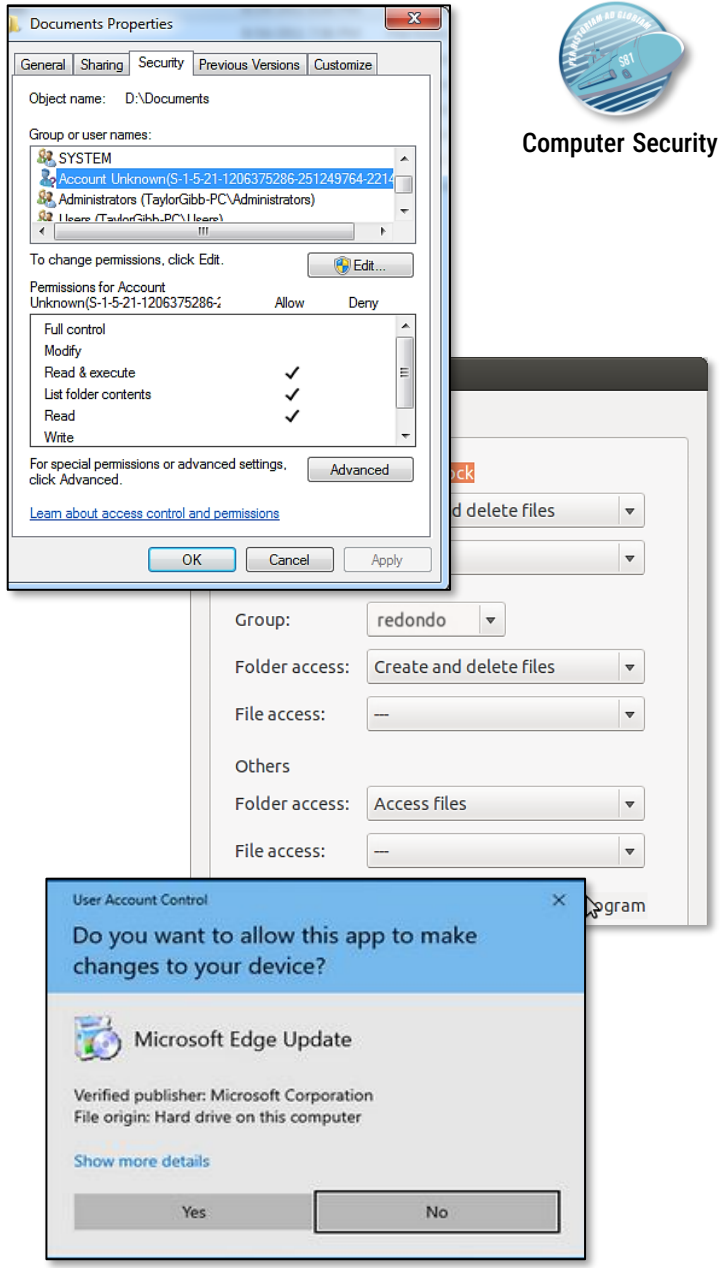
MANAGE IMPORTANT SOFTWARE

- **Enable task scheduling**
  - It is required for maintenance and security-supporting tasks
  - Normally is enabled by default, both in Windows and Linux (`crond` daemon) systems
- **Enable time synchronization services (NTP) (see ASR)**
  - Synchronizing time is essential for authentication services such as Kerberos
  - It is also important for **maintaining accurate logs** and properly auditing security breaches
  - Date/time is automatically managed by default: **Try to not to change them manually**
- **Control the resource consumption of the processes with appropriate tools**
  - E. g.: `sysstat`, `glances` in Linux
- **Disable compilers**
  - Compilers can be used by attackers to compile malware in a server
  - Disabling them also helps to protect from attacks that exploit its vulnerabilities
    - It also follows the rule of **not installing things that are not really needed**
    - An attacker may find a way to run its malicious code anyway

➢ Exercises **L5B4_SYSSTAT**, **L5B4_GLANCES**, **L5B2_COMPILERS**

Types of Access Control

- **DAC (Discretionary Access Control, "typical" permissions)**
  - Restrict access to objects with user identities or the groups they belong to
  - Access permission can be transferred to other entities
  - E. g. Linux files have read, write, and execute permissions
    - With three bits for user, group, and "others"
- **MAC (Mandatory Access Control)**
  - An entity (e. g. the OS) **constrains** the ability of a subject **to access or perform some sort of operation** on an object or target
  - E. g. Windows Mandatory Integrity Control (MIC) controls access to objects and works before evaluating file permissions (DAC)
    - Combines 4 integrity levels (**IL**) (**low**, **medium**, **high**, and **system**) with a mandatory policy to evaluate access
    - Standard users receive **medium**, privileged users receive **high**
    - User-started processes receive the "**low**" IL if its executable file IL is "**low**"
    - System services receive "**system**" integrity level
  - A process **cannot interact with another one that has a higher IL**

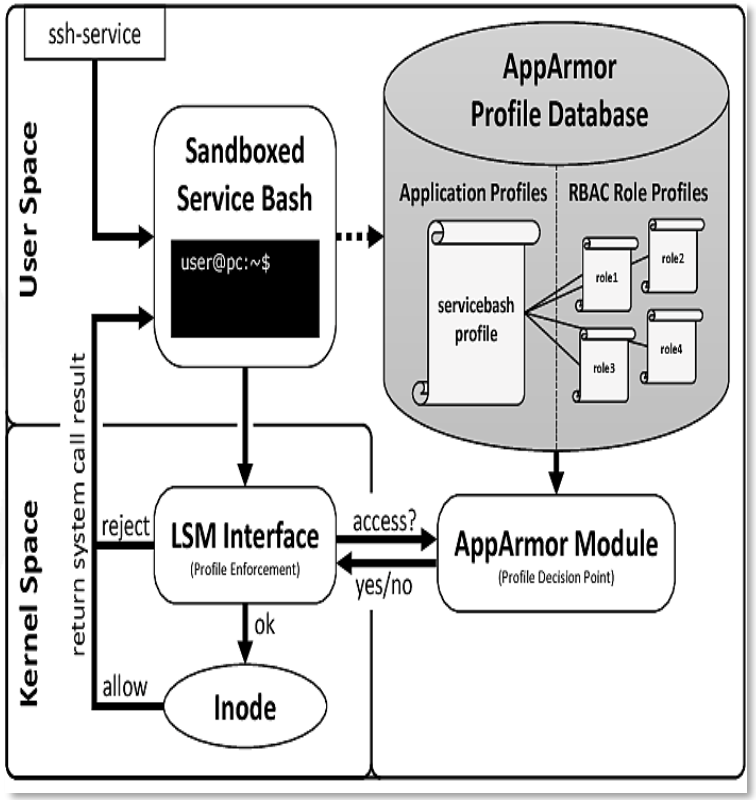


Source: https://helpdesk.athens.edu/hc/en-us/articles/4404584871575-Microsoft-Edge

➢ Exercise **L5B2_APPARMOR**

Mandatory Access Controls (MACs)

- **The Linux equivalent to Windows MIC is implemented as a kernel feature**
  - This guards against misconfigured or compromised programs
  - RedHat systems use **SELinux** and Debian-based use **AppArmor**
    - Both limit what files programs can access to…
    - And what actions they can do!
- **To ensure proper operation, you need these measures**
  - Ensure that it is enabled and "**enforcing**"
    - It has security policies designed to prevent processes from causing damage to the system, or elevating their privileges
  - Ensure all system processes are **covered** by their protection
- **OS permission bits and attributes of files and processes (Windows / Linux) are described in ASR**
  - Also, their association to users



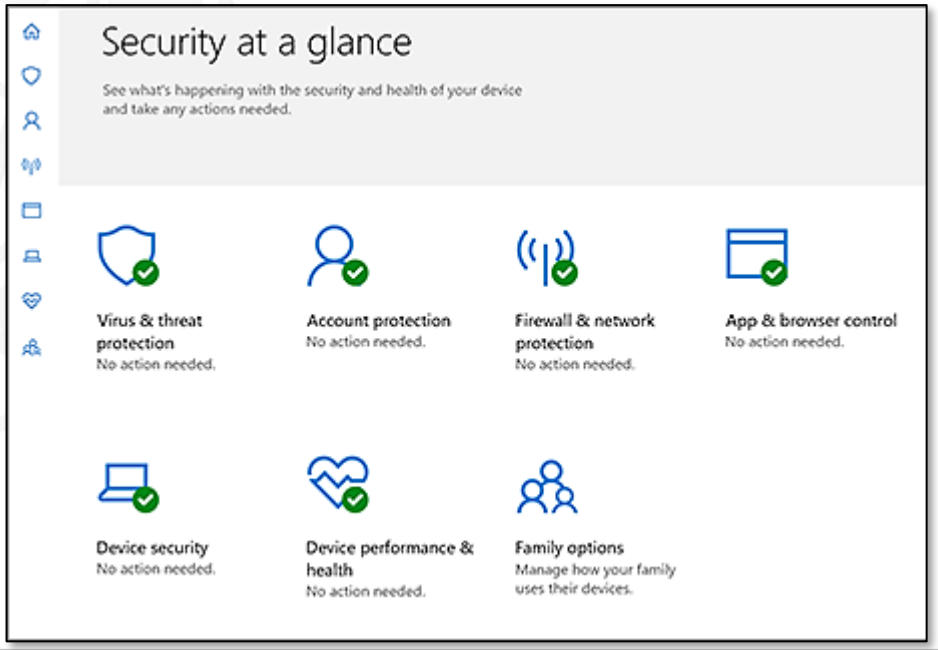
Source:
https://www.semanticscholar.org/paper/Using-RBAC-to-Enforce-the-Principle-of-Least-in-Kern-Anderl/c3438119a1d466f69568b48d6ae3f1664d1e3830

Programs: Malware

Seguridad de Sistemas Informáticos

- Installed programs must be periodically checked for **malware**
- In **Linux**, you can do this with
  - Linux Malware Detect: https://www.globo.tech/learning-center/install-use-lmd-clamav-ubuntu-16/
  - ClamAV: https://kifarunix.com/install-and-use-clamav-on-ubuntu-20-04/
    - ClamAV usually works in background
    - We need to check its log to check for detected infections
- **Windows** systems have **Windows Defender** fulfilling this role
- If in doubt, we can perform a thorough scan with **VirusTotal**
  - https://www.virustotal.com/es/

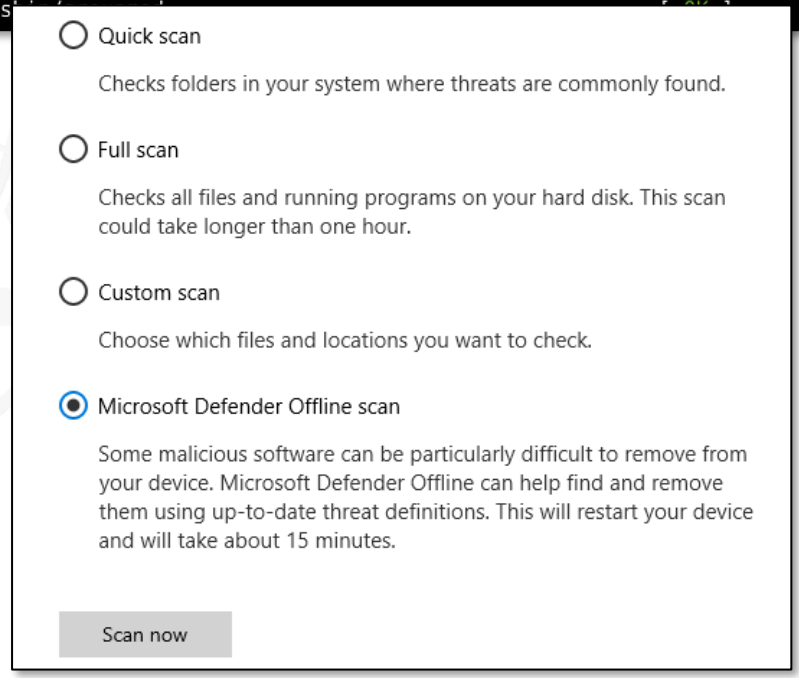
```
redondo@miw:~$ cat /var/log/clamav/freshclam.log
Thu Jul 16 16:33:51 2020 -> -----------------------------------
Thu Jul 16 16:33:51 2020 -> freshclam daemon 0.102.3 (OS: linux-gnu, ARCH: x86_6
4, CPU: x86_64)
Thu Jul 16 16:33:51 2020 -> ClamAV update process started at Thu Jul 16 16:33:51
 2020
Thu Jul 16 16:33:51 2020 -> daily database available for download (remote versio
n: 25875)
Thu Jul 16 16:33:57 2020 -> Testing database: '/var/lib/clamav/tmp.f35da/clamav-
d61d951eebdd284e3c30c2caa8440014.tmp-daily.cvd' ...
Thu Jul 16 16:34:07 2020 -> Database test passed.
Thu Jul 16 16:34:07 2020 -> daily.cvd updated (version: 25874, sigs: 3410927, f-
level: 63, builder: raynman)
Thu Jul 16 16:34:07 2020 -> main database available for download (remote version
: 59)
Thu Jul 16 16:34:12 2020 -> Testing database: '/var/lib/clamav/tmp.f35da/clamav-
38f2c15e309879a91acf2ccca95f72bf.tmp-main.cvd' ...
Thu Jul 16 16:34:17 2020 -> Database test passed.
Thu Jul 16 16:34:17 2020 -> main.cvd updated (version: 59, sigs: 4564902, f-leve
l: 60, builder: sigmgr)
Thu Jul 16 16:34:17 2020 -> bytecode database available for download (remote ver
sion: 331)
Thu Jul 16 16:34:17 2020 -> Testing database: '/var/lib/clamav/tmp.f35da/clamav-
0f7c356ca71f5dee810bc8384b4a4c36.tmp-bytecode.cvd' ...
Thu Jul 16 16:34:18 2020 -> Database test passed.
Thu Jul 16 16:34:18 2020 -> bytecode.cvd updated (version: 331, sigs: 94, f-leve
l: 63, builder: anvilleg)
```



Programs: Rootkits

- **Rootkits require special tools and attention**
  - In Windows, Windows Defender and System Guard protect from them
  - In Linux, we can check the system against rootkits with software like `chkrootkit` or `rkhunter`
  - They may also detect local exploits and backdoors
    - https://sourceforge.net/projects/rkhunter
    - https://computernewage.com/2015/01/10/detecta-rootkits-y-rastros-de-malware-en-linux-con-chkrootkit-y-rkhunter/
    - https://www.tecmint.com/install-rootkit-hunter-scan-for-rootkits-backdoors-in-linux/
- **We can check the local machine or mount disks in another to scan it externally**
  - This is advisable, as rootkits usually hide themselves when their system is running
  - Microsoft Defender Offline was created because of this
    - https://support.microsoft.com/en-us/help/17466/windows-microsoft-defender-offline-help-protect-my-pc

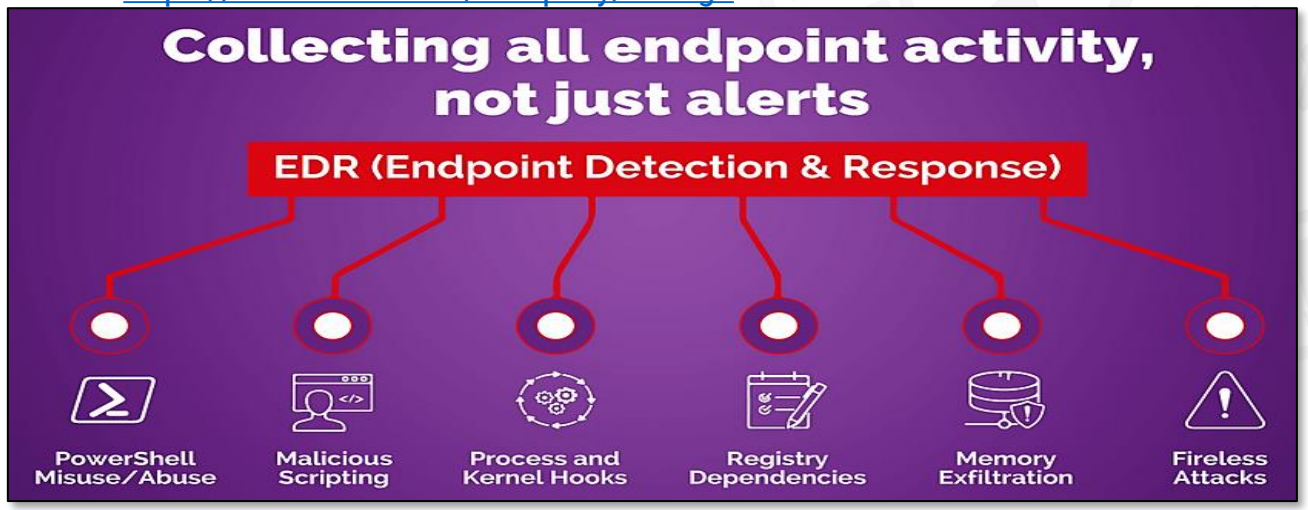
```
redondo@miw:~$ sudo rkhunter --check
[ Rootkit Hunter version 1.4.6 ]

Checking system commands...

  Performing 'strings' command checks
    Checking 'strings' command                          [ OK ]

  Performing 'shared libraries' checks
    Checking for preloading variables                   [ None found ]
    Checking for preloaded libraries                    [ None found ]
    Checking LD_LIBRARY_PATH variable                   [ Not found ]

  Performing file properties checks
    Checking for prerequisites                          [ OK ]
    /usr/sbin/adduser                                   [ OK ]
    /usr/sbin/chroot                                    [ OK ]
    /usr/sbin/cron                                      [ OK ]
    /usr/sbin/groupadd                                  [ OK ]
    /usr/sbin/groupdel                                  [ OK ]
    /usr/s
```



Endpoint Detection and Response (EDR/XDR) technology

- Continually monitors devices and responds to mitigate threats
- Identifies suspicious behavior and **Advanced Persistent Threats** (APTs) on endpoints (computing devices) in an environment
  - Collect and aggregate data from endpoints and other sources, and may direct them to SIEMs (**topic 5**)
  - Its main functionality is **alerting administrators about anomalies**, rather than protecting the endpoints
    - However, some products have real-time response to threats, detect malware injections, create element allow and block lists, and other advanced response capabilities
  - Some use the MITRE ATT&CK classification and framework for threats (**topics 5 and 7**)

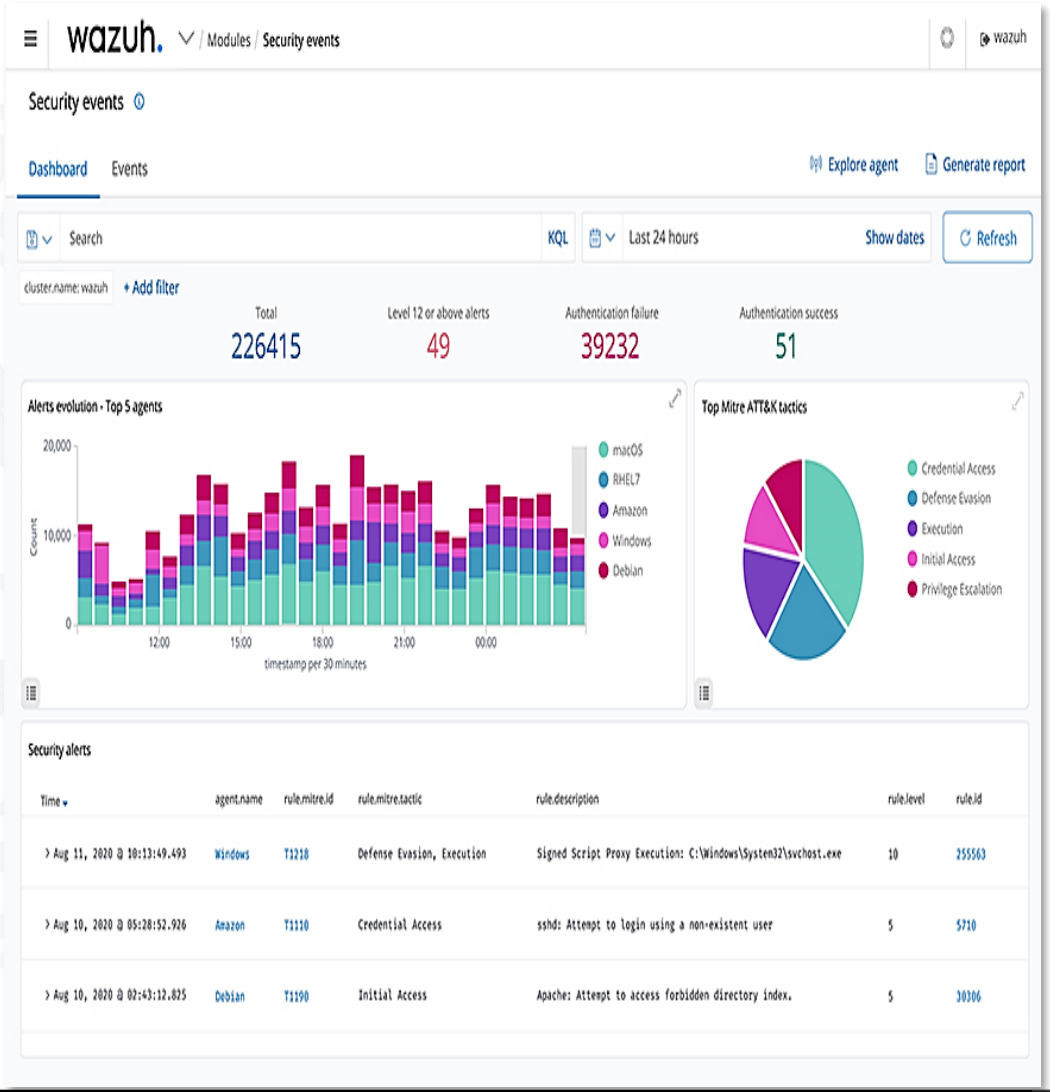
Fuente: https://ar.linkedin.com/company/absega



Fuente: https://www.dnsstuff.com/best-edr-solutions

EDR/XDR vs Antimalware

- **An antimalware is simpler and more limited in scope than an EDR**
  - In fact, several EDRs incorporate an antimalware product
- **An EDR may have more functionalities**
  - Scans, detects, and removes malware
  - Firewall
  - Multiple monitoring tools for several parts of the system (E. g.: **HIDS**, see previous image)
  - Vulnerability exploiting and data theft protection
- **The provide a more complete protection**
  - Follow a client-server architecture
  - Wazuh (https://wazuh.com/) is one of the most used XDR/SIEM solution, open source and free



➢ Exercise **L5B4_WAZUH**

Think you've understood it all? Evaluate yourself!

?

- **You can do the following**
  - *Understand that without control of what you install and without proper maintenance (updates), a system is going to be much more insecure*
  - *Understand the importance of scheduling security tasks, rather than relying on us to remember them*
  - *Understand the importance of properly synchronizing time, as it is imperative for many services*
  - *Understand why disabling compilers we don't use is another security measure*
  - *Be aware of the fundamental difference between DAC and MAC, and who decides what is or is not accessed in each case*
  - *Understand the importance of having malware scanning tools suitable for every type of malware*
  - *Understand the enormous importance of monitoring inappropriate behavior in an EDR/XDR system*

< Go to Index

Source: Bing Chat AI

Passwords >

USERS

Hardening user management

*Icons by Bing Chat AI

Departamento de Informática
Universidad de Oviedo

What are you going to find in this block?

- **In this block you will learn**
  - The main consideration when securely managing the **users** of a system
  - The huge difference between logging into a system as a **standard or high-privilege user**
  - The right measures to create **strong passwords and protect accounts**

General rules

- **As a rule, the less users that can log in, less intrusion possibilities**
  - User creation is covered in ASR
  - Therefore, we need to minimize the number of machine users that can access remotely or locally
  - There are even cases of **"throw-away" VMs** or containers with no login users
    - They run a process upon start and are regenerated entirely if something fails
- **If login users is a must…**
  - We should decrease the number of user with administrator privileges / sudoers to the minimum possible (or zero)
  - We don't want a lot of users with the ability of impersonating `root` via `sudo` calls
- **In Linux, if there must be sudoers, we could**
  - Just leave one sudoer user for administration purposes
  - Limit the number of commands that sudoers can `sudo` to, even only one

Accounts and Access

- **If an attacker gains interactive access to an account, they can perform any action or access any file that account has access**
  - Therefore, a secure system must prevent access to unauthorized users as much as possible
  - Specially to privileged accounts
- **Includes enabling Multi-Factor Authentication (MFA) on access services**
  - Like SSH: https://medium.com/mi-rincon/doble-factor-de-autenticacion-2fa-c2c8d728b978
  - We will talk more about MFA on **Topic 6**
- **Also improve access settings to make it as secure as possible**
  - For example, leaving the **most secure configuration options** possible in the SSH service
- **Remember: The easiest way to gain unauthorized access to a Linux system is to boot the server into single user mode**
  - Attackers can select a kernel to boot from the GRUB menu item

Session configuration files

- **When a user logs into an account, the system configures the user's session by reading several files**
  - Many of these files are in the user's home directory and may have weak permissions because user errors or misconfiguration
- **In Linux systems, a misconfigured `umask` value could result in files with excessive permissions accessible by unauthorized users**
  - `umask 027` is better from security perspective
  - `umask 077` is even better for `root` (only the owner can read or execute new files)
  - It avoids some common system administrator mistakes
  - It's harder for an attacker to run privilege escalation
  - The downside is when creating files/directories in a shared directory that are accessed by other users
  - In Linux, this can be achieved including in `/etc/profile` and in `/etc/bash.bashrc`: `umask 027`

LOGIN BANNERS

- **Login banners warn any possible intruders that may want to access a system that certain types of activity are illegal**
  - It also advises the authorized and legitimate users of their **obligations** (acceptable use of the computerized or networked environment(s))
  - Pre-logon warning messages can **deter** unauthorized use, increase security awareness, and provide a legal basis for prosecuting unauthorized access
- **E. g. The DoD (USA) uses this banner (Linux systems place this in `/etc/issue`)**

```
You are accessing a U.S. Government (USG) Information System (IS) that is provided for USG-authorized
use only. By using this IS (which includes any device attached to this IS), you consent to the
following conditions:
    - The USG routinely intercepts and monitors communications on this IS for purposes including, but
    not limited to, penetration testing, COMSEC monitoring, network operations and defense, personnel
    misconduct (PM), law enforcement (LE), and counterintelligence (CI) investigations.
    - At any time, the USG may inspect and seize data stored on this IS.
    - Communications using, or data stored on, this IS are not private, are subject to routine
    monitoring, interception, and search, and may be disclosed or used for any USG-authorized purpose.
    - This IS includes security measures (e.g., authentication and access controls) to protect USG
    interests -- not for your personal benefit or privacy.
    - Notwithstanding the above, using this IS does not constitute consent
```

Restrict root / Administrator logins

- **Normally, the administrator should access the system via a unique unprivileged account**
  - Only login in as an administrator when it is strictly necessary (**emergencies**)
  - In Linux systems it is advisable to use `su` or `sudo` to run privileged commands from sudoer accounts
- **Measures to harden `root` / `Administrator` logins**
  - **Only one account should have this role**
  - **Prevent direct `root` logins:** in Linux, the best course of action is disabling them (see image)
    - Use multifactor authentication (MFA) on privileged accounts
  - **Protect system account logins (service accounts):** not giving interactive login privileges to service accounts
    - Makes using them more difficult to attackers

It is not convenient that root has interactive login capabilities, it should be also nologin!

cat /etc/passwd

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run
systemd-network:x:101:103:systemd Network Management,,,:/run/sy
n
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/reso
syslog:x:103:107::/home/syslog:/usr/sbin/nologin
messagebus:x:104:108::/nonexistent:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
uuidd:x:106:112::/run/uuidd:/usr/sbin/nologin
```

User password management

Don’t let poor passwords ruin your day

*Icons by Bing Chat AI

Password policy

- **Even with its weaknesses, access via username / password is very common**
  - In Linux systems, this feature is supported by the `/etc/passwd` and `/etc/shadow` files (ASR)
  - Passwords are never stored in plain text (remember KDFs of **Topic 2**)
- **The password policy of a certain company is created by answering**
  - *What characters are permitted and forbidden to use within a password?*
  - *Is the user required to use characters from **different character sets** (lower and uppercase letters, digits and special symbols)?*
  - *How often can a user change their password?*
  - *How often can a user reuse a password? Do we maintain a **history of the user's previous passwords** so they cannot be reused again?*
  - *How quickly can a user change their password after a previous change?*
    - *Users may bypass their password history by changing their password multiple times until they can put the initial password again*
  - *When must a user change their password? After X days? After lockout due to excessive log on attempts?*
  - *How different must the new password be from the last one?*
  - *Is the user prevented from using his username or other account information in the password?*

Common password protection measures

- **Set a password expiration policy**
  - Requires setting a password warning, minimum and maximum age
  - **Password warning age** enables users to make the change before it expires
  - **Minimum password lifetime** helps prevent repeated password changes to defeat the password reuse
  - **Password maximum age** ensures users are required to periodically change their passwords (**currently using this is not recommended**, passwords should only change if they have been exposed)
- **Disable inactive accounts**
  - Ensures unused accounts are not available to attackers who may have compromised their credentials
- **Implement an adequate password strength and management policy**
  - In Windows systems, this is achieved by OS policies
    - If we are in a domain controller, we can set this once for all machines in a domain
  - In Linux systems this is normally achieved by configuring a module named PAM

Password strength policy

- **Password hashing algorithm**
  - **The most used algorithm is SHA-512**
  - Modern Linux and Windows distributions use secure hash algorithms by default (see **Topic 2**)
- **Failed password attempts**
  - **Capability to lock out user accounts after several erroneous passwords are entered**
  - Locking user accounts might facilitate DoS attacks, but prevents brute-force password guesses
- **Limit password reuse**
  - **Set how often an old password can be reused**
  - Do not allow users to reuse recent passwords that could have been compromised
  - For example, the Department of Defense of the US sets this in 5 passwords
    - A password cannot be used again until we have used 5 different passwords after that one

Password strength policy: Password quality requirements

- Using a complex / strong password helps to increase the time and resources required to compromise it
- Operating systems usually allows us to enforce these passwords controlling several variables
  - **Minimum characters required in a password**
    - The shorter the password, the lower the number of possible combinations that need to be tested before the password is compromised (hence weaker)
  - **Usage of digits, lowercase, and uppercase letters** in a password
  - **Minimum number of characters that must be different from the previous password**
  - **Maximum number of non alphanumeric characters** in a password
  - **Reject passwords which contain more than N equal consecutive characters**
- They should also be passwords that are not common or already leaked
  - Even if they meet all the conditions to be strong

Password policy implementation example: PAM module

- **Library that enables administrators to choose how applications authenticate users in Linux systems**
  - And thus strengthen its password policy
- **Offers multiple low-level authentication schemes into a high-level API**
  - However, modifications of the PAM can have unexpected consequences
    - Sometimes the system defaults must be restored if manual changes prevent the passwords to work normally
  - Tools like `authconfig` or `system-config-authentication` allows to do that
  - **User manual**
    - Using Pluggable Authentication Modules (PAM)
    - The Linux-PAM System Administrators' Guide
  - **More information**
    - Set a password policy in Red Hat Enterprise Linux 7
    - How to configure password complexity [...] using pam_passwdqc.so
    - Hardening Your System With Tools and Services
    - Configuring Services: PAM

User training

- **Training users on the prevention of security incidents is essential, but complex**
  - We need to find materials suitable for our needs
  - Users must understand the importance of this training process and that they are a key part of the company's security
- **There are many free resources available to train users of varying levels of technical expertise**
  - **INCIBE** has many training courses for different user profiles
    - E.g.: "**Experiencia senior**" (users with no knowledge): https://www.incibe.es/ciudadania/experiencia-senior
  - As part of our "**Cobra Kali**" initiative we have the "**P-45 Audaz**" project
    - **Cyber fraud training** (non-technical profiles) (in Spanish)
      - https://github.com/jose-r-lopez/Fraudes-y-Timos/wiki
      - Podcast and blog where a type of fraud is "dissected" weekly
      - Also on YouTube: https://www.youtube.com/channel/UCrB3D_97pYWlfBZBmCJiYzw
    - "**Técnicas de seguridad y prevención del fraude en mensajería electrónica**" course
      - In Spanish
      - https://uniovix.uniovi.es/course/view.php?id=62

Think you've understood it all? Evaluate yourself!

?

- **You can do the following**
  - *Understand that the fewer users who can log into the system, the greater security*
    - *Including minimizing the number of sudoers on **Linux***
  - *Understand that every remote access service to a system should have an MFA*
  - *Understand the importance of correctly configuring user session configuration files*
  - *Be aware that the warnings placed on the login screens have no real value when it comes to improving security*
  - *Understand why it is so important to operate as an elevated account for as little time as possible*
    - *Including prohibiting interactive access to it*
  - *Be aware of the rules to create a good password*
  - *Be aware of the rules for protecting accounts related to good password management*
    - *Expire passwords only in case of leakage, disable inactive accounts, force secure creation, limit reuse, block in the event of an excessive number of failed attempts...*
  - *Understand that, without adequate user training, these measures are insufficient*

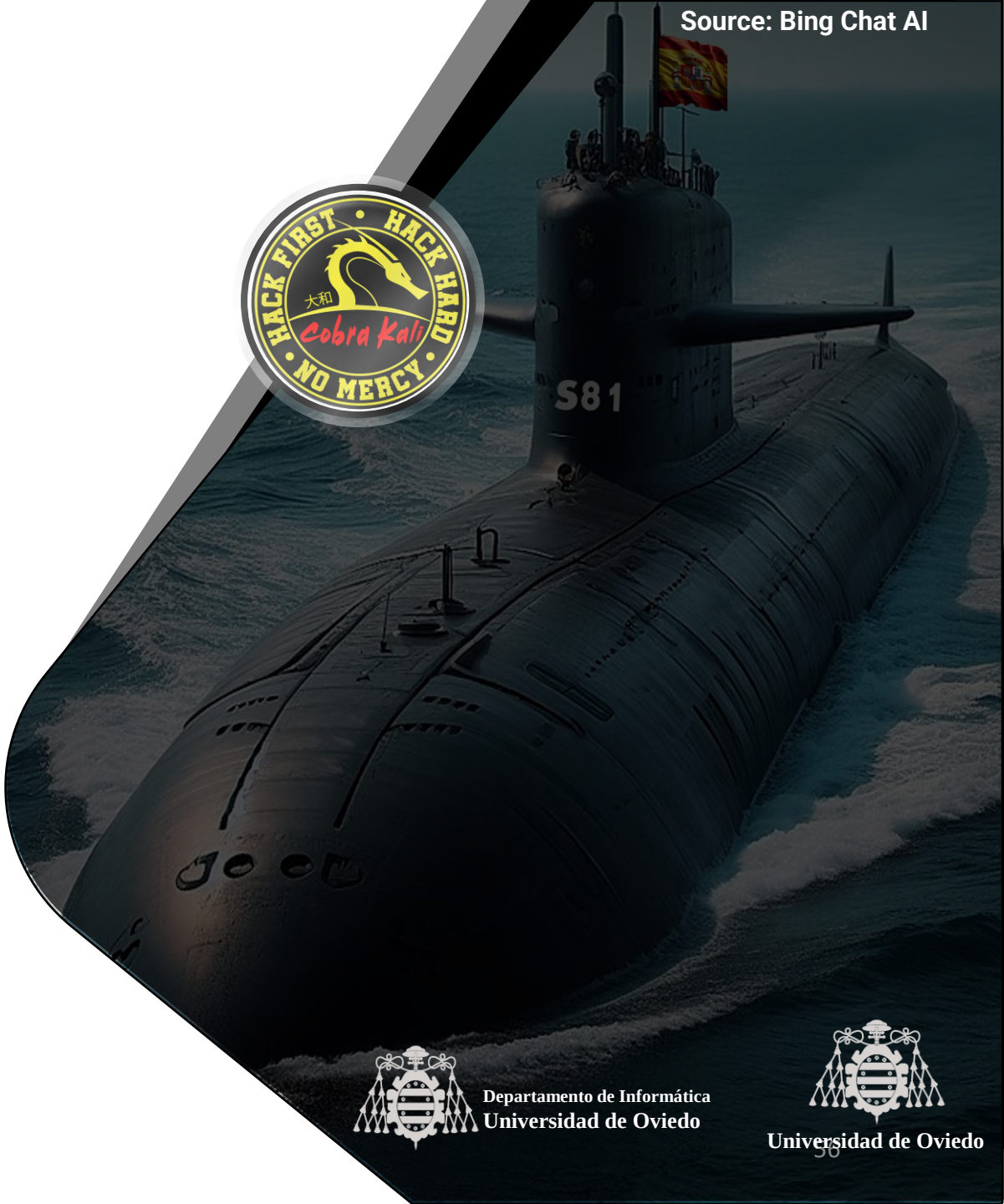
< Go to Index

FILES

Hardening the access to our data

*Icons by Bing Chat AI

Source: Bing Chat AI

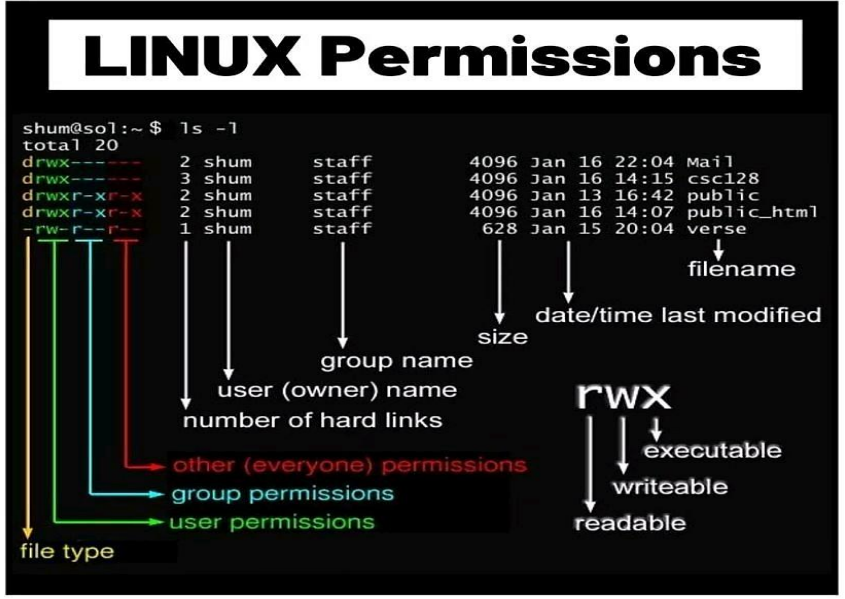


What are you going to find in this block?

- **In this block you will learn**
  - The importance of file and directory **permissions** in different OSs in order to maintain adequate security
  - The operations you need to do to **properly manage a file system**
  - How important it is to **control changes** to files that should not be modified in an OS

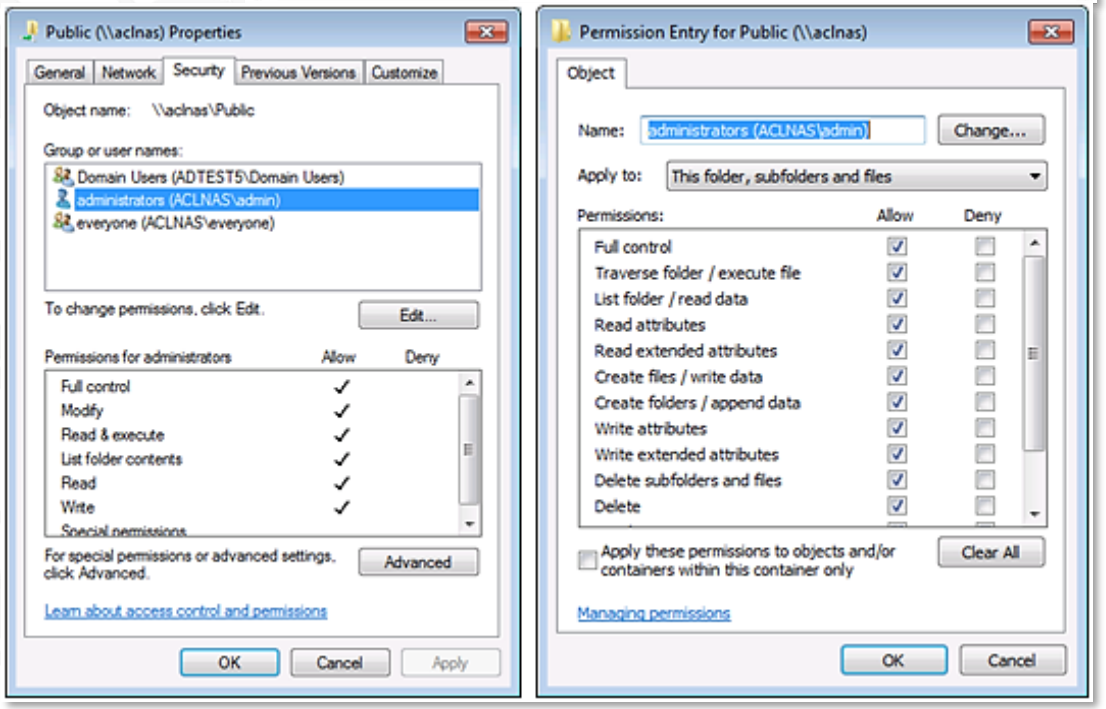
Critical files permissions and limits

Seguridad de Sistemas Informáticos



Source: https://www.devopsschool.com/blog/linux-tutorials-chmod-commands/

- **To properly understand this, you must remember the permission schema used by your OS**
  - The first image shows the typical **Linux** one
- **Windows uses a different schema, based on users that belong to groups**
  - Permissions can be assigned to files or directories to multiple users and/or groups, forming an **Access Control List** (ACL) (see second image)
  - There are multiple defined permissions allowing more granular file access than in **Linux**
- **Permissions are reviewed in SO and ASR**



Source: http://qnapsupport.net/qnap-uygulamalari/kullanici-ve-klasor-olusturma/klasor-olusturma-ve-yetkilendirme/how-to-use-windows-acl-to-manage-user-permissions-on-the-qnap-nas

CRITICAL FILES PERMISSIONS AND LIMITS

- **OS security relies on appropriate file and directory permissions**
  - They prevent unauthorized users from reading or modifying files to which they should not have access
  - Permissions for many files must be set **as restrictively as possible** to ensure sensitive information is properly protected
  - The default restrictive permissions for files which act as important security databases must be maintained, checking that they have not been changed (**indication of compromise, IoC**)
    - `passwd`, `shadow`, `group`, and `gshadow` (Linux) (`/etc` folder)
    - The `SAM` file (Windows) in the `C:\Windows\System32\config` folder
- **Ensure all files are owned by a user and group**: Files without an owner may be caused by
  - An intruder using tools to exfiltrate data or different malware
  - Incorrect software installation or failed removal
  - Failure to remove all files belonging to a deleted account

Critical files permissions and limits

- **About files and directories that are world-writable**
  - No file should have these permissions
    - Data in world-writable files can be modified by any system user, and supposes a high security risk
    - Files must use a combination of user / group permissions to allow legitimate access
  - In Linux, all world-writable directories must have their sticky bits set
    - Not doing it on public directories allows unauthorized users to delete files in its structure
- **Disable unused filesystems in Linux**
  - Linux has features for automatically add and remove filesystems on a running system
  - Enabling these carries some risk (always minimize software!)
- **Disable core dumps (large RAM images dumped into files)**
  - They may contain sensitive data present in program's memory when the dump was performed
    - Passwords, API keys, access tokens, active HTTP sessions IDs…
    - There are tools to extract information from dumps: the IFA course uses the Volatility Framework
      - https://cqureacademy.com/blog/forensics/memory-dump-analysis
      - http://manpages.ubuntu.com/manpages/bionic/man8/crash.8.html

Critical files permissions and limits: Unwanted file changes

- After the base OS installation, disk files can be divided in two categories
  - **Files that are modified frequently**: user files, development files, logs…
  - **Files that should not be modified** (or are modified only in specific circumstances)
    - OS system files, device drivers…are normally modified on OS **updates**
    - An unplanned modification of these files may indicate malware (e. g. ransomware)
- A **HIDS** (**Host-Based Intrusion Detection System**) can detect modifications to sensitive files
  - Detecting one may reveal malware presence before it is too late
  - Tripwire, Aide and OSSEC (previously mentioned) are three of the most famous ones
  - **More examples**: https://www.dnsstuff.com/host-based-intrusion-detection-systems
- Details about filesystems and their contents are explained in **ASR**

Think you've understood it all? Evaluate yourself!

?

- **You can do the following**
  - *Understand the differences between Linux and Windows permissions*
  - *Understand that there are files that must have very restrictive permissions due to the nature of the data they store*
    - *And that, if this is not the case, it is an indication of compromise on the part of the system*
  - *Understand that a write-in permission for everyone is a security risk*
  - *Understand that core dumps may have private information and should therefore be disabled*
  - *Understand that a HIDS is key to identifying changes in OS files that should not change except in very specific circumstances*

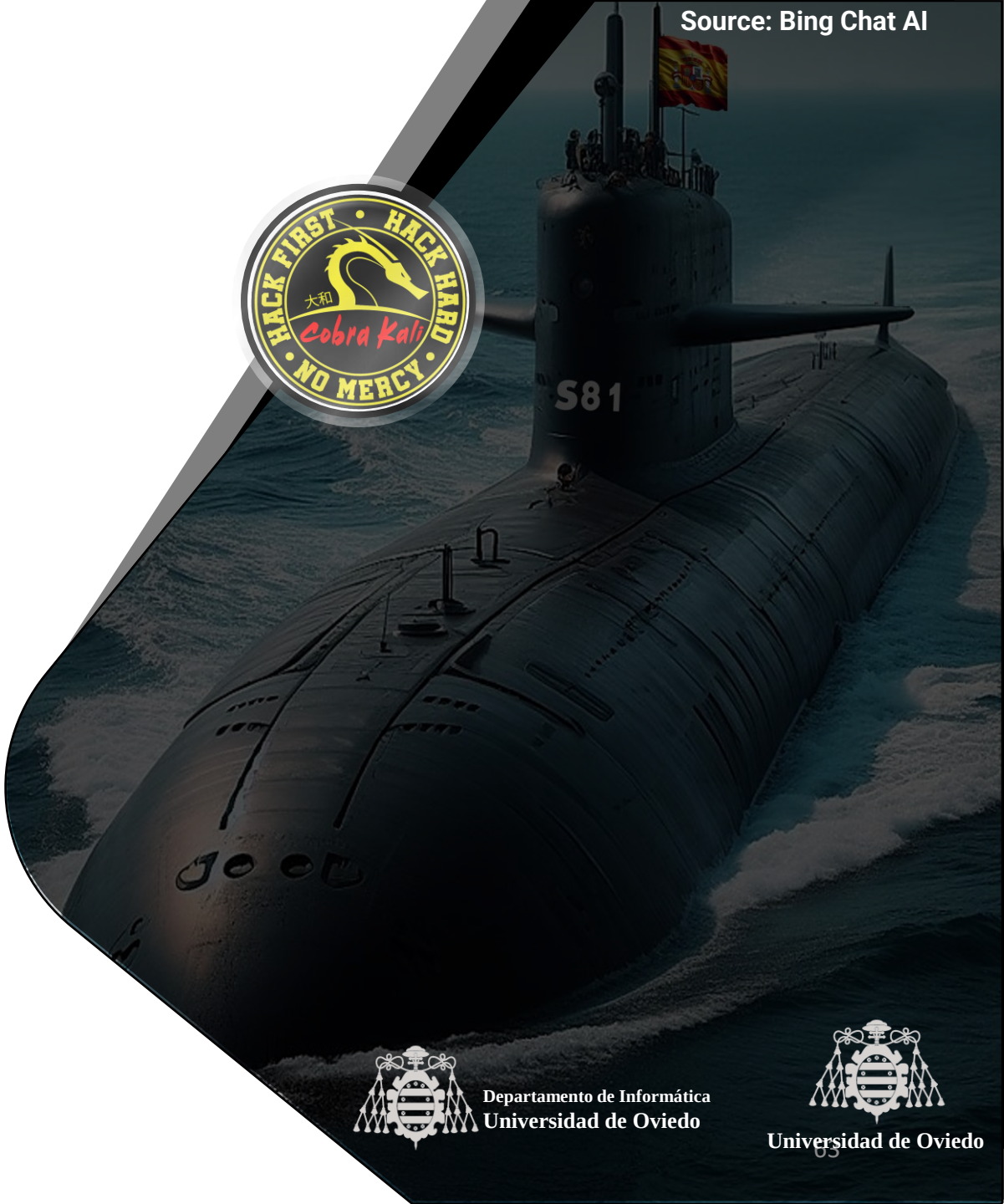
< Go to Index

How to evaluate and achieve security

Hardening in an efficient way

*Icons by Bing Chat AI

Source: Bing Chat AI



Departamento de Informática
Universidad de Oviedo

Universidad de Oviedo

IMPLEMENTING AND EVALUATING SECURITY

- **Strengthening an OS means incrementing its current security level**
  - This process is also known as **hardening**
- **To do that, we apply a series of security-enforcing operations to parts of the OS**
  - In this presentation we reviewed the most sensitive areas
  - Each of these security-enforcing operations is called a **security control**
  - Therefore, hardening any OS **means implementing a (extensive) list of security controls**
    - Previously checking if they are not already applied!
  - If we don't have a list of security controls available, we can always rely in **best-practice guides** for concrete software products as a back-up plan
    - Official Spanish ones can be consulted for free from the CCN-CERT website: https://www.ccn-cert.cni.es/informes/informes-de-buenas-practicas-bp/
- **This hardening process can be applied to any software product!**
  - Hardening software means obtaining its best list of security controls first
  - And later implement each one of them, if not already applied
    - Except if that breaks its legitimate functionality!

IMPLEMENTING AND EVALUATING SECURITY

- **The security control lists you use have to be adequate**
  - **Validated** by trustable third parties, **complete**: (covering the "sensitive" parts of the software and **properly structured**
  - To facilitate its readability and implementation, every security control of the list must at least have
    - A **rationale** explaining why each control is necessary
      - This way we know if it is worth implementing in case it collides with the legitimate functionality of the system
    - A precise procedure to **check or audit** if the security control is already implemented
    - A precise procedure to **implement** the security control
- **Please make sure you use trustable sources of security control lists**
  - We will see some sources on next topic
  - **Please avoid forums, StackOverflow, blog articles, etc. for doing this!**
- **Do not leave your OS or software without hardening using one of these lists**

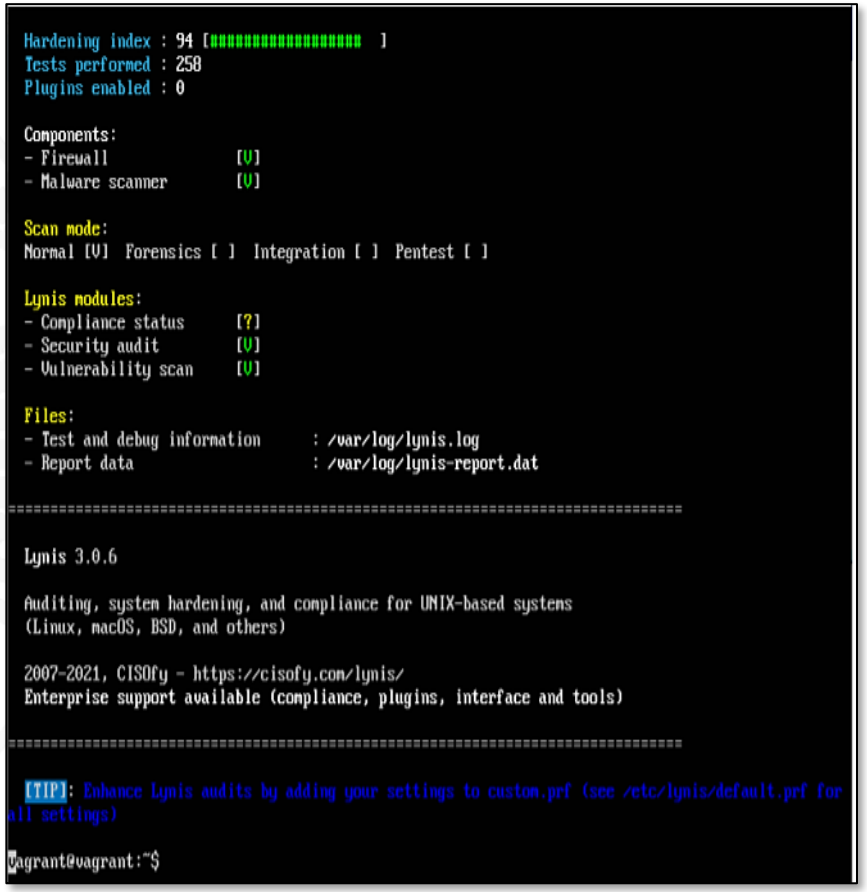
Implementing and evaluating security

- **So, having an extensive list of properly validated security controls is the only way of ensuring a proper security level**
  - But…this list **does not have to be applied manually**!
  - In fact, many security controls can be checked and applied **automatically**!
- **Automatically applying security controls means that there are a lot of situations in which a high level of hardening can be achieved…with little effort!**
  - In the next topic we will study tools and protocols that enable us to do that
  - And, if we need precise control, we can also mix automation with manual fine-tuning ☺
- **But what if I just want to evaluate the general security of an OS?**
  - Security control lists usually allow **an evaluation-only mode**
  - We can get an automated security evaluation with tools like Lynis (for Linux systems)
  - CLARA (https://www.ccn-cert.cni.es/es/soluciones-seguridad/clara.html) can be used on Windows (and certain Linux) systems

LYNIS

https://github.com/CISOfy/lynis

- **Performs an in-depth security scan on a local Linux, giving a "security score"**
  - The primary goal is to test security defenses and provide tips for further system hardening
  - Implements security auditing and vulnerability detection
  - Also scans for general system information, vulnerable software packages, and possible configuration issues
  - Compliance testing with security policies (e.g., ISO27001, PCI-DSS, HIPAA…), seen on **Topic 4**
- **The software (also) assists with**
  - Configuration, asset, and software patch management
  - System hardening and penetration testing (privilege escalation)
  - Intrusion detection
- **Tutorial**
  - https://geekytheory.com/tutorial-linux-audita-tu-sistema-con-lynis



➢ Exercises **L5B1_LYNIS1, L5B1_LYNIS2**

TOPIC 3
OPERATING SYSTEM SECURITY

Departamento de Informática
Universidad de Oviedo